



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

**“PSYCHOACOUSTICS: UNA WEB APP PER LA RICERCA NEL
CAMPO DELLA PSICOACUSTICA”**

Relatore: Prof. Mauro Migliardi

Laureando/a: Niccolò Orlandi

Correlatore: Prof. Massimo Grassi

ANNO ACCADEMICO 2021 – 2022

Data di laurea 21/09/2022

Indice

Sommario

.....	0
CORSO DI LAUREA IN INGEGNERIA INFORMATICA	0
“PSYCHOACOUSTICS: UNA WEB APP PER LA RICERCA NEL CAMPO DELLA PSICOACUSTICA”	0
Abstract	4
Introduzione	6
Capitolo 1	9
Modelli di sviluppo software	9
1.1 Modello a cascata.....	9
1.2 Modelli iterativi.....	10
1.3 Le metodologie agili	11
1.4 Metodologie Agili vs Metodologie classiche.....	12
Capitolo 2.....	13
Letteratura Psicoacustica.....	13
2.1 Introduzione	13
2.2 Il calcolo della soglia sensoriale	13
2.3 L’algoritmo staircase.....	18
2.4 Il metodo dei limiti.....	18
2.5 L’algoritmo Simple Up Down	19
2.6 L’algoritmo Transformed Up Down	20
Capitolo 3.....	23

Analisi dei requisiti.....	23
3.1 Anteprima del progetto.....	24
3.2 Test	26
3.2.1 Pure tone intensity discrimination	28
3.2.2 Pure tone frequency discrimination.....	29
3.2.3 Pure tone duration discrimination.....	30
3.2.4 Algoritmi per calcolo score:	31
3.3 Gestione utenti.....	31
Capitolo 4	33
Sviluppo della web application.....	33
4.1 Framework VS Scripting standalone	33
4.2 Sviluppo Front End.....	33
4.3 Sviluppo Back End.....	33
4.3.1 La API Web Audio	33
4.4 Connessione e comunicazione al database	35
Capitolo 5	39
Conclusioni e possibili sviluppi futuri.....	39
Bibliografia.....	41
Ringraziamenti.....	43
Appendice.....	43

Abstract

Psychoacoustics è una web-application che nasce dalla necessità dei ricercatori di rendere una toolbox matlab fruibile a chiunque, eliminando le limitazioni di compatibilità e di usabilità e permettendo contemporaneamente la raccolta e l'analisi dei dati ottenuti dagli studi e dai test effettuati.

Psychoacoustics contiene infatti un pacchetto di test predefiniti fondamentali per la ricerca nel campo della psicoacustica.

L'approccio alla modalità di progettazione e sviluppo della web app è stato il cosiddetto "metodo AGILE" che permette di minimizzare l'impatto di eventuali cambiamenti. La metodologia di realizzazione della web-app, si è perciò basata sulla ripetizione ciclica di tre fasi:

- Analisi dei requisiti
- Stesura documentazione
- Sviluppo
- Revisione con il committente (prof. Grassi)

Questi quattro passi sono stati poi ripetuti dopo ogni review.

I risultati ottenuti sono stati in linea con i requisiti documentati inizialmente, sono stati infatti creati i principali strumenti di ricerca necessari per effettuare i test più importanti cioè i test psicometrici su intensità, frequenza e durata del suono.

All'orizzonte però vi è ancora un ampio margine di miglioramento per esempio:

- L'incremento della sicurezza della web app, i dati sensibili sono oscurati ma ci sono sicuramente problematiche di sicurezza non completamente sviscerate a causa della loro complessità.
- L'introduzione di un metodo di sintetizzazione del suono mediante vettori, così da rendere la generazione dei segnali quanto più personalizzabile possibile.
- L'introduzione di rumori sovrapposti ai segnali dei test.

Introduzione

L'idea di creare la web app è nata dalla necessità di permettere a chiunque di sfruttare i servizi messi a disposizione dalla toolbox matlab precedentemente ideata dal professor Grassi (committente della web-app) oggi deprecata e difficilmente utilizzabile a causa della necessità di installare degli ambienti di lavoro specifici, spesso difficilmente sfruttabili dalle persone con meno esperienza o conoscenza tecnica.

L'obiettivo prefissato è di progettare e sviluppare una web app per la ricerca in campo psicoacustico a partire dalle specifiche della toolbox e creare un database per permettere i ricercatori di salvare i dati ottenuti.

Data l'impossibilità di confronto con la toolbox è stata fatta un'analisi dei requisiti iniziale assieme al committente, così da avere un'idea generale delle sue necessità, successivamente è stata applicata una particolare metodologia di progettazione e sviluppo software chiamata metodologia agile (Descritta nel capitolo 1) che permette il continuo affinamento dei requisiti e feedback del committente. La metodologia è stata particolarmente utile poiché ha permesso al professore di essere costantemente a conoscenza di modifiche e migliorie effettuate potendo così partecipare attivamente alle fasi di progettazione e sviluppo della web-app.

Inizialmente il prof Grassi ha spiegato una parte di letteratura psicoacustica (Introdotta nel capitolo 2) necessaria alla comprensione dei test e del loro funzionamento.

Si è quindi passati all'analisi iniziale dei requisiti (Trattati nel capitolo 3) dove sono stati individuati i fattori principali quali la facilità di utilizzo per l'utente, i test necessari ai ricercatori, la necessità di storage e manipolazione dei dati ottenuti.

Le prospettive di impiego sono varie, può essere usata per fare ricerca, ma anche durante le lezioni per mostrare agli studenti un'applicazione pratica dei concetti di psicoacustica studiati a lezione.

Sarà utile in modo particolare per semplificare il lavoro di chiunque avrebbe voluto usare la toolbox ma non era sufficientemente esperto per creare i propri test o per installarla.

Attualmente i test a disposizione sono quelli su ampiezza, frequenza e durata del suono mentre gli algoritmi per il calcolo della soglia di percezione uditiva sono SimpleUpDown e TransformedUpDown, ma è in continuo sviluppo e miglioramento (Spiegati nel capitolo 3). Successivamente è iniziato lo sviluppo della web app, suddivisa in Front-End Back-End e connessione con il database (Descritti nel capitolo 4).

La web app ottenuta si distingue dalla toolbox poiché è utilizzabile da un maggior numero di utenti anche senza competenze tecniche specifiche, non è necessario installare l'ambiente di lavoro e permette lo storage dei dati raccolti dagli utenti, ha però un unico svantaggio, il tempo necessario per la creazione dei test, infatti per lo sviluppo del test è necessario avere competenze tecniche maggiori rispetto a prima, e più tempo per ricercare delle funzionalità che possano aiutare il programmatore nella sintetizzazione dei suoni (Argomento trattato nel capitolo 5). In futuro sono in previsione alcuni miglioramenti come l'inserimento di nuovi test, e l'introduzione di ulteriori algoritmi di sintetizzazione del suono e di rumori che consentano di migliorarne le performance (Trattati nel capitolo 5).

Capitolo 1

Modelli di sviluppo software

«Un solo giorno con un bravo insegnante è meglio di mille giorni di studio diligente.»

(Proverbio Giapponese)

Lo sviluppo di un software è un processo lungo e complesso e necessita:

- La realizzazione del prodotto con le tecnologie, il tempo e le risorse economiche a disposizione del committente.
- L'interazione tra committenza e produttore.
- La definizione dei requisiti del prodotto.

E' facile notare che ci sono quindi molteplici agenti che agiscono in parallelo e si influenzano reciprocamente rendendo necessaria una meticolosa gestione delle risorse a disposizione.

Nel corso della storia sono quindi stati creati dei modelli per lo sviluppo di un software che tengono conto dei fattori precedentemente elencati; di seguito si riportano i più importanti:

- Modello a cascata (Waterfall)
- Modello a cascata ibrido (Hybrid waterfall)
- Modello incrementale.
- Modello iterativo.
- Modello agile.

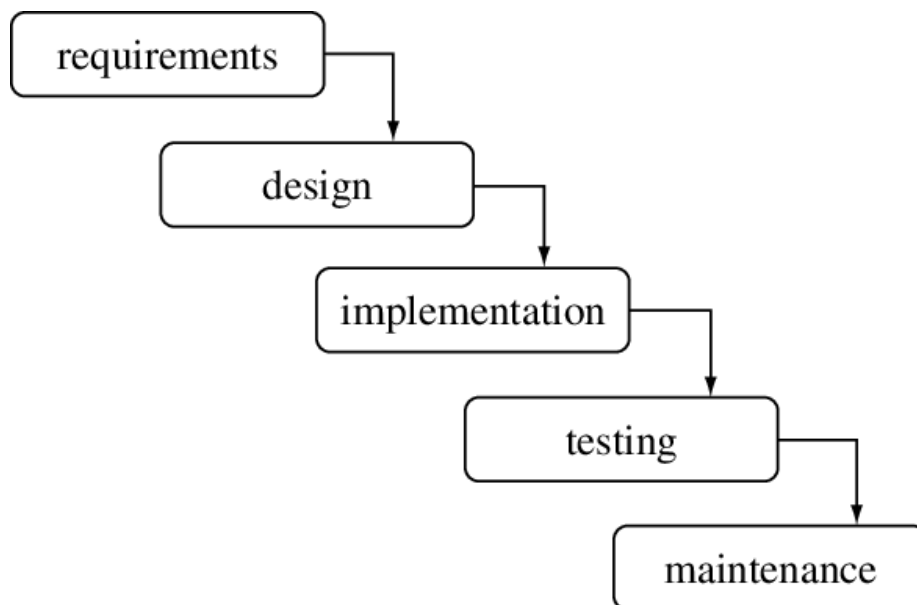
1.1 Modello a cascata

Nei modelli a cascata l'utente è coinvolto solo durante la raccolta dei requisiti ed il collaudo per l'accettazione finale. Quindi l'utente elenca all'analista i requisiti di cui ha bisogno ed approva analisi e disegno del progetto.

Ciò può essere problematico in quanto spesso il committente non sa con certezza di cosa ha bisogno e avrà bisogno di fare delle variazioni ai requisiti ma, non essendo coinvolto nella

fase di sviluppo, non avrà modo di rendersi conto dei cambiamenti desiderati e di comunicarli al team.

Inoltre, eventuali variazioni ai requisiti emergeranno durante la fase di collaudo comportando così lo stravolgimento del prodotto e sprecando tempo e soldi per implementare i cambiamenti.



1.2 Modelli iterativi

Nei modelli iterativi si procede a piccoli cicli chiamati "sprint" durante i quali il committente è costantemente coinvolto rendendolo parte integrante del progetto.

Ogni ciclo raccoglie discute e valuta i requisiti, integrandone di nuovi se necessario, in questo modo è possibile fare le variazioni appena se ne evidenzia la necessità.



1.3 Le metodologie agili

Le metodologie agili sono state create per trovare una soluzione alla sempre maggiore complessità dei sistemi da progettare e si basa su dodici principi fondamentali:

- Massima priorità nel soddisfare il cliente rilasciando software di valore fin da subito e in modo continuativo
- Accogliere i cambiamenti dei requisiti anche a stadi avanzati dello sviluppo sfruttando i cambiamenti a favore del vantaggio competitivo del cliente
- Consegna frequente di software funzionante con cadenza variabile da due settimane a due mesi con preferenza ai periodi brevi
- Committente e sviluppatori lavorano assieme per tutta la durata del progetto
- Progetti fondati su individui motivati dando loro l'ambiente e il supporto di cui hanno bisogno e confidando nelle loro capacità
- Comunicazione "faccia a faccia" con il team per rendere la comunicazione più efficace
- Software funzionante usato come principale metro di misura del progresso
- Promozione di uno sviluppo sostenibile: sponsor, sviluppatori e utenti dovrebbero essere in grado di mantenere indefinitamente un ritmo costante
- Attenzione all'eccellenza tecnica e alla buona progettazione in modo da esaltare l'agilità
- Massimizzazione della quantità di lavoro non svolto attraverso la semplicità
- Auto-organizzazione del team per rendere architetture, requisiti e progettazione migliori
- Riflessione sul miglioramento dell'efficienza a intervalli regolari e conseguente adattamento del comportamento



1.4 Metodologie Agili vs Metodologie classiche

Le metodologie classiche tendono ad analizzare un progetto nella sua totalità, per prevedere funzionalità future, con un grande sforzo nella fase di analisi.

Le metodologie “agili”, al contrario, tendono a considerare piccole parti del progetto ed a completarle velocemente, ottenere subito risultati tangibili, per completare il progetto attraverso iterazioni successive.

È inoltre importante evidenziare che nei modelli a cascata ogni fase richiede che la precedente sia completata, le continue richieste di variazioni dell’utente o del cliente, richiedono spesso che si debba tornare a fasi precedenti per meglio definire alcuni requisiti.

Un'altra differenza è lo scopo del processo di analisi dei due modelli: nelle metodologie classiche il processo di analisi ha un carattere predittivo, cioè cerca di prevedere future funzionalità necessarie, le metodologie agili invece hanno un carattere adattivo in quanto viene riconosciuta l'impossibilità di predire tutte le future possibili funzionalità integrative

Alcuni esempi di metodologie agile sono:

Extreme Programming, SCRUM, DSDM e Agile Modeling

Capitolo 2

Letteratura Psicoacustica

Per dare una migliore comprensione del funzionamento della web app si introduce ora un estratto della letteratura psicoacustica:

2.1 Introduzione

Gli psicofisici necessitano spesso di calcolare soglie sensoriali di vario tipo e per farlo, negli anni, gli studi hanno portato al consolidamento di due diverse procedure principali: le procedure adattive e le procedure non adattive.

Le procedure adattive sono le tecniche più efficienti nella misurazione della soglia sensoriale (e.g., Watson and Fitzhugh, 1990)¹. Queste tecniche vengono impiegate dagli psicofisici sin dagli inizi (Fechner, 1889)² e vengono ancora sviluppate oggi (e.g., Faes et al., 2007)³. Non sono però usate quanto ci si potrebbe aspettare, la ragione potrebbe essere la mancanza di tools di ricerca per implementare queste procedure in modo semplice ed efficace.

2.2 Il calcolo della soglia sensoriale

In psicoacustica esistono due tipi di soglie: rilevamento e discriminazione.

La soglia di rilevamento è il livello minimo di stimolo rilevabile in assenza di qualunque altro stimolo della stessa specie. In altre parole, la soglia di rilevamento segna l'inizio della sensazione di un dato stimolo.

La soglia di discriminazione è la minima differenza rilevabile tra due livelli di stimoli.

Pertanto, per un dato continuum sensoriale, la soglia di discriminazione taglia i gradini in cui è suddiviso il continuum sensoriale.

La soglia di rilevamento può essere stimata tramite test sì/no o in alternativa tramite compiti a scelta forzata (chiamati nAFC, dove n è il numero di alternative).

La soglia di discriminazione, invece, è stimata principalmente tramite test nAFC. Usando il metodo sì/no il soggetto è sottoposto a una successione di diversi livelli di stimolo (da sopra la soglia sensoriale a sotto la soglia sensoriale) e gli viene chiesto di dire se ha sentito lo stimolo (sì o no). Nei test nAFC il soggetto è sottoposto ad una serie di n stimoli di livelli differenti. Nei test nAFC uno stimolo è variabile e cambia nel corso del test, mentre il livello del secondo stimolo è standard. La differenza tra suono standard e variabile varia da sopra a

sotto la soglia sensoriale. Dopo ogni riproduzione dei suoni viene chiesto all'utente di scegliere quale dei due suoni era quello variabile.

In figura 1 un ipotetico risultato di un test si/no i risultati del soggetto sono descritti dalla logistic function (curva tratteggiata). Si osservi che il soggetto ha generato dei falsi allarmi (descritti in seguito nel testo) poiché nel livello di stimolo 0 possiamo ancora osservare un certo numero di risposte positive (sì).

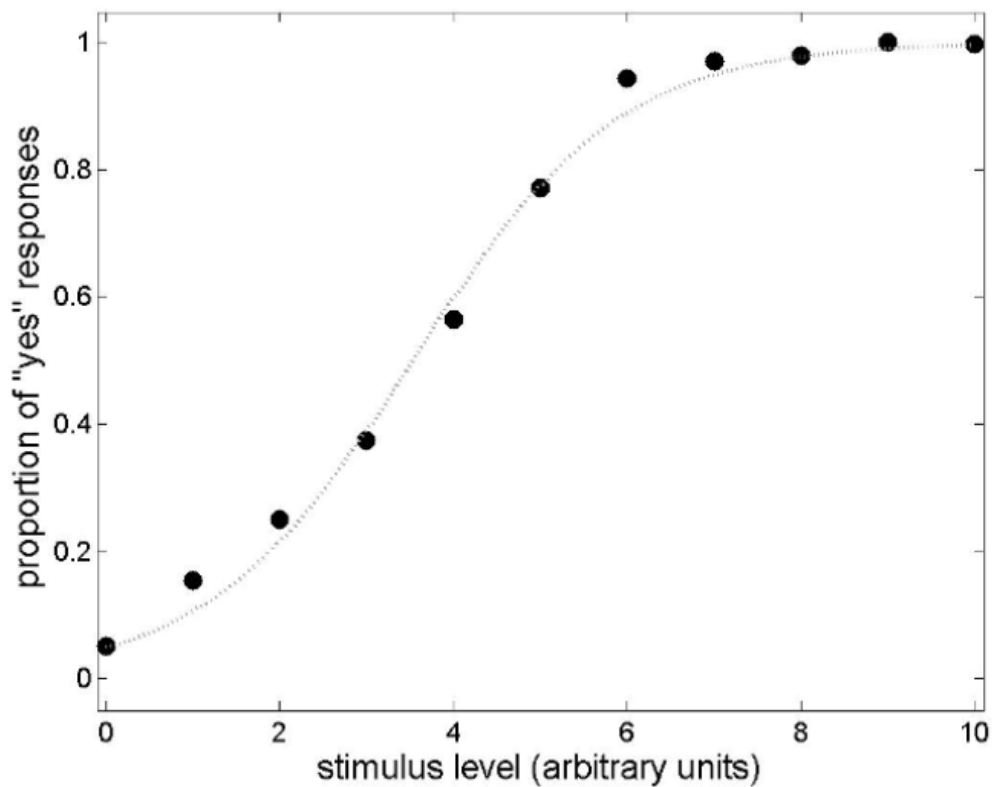
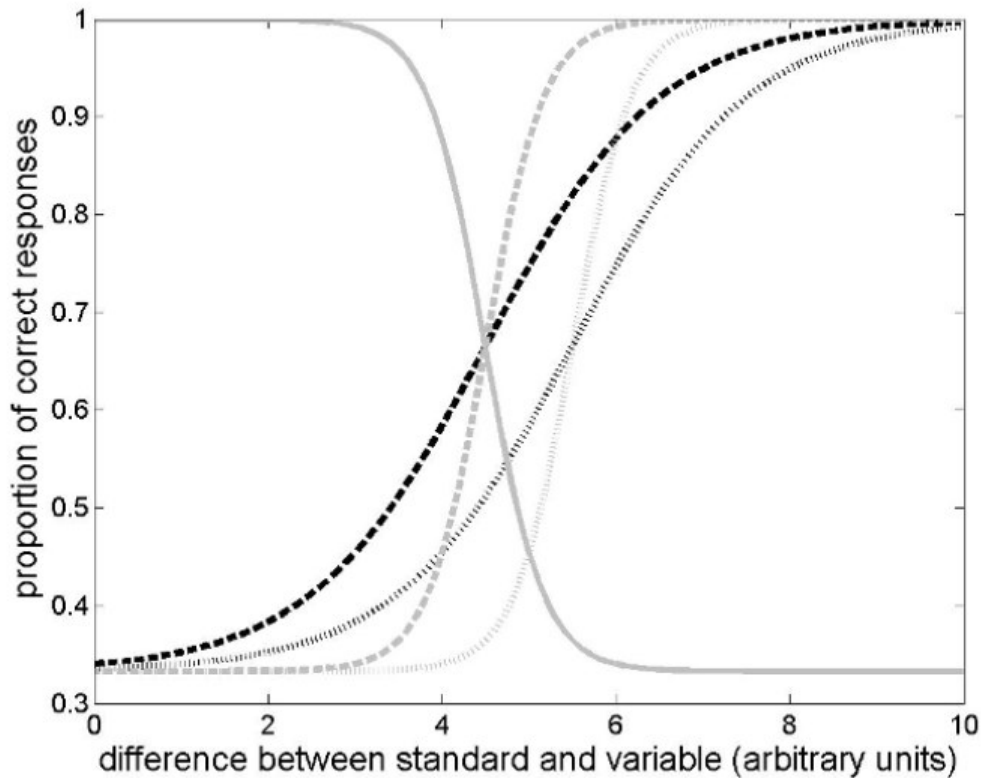


Figura 2 Cinque logistic psychometric functions. Le grigie o nere hanno pendenza identica. Le funzioni punteggiate hanno puntomedio identico, la linea grigia continua ha invece pendenza negativa



Tra le funzioni sigmoide la logistic è la più usata a causa della sua semplicità computazionale. La formula è la seguente:

$$f(x) = \frac{1}{1 + e^{\beta(\alpha+x)}}$$

Inoltre, la corrispondente funzione psicometrica è

$$\psi = \gamma + (1 - \lambda - \gamma) \left(\frac{1}{1 + e^{\beta(\alpha+x)}} \right)$$

Nella funzione logistica psicometrica, α (generalmente indicata come punto medio) permette lo spostamento della funzione lungo l'asse del livello dello stimolo. Corrisponde alla media tra γ e λ [i.e., $\Psi(\alpha) = (\gamma + \lambda) / 2$].

β è la funzione pendenza, i.e., il tasso di cambiamento nelle performance del soggetto con il livello di stimolo. Più grande è il valore assoluto di β , più ripida sarà la funzione psicometrica. Inoltre, per valori positivi di β la funzione aumenta mentre per valori negativi diminuisce (vedi figura 2). γ e λ entrano in gioco per adattare la funzione alle necessità psicologiche. γ assume significati diversi a seconda di come viene impiegata. C'è infatti una grande differenza tra le due metodologie, nei test sì/no il criterio di risposta del soggetto non è sotto il controllo dello sperimentatore, al contrario lo è nei compiti nAFC (Green e Swets 4,

1966; Stainslaw & Todorov, 1999⁶). La regione di questa differenza è che in un test sì/no, quando una risposta si viene ricevuta per un livello di stimolo molto basso, è difficile attribuire questa risposta a un pregiudizio verso la risposta sì. Le risposte distorte sono dette falsi allarmi e incidono sul limite inferiore della funzione psicometrica, che può assumere valori maggiori di zero (figura 1). In altre parole, la probabilità di ottenere una risposta sì in assenza dello stimolo è maggiore di zero (Green, 1993)⁵. Quindi, nei test sì/no γ corrisponde al tasso di falsi allarmi dei soggetti. I falsi allarmi sono assenti nei test nAFC (Green & Swets⁴, 1966; Stainslav e Todorov 1999⁶). Nelle attività nAFC, il livello degli standard è sempre diverso dal livello della variabile e le prove possono avere quindi solo risposte corrette o errate. Quando la differenza del di livello tra standard e variabile è inferiore alla soglia del soggetto (cioè “quando il livello di stimolo è basso”) la probabilità che il soggetto restituisca la risposta corretta è determinata dal caso e il caso dipenderà dal numero di alternative. Per questo motivo, nei test nAFC, γ corrisponde al livello casuale, cioè il reciproco del numero di alternative (es 50% per 2AFC, 33% per 3AFC, 25% per 4AFC e così via).

Il significato di λ al contrario è indipendente dal compito e rimanda ad un altro errore.

Sia nei test sì/no che quelli nAFC, i soggetti potrebbero commettere errori indipendentemente dal livello dello stimolo, sono le disattenzioni (es Voglio rispondere “A” ma premo erroneamente il tasto “B”). Si stima che le cadute d’attenzione siano una piccola percentuale delle risposte del soggetto (es 1-5%, Saberi & Green, 1997⁸; Wichmann & Hill, 2001)⁷ e possono influenzare l’adattamento della funzione psicometrica diminuendo il limite superiore della funzione (vedi Wichmann & Hill, 2001 per un’ampia discussione)⁷. Le disattenzioni sono particolarmente problematiche ad alti livelli di stimoli – in test sì/no – o elevate differenze nel livello tra lo standard e le variabili – nei test nAFC (Saberi & Green, 1997⁸; Whichmann & Hill, 2001)⁷. I ricercatori sono spesso interessati a stimare un singolo punto della funzione psicometrica, che è la soglia del soggetto. In termini probabilistici, la soglia corrisponde ad un punto arbitrario della funzione psicometrica p (di seguito denominato p -target) incluso tra il limite inferiore e superiore della funzione (cioè, γ e λ). In altre parole, quando vogliamo stimare una soglia, cerchiamo il livello di stimolo che elicit la proporzione p -target di risposte sì (o corrette). Treutwein (1995) propone che il p -target dovrebbe essere nel mezzo della funzione psicometrica (es 50% per compiti sì/no, 75% per 2AFC, 66% per 3AFC, ecc.). Tuttavia, altri autori suggeriscono che dovrebbero essere presi di mira valori più elevati (Green, 1990⁹; Baker & Rosen 1998, 2001¹¹; Amitay, Irwin, Hawkey, Cowan e Moore, 2006¹⁰).

Le soglie possono essere stimate mediante due classi di procedure: adattive e non adattive.

Nelle procedure adattive, ad esempio il metodo degli stimoli costanti, i livelli (o differenze tra livello standard e variabile) sono preimpostati prima dell'inizio dell'esperimento. Gli stimoli dovrebbero estendersi da sotto a sopra la soglia del soggetto.

Durante l'esperimento, gli stimoli vengono presentati al soggetto in ordine casuale e la proporzione di risposte sì (o corrette) viene calcolata per ogni stimolo. In altre parole, la soglia del soggetto sarà interpolata da una funzione psicometrica completamente discretizzata, la misurazione della soglia è costosa in termini di tempo di esperimento. Questo rappresenta il principale inconveniente di questa classe di procedure quando lo sperimentatore ha bisogno di stimare solo la soglia del soggetto. Un altro inconveniente è che spesso lo sperimentatore sa dove la soglia può essere. Pertanto, potrebbe definire male i livelli degli stimoli dell'esperimento. Per i motivi di cui sopra, quando hanno bisogno di stimare una soglia, gli psicofisici preferiscono procedure adattive rispetto a quelle non adattive. Nelle procedure adattive, i livelli di stimolo sono selezionati contemporaneamente all'esecuzione dell'esperimento, a seconda delle risposte del soggetto. Le procedure adattive massimizzano il rapporto tra il numero di stimoli presentati a/vicino alla soglia e numero di stimoli presentati lontano dalla soglia. Le procedure adattive possono essere divise in due tipi: non parametrico (noto anche come staircase) e parametrico. L'unica condizione che permette di usare procedure non parametriche è che la funzione psicometrica sia monotona. Le procedure parametriche, al contrario, fanno più ipotesi. Per esempio, assumono la forma della funzione psicometrica. Esempi di procedure non parametriche sono il metodo dei limiti di Fechner (1889)², il simple up down di von Békésy (1947) e il transformed up-down di Levitt (1971).

Esempi di procedure parametriche sono il PEST, di Taylor e Creelman, (1967), il PEST migliorato di Pentland (1980) e il QUEST di Watson e Pelli, (1983), e il metodo di massima verosimiglianza (12, Green 1990, 1993)⁹. Le procedure non parametriche sono generalmente più utilizzate di quelle parametriche, anche se comportano alcuni svantaggi. Il principale è che tendono a richiedere più tempo (ad esempio 13, Amitay et al., 2006¹⁰; Leek, 2001¹²). Tuttavia, le procedure non parametriche sono teoricamente semplici e possono essere più facilmente implementate tramite software convenzionali (ad es. 14, E-Prime), mentre le procedure parametriche sono teoricamente più complesse e richiedono un livello più avanzato di capacità di programmazione.

2.3 L'algoritmo staircase

Gli algoritmi Staircase sono le più vecchie procedure adattive psicoacustiche. Possono essere distinte in tre diverse varianti: Il metodo dei limiti (Fechner, 1889)², il Simple Up Down (15, von Békésy, 1947; Dixon and Mood, 1948¹³) ed il Transformed Up Down (16, Wetherill and Levitt, 1965; Levitt, 1971)¹⁴. Gli staircase possono essere usati per stimare sia la soglia di rilevazione che la soglia di discriminazione. Inoltre, possono essere usati con test sì/no e con test nAFC. Dato che gli staircase sono stati usati per decenni, sono stati ampiamente analizzati (e.g.17, Garcia-Perez, 1998, 2002, 2009¹⁶) e se comparati con le procedure moderne (18, Amitay et al., 2006¹⁰; Kollmeier et al., 1988; Marvit et al., 2003)¹⁵, si evince che gli staircase sono ancora un metodo molto affidabile per la stima delle soglie sensoriali.

In alcuni casi gli autori hanno suggerito dei possibili miglioramenti agli staircase (e.g.19, Brown, 1996; Garcia-Perez, 2009¹⁶).

Sfortunatamente però, questi miglioramenti non trovarono un'applicazione pratica o un impatto reale nell'uso laboratoriale di tutti i giorni. Perciò in seguito verranno descritti nei loro usi più comuni in PSYCHOACOUSTICS-WEB

2.4 Il metodo dei limiti

Il metodo dei limiti è stato creato da Fechner (1889)². Per descrivere questo metodo, si consideri il caso in cui si voglia stimare la soglia di discriminazione in frequenza di un tono puro di 1kHz. Ci sono due tipi di stimoli: uno standard e uno variabile. La frequenza dello stimolo standard è fissa. La frequenza della variabile è superiore allo standard di un determinato valore Δf , che viene modificato in modo adattivo durante l'esperimento in funzione della risposta del soggetto.

In ogni prova, gli stimoli standard e variabile sono presentati in ordine casuale e al soggetto viene chiesto di riportare il suono più alto. Ogni volta che la risposta del soggetto è corretta, Δf viene ridotto. Ad una certa prova n , la risposta del soggetto sarà errata poiché Δf sarà al di sotto della sua soglia sensoriale (e perché il soggetto non ha indovinato la risposta giusta). Questo è chiamato "reversal" (letteralmente "inversione") perché è il momento in cui, dopo una serie di risposte corrette se ne registra una sbagliata.

L'inversione è anche il punto in cui abbiamo varcato la soglia del soggetto perché siamo passati dall'ultima risposta positiva alla prima negativa.

La soglia corrispondente alla media tra Δf_n e Δf_{n-1} ; cioè la media tra il livello di stimolazione prima e dopo l'inversione (Figura 3, grafico a sinistra, prova 8-9). Attraverso questo calcolo, il metodo dei limiti restituisce il livello di stimolo corrispondente al 50% della funzione psicometrica. Infatti, il calcolo della soglia viene effettuato con l'ultimo livello restituendo una risposta corretta (cioè il 100% della funzione psicometrica) e con il primo livello restituire una risposta errata (cioè 0% della risposta psicometrica).

Teoricamente, il metodo dei limiti (ma anche il simple e il transformed up-down) può essere eseguito anche nell'ordine opposto; il primo livello della variabile può essere al di sotto della soglia attesa e nelle prove successive il livello della variabile verrebbe aumentato anziché diminuito. Tuttavia, nella psicoacustica, la soglia è raramente avvicinata dal basso.

Se I valori iniziali delle modifiche Δf vengono accuratamente selezionati, il metodo dei limiti è il metodo più veloce e semplice in psicofisica per avvicinarsi alla soglia. In ogni caso, la rapidità del metodo è superata dall'influenza del caso nei compiti nAFC e nell'influenza dei falsi positivi nei test a risposta chiusa (sì/no) (19, Gescheider 2003¹⁶).

Per questi motivi oggi il metodo dei limiti è usato raramente o mai.

2.5 L'algoritmo Simple Up Down

Per risolvere alcuni degli inconvenienti del metodo dei limiti, Premio Nobel von Békésy (1947) propose il simple up-down (vedi anche 20, Dixon Mood 1948¹³).

Il Simple up-down non smette di tracciare la soglia alla prima inversione: quando si verifica la prima inversione, la procedura continua a tracciare la soglia fino a quando non si verifica un numero prefissato di inversioni. Per vederne il funzionamento consideriamo ancora l'esempio di discriminazione di frequenza. Quando il soggetto restituisce la scelta corretta, Δf viene ridotto, e quando il soggetto restituisce un errore nella risposta, registriamo la prima inversione. Tuttavia, contrariamente al metodo dei limiti l'esperimento non si ferma qui e al soggetto viene presentato almeno uno stimolo in più con Δf aumentato. Ad esempio, potremmo presentare di nuovo lo stesso stimolo presentato prima dell'inversione (Figura 1, grafico a destra, prova 9-10). Riassumendo, ogni volta in cui la risposta è corretta Δf viene ridotto mentre ogni volta che la risposta è errata si aumenta Δf .

Analogamente al metodo dei limiti, il metodo simple up-down traccia anche il 50% della funzione psicometrica. Si noti che il metodo dei limiti è in realtà un caso particolare del simple up-down: è un simple up down in cui ci si ferma alla prima inversione.

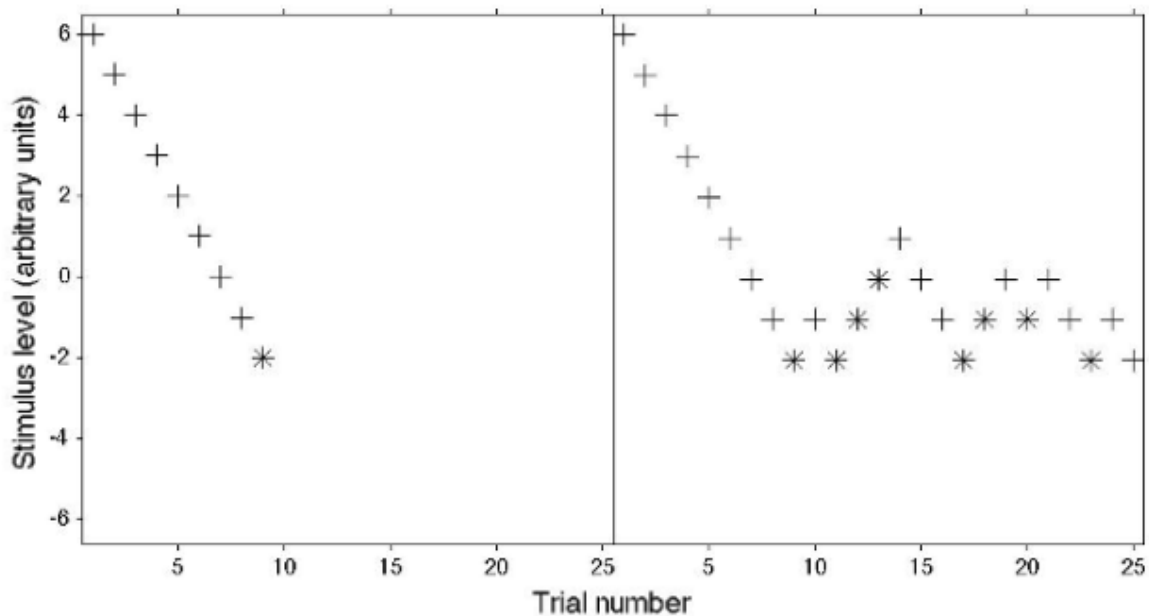


Figura 3 Ipotetica rilevazione della soglia con il metodo dei limiti (a sinistra) e con il simple up down (a destra). Il segno piú rappresenta le risposte corrette mentre l'asterisco quelle errate. Si noti che i tracciamenti della soglia sono identici fino alla prova n. 9.

Entrambi i tracciamenti iniziano con un livello di stimolo 6 e riducono il livello della dimensione del passo di 1 a ciascuna inversione.

2.6 L'algorithmo Transformed Up Down

Il transformed up-down (21, Wetherill e Levitt, 1965; Levitt 1971)¹⁴, può tracciare diversi punti della funzione psicometrica. Questo perché il così detto up and down che permette di tracciare la soglia ne causa anche uno sbilanciamento. Infatti, sia nel metodo dei limiti che nel simple up-down, il cambiamento nel rilevamento della soglia è bilanciato, cioè lo stimolo variabile va verso la soglia dopo una risposta corretta e si allontana dalla soglia dopo una risposta errata. Per questo motivo il simple up down è anche definito come procedura 1-up 1-down. Nel transformed up-down, invece, lo stimolo variabile si sposta verso il basso, verso la soglia per due (o piú) risposte positive mentre sale dopo una sola risposta negativa.

Supponiamo che la probabilità che uno stimolo dia luogo a una risposta positiva sia p .

In questo caso il transformed up down suggerisce di spostarsi verso il basso quando il soggetto restituisce due o piú risposte positive e salire quando produce una risposta negativa.

Pertanto, la probabilità di scendere, verso la soglia diventa p^2 mentre la probabilità di salire lontano dalla soglia è $1-p$ (cioè una sola risposta negativa) o $p(1-p)$, ovvero una risposta positiva seguita da una risposta negativa.

In sintesi:

$$P2 = (1-p) + p(1-p) = 1-p^2 = 0.707$$

Per questo motivo il metodo 2-down 1-up traccia il 70,7% della funzione psicometrica.

Ci sono molte possibili varianti di questo metodo, il più popolare insieme al 2-down 1-up è il 3-down 1-up che traccia il 79,4% della funzione psicometrica (i.e. = 0.794). Va notato che ogni volta che il numero di risposte in discesa aumenta (es. Da 2-down a 3-down), la durata dell'esperimento aumenta perché ogni gruppo di risposte "verso il basso" è allungato di almeno una prova.

I Transformed up-down sono probabilmente le procedure più comunemente utilizzate in psicofisica.

Diventano particolarmente interessanti quando l'algoritmo viene utilizzato in attività a scelta forzata. Ad esempio, se adottiamo un'attività 2AFC la performance minima che dobbiamo aspettarci è del 50% perché quando il livello dello stimolo è estremamente basso il soggetto può ancora indovinare per caso la risposta corretta, tuttavia poiché il transformed up-down traccia i punti più alti della funzione psicometrica (es. 70,7% o 79,4%) il ricercatore può ancora raccogliere una buona misura sensoriale che non risente molto del caso.

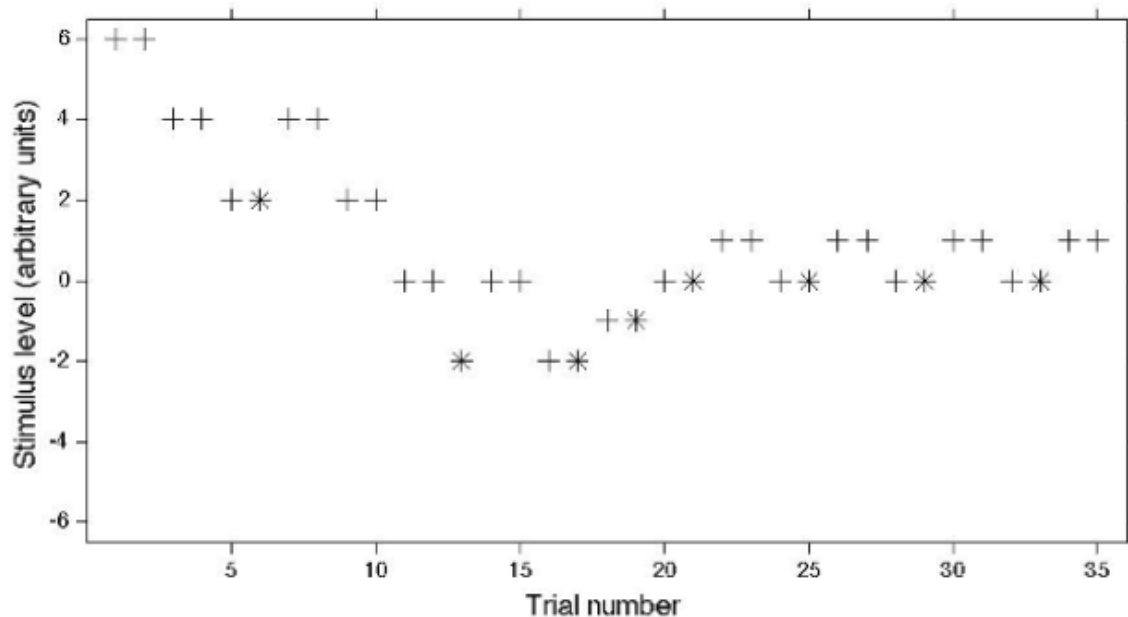


Figura 4. Ipotetico rilevamento della soglia con il transformed up down. Il segno più rappresenta le risposte corrette mentre l'asterisco rappresenta le risposte errate. Il livello di stimolo iniziale è 6. Il numero di inversioni è dodici. I primi quattro reversals vengono eseguiti con una dimensione del passo di 1 e gli otto successivi vengono eseguiti con una dimensione del passo di 0,5. Nota come la regola trasformata allunga il rilevamento della

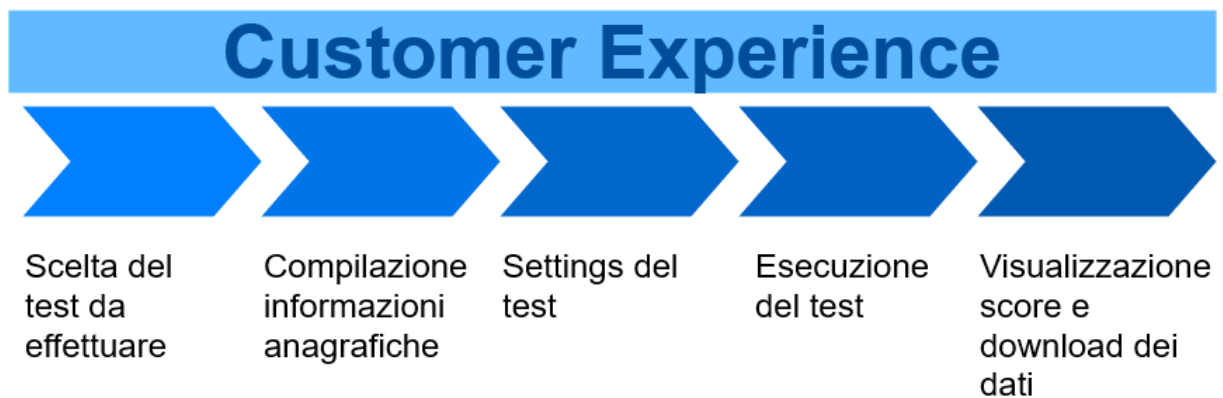
soglia rispetto al metodo dei limiti o al simple up-down.

Capitolo 3

Analisi dei requisiti

Nel metodo di sviluppo agile l'analisi dei requisiti viene continuamente raffinata mediante delle review con il committente:

di seguito si riporta lo schema customer experience ottenuta dall'analisi dei requisiti dopo l'ultima review.



3.1 Anteprima del progetto

La funzione principale della web app è la somministrazione di diversi tipi di test in ambito psicoacustico:

Di seguito si riporta la schermata di settings della web app:

The screenshot shows a settings interface for an experiment. It is titled "Set the characteristics of the experiment".

- Set the characteristics of the standard tone:** Amplitude: -20 dB, Frequency: 1000 Hz, Duration: 500 ms.
- Set the characteristics of the experiment:** n. of blocks: 3, nAFC: 2, ITI: 1000 ms, ISI: 500 ms, Delta: 12 dB.
- Set the characteristics of the staircase:** Radio buttons for SimpleUpDown, TwoDownOneUp (selected), and ThreeDownOneUp. Checkboxes for Noise (unchecked) and FeedBack (checked). Input fields for First factor: 2, Second factor: 1.414, First reversals: 4, Second reversals: 8, and Reversal threshold: 8.

At the bottom, there are two large red buttons: "BACK" and "START".

- **Amplitude**
Definisce l'ampiezza massima del suono creato, misurata in decibel
- **Frequency**
Definisce la frequenza del suono, misurata in Hertz
- **Duration**
Definisce la durata del suono, misurata in millisecondi
- **n. of blocks**
Definisce il numero di volte per cui viene ripetuto l'intero blocco di test
- **nAFC**
Definisce il numero di suoni che vengono proposti come possibile scelta, i diversi suoni sono sempre due ma quello errato viene ripetuto nAFC-1 volte
- **ITI**

Acronimo di Inter-Trial Interval, stabilisce il tempo intercorso tra la risposta dell'utente e la riproduzione dei nuovi suoni

- ISI

Acronimo di Inter-Stimulus Interval, stabilisce il tempo intercorso tra i suoni riprodotti

- Delta

Definisce la differenza iniziale tra il suono corretto e quello errato

- First factor

Fattore moltiplicativo che verrà utilizzato per i primi reversals

- First reversals

Numero che identifica quanti sono i reversals che useranno il first factor

- Second factor

Fattore moltiplicativo che verrà utilizzato per gli ultimi reversals

- Second reversals

Numero che identifica quanti sono i reversals che useranno il second factor

- Reversal threshold

Specifica il numero di reversals che vengono impiegati per il calcolo del threshold di ogni blocco di test

- Algoritmo

Specifica il criterio di avanzamento del test

È inoltre a disposizione dei ricercatori una funzionalità che permette il download dei dati dei test in formato CSV in modo da facilitare la creazione di statistiche e la raccolta di informazioni per futuri esperimenti aggiuntivi.

3.2 Test

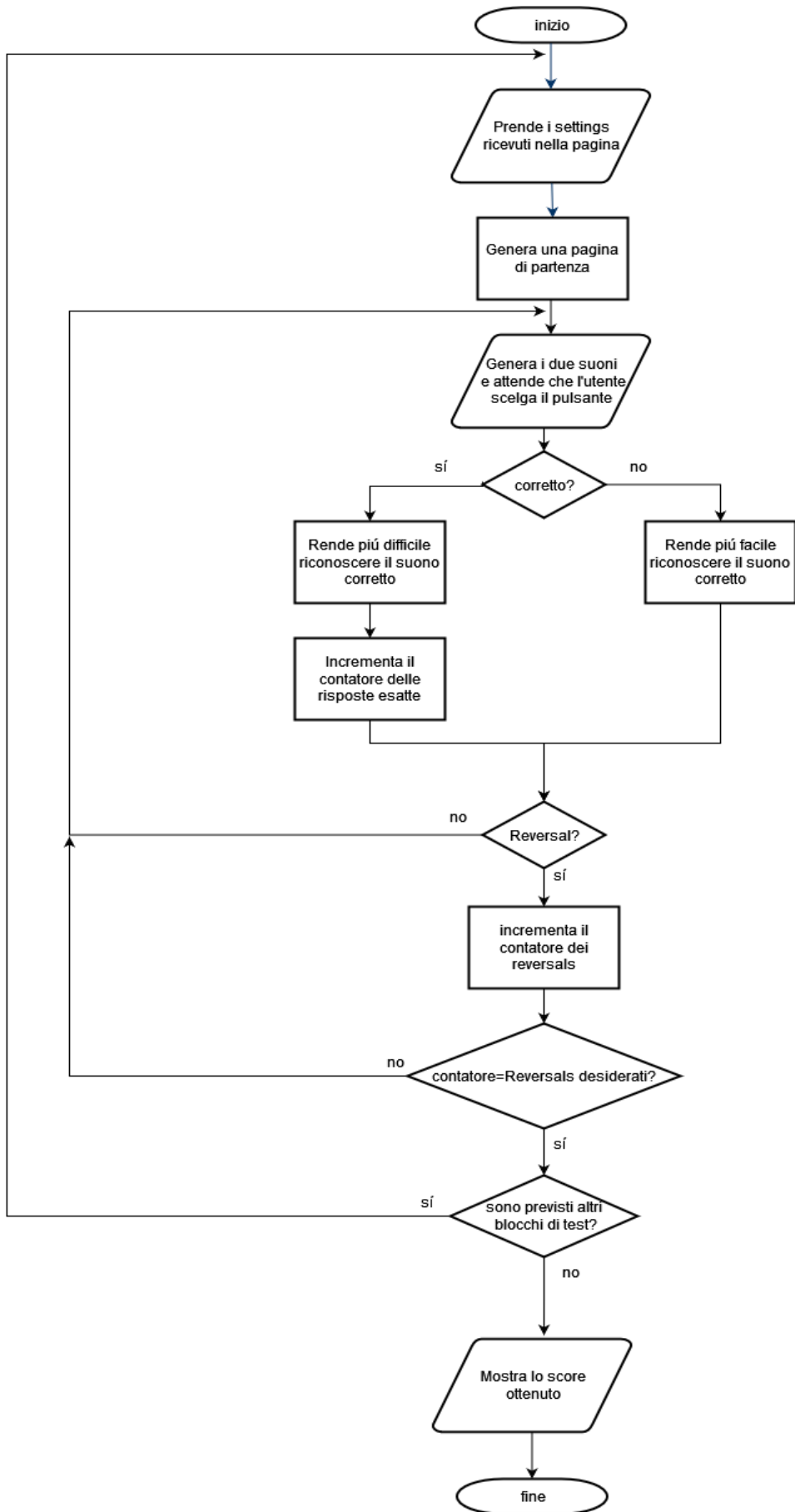
Tutti i test propongono una serie di suoni che si differenziano per una specifica caratteristica che può essere intensità, frequenza o durata del suono.

Ogni test può essere impostato dal suo creatore (che non sempre corrisponde all'utente che esegue il test) secondo una serie di parametri della letteratura psicoacustica.

Attualmente i test disponibili sono:

- Pure tone intensity discrimination
- Pure tone frequency discrimination
- Pure tone duration discrimination

In figura un flow chart semplificato del funzionamento dei test:



3.2.1 Pure tone intensity discrimination

Il test pure tone intensity discrimination serve a studiare la sensibilità dell'utente alle variazioni di intensità del suono.

All'inizio del test si valuta la differenza tra valore del suono variabile e valore del suono standard

$$\Delta = VarAmp - StdAmp$$

A questo punto si generano i suoni definiti dalle rispettive ampiezze VarAmp ed StdAmp e poi li si riproduce con un tempo di attesa tra i due determinato dal valore di ISI.

Si osserva che nAFC è il numero di suoni che vengono usati per il test e che anche se il numero è maggiore di due, il numero di suoni generati sarà sempre lo stesso ma il suono standard verrà riprodotto nAFC-1 volte, ciò serve per aumentare maggiormente la difficoltà del test e il livello di attenzione per dare le risposte.

Si attende poi che l'utente scelga quale dei due suoni è più intenso.

Una volta ricevuta la risposta:

- Se la risposta è corretta: si modifica l'ampiezza variabile in questo modo:

$$VarAmp = StdAmp + \frac{\Delta}{Factor}$$

In modo da rendere più difficile la distinzione dei suoni

- Se la risposta è errata: si modifica l'ampiezza variabile in questo modo:

$$VarAmp = StdAmp + \Delta * Factor$$

In modo da semplificare la differenza dei due suoni

Successivamente si ripete il procedimento di generazione e riproduzione dei suoni fino all'ottenimento del numero desiderato di reversals.

Si osserva che a volte possono essere utilizzati due fattori diversi per i reversals iniziali e quelli finali, questo perché i dati più importanti vengono dati dai reversals finali, cioè quelli in cui l'utente fatica maggiormente a capire quale suono sia quello giusto.

Possono essere inoltre settati più blocchi di test, si ripete cioè l'intero test per più volte, per avere una maggiore certezza dei risultati ottenuti e minimizzare l'impatto delle risposte date a caso, il numero di ripetizioni del test è dato dalla variabile number of blocks mentre il tempo che intercorre tra un test e il successivo è dato dal valore di ITI.

Una volta terminati tutti i blocchi verrà mostrato a video il valore della soglia all'intensità del suono dell'utente e data la possibilità di salvare i dati generati durante il test

3.2.2 Pure tone frequency discrimination

Il test pure tone frequency discrimination serve a studiare la sensibilità dell'utente alle variazioni di frequenza del suono.

All'inizio del test si valuta la differenza tra valore del suono variabile e valore del suono standard

$$\Delta = VarFreq - StdFreq$$

A questo punto si generano i suoni definiti dalle rispettive frequenze VarFreq ed StdFreq e poi li si riproduce con un tempo di attesa tra i due determinato dal valore di ISI.

Si osserva che nAFC è il numero di suoni che vengono usati per il test e che anche se il numero è maggiore di due, il numero di suoni generati sarà sempre lo stesso ma il suono standard verrà riprodotto nAFC-1 volte, ciò serve per aumentare maggiormente la difficoltà del test e il livello di attenzione per dare le risposte.

Si attende poi che l'utente scelga quale dei due suoni è più acuto.

Una volta ricevuta la risposta:

- Se la risposta è corretta: si modifica la frequenza variabile in questo modo:

$$VarFreq = StdFreq + \frac{\Delta}{Factor}$$

In modo da rendere più difficile la distinzione dei suoni

- Se la risposta è errata: si modifica la frequenza variabile in questo modo:

$$VarFreq = StdFreq + \Delta * Factor$$

In modo da semplificare la differenza dei due suoni

Successivamente si ripete il procedimento di generazione e riproduzione dei suoni fino all'ottenimento del numero desiderato di reversals.

Si osserva che a volte possono essere utilizzati due fattori diversi per i reversals iniziali e quelli finali, questo perché i dati più importanti vengono dati dai reversals finali, cioè quelli in cui l'utente fatica maggiormente a capire quale suono sia quello giusto.

Possono essere inoltre settati più blocchi di test, si ripete cioè l'intero test per più volte, per avere una maggiore certezza dei risultati ottenuti e minimizzare l'impatto delle risposte date a caso, il numero di ripetizioni del test è dato dalla variabile number of blocks mentre il tempo che intercorre tra un test e il successivo è dato dal valore di ITI.

Una volta terminati tutti i blocchi verrà mostrato a video il valore della soglia alla frequenza del suono dell'utente e data la possibilità di salvare i dati generati durante il test

3.2.3 Pure tone duration discrimination

Il test pure tone duration discrimination serve a studiare la sensibilità dell'utente alle variazioni di durata del suono.

All'inizio del test si valuta la differenza tra valore del suono variabile e valore del suono standard

$$Delta = VarDur - StdDur$$

A questo punto si generano i suoni definiti dalle rispettive durate $VarDur$ ed $StdDur$ e poi li si riproduce con un tempo di attesa tra i due determinato dal valore di ISI.

Si osserva che $nAFC$ è il numero di suoni che vengono usati per il test e che anche se il numero è maggiore di due, il numero di suoni generati sarà sempre lo stesso ma il suono standard verrà riprodotto $nAFC-1$ volte, ciò serve per aumentare maggiormente la difficoltà del test e il livello di attenzione per dare le risposte.

Si attende poi che l'utente scelga quale dei due suoni è più lungo.

Una volta ricevuta la risposta:

- Se la risposta è corretta: si modifica la durata variabile in questo modo:

$$VarDur = StdDur + \frac{Delta}{Factor}$$

In modo da rendere più difficile la distinzione dei suoni

- Se la risposta è errata: si modifica la durata variabile in questo modo:

$$VarDur = StdDur + Delta * Factor$$

In modo da semplificare la differenza dei due suoni

Successivamente si ripete il procedimento di generazione e riproduzione dei suoni fino all'ottenimento del numero desiderato di reversals.

Si osserva che a volte possono essere utilizzati due fattori diversi per i reversals iniziali e quelli finali, questo perché i dati più importanti vengono dati dai reversals finali, cioè quelli in cui l'utente fatica maggiormente a capire quale suono sia quello giusto.

Possono essere inoltre settati più blocchi di test, si ripete cioè l'intero test per più volte, per avere una maggiore certezza dei risultati ottenuti e minimizzare l'impatto delle risposte date a caso, il numero di ripetizioni del test è dato dalla variabile number of blocks mentre il tempo che intercorre tra un test e il successivo è dato dal valore di ITI.

Una volta terminati tutti i blocchi verrà mostrato a video il valore della soglia alla durata del suono dell'utente e data la possibilità di salvare i dati generati durante il test

3.2.4 Algoritmi per calcolo score:

La web app implementa diversi algoritmi per lo svolgimento dei test che funzionano sfruttando i reversals:

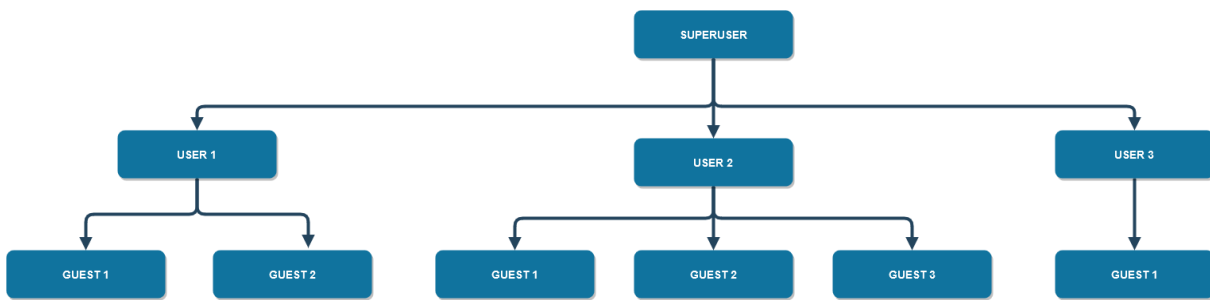
- Simple Up Down
Funziona esattamente come descritto dalla letteratura psicoacustica
- Two Down One Up
È un caso particolare di transformed up down, che usa due risposte esatte per aumentare la difficoltà e una risposta errata per diminuirla
- Three Down One Up
- È un caso particolare di transformed up down, che usa tre risposte esatte per aumentare la difficoltà e una risposta errata per diminuirla

3.3 Gestione utenti

Sono previsti tre tipi di utenti con privilegi crescenti:

- Guest
Usa il programma occasionalmente, generalmente è il soggetto a cui viene sottoposto l'esperimento
- Account Owner
Possiede un account, somministra test ma può utilizzare e scaricare solo i dati delle persone a cui ha sottoposto l'esperimento
- Superuser
Come l'account owner ma può utilizzare e scaricare tutti i dati del database, può essere nominato solo da un altro superuser, è il massimo livello di autorizzazione possibile e generalmente viene dato a ricercatori o personale che necessita di grandi quantità di dati

Per maggiore chiarezza si mostra l'organigramma della gestione degli utenti dove si mostra l'accesso ai dati di ogni livello



Capitolo 4

Sviluppo della web application

4.1 Framework VS Scripting standalone

Dopo un'attenta valutazione di vantaggi e svantaggi si è deciso di non usare framework, in quanto le funzionalità non sono così avanzate e/o complesse:

La web app è single task, non necessitiamo di un selector engine (utilizzato per interrogare il dominio sulla presenza di tag specifici nel codice e.g. \$('div') cerca e trova tutti i tag 'div' presenti nella pagina, sfruttando il jQuery selector engine) e non vogliamo che la pagina sia troppo pesante per il browser.

Ma soprattutto durante lo sviluppo è stato possibile imparare a sviluppare sul web sfruttando tutti i vari linguaggi utilizzati e ampliando così il bagaglio di conoscenze.

4.2 Sviluppo Front End

Si è deciso di iniziare lo sviluppo con il Front-End per facilitare il committente nell'immaginazione di tutte le pagine desiderate, questo si è rivelato particolarmente efficace in quanto si è ridotto il tempo necessario alle review iniziali poiché i cambiamenti erano estremamente evidenti e le modifiche potevano essere effettuati 'live' durante le call, rendendo possibile avere un'elevata flessibilità

Per lo sviluppo Front-End sono stati usati i linguaggi HTML e CSS accoppiati a Bootstrap, che è stato un grande aiuto per velocizzare il processo di creazione delle pagine ed evitare errori banali durante lo sviluppo (e.g. errori nei tag, classi, id ecc...)

4.3 Sviluppo Back End

Il linguaggio principale usato per il back end lato client è Javascript, per la creazione dei suoni è stata usata la API Web Audio, che ha permesso di sfruttare delle funzionalità a volte anche complesse nella sintetizzazione dei suoni.

4.3.1 La API Web Audio

Le API Web Audio sono uno strumento avanzato per la sintetizzazione dei suoni e, in generale creare applicazioni audio complesse.

IMPORTANTE: Come è facile verificare dal sito <https://caniuse.com/?search=web-audio-api> le API hanno una buona diffusione: solo Internet Explorer (compresa la versione 11) e Opera Mini non le supportano.

Queste API colmano un vuoto che non poteva essere certo coperto dal tag HTML 5 *audio*, che consente principalmente di riprodurre un brano musicale in streaming: l'intento di questo strumento, del quale possiamo leggere la versione attuale delle specifiche in <https://webaudio.github.io/web-audio-api/>, è quello di generare suoni partendo da zero oppure manipolando file audio esistenti, o una sorgente sonora tramite l'ingresso di linea. Il concetto alla base di web audio è il collegamento di oggetti denominati `AudioNode` che fondamentalmente possono essere di tre tipi:

- `Source node` - generano il suono, sono dotati di uscite ma non di ingressi
- Nodi di trasferimento - trasferiscono l'audio verso la destinazione finale, tipicamente l'hardware audio
- Nodi per manipolazione del suono - per esempio filtri, sono dotati di ingressi ed uscite

Il percorso del suono attraverso questi nodi si chiama `Routing` e può essere riassunto in tre step molto simili alla generazione di suoni reali dagli strumenti musicali:

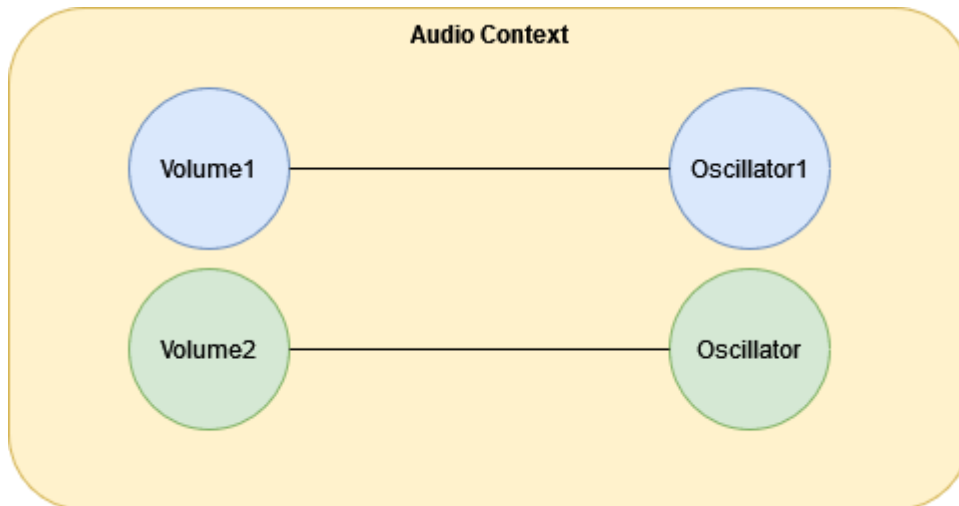
- Sintetizzazione del suono
- Manipolazione del suono
- Invio al sistema di riproduzione

La web app utilizza solo alcune delle funzionalità fornite dall'API in quanto il generatore di suoni per i test non necessita di funzionalità particolarmente complesse.

Innanzitutto, viene creato il cosiddetto ambiente `Audio Context` dentro il quale è possibile creare gli elementi necessari alla sintetizzazione dei suoni, sfruttiamo poi il nodo `Gain`, un nodo che permette di avere un guadagno in ampiezza del suono e lo chiamiamo volume successivamente si crea il nodo `oscillator` dove si imposta la frequenza e la durata del suono che si vuole sintetizzare, si collegano i due nodi mediante la funzione `connect()` e si riproduce il suono sintetizzato.

Per semplicità si mostra un grafico che mostra la struttura usata per creazione dei suoni

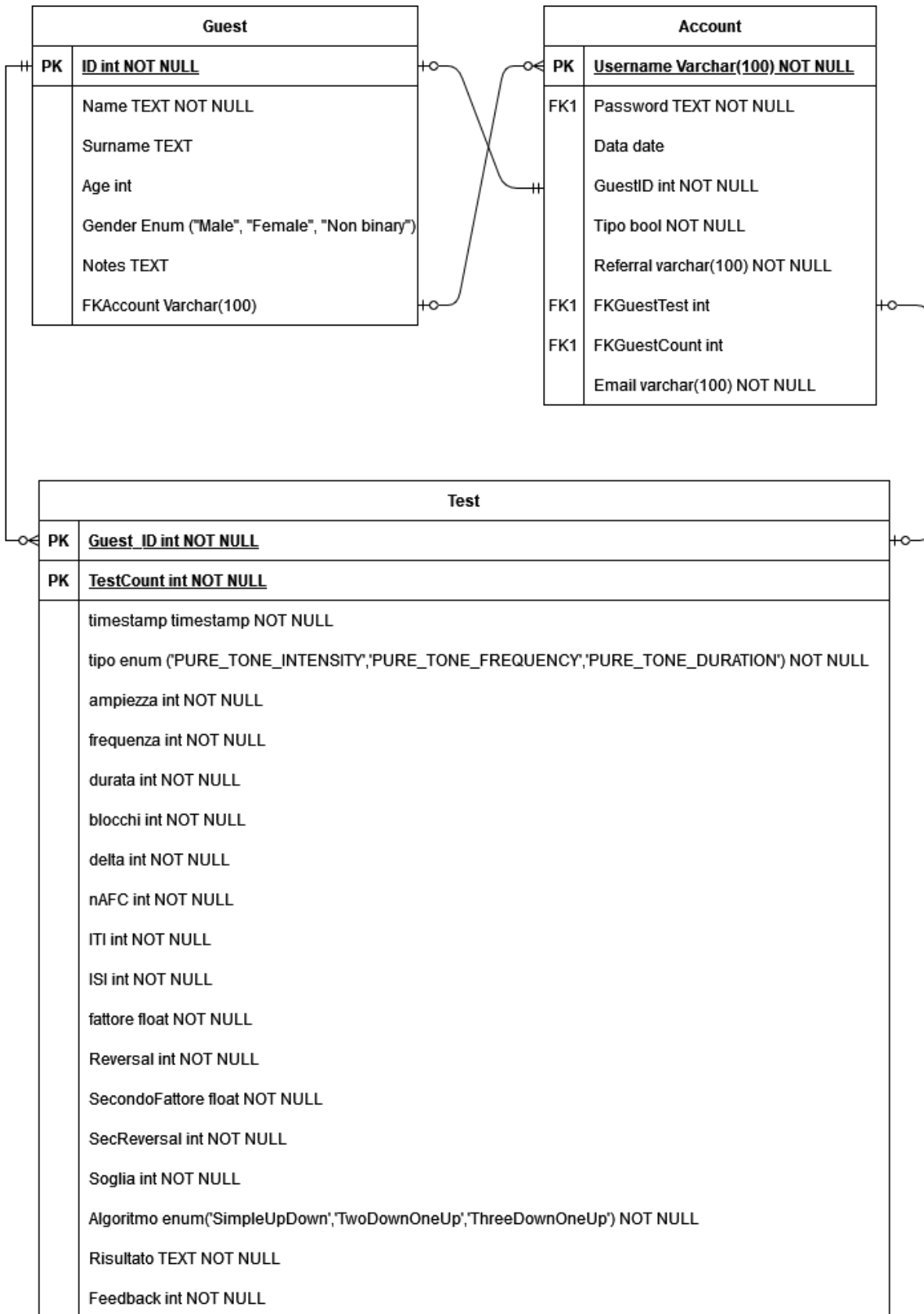
NOTA: Il codice verrà incorporato in appendice



Sono stati creati due oscillatori diversi in quanto la API web audio funziona solo come evento dopo aver premuto un pulsante, rendendo impossibile avere cicli che continuano a riprodurre suoni senza eventi che li generano. Per ovviare questo problema sono stati creati due diversi oscillatori e guadagni che lavorano in parallelo, e generano i suoni in sequenze diverse che non si sovrappongono, il risultato finale è come quello che genererebbe un unico oscillatore che genera più suoni.

4.4 Connessione e comunicazione al database

Il back end lato client è stato sviluppato attraverso il linguaggio PHP, è stata la parte finale del progetto, in quanto una volta ottenuta la conferma del prof Grassi sulla validità del front end e verificata la sua soddisfazione abbiamo potuto lavorare per rendere responsive il sito e iniziare la fase di creazione del database. In questo modo si è potuto avere completa libertà nello sviluppo in quanto bisognava solo implementare le funzionalità precedentemente identificate dal front end.

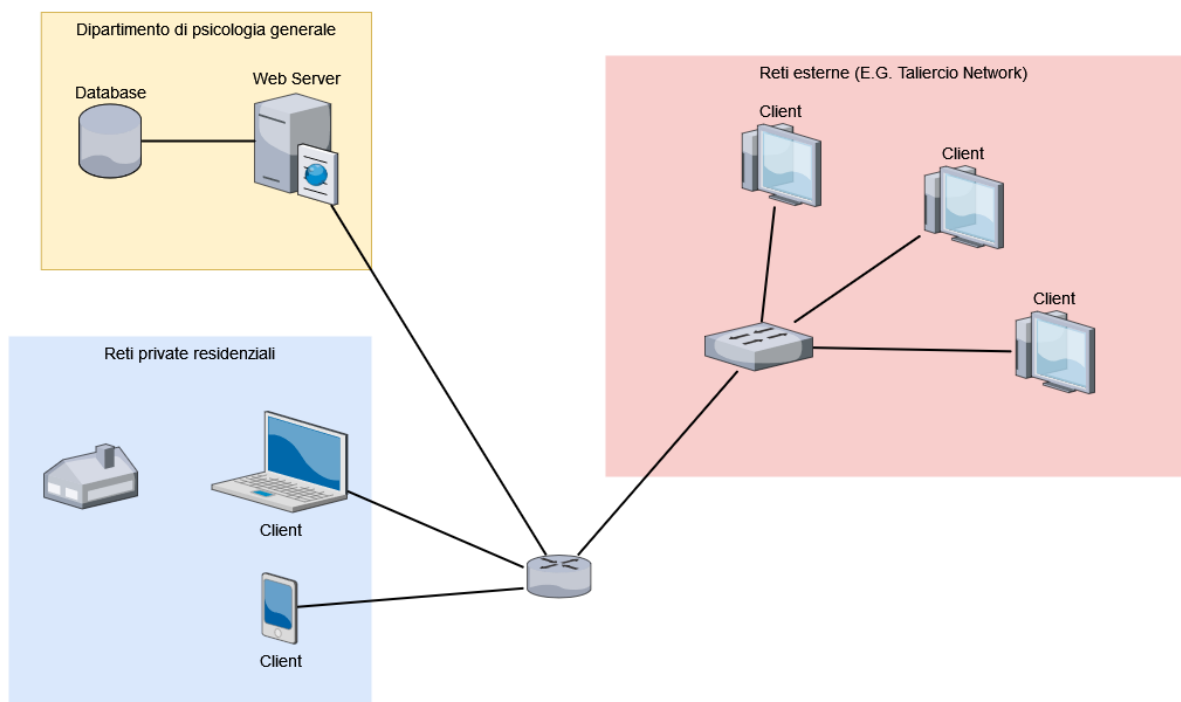


La web app sfrutta inoltre un server SMTP (Simple Mail Transfer Protocol) che gestisce le operazioni di cambio delle password dell'account.

Ciò è molto utile poiché in fase di user testing e beta testing i futuri gestori del sito non dovranno cambiare e informare manualmente le persone dell'avvenuto cambio della password dell'account, sollevandoli da un lavoro gravoso e ripetitivo.

Il dominio ci è stato gentilmente fornito dal dipartimento di psicologia generale, il sito è reperibile al seguente indirizzo: <http://psychoacoustics.dpg.psy.unipd.it>

Schema architetturale Psychoacoustics



È stato implementato un sistema anti sql-injection per tutti gli input presenti, così da avere un buon grado di sicurezza lato server.

È inoltre in previsione la richiesta di https, così da inserire un captcha prima dell'accesso al sito, in modo da evitare attacchi mirati al riempimento del database con file inutili o casuali

Capitolo 5

Conclusioni e possibili sviluppi futuri

Gli obiettivi posti dall'analisi dei requisiti sono stati ottimamente raggiunti, la web app permette di utilizzare tutti i test principali richiesti per lo studio in psicoacustica e sfrutta tutti gli algoritmi minimi necessari ai ricercatori nel calcolo dello score.

La web app è stata ideata sin dal principio con il proposito di essere migliorata in futuro e contiene quindi codice ben commentato, comprensibile e riutilizzabile in futuro così da semplificare il lavoro di comprensione di chiunque voglia ampliarne le funzionalità e contemporaneamente permettere di sfruttare il codice già scritto e testato.

In futuro, invece, opportunità di miglioramento è data dall'aggiunta di nuove funzionalità e test. In previsione ci sono test con rumori, generazione di suoni mediante l'uso di vettori e implementazioni di altri algoritmi per il calcolo della soglia sensoriale.

Per quanto riguarda la sicurezza della web app si consiglia l'inserimento di un captcha prima dell'accesso alla pagina così da evitare problemi dovuti a bot e di inserire un sistema per il criptaggio delle password (nel metodo post di php)

«Tra vent'anni non sarete delusi delle cose che avete fatto ma da quelle che non avete fatto.

Allora levate l'ancora, abbandonate i porti sicuri, catturate il vento nelle vostre vele.

Esplorate. Sognate. Scoprite.»

(Mark Twain)

Bibliografia

- ¹ Watson, A. B., and Fitzhugh, A. (1990). “The method of constant stimuli is inefficient”
- ² Fechner, G. T. (1889)
- ³ Faes, L., Nollo, G., Ravelli, F., Ricci, L., Vescovi, M., Turatto, M., Pavani, F. and Antolini, R. (2007). “Small-sample characterization of stochastic approximation staircases in forced-choice adaptive threshold estimation”
- ⁴ Green, D. M., & Swets, J. A. (1966). Signal detection theory and psychophysics.
- ⁵ Green, D. M. (1993). “A maximum-likelihood method for estimating thresholds in a yes-no task”
- ⁶ Stanislaw, H., & Todorov, N. (1999). “Calculation of signal detection theory measures.”
- ⁷ Wichmann, F. A., and Hill, N. J. (2001). “The psychometric function: I. Fitting, sampling, and goodness of fit.”
- ⁸ Saberi, K., & Green, D. M. (1997). “Evaluation of maximum-likelihood estimators in nonintensive auditory psychophysics.”
- ⁹ Green, D. M. (1990). “Stimulus selection in adaptive psychophysical procedures”
- ¹⁰ Amitay, S., Irwin, A., Hawkey, D. J. Cowan, J. A., and Moore, D. R. (2006). “A comparison of adaptive procedures for rapid and reliable threshold assessment and training in naïve listeners,” J.
- ¹¹ Baker, R. J., & Rosen, S. (1998). “Minimizing the boredom by maximising likelihood – Efficient estimation of masked threshold.”
- ¹² Leek, M. R. (2001). “Adaptive procedures in psychophysical research”
- ¹³ Dixon, J. W., and Mood A. M. (1948). “A Method for Obtaining and Analyzing Sensitivity Data”
- ¹⁴ Levitt, H. (1971). “Transformed up–down methods in psychoacoustics”
- ¹⁵ Marvit, P., Florentine, M., and Buus, S. (2003). “A comparison of psychophysical procedures for level-discrimination thresholds,” J. Acoust. Soc. Am.
- ¹⁶ García-Pérez, M. A. (2009). “Denoising forced-choice detection data”
- ¹⁷ Gescheider, G. A. (2003). Psychophysics: the fundamentals (Lawrence Erlbaum Associates, Hillsdale).

Ringraziamenti

Ringrazio il professor Mauro Migliardi per l'opportunità concessa nel fare una tesi di questa dimensione, portandomi ad avere maggiore conoscenza ed esperienza nella progettazione e sviluppo web e il professor Massimo Grassi per l'occasione che mi è stata offerta nella partecipazione ad un progetto così rilevante nel campo della psicoacustica.

Appendice

ampstest.php

```
<!doctype html>
<html lang="en">
  <head>
    <?php session_start(); ?>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="icon" type="image/x-icon" href="files/logo.png">

    <!-- Bootstrap CSS -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
    <link rel="stylesheet" href="css/test.css<?php if
(isset($_SESSION['version'])) echo "?{$_SESSION['version']}"; ?>">

    <title>Psychoacoustics-web - Amplitude test</title>

    <script>
      // pass info from php session to js
      var amp = parseFloat(<?php echo $_SESSION["amplitude"]; ?>);
      var freq = parseFloat(<?php echo $_SESSION["frequency"]; ?>);
      var dur = parseFloat(<?php echo $_SESSION["duration"]; ?>);
      var mod = parseFloat(<?php echo $_SESSION["modulation"]; ?>);
      //var phase = <?php echo $_SESSION["phase"]; ?>;
      var blocks = parseInt(<?php echo $_SESSION["blocks"]; ?>);
      var delta = parseFloat(<?php echo $_SESSION["delta"]; ?>);
      var nAFC = parseInt(<?php echo $_SESSION["nAFC"]; ?>);
      var ITI = parseInt(<?php echo $_SESSION["ITI"]; ?>);
      var ISI = parseInt(<?php echo $_SESSION["ISI"]; ?>);
```

```

    var feedback = <?php echo $_SESSION["checkFb"]; ?>;
    var saveSettings = <?php echo $_SESSION["saveSettings"]; ?>;
    var factor = parseFloat(<?php echo $_SESSION["factor"]; ?>);
    var secondFactor = parseFloat(<?php echo $_SESSION["secFactor"];
?>);

    var reversals = parseInt(<?php echo $_SESSION["reversals"]; ?>);
    var secondReversals = parseInt(<?php echo
$_SESSION["secReversals"]; ?>);
    var reversalThreshold = parseInt(<?php echo
$_SESSION["threshold"]; ?>);
    var algorithm = <?php echo "{$_SESSION["algorithm"]}"; ?>;
    var currentBlock = parseInt(<?php
if(isset($_SESSION["currentBlock"])) echo $_SESSION["currentBlock"]+1; else
echo "1"?>);

</script>
<script type="text/javascript" src="js/soundsAmplitude.js<?php if
(isset($_SESSION['version'])) echo "{$_SESSION['version']}"; ?>"
defer></script>
</head>

<body>
    <!-- Option 1: Bootstrap Bundle with Popper -->
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min
.js" integrity="sha384-
ka7Sk0GlIn4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENB00LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>

    <div class="window" id="StartingWindow">
        <h2>Ready?</h2>
        <button type="button" class="btn btn-success" id="start"
onclick="start()">Let's start!</button>
    </div>

    <form action="" id="PlayForm">
        <h1>Which is the loudest tone?</h1>
        <button type="button" class="btn btn-success" id="button1" onclick
= "select(1)" disabled>1° sound</button>
        <button type="button" class="btn btn-danger" id="button2" onclick
= "select(2)" disabled>2° sound</button>
        <?php
            $colors = ["#198754", "#dc3545", "#0d6efd", "#e0b000",
"#a000a0", "#ff8010", "#50a0f0", "#703000", "#606090"];
            for($i = 3; $i<=intval($_SESSION['nAFC']); $i++){

```

```

        echo "<button type='button' class='btn btn-success'
style='background-color: ".$colors[($i-1)%count($colors)]."' id='button{$i}'
onclick = 'select({$i})' disabled>{$i}° sound</button>";
    }
    ?>
</form>
<div class='alert alert-danger' id="wrong">Wrong!</div>
<div class='alert alert-success' id="correct">Correct!</div>
</body>
</html>

```

demographicData.php

```

<!doctype html>
<html lang="en">
  <head>

    <?php
      include "php/config.php";
      session_start();
      if(!isset($_GET["test"]) && !isset($_GET['ref']))
        header("Location: index.php");
    ?>

    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="icon" type="image/x-icon" href="files/logo.png">

    <!-- Bootstrap CSS -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfSpd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
    <link rel="stylesheet" href="css/staircaseStyle.css<?php if
(isset($_SESSION['version'])) echo "?{"$_SESSION['version']}"; ?>">
    <script type="text/javascript" src="js/funzioni.js<?php if
(isset($_SESSION['version'])) echo "?{"$_SESSION['version']}"; ?>"></script>

    <title>Psychoacoustics-web - Personal info</title>

    <script>
      function verifyRef(){
        var display = true;
        var logged = <?php if(isset($_SESSION['usr'])) echo "true";
else echo "false"; ?>;

```

```

        if(document.getElementById("ref").value != "" && logged){
            if(document.getElementById("name").innerHTML.slice(-
1)!="*")
                document.getElementById("name").innerHTML += "*";
            display = true;
        }else if(document.getElementById("ref").value == "" &&
logged){
            display = false;
        }
        updatePage(display);
    }

    function insertRef(){
        <?php
            if(isset($_SESSION['usr'])){
                try{
                    $conn = new mysqli($host, $user, $password,
$dbname);

                    if ($conn->connect_errno)
                        throw new Exception('DB connection failed');
                    mysqli_set_charset($conn, "utf8");

                    $sql="SELECT Referral as ref FROM account WHERE
Username='{$_SESSION['usr']}'";

                    $result = $conn->query($sql);
                    $row = $result->fetch_assoc();
                }catch(Exception $e){
                    header("Location: index.php?err=db");
                }
            }
            ?>
            document.getElementById("ref").value=<?php echo
"". $row['ref']. "" ; ?>;
            verifyRef();
        }
    </script>
</head>
<body onload="verifyRef()">

    <?php
        //se si sceglie un username già esistente verrà messo "?err=1"
nell'url
        if(isset($_GET['err'])){
            if($_GET['err']==0)
                echo "<div class='alert alert-danger'>Some inserted
characters aren't allowed</div>";
            if($_GET['err']==1)

```

```

        echo "<div class='alert alert-danger'>The name field is
required</div>";
        if($_GET['err']==2)
            echo "<div class='alert alert-danger'>The name field is
required when using a referral code</div>";
        else if($_GET['err']==3)
            echo "<div class='alert alert-danger'>Invalid referral
code</div>";
    }
?>

<div class="container p-4" style="margin-top:10%" >
    <div class="row gx-4">
        <div class="col">
            <div class=" p-3 border bg-light">
                <form name="staircase" method="post"
action="php/personalInfoValidation.php<?php
                    if(isset($_GET["test"]))
                        echo "?test=".$_GET["test"];
                    ?>">
                    <!-- Contenuto dello slot, qui vanno inseriti tutti i
bottoni e i check box del primo slot -->

                    <h1>Personal Informations</h1>

                    <!-- Label -->
                    <div class="input-group flex-nowrap
conditionalDisplay">
                        <span class="input-group-text" id="name"
>Name<?php if(!isset($_SESSION['usr'])) echo "*"; ?></span>
                        <input type="text" class="form-control"
id="inputName" placeholder="Name" name="name">
                    </div>

                    <div class="input-group flex-nowrap
conditionalDisplay">
                        <span class="input-group-text"
id="surname" >Surname</span>
                        <input type="text" class="form-control"
id="inputSurname" placeholder="Surname" name="surname">
                    </div>

                    <div class="input-group flex-nowrap
conditionalDisplay">
                        <span class="input-group-text" id="age" >Age</span>

```



```

        <input type="text" class="form-control"
id="inputAge" placeholder="Age" name="age">
    </div>

    <div class="input-group flex-nowrap
conditionalDisplay">
        <span class="input-group-text" id="gender"
>Gender</span>
        <select name='gender' class="form-select">
            <option disabled="disabled" selected value="null"
id="NullGender">Select your gender</option>
            <?php
                try{
                    $conn = new mysqli($host, $user,
$password, $dbname);

                    if ($conn->connect_errno)
                        throw new Exception('DB connection
failed');

                    mysqli_set_charset($conn, "utf8");

                    $sql="SELECT COLUMN_TYPE AS ct FROM
INFORMATION_SCHEMA.COLUMNS WHERE TABLE_SCHEMA = 'psychoacoustics_db' AND
TABLE_NAME = 'guest' AND COLUMN_NAME = 'gender'";
                    $result=$conn->query($sql);
                    $row=$result->fetch_assoc();//questa query
da un risultato di tipo enum('Male','Female','Non-Binary')

                    //metto i valori in un array
                    $values = substr($row['ct'], 5, -
1);//tolgo "enum(" e ")"
                    $values = str_replace("'", "",
$values);//tolgo gli apici
                    $list = explode(",", $values);//divido in
una lista in base alle virgole

                    //creo un'opzione per ogni possibile
valore
                    foreach($list as $elem)
                        echo "<option
value='".strtoupper($elem)."'>".strtoupper($elem)."</option>";
                }catch(Exception $e){
                    header("Location: index.php?err=db");
                }
            ?>
        </select>
    </div>

```

```

        <div class="input-group flex-nowrap
conditionalDisplay" id="notesDiv">
            <span class="input-group-text"
id="notes">Notes</span>
            <input type="text" class="form-control"
id="inputNotes" placeholder="Notes" name="notes">
        </div>

        <?php if(isset($_SESSION['usr'])) echo '<button
type="button" class="btn btn-primary btn-lg m-3 refButton"
onclick="insertRef()">USE MINE</button>'; ?>
        <div class="input-group flex-nowrap referral">
            <span class="input-group-text" id="notes">Invite
code</span>
            <input type="text" class="form-control" id="ref"
name="ref" onkeyup="verifyRef()" value="<?php if(isset($_GET['ref'])) {echo
$_GET['ref'];}?>">
        </div>

        <div class="form-check">
            <input class="form-check-input" type="checkbox"
value="" id="flexCheckDefault" name="checkSave" checked>
            <label class="form-check-label"
for="flexCheckDefault">Save results</label>
        </div>

        <button type="button" class="btn btn-primary btn-lg m-
3" onclick = "location.href='index.php'">BACK</button>
        <button type="submit" class="btn btn-primary btn-lg m-
3" >NEXT</button>

    </form>
</div>
</div>
</div>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min
.js" integrity="sha384-
MrCw6ZMFY1zcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
</body>
</html>

```

Durtest.php

```

<!doctype html>
<html lang="en">
<head>

```

```

<?php session_start(); ?>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" type="image/x-icon" href="files/logo.png">

<!-- Bootstrap CSS -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
  <link rel="stylesheet" href="css/test.css<?php if
(isset($_SESSION['version'])) echo "{$_SESSION['version']}"; ?>">

<title>Psychoacoustics-web - Duration test</title>

<script>
  // pass info from php session to js
  var amp = parseFloat(<?php echo $_SESSION["amplitude"]; ?>);
  var freq = parseFloat(<?php echo $_SESSION["frequency"]; ?>);
  var dur = parseFloat(<?php echo $_SESSION["duration"]; ?>);
  var mod = parseFloat(<?php echo $_SESSION["modulation"]; ?>);
  //var phase = <?php echo $_SESSION["phase"]; ?>;
  var blocks = parseInt(<?php echo $_SESSION["blocks"]; ?>);
  var delta = parseFloat(<?php echo $_SESSION["delta"]; ?>);
  var nAFC = parseInt(<?php echo $_SESSION["nAFC"]; ?>);
  var ITI = parseInt(<?php echo $_SESSION["ITI"]; ?>);
  var ISI = parseInt(<?php echo $_SESSION["ISI"]; ?>);
  var feedback = <?php echo $_SESSION["checkFb"]; ?>;
  var saveSettings = <?php echo $_SESSION["saveSettings"]; ?>;
  var factor = parseFloat(<?php echo $_SESSION["factor"]; ?>);
  var secondFactor = parseFloat(<?php echo $_SESSION["secFactor"];
?>);

  var reversals = parseInt(<?php echo $_SESSION["reversals"]; ?>);
  var secondReversals = parseInt(<?php echo
$_SESSION["secReversals"]; ?>);
  var reversalThreshold = parseInt(<?php echo
$_SESSION["threshold"]; ?>);
  var algorithm = <?php echo "{$_SESSION["algorithm"]}"; ?>;
  var currentBlock = parseInt(<?php
if(isset($_SESSION["currentBlock"])) echo $_SESSION["currentBlock"]+1; else
echo "1"?>);

</script>
<script type="text/javascript" src="js/soundsDuration.js<?php if
(isset($_SESSION['version'])) echo "{$_SESSION['version']}"; ?>"
defer></script>

```

```

</head>

<body>
  <!-- Option 1: Bootstrap Bundle with Popper -->
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min
.js" integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENB00LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>

  <div class="window" id="StartingWindow">
    <h2>Ready?</h2>
    <button type="button" class="btn btn-success" id="start"
onclick="start()">Let's start!</button>
  </div>

  <form action="" id="PlayForm">
    <H1>Which is the longest tone?</H1>
    <button type="button" class="btn btn-success" id="button1" onclick
= "select(1)" disabled>1° sound</button>
    <button type="button" class="btn btn-danger" id="button2" onclick
= "select(2)" disabled>2° sound</button>
    <?php
        $colors = ["#198754", "#dc3545", "#0d6efd", "#e0b000",
"#a000a0", "#ff8010", "#50a0f0", "#703000", "#606090"];
        for($i = 3; $i<=intval($_SESSION['nAFC']); $i++){
            echo "<button type='button' class='btn btn-success'
style='background-color: ".$colors[($i-1)%count($colors)].";' id='button{$i}'
onclick = 'select({$i})' disabled>{$i}° sound</button>";
        }
    ?>
  </form>
  <div class='alert alert-danger' id="wrong">Wrong!</div>
  <div class='alert alert-success' id="correct">Correct!</div>
</body>
</html>

```

Freptest.php

```

<!doctype html>
<html lang="en">
  <head>
    <?php session_start(); ?>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="icon" type="image/x-icon" href="files/logo.png">

```

```

<!-- Bootstrap CSS -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
<link rel="stylesheet" href="css/test.css<?php if
(isset($_SESSION['version'])) echo "{$_SESSION['version']}"; ?>">

<title>Psychoacoustics-web - Frequency test</title>

<script>
// pass info from php session to js
var amp = parseFloat(<?php echo $_SESSION["amplitude"]; ?>);
var freq = parseFloat(<?php echo $_SESSION["frequency"]; ?>);
var dur = parseFloat(<?php echo $_SESSION["duration"]; ?>);
var mod = parseFloat(<?php echo $_SESSION["modulation"]; ?>);
//var phase = <?php echo $_SESSION["phase"]; ?>;
var blocks = parseInt(<?php echo $_SESSION["blocks"]; ?>);
var delta = parseFloat(<?php echo $_SESSION["delta"]; ?>);
var nAFC = parseInt(<?php echo $_SESSION["nAFC"]; ?>);
var ITI = parseInt(<?php echo $_SESSION["ITI"]; ?>);
var ISI = parseInt(<?php echo $_SESSION["ISI"]; ?>);
var feedback = <?php echo $_SESSION["checkFb"]; ?>;
//var noise = <?php echo $_SESSION["checkNoise"]; ?>;
var saveSettings = <?php echo $_SESSION["saveSettings"]; ?>;
var factor = parseFloat(<?php echo $_SESSION["factor"]; ?>);
var secondFactor = parseFloat(<?php echo $_SESSION["secFactor"];
?>);

var reversals = parseInt(<?php echo $_SESSION["reversals"]; ?>);
var secondReversals = parseInt(<?php echo
$_SESSION["secReversals"]; ?>);
var reversalThreshold = parseInt(<?php echo
$_SESSION["threshold"]; ?>);
var algorithm = <?php echo "{$_SESSION["algorithm"]}"; ?>;
var currentBlock = parseInt(<?php
if(isset($_SESSION["currentBlock"])) echo $_SESSION["currentBlock"]+1; else
echo "1"?>);

</script>
<script type="text/javascript" src="js/soundsFrequency.js<?php if
(isset($_SESSION['version'])) echo "{$_SESSION['version']}"; ?>"
defer></script>
</head>

<body>
<!-- Option 1: Bootstrap Bundle with Popper -->

```

```

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min
.js" integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENB00LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>

    <div class="window" id="StartingWindow">
        <h2>Ready?</h2>
        <button type="button" class="btn btn-success" id="start"
onclick="start()">Let's start!</button>
    </div>

    <form action="" id="PlayForm">
        <H1>Which is the highest pitch tone?</H1>
        <button type="button" class="btn btn-success" id="button1" onclick
= "select(1)" disabled>1° sound</button>
        <button type="button" class="btn btn-danger" id="button2" onclick
= "select(2)" disabled>2° sound</button>
        <?php
            $colors = ["#198754", "#dc3545", "#0d6efd", "#e0b000",
"#a000a0", "#ff8010", "#50a0f0", "#703000", "#606090"];
            for($i = 3; $i<=intval($_SESSION['nAFC']); $i++){
                echo "<button type='button' class='btn btn-success'
style='background-color: ".$colors[(($i-1)%count($colors)].";' id='button{$i}'
onclick = 'select({$i})' disabled>{$i}° sound</button>";
            }
        ?>
    </form>
    <div class='alert alert-danger' id="wrong">Wrong!</div>
    <div class='alert alert-success' id="correct">Correct!</div>
</body>
</html>

```

Gapstest.php

```

<!doctype html>
<html lang="en">
    <head>
        <?php session_start(); ?>
        <!-- Required meta tags -->
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link rel="icon" type="image/x-icon" href="files/logo.png">

        <!-- Bootstrap CSS -->
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-

```

```

1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
    <link rel="stylesheet" href="css/test.css<?php if
(isset($_SESSION['version'])) echo "?{$_SESSION['version']}"; ?>">

    <title>Psychoacoustics-web - Amplitude test</title>

    <script>
        // pass info from php session to js
        var amp = parseFloat(<?php echo $_SESSION["amplitude"]; ?>);
        var freq = parseFloat(<?php echo $_SESSION["frequency"]; ?>);
        var dur = parseFloat(<?php echo $_SESSION["duration"]; ?>);
        var mod = parseFloat(<?php echo $_SESSION["modulation"]; ?>);
        //var phase = <?php echo $_SESSION["phase"]; ?>;
        var blocks = parseInt(<?php echo $_SESSION["blocks"]; ?>);
        var delta = parseFloat(<?php echo $_SESSION["delta"]; ?>);
        var nAFC = parseInt(<?php echo $_SESSION["nAFC"]; ?>);
        var ITI = parseInt(<?php echo $_SESSION["ITI"]; ?>);
        var ISI = parseInt(<?php echo $_SESSION["ISI"]; ?>);
        var feedback = <?php echo $_SESSION["checkFb"]; ?>;
        var saveSettings = <?php echo $_SESSION["saveSettings"]; ?>;
        var factor = parseFloat(<?php echo $_SESSION["factor"]; ?>);
        var secondFactor = parseFloat(<?php echo $_SESSION["secFactor"];
?>);

        var reversals = parseInt(<?php echo $_SESSION["reversals"]; ?>);
        var secondReversals = parseInt(<?php echo
$_SESSION["secReversals"]; ?>);
        var reversalThreshold = parseInt(<?php echo
$_SESSION["threshold"]; ?>);
        var algorithm = <?php echo "'{$_SESSION["algorithm"]}"; ?>;
        var currentBlock = parseInt(<?php
if(isset($_SESSION["currentBlock"])) echo $_SESSION["currentBlock"]+1; else
echo "1"?>);

    </script>
    <script type="text/javascript" src="js/noisesGap.js<?php if
(isset($_SESSION['version'])) echo "?{$_SESSION['version']}"; ?>"
defer></script>
</head>

<body>
    <!-- Option 1: Bootstrap Bundle with Popper -->
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min
.js" integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>

```

```

    <div class="window" id="StartingWindow">
        <h2>Ready?</h2>
        <button type="button" class="btn btn-success" id="start"
onclick="start()">Let's start!</button>
    </div>

    <form action="" id="PlayForm">
        <H1>Which is the noise with the gap?</H1>
        <button type="button" class="btn btn-success" id="button1" onclick
= "select(1)" disabled>1° sound</button>
        <button type="button" class="btn btn-danger" id="button2" onclick
= "select(2)" disabled>2° sound</button>
        <?php
            $colors = ["#198754", "#dc3545", "#0d6efd", "#e0b000",
"#a000a0", "#ff8010", "#50a0f0", "#703000", "#606090"];
            for($i = 3; $i<=intval($_SESSION['nAFC']); $i++){
                echo "<button type='button' class='btn btn-success'
style='background-color: ".$colors[(($i-1)%count($colors)].";' id='button{$i}'
onclick = 'select({$i})' disabled>{$i}° sound</button>";
            }
        ?>
    </form>
    <div class='alert alert-danger' id="wrong">Wrong!</div>
    <div class='alert alert-success' id="correct">Correct!</div>
</body>
</html>

```

Index.php

```

<!doctype html>
<html lang="en">
    <head>
        <?php
            session_start();
            $_SESSION['version'] = 4; //change this number to force cache
update
        ?>

        <!-- Required meta tags -->
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link rel="icon" type="image/x-icon" href="files/logo.png">

        <!-- Bootstrap CSS -->
        <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-

```



```

EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWFspD65VohhpUuCOmLASjC"
crossorigin="anonymous">
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">

  <link rel="stylesheet" href="css/style.css<?php if
(isset($_SESSION['version'])) echo "?".$_SESSION['version']; ?>">
  <script type="text/javascript" src="js/funzioni.js<?php if
(isset($_SESSION['version'])) echo "?".$_SESSION['version']; ?>"></script>

  <title>Psychoacoustics-web</title>

</head>

<body>
  
  <!-- Barra navigazione -->
  <nav class="navbar navbar-dark bg-dark">
    <div class="container-fluid" >
      <a class="navbar-brand" href="index.php" >
        
          PSYCHOACOUSTICS-WEB

      </a>
      <form class="container-fluid logButtons">
        <?php
          if(!isset($_SESSION["usr"])){
            if(isset($_SESSION["idGuest"]))
              unset($_SESSION["idGuest"]);
            echo "<button class='btn btn-outline-danger \"
type='button\" onclick='location.href='register.php'\" >Sign Up</button>";
            echo "<button class='btn btn-outline-success me-2\"
type='button\" onclick='location.href='login.php'\">Log In</button>";
          }else{
            echo "<label class='welcomeMessage'>Welcome
".$_SESSION['usr'].</label>";
            echo "<a class='settings' href='userSettings.php'><i
class='material-icons rotate'>settings</i></a>";
            echo "<button class='btn btn-outline-primary
yourTests\" type='button\" onclick='location.href='yourTests.php'\">Your
tests</button>";
            echo "<button class='btn btn-outline-danger logout\"
type='button\" onclick='location.href='php/logout.php'\">Log Out</button>";
          }
        ?>
      </form>

```

```

        </div>
    </nav>

    <?php
    if(isset($_GET['err'])){
        if($_GET['err']==1)
            echo "<div class='alert alert-danger'>Access denied, attempt
logged</div>";
        if($_GET['err']==2)
            echo "<div class='alert alert-danger'>Something went
wrong</div>";
        if($_GET['err']=="db")
            echo "<div class='alert alert-danger'>Something went wrong
while trying to connect to the database, please contact an
administator</div>";
    }
    ?>
    <!-- Descrizione e presentazione -->

    <div class="container-fluid">

        <h3 class="display-3 descriptionTitle">
            Welcome to PSYCHOACOUSTICS-WEB
        </h3>
        <h5 class="display-8 description">
            PSYCHOACOUSTICS-WEB is a web developed tool to measure
auditory sensory thresholds for a
            variety of classic tasks. You can run each test as a guest or
you can create your personal
            account and costumize the toolbox for your own research.
Please refer to the <a href="files/PSYCHOACOUSTICS-WEB_manual.pdf">instruction
manual</a> for further info on how to use the toolbox.
            <br><br>
            The tasks estimate the intensity, frequency and duration
discrimination threshold for a pure tone.
            In the tasks you can set up the characteristics of the tone as
well as
            the characteristics of the adaptive staircase. The tests
implement the
            following adaptive staircase algorithms: simple up-down, 2-
down 1-up,
            and 3-down 1-up. Please refer to Levitt (JASA, 1971) for more
info on
            these adaptive staircases.
        </h5>
    </div>

    <!-- cards -->

```

```

    <div class="cards">
      <div class="card">
        <div class="card-body"
onclick="location.href='demographicData.php?test=amp'">
          <h5 class="card-title" >Pure tone intensity
discrimination</h5>
          <p>Click here to run the test</p>
        </div>
      </div>

      <div class="card">
        <div class="card-body"
onclick="location.href='demographicData.php?test=freq'">
          <h5 class="card-title">Pure tone frequency
discrimination</h5>
          <p>Click here to run the test</p>
        </div>
      </div>

      <div class="card">
        <div class="card-body"
onclick="location.href='demographicData.php?test=dur'">
          <h5 class="card-title">Pure tone duration
discrimination</h5>
          <p>Click here to run the test</p>
        </div>
      </div>

      <div class="card">
        <div class="card-body"
onclick="location.href='demographicData.php?test=gap'">
          <h5 class="card-title">White noise gap detection</h5>
          <p>Click here to run the test</p>
        </div>
      </div>

      <div class="card">
        <div class="card-body"
onclick="location.href='demographicData.php?test=ndur'">
          <h5 class="card-title">White noise duration
discrimination</h5>
          <p>Click here to run the test</p>
        </div>
      </div>
    </div>

```

```

        <!-- Option 1: Bootstrap Bundle with Popper -->
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min
.js" integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
    </body>
</html>

```

Info.php

```

<!doctype html>
<html lang="en">
<head>
    <?php
    session_start();
    include "php/config.php";
    if (!isset($_GET["test"]) && !isset($_SESSION['test']))
        header("Location: index.php");
    ?>

    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="icon" type="image/x-icon" href="files/logo.png">

    <!-- Bootstrap CSS -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet"
        integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWFspD3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">

    <link rel="stylesheet"
        href="css/staircaseStyle.css<?php if (isset($_SESSION['version']))
echo "?{$_SESSION['version']}"; ?>">
    <title>Psychoacoustics-web - Test settings</title>

    <?php
    try {
        $conn = new mysqli($host, $user, $password, $dbname);
        if ($conn->connect_errno)
            throw new Exception('DB connection failed');
        mysqli_set_charset($conn, "utf8");

        $sql = "SELECT Type, Amplitude as amp, Frequency as freq, Duration as
dur, Modulation as modu, ISI, blocks, Delta, nAFC,

```

```

        Factor as fact, Reversal as rev, SecFactor as secfact,
SecReversal as secrev,
        Threshold as thr, Algorithm as alg

        FROM test

        WHERE Guest_ID='{$_SESSION['test']['guest']}' AND
Test_count='{$_SESSION['test']['count']}'";
    $result = $conn->query($sql);
    $row = $result->fetch_assoc();

    if ($row['Type'] == 'PURE_TONE_INTENSITY')
        $type = "amplitude";
    else if ($row['Type'] == 'PURE_TONE_FREQUENCY')
        $type = "frequency";
    else if ($row['Type'] == 'PURE_TONE_DURATION')
        $type = "duration";
    else if ($row['Type'] == 'WHITE_NOISE_GAP')
        $type = "gap";
    else if ($row['Type'] == 'WHITE_NOISE_DURATION')
        $type = "nduration";
} catch (Exception $e) {
    header("Location: index.php?err=db");
}
?>
<script>
    var type = "<?php echo $type; ?>";
    var amp = parseFloat(<?php echo $row["amp"]; ?>);
    var freq = parseFloat(<?php echo $row["freq"]; ?>);
    var dur = parseFloat(<?php echo $row["dur"]; ?>);
    var mod = parseFloat(<?php echo $row["modu"]; ?>);
    var delta = parseFloat(<?php echo $row["Delta"]; ?>);
    var ISI = parseInt(<?php echo $row["ISI"]; ?>);
    var nAFC = parseInt(<?php echo $row["nAFC"]; ?>);
</script>

<script type="text/javascript"
    src="js/testPreview.js<?php if (isset($_SESSION['version'])) echo
"?{$_SESSION['version']}"; ?>"
    defer></script>
</head>
<body>


<div class="info container">
    <h2 class="info title">Hi <?php echo $_SESSION['name']; ?></h2>
    <p class="info text">

```

You will now do an acoustic test that will measure your sensibility to the `<?php echo $type; ?>` of a sound.

```
<br><br>
```

During the test you will be asked a series of questions. In each question you will

hear `<?php echo $row['nAFC']; ?>` sounds and will have to choose which of them was the

```
<?php
if ($type == 'amplitude')
    echo "loudest";
else if ($type == 'frequency')
    echo "highest pitch";
else if ($type == 'duration')
    echo "longest";
?>.
```

```
<br><br>
```

```
<?php
if ($row['nAFC'] > 2)
    echo "Only o";
else
    echo "O";
?>
```

ne of the sounds will be

```
<?php
if ($type == 'amplitude')
    echo "louder";
else if ($type == 'frequency')
    echo "higher pitch";
else if ($type == 'duration')
    echo "longer";
?>
```

than the other

```
<?php
if ($row['nAFC'] > 2)
    echo "s";
?>.
```

It will have a

```
<?php
if ($type == 'amplitude')
    echo "n intensity of " . (floatval($row['amp']) +
floatval($row['Delta'])) . " dB";
else if ($type == 'frequency')
    echo " frequency of " . (floatval($row['freq']) +
floatval($row['Delta'])) . " Hz";
else if ($type == 'duration')
    echo " duration of " . (floatval($row['dur']) +
floatval($row['Delta'])) . " ms";
?>.
```

```
<?php
while the other
if ($row['nAFC'] > 2)
    echo "s";
```

```

?>
will have a<?php
if ($type == 'amplitude')
    echo "n intensity of " . floatval($row['amp']) . " dB (the maximum
amplitude is 0dB, which corresponds to
        '1' in decimal, while a negative number in decibel
corresponds to a number near '0' in decimal
        (the higher is the absolute value, the nearest is
to 0));
else if ($type == 'frequency')
    echo " frequency of " . floatval($row['freq']) . " Hz";
else if ($type == 'duration')
    echo " duration of " . floatval($row['dur']) . " ms";
?>.

```


The number of questions that will be asked depends on the number of reversals that was set.

Every time you give a correct answer and then a wrong answer or vice versa, there will be a reversal (the correctness direction changes).

In this case there will be <?php echo \$row['rev']; ?> reversal<?php if (\$row['rev'] > 1) echo "s"; ?> using a factor of

```

<?php echo $row['fact']; ?> and then <?php echo $row['secrev']; ?>
reversal<?php if ($row['secrev'] > 1) echo "s"; ?> using a factor of
<?php echo $row['secfact']; ?>.

```


But what are these factors?

Every time you give

```

<?php

```

```

if ($row['alg'] == 'SimpleUpDown')
    echo "1";
else if ($row['alg'] == 'TwoDownOneUp')
    echo "2";
else if ($row['alg'] == 'ThreeDownOneUp')
    echo "3";

```

```

?>

```

```

right answer<?php

```

```

if ($row['alg'] != 'SimpleUpDown')
    echo "s";

```

```

?>

```

or a wrong answer, the difficulty changes. The right answers make the test harder, reducing the difference

between the

correct sound and the wrong one, the wrong answers make it easier, increasing the difference.


```

    The test will be repeated <?php echo $row['blocks']; ?> time<?php if
($row['blocks'] > 1) echo "s"; ?>. Each
    time you will see your threshold,
    a number that summarizes your level of sensibility. It is calculated
only using the
    last <?php echo $row['thr']; ?>
    reversal<?php if ($row['rev'] > 1) echo "s"; ?>.
    <br>
    At the end you can download a csv file that summarizes all the tests
done (and also a file with the details of
    each test,
    if you are logged in with an account).
    <br><br>
    <?php if ($checkFb) echo " You will receive a feedback to advice
whether the response is correct or not."; ?>
</p>
    <div class="info test">
        <div class="info test-title">
            <h5>Test preview</h5>
            <button id="playTest" class="btn btn-light"
onclick="random()">Start test preview</button>
        </div>
        <div class="info test-preview">
            <form action="" class="info test-preview-form" id="PlayForm">
                <h1><?php
                    if ($type == 'amplitude')
                        echo "Which is the loudest tone?";
                    else if ($type == "frequency")
                        echo "Which is the highest pitch tone?";
                    else if ($type == "duration")
                        echo "Which is the longest tone?";
                    else if ($type == "gap")
                        echo "Which is the noise with the gap?";
                    else if ($type == "nduration")
                        echo "Which is the longest noise?";
                    ?></h1>
                <button type="button" class="btn btn-success" id="button1"
onclick="select(1)" disabled>1° sound
                </button>
                <button type="button" class="btn btn-danger" id="button2"
onclick="select(2)" disabled>2° sound</button>
                <?php
                    $colors = ["#198754", "#dc3545", "#0d6efd", "#e0b000",
"#a000a0", "#ff8010", "#50a0f0", "#703000", "#606090"];
                    for ($i = 3; $i <= intval($row['nAFC']); $i++) {
                        echo "<button type='button' class='btn btn-success'
style='background-color:" . $colors[($i - 1) % count($colors)] . "; border-

```



```

color:" . $colors[($i - 1) % count($colors)] . "' id='button{$i}' onclick =
'select({$i})' disabled>{$i}° sound</button>";
    }
    ?>
</form>
<div class='alert' id="alert"></div>
</div>
</div>
<form action="php/soundSettingsValidation.php" name="Settings"
method="post">

    <button type="button" class="btn btn-primary btn-lg m-3
soundSettingsButton"
        onclick="location.href='demographicData.php'">BACK
    </button>
    <button type="submit" class="btn btn-primary btn-lg m-3
soundSettingsButton">START</button>
</form>
</div>
</body>
</html>

```

Login.php

```

<!DOCTYPE html>
<html>
    <head>
        <?php session_start(); ?>
        <!-- Required meta tags -->
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">

        <title>Psychoacoustics-web - Login</title>
        <link rel="icon" type="image/x-icon" href="files/logo.png">

        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
        <link rel="stylesheet" href="css/formStyle.css<?php if
(isset($_SESSION['version'])) echo "{$_SESSION['version']}"; ?>">
    </head>
    <body>
        <?php
            //se si sceglie un username già esistente verrà messo "?err=1"
nell'url
            if(isset($_GET['err'])){

```

```

        if ($_GET['err']==0)
            echo "<div class='alert alert-danger'>Some inserted
characters aren't allowed</div>";
        if ($_GET['err']==1)
            echo "<div class='alert alert-danger'>Incorrect username
or password</div>";
    }
    ?>
    <form method="post" action="php/log.php">
        <h1>Login</h1>
        <div class="input-group mb-3">
            <span class="input-group-text" id="basic-
addon1">Username</span>
            <input type="text" class="form-control" placeholder="Username"
aria-label="Username" aria-describedby="basic-addon1" required name="usr">
        </div>
        <div class="input-group mb-3">
            <span class="input-group-text" id="basic-
addon1">Password</span>
            <input type="password" class="form-control"
placeholder="Password" aria-label="Username" aria-describedby="basic-addon1"
required name="psw">
        </div>

        <!--input type="password" id="password" placeholder="Password"
name="password"-->
        <button type="submit" class="btn btn-primary btn-lg m-1">Log
in</button>
    </form>
</body>
</html>

```

Durtest.php

```

<!doctype html>
<html lang="en">
    <head>
        <?php session_start(); ?>
        <!-- Required meta tags -->
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link rel="icon" type="image/x-icon" href="files/logo.png">

        <!-- Bootstrap CSS -->
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-

```

```

1BmE4kWbq78iYhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
    <link rel="stylesheet" href="css/test.css<?php if
(isset($_SESSION['version'])) echo "{$_SESSION['version']}"; ?>">

    <title>Psychoacoustics-web - Duration test</title>

    <script>
        // pass info from php session to js
        var amp = parseFloat(<?php echo $_SESSION["amplitude"]; ?>);
        var freq = parseFloat(<?php echo $_SESSION["frequency"]; ?>);
        var dur = parseFloat(<?php echo $_SESSION["duration"]; ?>);
        var mod = parseFloat(<?php echo $_SESSION["modulation"]; ?>);
        //var phase = <?php echo $_SESSION["phase"]; ?>;
        var blocks = parseInt(<?php echo $_SESSION["blocks"]; ?>);
        var delta = parseFloat(<?php echo $_SESSION["delta"]; ?>);
        var nAFC = parseInt(<?php echo $_SESSION["nAFC"]; ?>);
        var ITI = parseInt(<?php echo $_SESSION["ITI"]; ?>);
        var ISI = parseInt(<?php echo $_SESSION["ISI"]; ?>);
        var feedback = <?php echo $_SESSION["checkFb"]; ?>;
        var saveSettings = <?php echo $_SESSION["saveSettings"]; ?>;
        var factor = parseFloat(<?php echo $_SESSION["factor"]; ?>);
        var secondFactor = parseFloat(<?php echo $_SESSION["secFactor"];
?>);

        var reversals = parseInt(<?php echo $_SESSION["reversals"]; ?>);
        var secondReversals = parseInt(<?php echo
$_SESSION["secReversals"]; ?>);
        var reversalThreshold = parseInt(<?php echo
$_SESSION["threshold"]; ?>);
        var algorithm = <?php echo "'{$_SESSION["algorithm"]}"; ?>;
        var currentBlock = parseInt(<?php
if(isset($_SESSION["currentBlock"])) echo $_SESSION["currentBlock"]+1; else
echo "1"?>);

    </script>
    <script type="text/javascript" src="js/noisesDuration.js<?php if
(isset($_SESSION['version'])) echo "{$_SESSION['version']}"; ?>"
defer></script>
</head>

<body>
    <!-- Option 1: Bootstrap Bundle with Popper -->
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min
.js" integrity="sha384-
ka7Sk0Gln4gmtz2MlQnik11wXgYs0g+OMhuP+IlRH9sENB00LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>

```

```

    <div class="window" id="StartingWindow">
        <h2>Ready?</h2>
        <button type="button" class="btn btn-success" id="start"
onclick="start()">Let's start!</button>
    </div>

    <form action="" id="PlayForm">
        <H1>Which is the longest noise?</H1>
        <button type="button" class="btn btn-success" id="button1" onclick
= "select(1)" disabled>1° sound</button>
        <button type="button" class="btn btn-danger" id="button2" onclick
= "select(2)" disabled>2° sound</button>
        <?php
            $colors = ["#198754", "#dc3545", "#0d6efd", "#e0b000",
"#a000a0", "#ff8010", "#50a0f0", "#703000", "#606090"];
            for($i = 3; $i<=intval($_SESSION['nAFC']); $i++){
                echo "<button type='button' class='btn btn-success'
style='background-color: ".$colors[($i-1)%count($colors)].";' id='button{$i}'
onclick = 'select({$i})' disabled>{$i}° sound</button>";
            }
        ?>
    </form>
    <div class='alert alert-danger' id="wrong">Wrong!</div>
    <div class='alert alert-success' id="correct">Correct!</div>
</body>
</html>

```

Register.php

```

<!DOCTYPE html>
<html>
    <head>
        <?php
            include "php/config.php";
            session_start()
        ?>
        <!-- Required meta tags -->
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">

        <link rel="icon" type="image/x-icon" href="files/logo.png">
        <title>Psychoacoustics-web - Register</title>

        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">

```

```

    <link rel="stylesheet" href="css/formStyle.css<?php if
(isset($_SESSION['version'])) echo "?{"$_SESSION['version']}"; ?>">
</head>
<body>
    <?php
        //se si sceglie un username già esistente verrà messo "?err=1"
nell'url
        if(isset($_GET['err'])){
            if ($_GET['err']==0)
                echo "<div class='alert alert-danger'>Some inserted
characters aren't allowed</div>";
            if ($_GET['err']==1)
                echo "<div class='alert alert-danger'>Username already
taken</div>";
        }
    ?>
    <form method="post" action="php/registering.php"
onsubmit="validation()">
        <h1>Registrazione</h1>
        <div class="input-group mb-3">
            <span class="input-group-text" id="basic-
addon1">Username*</span>
            <input type="text" class="form-control" placeholder="Username"
required name="usr">
        </div>
        <div class="input-group mb-3">
            <span class="input-group-text" id="basic-
addon1">Password*</span>
            <input type="password" class="form-control"
placeholder="Password" required name="psw">
        </div>
        <div class="input-group mb-3">
            <span class="input-group-text" id="basic-addon1">Email*</span>
            <input type="text" class="form-control" placeholder="Email"
required name="email">
        </div>
        <div class="input-group mb-3">
            <span class="input-group-text" id="basic-addon1">Name*</span>
            <input type="text" class="form-control" placeholder="Name"
required name="name">
        </div>
        <div class="input-group mb-3">
            <span class="input-group-text" id="basic-
addon1">Surname</span>
            <input type="text" class="form-control" placeholder="Surname"
name="surname">
        </div>
        <div class="input-group mb-3">

```

```

        <span class="input-group-text" id="basic-addon1">Birth
date</span>
        <input type="date" class="form-control" name="date">
    </div>
    <select name='gender' class="form-select">
        <option disabled="disabled" selected value="null"
id="NullGender">Select your gender</option>
        <?php
            try{
                $conn = new mysqli($host, $user, $password, $dbname);
                if ($conn->connect_errno)
                    throw new Exception('DB connection failed');
                mysqli_set_charset($conn, "utf8");

                $sql="SELECT COLUMN_TYPE AS ct FROM
INFORMATION_SCHEMA.COLUMNS WHERE TABLE_SCHEMA = 'psychoacoustics_db' AND
TABLE_NAME = 'guest' AND COLUMN_NAME = 'gender';";
                $result=$conn->query($sql);
                $row=$result->fetch_assoc();//questa query da un
risultato di tipo enum('Male','Female','Non-Binary')

                //metto i valori in un array
                $values = substr($row['ct'], 5, -1);//tolgo "enum(" e
)"
                $values = str_replace("'", "", $values);//tolgo gli
apici
                $list = explode(",", $values);//divido in una lista in
base alle virgole

                //creo un'opzione per ogni possibile valore
                foreach($list as $elem)
                    echo "<option
value='".strtoupper($elem)."'>".strtoupper($elem)."</option>";
                }catch(Exception $e){
                    header("Location: index.php?err=db");
                }
            }
        ?>
    </select>
    <div class="input-group mb-3">
        <span class="input-group-text" id="basic-addon1">Notes</span>
        <input type="text" class="form-control" placeholder="Notes"
name="notes">
    </div>
    <button type="submit" class="btn btn-primary btn-lg m-
1">Register</button>
</form>
</body>

```

```

<script>
    function validation(){
        document.getElementById("NullGender").disabled=false;
    }
</script>
</html>

```

Results.php

```

<!doctype html>
<html lang="en">

    <head>
        <?php
            session_start();
        ?>
        <!-- Required meta tags -->
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link rel="icon" type="image/x-icon" href="files/logo.png">

        <!-- Bootstrap CSS -->
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
        <link rel="stylesheet" href="css/staircaseStyle.css<?php if
(isset($_SESSION['version'])) echo "?".$_SESSION['version']; ?>">

        <title>Psychoacoustics-web - Test results</title>

    </head>
    <body>
        <?php
            //controllo errori
            if(isset($_GET['err'])){
                if($_GET['err']=="1")
                    echo "<div class='alert alert-danger' style='float:left;
width:95%'>'Save result' wasn't checked but 'Save settings' was, Settings
can't be saved without saving the results
                    <br>Result and settings weren't saved</div>";
            }
        ?>
        <div class="container p-4">
            <div class="row gx-4">
                <div class="col">
                    <div class="p-3 border bg-light resultsBox">

```

```

        <h2>Your threshold is
        <?php
            if(isset($_SESSION['score']))
                if(strrpos($_SESSION['score'],";"))
                    echo
substr($_SESSION['score'],strrpos($_SESSION['score'],";")+1);
                else
                    echo $_SESSION['score'];
        ?>
    </h2>

    <?php
        if(isset($_GET['continue'])){
            if(!$_GET['continue']){
                if(isset($_SESSION['usr']))
                    echo "<button type='button' class='btn
btn-primary btn-lg m-3'
onclick='location.href=\"php/download.php?format=complete\" '>Download
data</button>";

                    echo "<button type='button' class='btn
btn-primary btn-lg m-3'
onclick='location.href=\"php/download.php?format=reduced\" '>Download data
(thresholds only)</button>";

                    echo "<button type='button' class='btn
btn-primary btn-lg m-3' onclick='location.href=\"index.php\" '>Home</button>";
            }else{
                $page = "test.php";
                if($_SESSION['type'] ==
"PURE_TONE_FREQUENCY")
                    $page = "freq".$page;
                if($_SESSION['type'] ==
"PURE_TONE_INTENSITY")
                    $page = "amp".$page;
                if($_SESSION['type'] ==
"PURE_TONE_DURATION")
                    $page = "dur".$page;
                if($_SESSION['type'] == "WHITE_NOISE_GAP")
                    $page = "gap".$page;
                if($_SESSION['type'] ==
"WHITE_NOISE_DURATION")
                    $page = "ndur".$page;
                echo "<button type='button' class='btn
btn-primary btn-lg m-3'
onclick='location.href=\"{$page}\" '>Continue</button>";
            }
        }
    ?>

```



```

        <p style="margin-bottom:5%;"></p>
    </div>
</div>
</div>
</div>
</body>
</html>

```

soundsSettings.php

```

<!doctype html>
<html lang="en">
  <head>
    <?php
      session_start();
      include "php/config.php";
      if (!isset($_GET["test"]) && !isset($_SESSION['test']))
        header("Location: index.php");
    ?>

    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="icon" type="image/x-icon" href="files/logo.png">

    <!-- Bootstrap CSS -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet"
      integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIIm9Na00Yz1ztcQTWfSpd3yD65VohhpUuCOmLASjC"
crossorigin="anonymous">

    <link rel="stylesheet"
      href="css/staircaseStyle.css<?php if
(isset($_SESSION['version'])) echo "?".$_SESSION['version']; ?>">

    <title>Psychoacoustics-web - Test settings</title>

  </head>
  <body>
    <?php
      //controllo errori
      if (isset($_GET['err'])) {
        if ($_GET['err'] == "amp1")
          echo "<div class='alert alert-danger'>The amplitude field is
required</div>";
        else if ($_GET['err'] == "amp2")

```

```

        echo "<div class='alert alert-danger'>The amplitude value must
be a number</div>";
    else if ($_GET['err'] == "amp3")
        echo "<div class='alert alert-danger'>The amplitude value
can't be a positive value (maximum is 0dB)</div>";
    else if ($_GET['err'] == "freq1")
        echo "<div class='alert alert-danger'>The frequency field is
required</div>";
    else if ($_GET['err'] == "freq2")
        echo "<div class='alert alert-danger'>The frequency value must
be a positive number</div>";
    else if ($_GET['err'] == "dur1")
        echo "<div class='alert alert-danger'>The duration field is
required</div>";
    else if ($_GET['err'] == "dur2")
        echo "<div class='alert alert-danger'>The duration value must
be a positive number</div>";
    else if ($_GET['err'] == "mod1")
        echo "<div class='alert alert-danger'>The modulation field is
required</div>";
    else if ($_GET['err'] == "mod2")
        echo "<div class='alert alert-danger'>The modulation value
must be a positive number</div>";
    else if ($_GET['err'] == "numblock1")
        echo "<div class='alert alert-danger'>The n. of blocks field
is required</div>";
    else if ($_GET['err'] == "numblock2")
        echo "<div class='alert alert-danger'>The n. of blocks value
must be a positive number</div>";
    else if ($_GET['err'] == "delta1")
        echo "<div class='alert alert-danger'>The delta field is
required</div>";
    else if ($_GET['err'] == "delta2")
        echo "<div class='alert alert-danger'>The delta value must be
a positive number</div>";
    else if ($_GET['err'] == "delta3")
        echo "<div class='alert alert-danger'>The delta value is too
high</div>";
    else if ($_GET['err'] == "ITI1")
        echo "<div class='alert alert-danger'>The ITI field is
required</div>";
    else if ($_GET['err'] == "ITI2")
        echo "<div class='alert alert-danger'>The ITI value must be a
number greater than or equal to 1000</div>";
    else if ($_GET['err'] == "ISI1")
        echo "<div class='alert alert-danger'>The ISI field is
required</div>";
    else if ($_GET['err'] == "ISI2")

```

```

        echo "<div class='alert alert-danger'>The ISI value must be a
positive number</div>";
        else if ($_GET['err'] == "nAFC1")
            echo "<div class='alert alert-danger'>The nAFC field is
required</div>";
        else if ($_GET['err'] == "nAFC2")
            echo "<div class='alert alert-danger'>The nAFC value must be a
number greater than or equal to 2</div>";
        else if ($_GET['err'] == "nAFC3")
            echo "<div class='alert alert-danger'>The nAFC value can't be
greater than 9</div>";
        else if ($_GET['err'] == "factor1")
            echo "<div class='alert alert-danger'>The factor field is
required</div>";
        else if ($_GET['err'] == "factor2")
            echo "<div class='alert alert-danger'>The factor value must be
a number grater than the second factor</div>";
        else if ($_GET['err'] == "secFactor1")
            echo "<div class='alert alert-danger'>The second factor field
is required</div>";
        else if ($_GET['err'] == "secFactor2")
            echo "<div class='alert alert-danger'>The second factor value
must be a number lower than the factor</div>";
        else if ($_GET['err'] == "rev1")
            echo "<div class='alert alert-danger'>The reversals field is
required</div>";
        else if ($_GET['err'] == "rev2")
            echo "<div class='alert alert-danger'>The reversals value must
be a positive number</div>";
        else if ($_GET['err'] == "secRev1")
            echo "<div class='alert alert-danger'>The second reversals
field is required</div>";
        else if ($_GET['err'] == "secRev2")
            echo "<div class='alert alert-danger'>The second reversals
value must be a positive number</div>";
        else if ($_GET['err'] == "threshold1")
            echo "<div class='alert alert-danger'>The reversal threshold
field is required</div>";
        else if ($_GET['err'] == "threshold2")
            echo "<div class='alert alert-danger'>The reversal threshold
value must be a positive number</div>";
        else if ($_GET['err'] == "threshold3")
            echo "<div class='alert alert-danger'>The reversal threshold
value can't be more than the sum of 'Reversals' value and 'Second reversal'
value</div>";
    }

    if (isset($_GET['test']))

```

```

$type = $_GET['test'];

if (isset($_SESSION['usr'])) {
    try {
        $conn = new mysqli($host, $user, $password, $dbname);
        if ($conn->connect_errno)
            throw new Exception('DB connection failed');
        mysqli_set_charset($conn, "utf8");

        $sql = "SELECT test.Amplitude as amp, test.Frequency as freq,
test.Duration as dur, test.Modulation as modu, test.blocks as blocks,
                    test.nAFC, test.ITI, test.ISI, test.Factor as
fact, test.Reversal as rev,
                    test.SecFactor as secfact, test.SecReversal as
secrev, test.Algorithm as alg

                    FROM test
                    INNER JOIN account ON
account.fk_GuestTest=test.Guest_ID AND account.fk_TestCount=test.Test_count

                    WHERE account.Username='{$_SESSION['usr']}'";
        $result = $conn->query($sql);
        $row = $result->fetch_assoc();
    } catch (Exception $e) {
        header("Location: index.php?err=db");
    }
} else
    $row = false;
?>
<div class="container" style="margin-top:1%">
    <div class="row gx-4">
        <div class="col">
            <div class="p-3 border bg-light">
                <h2>Set the characteristics of the experiment</h2>
                <form action="
                    <?php
                        if(isset($_SESSION['updatingSavedSettings']))
&& $_SESSION['updatingSavedSettings']=true)
                            echo
"php/updatingSavedSettings.php?test=".$type;
                        else
                            echo
"php/soundSettingsValidation.php?test=".$type;
                    ?>"
                    name="Settings" method="post">

                    <!-- Primo slot di setting -->
                    <div class="container p-4">

```

```

        <div class="row gx-4">
            <div class="col">
                <div class="p-3 border bg-light
little1">
                    <h6>Set the characteristics of the
standard tone</h6>
                    <!-- Contenuto dello slot, qui
vanno inseriti tutti i bottoni e i check box del secondo slot -->
                    <div class="input-group flex-
nowrap"
                        title="dB of the standard
tone, 0dB = 1 is the maximum value">
                        <span class="input-group-
text">Amplitude</span>
                        <input type="text"
class="form-control" name="amplitude" id="amplitude"
                            value="<?php
                            if ($row)
                                echo $row['amp'];
                            else
                                echo "-20";
                            ?>"
                        >
                        <span class="input-group-
text">dB</span>
                    </div>
                    <div class="input-group flex-
nowrap"
                        title="Hz of the standard
tone, a higher frequency makes the sound sharper"
                        <?php if ($type == "gap" ||
$type == "ndur") echo 'style = "display: none"'?>>
                        <span class="input-group-
text">Frequency</span>
                        <input type="text"
class="form-control" name="frequency" id="frequency"
                            value="<?php
                            if ($row)
                                echo $row['freq'];
                            else
                                echo "1000";
                            ?>"
                        >
                    </div>
                </div>
            </div>
        </div>

```

```

        <span class="input-group-
text">Hz</span>
    </div>

    <div class="input-group flex-
nowrap"
        title="ms of the standard
tone, a higher value makes the sound last longer">
        <span class="input-group-
text">Duration</span>
        <input type="text"
class="form-control" name="duration" id="duration"
            value="<?php
            if ($row)
                echo $row['dur'];
            else
                echo "500";
            ?>"
        >
        <span class="input-group-
text">ms</span>
    </div>

    <div class="input-group flex-
nowrap"
        title="ms of the onset and
offset ramp of the standard tone, a higher value makes the initial and final
transition slower">
        <span class="input-group-
text">Onset/Offset Ramp</span>
        <input type="text"
class="form-control" name="modulation" id="modulation"
            value="<?php
            if ($row)
                echo $row['modu'];
            else
                echo "10";
            ?>"
        >
        <span class="input-group-
text">ms</span>
    </div>

    <!-- <div class="input-group flex-
nowrap">

```

```

        <span class="input-group-
text">Starting phase</span>
        <input type="text"
class="form-control" name="phase" id="phase" placeholder="Standard" aria-
label="Username" aria-describedby="addon-wrapping" value="0">
        <span class="input-group-
text">°</span>
    </div> -->
</div>
</div>
</div>
</div>
</div>

<!-- Secondo slot di setting -->
<div class="container p-4">
    <div class="row gx-4">
        <div class="col">
            <div class="p-3 border bg-light
little2">
                <h6>Set the characteristics of the
experiment</h6>
                <!-- Contenuto dello slot, qui
vanno inseriti tutti i bottoni e i check box del secondo slot -->
                <div class="input-group flex-
nowrap"
                    title="how many times the
test will be repeated">
                    <span class="input-group-
text">
                        n. of blocks
                    </span>
                    <input type="text"
class="form-control" name="blocks" id="blocks"
                        value="<?php
                        if ($row)
                            echo
$row['blocks'];
                        else
                            echo "3";
                        ?>"
                    >
                </div>

```

```

        <div class="input-group flex-
nowrap" title="how many sounds will be played">
        <span class="input-group-
text">nAFC</span>
        <input type="text"
class="form-control" name="nAFC" id="nAFC"
        value="<?php
        if ($row)
            echo $row['nAFC'];
        else
            echo "2";
        ?>"
        >
    </div>

    <div class="input-group flex-
nowrap"
        title="the time before the
new sounds are played (ms)">
        <span class="input-group-
text">ITI</span>
        <input type="text"
class="form-control" name="ITI" id="ITI"
        value="<?php
        if ($row)
            echo $row['ITI'];
        else
            echo "1000";
        ?>"
        >
        <span class="input-group-
text">ms</span>
    </div>

    <div class="input-group flex-
nowrap" title="the time between two sounds (ms)">
        <span class="input-group-
text">ISI</span>
        <input type="text"
class="form-control" name="ISI" id="ISI"
        value="<?php
        if ($row)
            echo $row['ISI'];
        else
            echo "500";
        ?>"

```



```

        >
        <span class="input-group-
text">ms</span>
    </div>

    <div class="input-group flex-
nowrap"
        title="the starting
difference between the sounds">
        <span class="input-group-
text">Delta</span>
        <input type="text"
class="form-control" name="delta" id="level"
            value="<?php
            if ($type == "amp")
                echo "12";
            else if ($type ==
"freq")
                echo "200";
            else if ($type == "dur"
|| $type == "ndur")
                echo "300";
            else if ($type ==
"gap")
                echo "100";
            ?>"
        >
        <span class="input-group-
text">
            <?php
            if ($type == "amp")
                echo "dB";
            else if ($type == "freq")
                echo "Hz";
            else if ($type == "dur" ||
$type == "ndur")
                echo "ms";
            else if ($type == "gap")
                echo "ms";
            ?>
        </span>
    </div>
</div>
</div>
</div>
</div>
</div>

```

```

        <!-- Terzo slot di setting -->
        <div class="container p-4">
            <div class="row gx-4">
                <div class="col">
                    <div class="p-3 border bg-light">
                        <h6>Set the characteristics of the
staircase</h6>

                        <!-- Contenuto dello slot, qui
vanno inseriti tutti i componenti del terzo slot -->

                        <!-- input boxes, sono raggruppati
in un div che sta sulla destra-->

                        <div class="right-div">
                            <div class="left-div">
                                <div class="input-group
flex-nowrap"
                                title="the changing
factor for the first raversals">
                                    <span class="input-
group-text">First factor</span>
                                    <input type="text"
class="form-control" name="factor" id="factor"
value="<?php
if ($row)
    echo
$row['fact'];
else
    echo "2";
?>"
                                    >
                                </div>
                                <div class="input-group
flex-nowrap"
                                title="for how many
reversals the algorithm will use the first factor">
                                    <span class="input-
group-text">First reversals</span>
                                    <input type="text"
class="form-control" name="reversals" id="reversals"
value="<?php
if ($row)
    echo
$row['rev'];
else
    echo "4";
?>"
                                    >
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>

```

```

        ?>"
    >
    </div>
</div>
<div class="right-div">
    <div class="input-group
flex-nowrap"
        title="the changing
factor for the second raversals">
    <span class="input-
group-text">Second factor</span>
    <input type="text"
class="form-control" name="secFactor"
        id="secondFacto
r"
        value="<?php
if ($row)
    echo
$row['secfact'];
        else
    echo
"1.414";
        ?>"
    >
    </div>
<div class="input-group
flex-nowrap"
        title="for how many
reversals the algorithm will use the second factor">
    <span class="input-
group-text">Second reversals</span>
    <input type="text"
class="form-control" name="secReversals"
        id="reversals"
        value="<?php
if ($row)
    echo
$row['secrev'];
        else
    echo "8";
        ?>"
    >
    </div>
</div>
<div class="input-group flex-
nowrap">

```

```

        <span class="input-group-
text">Reversal threshold</span>
        <input type="text"
class="form-control" name="threshold" id="reversalsTh"
        value="<?php
        if ($row)
            echo
$row['secrev'];
        else
            echo "8";
        ?>"
        >
    </div>
</div>

<!-- Radios, sono raggruppati in
un div che sta sulla sinistra-->
    <div class="left-div">
        <div class="form-check"
            title="every correct
answer increases the difficulty of the test, every wrong answer makes it
easier">
            <input class="form-check-
input" type="radio" name="algorithm"
                value="SimpleUpDown
" id="alg"
                <?php
                if ($row &&
$row['alg'] == "SimpleUpDown")
                    echo "checked";
                ?>
            >
            <label class="form-check-
label" for="flexRadioDefault1">
                SimpleUpDown
            </label>
        </div>
        <div class="form-check"
            title="two consecutive
correct answers increase the difficulty of the test, every wrong answer makes
it easier">
            <input class="form-check-
input" type="radio" name="algorithm"
                value="TwoDownOneUp
" id="alg"
                <?php
                if (($row &&
$row['alg'] == "TwoDownOneUp") || !$row)

```

```

                echo "checked";
            ?>
        >
        <label class="form-check-
label" for="flexRadioDefault1">
                TwoDownOneUp
            </label>
        </div>
        <div class="form-check"
            title="three consecutive
correct answers increase the difficulty of the test, every wrong answer makes
it easier">
            <input class="form-check-
input" type="radio" name="algorithm"
                value="ThreeDownOne
Up" id="alg"
                <?php
                if ($row &&
$row['alg'] == "ThreeDownOneUp")
                    echo "checked";
                ?>>
            <label class="form-check-
label" for="flexRadioDefault1">
                ThreeDownOneUp
            </label>
        </div>

        <!-- Checkbox -->
        <div class="form-check
checkboxes">
            <div class="form-check"
                title="if checked a
message will tell if you choose the correct sound">
                <input class="form-
check-input" type="checkbox" id="cb" name="checkFb"
                    checked>
                <label class="form-
check-label" for="cb">
                    FeedBack
                </label>
            </div>
            <?php
            if
(isset($_SESSION['usr']) && !(isset($_SESSION['updatingSavedSettings']) &&
$_SESSION['updatingSavedSettings']=true))

```

```

                                echo '<div
class="form-check" title="if checked the settings will be saved and used as
default for the next tests">
                                <input
class="form-check-input" type="checkbox" id="saveSettings"
name="saveSettings">
                                <label
class="form-check-label" for="saveSettings">
                                Save
settings
                                </label>
                                </div>;
                                ?>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
                                <!-- i bottoni sono fuori dal terzo slot -->
                                <button type="button" class="btn btn-primary btn-
lg m-3 soundSettingsButton"
                                onclick="location.href='demographicData.ph
p'">BACK
                                </button>
                                <button type="submit" class="btn btn-primary btn-
lg m-3 soundSettingsButton">
                                <?php
                                if(isset($_SESSION['updatingSavedSettings'
]) && $_SESSION['updatingSavedSettings']=true)
                                echo "SAVE SETTINGS";
                                else
                                echo "START";
                                ?>
                                </button>
                                </form>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
                                <!-- Bootstrap Bundle with Popper -->
                                <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min
.js"
                                integrity="sha384-
MrcW6ZMFY1zcla8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"

```

```
crossorigin="anonymous"></script>
```

```
</body>
```

```
</html>
```

userSettings.php

```
<!doctype html>
<html lang="en">
  <head>
    <?php
      session_start();
      include "php/config.php";
    ?>

    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="icon" type="image/x-icon" href="files/logo.png">

    <!-- Bootstrap CSS -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
      <link rel="stylesheet" href="css/style.css<?php if
(isset($_SESSION['version'])) echo "{$_SESSION['version']}"; ?>">
      <script type="text/javascript" src="js/funzioni.js<?php if
(isset($_SESSION['version'])) echo "{$_SESSION['version']}"; ?>"></script>

    <title>Psychoacoustics-web - User settings</title>

  </head>

  <body>

    <!-- Barra navigazione -->
    <nav class="navbar navbar-dark bg-dark">
      <div class="container-fluid" >
        <a class="navbar-brand" href="index.php" >
          
          PSYCHOACOUSTICS
        </a>
        <form class="container-fluid logButtons">
          <label class='welcomeMessage'>Welcome <?php echo
$_SESSION['usr'];?></label>
```

```

        <button class="btn btn-outline-primary yourTests"
type="button" onclick="location.href='yourTests.php'">Your tests</button>
        <button class="btn btn-outline-danger logout" type="button"
onclick="location.href='php/logout.php'">Log Out</button>
    </form>
</div>
</nav>

<?php
    //se si sceglie un username già esistente verrà messo "?err=1"
nell'url
    if(isset($_GET['err'])){
        if ($_GET['err']==0)
            echo "<div class='alert alert-danger'>Some inserted
characters aren't allowed</div>";
        if ($_GET['err']==1)
            echo "<div class='alert alert-danger'>Username already
taken</div>";
        if ($_GET['err']==2)
            echo "<div class='alert alert-danger'>Wrong
password</div>";
        if ($_GET['err']==3)
            echo "<div class='alert alert-success'>Password
changed</div>";
        if ($_GET['err']==4)
            echo "<div class='alert alert-success'>Test settings
changed</div>";
    }
    try{
        $conn = new mysqli($host, $user, $password, $dbname);
        if ($conn->connect_errno)
            throw new Exception('DB connection failed');
        mysqli_set_charset($conn, "utf8");

        $sql = "SELECT referral, name, surname, date, gender, notes,
email
                FROM account INNER JOIN guest ON account.Guest_ID =
guest.ID
                WHERE username='".$$_SESSION['usr']."'";
        $result = $conn->query($sql);
        $row = $result->fetch_assoc();
        $ref = $row['referral'];
        $name = $row['name'];
        $sur = $row['surname'];
        $date = $row['date'];
        $gender = $row['gender'];
        $notes = $row['notes'];
        $email = $row['email'];

```



```

    }catch(Exception $e){
        header("Location: index.php?err=db");
    }
?>

<form action="php/newReferral.php" class="settingForm ref">
    <div class="input-group mb-3">
        <span class="input-group-text title" onclick="copy('ref')"
title="click to copy">Invite code</span>
        <span class="input-group-text link" id="ref"
onclick="copy('ref')" title="click to copy"><?php echo $ref; ?></span>
    </div>
    <div class="input-group mb-3">
        <span class="input-group-text title" onclick="copy('link')"
title="click to copy">Link</span>
        <span class="input-group-text link" id="link"
onclick="copy('link')" title="click to copy">
            psychoacoustics.dpg.psy.unipd.it/sito/demographicData.php?
ref=<?php echo $ref; ?>
        </span>
    </div>

    <button type="submit" class="btn btn-primary btn-lg m-3">Change
invite code</button>
    <select name='gender' class="form-select"
onchange="updateLink('<?php echo $ref; ?>')" id="testType">
        <option value='amp' selected>Pure tone intensity</option>
        <option value='freq'>Pure tone frequency</option>
        <option value='dur'>Pure tone duration</option>
        <option value='gap'>Noise Gap</option>
        <option value='ndur'>Noise Duration</option>
    </select>
    <button type="button" class="btn btn-primary btn-lg m-3"
onclick="window.location='php/updateSavedSettings.php?test='+document.getEleme
ntById('testType').value">Change test settings</button>

</form>
<?php
    try{
        $sql = "SELECT Type FROM account WHERE
Guest_ID='{$_SESSION['idGuest']}' AND Username='{$_SESSION['usr']}'";
        $result=$conn->query($sql);
        $row=$result->fetch_assoc();
        if($row['Type'] == 1){
            echo '<form action="php/newUsername.php" method="POST"
class="settingForm ref">
                <div class="input-group mb-3">

```

```

        <span class="input-group-text title"
onclick="copy('\ref\')" title="Username">Username</span>
        <input type="text" class="form-control"
placeholder="Username" name="username">
    </div>

        <button type="submit" class="btn btn-primary btn-lg
m-3">Create new Superuser</button>
    </form>';
    }
    }catch(Exception $e){
        header("Location: index.php?err=db");
    }
    ?>

    <form action="php/changePsw.php" method= "post" class="settingForm">
        <div class="input-group mb-3">
            <span class="input-group-text">Old password</span>
            <input type="password" class="form-control" placeholder="Old
password" name="oldPsw">
        </div>
        <div class="input-group mb-3">
            <span class="input-group-text">New password</span>
            <input type="password" class="form-control" placeholder="New
password" name="newPsw">
        </div>
        <button type="submit" class="btn btn-primary btn-lg m-3">Change
Password</button>
    </form>

    <form method="post" action="php/saveSettings.php" class="settingForm">
        <div class="input-group mb-3">
            <span class="input-group-text">Username</span>
            <input type="text" class="form-control" name="usr"
value="<?php echo $_SESSION['usr']; ?>">
        </div>
        <div class="input-group mb-3">
            <span class="input-group-text">Email</span>
            <input type="text" class="form-control" name="email"
value="<?php echo $email; ?>">
        </div>

        <div class="input-group mb-3">
            <span class="input-group-text">Name</span>
            <input type="text" class="form-control" name="name"
value="<?php echo $name; ?>">
        </div>
        <div class="input-group mb-3">

```

```

        <span class="input-group-text">Surname</span>
        <input type="text" class="form-control" name="surname"
value="<?php echo $sur; ?>">
    </div>
    <div class="input-group mb-3">
        <span class="input-group-text">Birth date</span>
        <input type="date" class="form-control" name="date"
value="<?php echo $date; ?>">
    </div>
    <select name='gender' class="form-select">
        <option disabled="disabled" value="null" id="NullGender" <?php
if($gender=="NULL") echo "selected"; ?>>Select your gender</option>
        <?php
            try{
                $sql="SELECT COLUMN_TYPE AS ct FROM
INFORMATION_SCHEMA.COLUMNS WHERE TABLE_SCHEMA = 'psychoacoustics_db' AND
TABLE_NAME = 'guest' AND COLUMN_NAME = 'gender';";
                $result=$conn->query($sql);
                $row=$result->fetch_assoc();//questa query da un
risultato di tipo enum('Male','Female','Non-Binary')

                //metto i valori in un array
                $values = substr($row['ct'], 5, -1);//tolgo "enum(" e
)"
                $values = str_replace("'", "", $values);//tolgo gli
apici
                $list = explode(",", $values);//divido in una lista in
base alle virgole

                //creo un'opzione per ogni possibile valore
                foreach($list as $elem)
                    echo "<option
value='".strtoupper($elem)."'>".strtoupper($elem)."</option>";
                }catch(Exception $e){
                    header("Location: index.php?err=db");
                }
            ?>
    </select>
    <div class="input-group mb-3 notes">
        <span class="input-group-text">Notes</span>
        <input type="text" class="form-control" placeholder="Notes"
name="notes" value="<?php echo $notes; ?>">
    </div>
    <button type="submit" class="btn btn-primary btn-lg m-
3">Save</button>
</form>
</body>
</html>

```

yourTest.php

```
<!doctype html>
<html lang="en">

  <head>
    <?php
      session_start();
      if(!isset($_SESSION['usr']) || !isset($_SESSION['idGuest']))
        header("Location: index.php");
      include "php/config.php";
    ?>

    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="icon" type="image/x-icon" href="files/logo.png">

    <!-- Bootstrap CSS -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWFspD3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
    <link rel="stylesheet" href="css/style.css<?php if
(isset($_SESSION['version'])) echo "?{"$_SESSION['version']}"; ?>">

    <title>Psychoacoustics-web - Test results</title>
  </head>
  <body>

    <nav class="navbar navbar-dark bg-dark">
      <div class="container-fluid" >
        <a class="navbar-brand" href="index.php" >
          
            PSYCHOACOUSTICS
        </a>
        <form class="container-fluid logButtons">
          <?php
            echo "<label class='welcomeMessage'>Welcome
".$_SESSION['usr']."</label>";
            echo "<button class=\"btn btn-outline-danger logout\"
type=\"button\" onclick=\"location.href='php/logout.php'\">Log Out</button>";
          ?>
        </form>
      </div>
    </nav>
  </body>
</html>
```

```

    </div>
</nav>

<h1>Welcome <?php echo $_SESSION['usr'];?></h1>

<button type='button' class='btn btn-primary btn-lg m-3'
onclick='location.href="php/downloadYours.php?all=1"'>Download all your
data</button>
<button type='button' class='btn btn-primary btn-lg m-3'
onclick='location.href="php/downloadYours.php?all=0"'>Download all your
guest's data</button>

<?php
    try{
        $conn = new mysqli($host, $user, $password, $dbname);

        if ($conn->connect_errno)
            throw new Exception('DB connection failed');

        mysqli_set_charset($conn, "utf8");

        $usr = $_SESSION['usr'];
        $id = $_SESSION['idGuest'];

        $sql = "SELECT Type FROM account WHERE Guest_ID='$id' AND
Username='$usr'";
        $result=$conn->query($sql);
        $row=$result->fetch_assoc();
        if($row['Type'] == 1){
            echo "<button type='button' class='btn btn-primary btn-lg
m-3' onclick='location.href=\"php/downloadAll.php\"'>Download all the data in
the database</button>";
        }
    }catch(Exception $e){
        header("Location: index.php?err=db");
    }
?>

<h3>Your results</h3>

<table>
    <tr class="tableHeader">
        <td>Test</td>
        <td>Time</td>
        <td>Type</td>
    </tr>
    <?php
        try{

```

```

        $sql = "SELECT Test_count, Timestamp, Type FROM test WHERE
Guest_ID='$id'";
        $result=$conn->query($sql);
        while($row=$result->fetch_assoc()){
            echo "<tr>";
            echo "<td>".$row["Test_count"]."</td>";
            echo "<td>".$row["Timestamp"]."</td>";
            echo "<td>".$row["Type"]."</td>";
            echo "</tr>";
        }
    }catch(Exception $e){
        header("Location: index.php?err=db");
    }
    ?>
</table>

<h3>Your guest's results</h3>

<table>
    <tr class="tableHeader">
        <td>Name</td>
        <td>Test</td>
        <td>Time</td>
        <td>Type</td>
    </tr>
    <?php
        try{
            $sql = "SELECT Name, Test_count, Timestamp, Type FROM test
INNER JOIN guest ON Guest_ID=ID WHERE fk_guest='{$_SESSION['usr']}'";
            $result=$conn->query($sql);
            while($row=$result->fetch_assoc()){
                echo "<tr>";
                echo "<td>".$row["Name"]."</td>";
                echo "<td>".$row["Test_count"]."</td>";
                echo "<td>".$row["Timestamp"]."</td>";
                echo "<td>".$row["Type"]."</td>";
                echo "</tr>";
            }
        }catch(Exception $e){
            header("Location: index.php?err=db");
        }
    ?>
</table>

    <button type="button" class="btn btn-primary btn-lg m-3" id="home"
onclick = "location.href='index.php'">Home</button>
</body>
</html>

```

formStyle.css

```
.alert{
  margin:2%;
}

h1{
  margin-bottom: 20px;
}

form {
  padding: 40px;
  background: #DDD;
  border: 2px solid #000;
  width: 50%;
  margin: 30px auto;
  border-radius: 10px;
}

.form-select{
  margin-bottom: 3%;
}

@media all and (max-width: 991px) {
  form {
    width: 90%;
    padding: 3%;
    margin: 5%;
    margin-top:15%;
  }
}
```

staircaseStyle.css

```
.alert{
  margin:2%;
}

.wallpaper{
  position:fixed;
  top:0%;
  left:0%;
  width: 100%;
  height: 110%;
  object-fit:cover;
  opacity: 0.6;
  z-index: -1;
  overflow: hidden;
```

```

}

.btn-primary.btn-lg.m-3{
  color: white;
  font-weight: lighter;
  text-align: center;
  cursor: pointer;
  background-color:darkred;
  border: 0;
}

.btn-primary.btn-lg.m-3:hover{
  background-color:#50000A;
}

.btn-primary.btn-lg.m-3:focus{
  box-shadow: 0px 0px 3px #212529;
}

input[type="checkbox"], input[type="radio"]{
  border: 1px solid #212529;
}

input[type="checkbox"]:checked, input[type="radio"]:checked{
  background-color:darkred;
  border: 1px solid #212529;
}

input[type="checkbox"]:focus, input[type="radio"]:focus{
  border: 1px solid #212529;
  box-shadow: 0px 0px 3px #212529;
}

.input-group{
  margin-top: 10px;
}

.input-group.flex-nowrap{
  width: 23%;
  float: left;
  margin-right: 2%; /*anche il margine si adatta con la dimensione dello
schermo*/
}

#notesDiv{
  /*a differenza degli altri input box della pagina per le info anagrafiche,
questo è più largo,

```



```

    quindi ha una classe in più con una proprietà in più (la larghezza)*/
    width: 98%;
}

.form-check{
    margin-top: 1.5%;
    width: 23%;
    float: right; /*il checkbox sta sulla destra in entrambe le pagine html*/
}

.btn.btn-primary.btn-lg.m-3.soundSettingsButton{
    /*i pulsanti in soundsettings hanno una larghezza e dei margini diversi,
    quindi usano una classe in più*/
    float: left;
    margin: 5% !important;
    margin-top: 0px !important;
    margin-bottom: 2% !important;
    width: 40%;
}

.little1, .little2{
    /*in soundsettings i primi due gruppi di input box (i più piccoli)
    contengono solo oggetti float,
    quindi hanno bisogno di avere una dimensione maggiore di 100% per coprire
    tutto. Gli oggetti con
    proprietà float non occupano spazio e quindi il div crede di essere vuoto
    e diventa troppo piccolo*/
    height: 130% !important;
}

.p-3.border.bg-light.little1 .input-group.flex-nowrap{
    width: 31%;
}

.p-3.border.bg-light.little2 .input-group.flex-nowrap{
    width: 31%;
}

.p-3.border.bg-light{
    height: 99%;
    border-radius: 20px;
}

.right-div{
    /*il right div sta a destra*/
    width: 48%;
    float: right;
    margin-bottom: 2%;
}

```

```

margin-left: 1%;
}

.right-div .input-group.flex-nowrap{
  /*gli input box all'interno del right div stanno a destra*/
  width: 100%;
  float: left;
}

.left-div{
  /*il left div sta a sinistra*/
  width: 48%;
  float: left;
  margin-right: 1%;
}

.left-div .form-check{
  /*normalmente i checkbox hanno width 23% (vedasi sopra), ma dentro al left
  div devono occupare tutto lo spazio*/
  width: 100%;
}

.referral{
  float: right !important;
  width: 35% !important;
  margin: 2% !important;
}

.refButton{
  float: right;
  margin: 2% !important;
  padding: 6px 1% 6px !important;
  font-size: 1.1rem;
}

.checkboxes{
  margin-top: 5%;
  padding-left: 0%;
  margin-bottom: 2%;
}

.info.Checkbox{
  width: 80%;
  float: left;
  margin: 5%;
  margin-top: 2%;
  margin-bottom: 2%;
}

```

```
.info.container{
  margin: 1%;
  max-width: 95%;
  padding: 0;
}

.info.title{
  margin: 5%;
  margin-top:3%;
  margin-bottom:2%;
}

.info.text{
  margin: 4%;
  margin-top:2%;
  margin-bottom:0;
  background-color: rgba(255,255,255,0.85);
  border-radius: 30px;
  padding: 2%;
}

.info.test {
  margin: 3% 4% 0;
}

.info.test-title {
  display: flex;
  flex-wrap: nowrap;
  align-items: center;
  gap: 32px;
}

.info.test-title h5 {
  margin: 0;
}

.info.test-preview {
  background-color: white;
  border-radius: 30px;
  margin-top: 8px;
  padding: 2%;
}

.info.test-preview-form {
  padding: 3%;
  background: #DDD;
  border: 2px solid #000;
}
```

```

width: 50%;
margin: 30px auto;
border-radius: 10px;
text-align: center;
}

#alert {
margin: 40%;
margin-top: 0px;
margin-bottom: 0px;
text-align: center;
visibility: hidden;
}

.resultsBox{
margin-top: 10%;
text-align: center;
height: inherit !important;
}

.resultsBox > *{
margin-top: 5% !important;
}

/*quando la finestra è più piccola di 991 pixel, viene usato questo css*/
@media all and (max-width: 991px) {

    /*gli input box ora prendono tutta la riga, come le note*/
    .input-group.flex-nowrap{
width: 98% !important;
float: none;
margin-right: 2%;
}

    .right-div{
/*non sta più a destra, non c'è spazio per fare due colonne quindi
vengono impilati verticalmente,
quindi tolgo tutti i float*/
width: 95%;
float: none;
margin-bottom: 5%;
}

    .right-div .input-group.flex-nowrap{
width: 95%;
float: none;
}
}

```

```

.left-div{
    width: 95%;
    float: none;
}

.p-3.border.bg-light{
    height: 101%;
}

.form-check{
    /*per evitare che si sovrapponga ad altri oggetti*/
    float: none;
    margin: 5%;
    width: 95%;
}

.referral{
    margin: 0 !important;
    margin-right: 2% !important;
    margin-top: 5% !important;
    float: none !important;
    width: 98% !important;
}

.refButton{
    min-width: 50px;
    width: 20% !important;
    margin-top: 5% !important;
}
}

```

style.css

```

body{
    background-color: #fff;
}

.alert{
    margin:2%;
}

.wallpaper{
    position:absolute;
    top:40px;
    left:0%;
    width: 100%;
    height: 100%;
    object-fit:cover;
}

```

```

    opacity: 0.6;
    z-index: -1;
    overflow: hidden;
}

.navbar{
    box-shadow: 0px 2px 5px darkred;
}

.btn-outline-success{
    float:right;
    border-color: white;
    color: white;
}

.btn-outline-danger{
    float:right;
    border-color: white;
    color: white;
}

.btn-primary.btn-lg.m-3{
    color: white;
    font-weight: lighter;
    text-align: center;
    cursor: pointer;
    background-color:darkred;
    border: 0;
}

.btn-primary.btn-lg.m-3:hover{
    background-color:#50000A;
}

.btn-primary.btn-lg.m-3:focus{
    box-shadow: 0px 0px 3px #212529;
}

.darkred{
    background-color: darkred;
    border-color: black;
}

.logout{
    margin-left: 2%;
    margin-top:0px;
}

```

```
.yourTests{
  float: right;
  margin-left: 2%;
  margin-top:0px;
  border-color: white;
  color: white;
}

.logButtons{
  text-align:right;
  width: 60%;
  margin-left: auto;
  margin-right: 0;
}

.welcomeMessage{
  color:white;
  margin-top: 7px;
  margin-left: 0;
  margin-right: auto;
  text-align:center;
  text-decoration: none;
}

.settings{
  float:right;
  margin: 1%;
  color: grey;
  margin-left:2%;
}

.rotate{
  transform: rotate(-45deg);
  transition: all 0.3s ease-in-out 0s;
}

.settings:hover{
  color: white;
}

.rotate:hover{
  transform: rotate(45deg);
  transition: all 0.3s ease-in-out 0s;
}

.navbar-brand{
  width:40%;
  margin-right: 0;
```

```

}

.description{
  color:black;
  padding: 3%;
  margin: 1%;
  font-weight: 400;
  background-color: rgba(255,255,255,0.5);
  border-radius: 30px;
}

.descriptionTitle{
  color:black;
  padding: 3%;
  padding-bottom:1%;
  font-size:60px;
  font-weight: 400;
  text-align:center;
  margin-top: 2%;
}

.cards{
  margin:2%;
}

.card{
  float: left;
  width: 30%;
  margin: 1.5%;
  margin-top: 3px;
  box-shadow: 3px 3px 5px darkred;
  color: white;
  font-weight: lighter;
  text-align: center;
  cursor: pointer;
  background-color:#212529;
}

.card:hover{
  background-color:darkred;
  color: white;
  box-shadow: 10px 10px 5px #212529;
}

.settingForm{
  padding: 40px;
  background: #DDD;
  border: 2px solid #000;
}

```



```

width: 90%;
margin: 30px auto;
border-radius: 10px;
}

.form-select{
margin: 2%;
width: 45%;
float: left;
}

.input-group.mb-3{
margin: 2%;
width: 45%;
float: left;
}

.input-group.mb-3.notes{
margin: 2%;
width: 94%;
float: left;
}

.input-group-text.link{
width: 70%;
display: initial;
/*da attivare quando ci sarà https
overflow:hidden;
white-space:nowrap;
text-overflow:ellipsis;
/**/
overflow:scroll;
}

.settingForm.ref .input-group-text.title{
width:30%;
}

h1, h3{
margin:2%;
margin-top:5%;
}

table{
margin:2%;
width: 95%;
}

```

```

tr:nth-child(even) {background: #CCC}
tr:nth-child(odd) {background: #EEE}

td{
  text-align: center;
  padding:1%;
  padding-top: 2%;
  padding-bottom: 2%;
}

.tableHeader{
  background: #fff !important;
  font-weight: 700;
}

@media all and (max-width: 900px) {
  .wallpaper{
    top:22%;
  }

  .navbar-brand{
    width: 95%;
    margin: 2%;
    text-align: center;
    font-size:6vw;
  }

  .logButtons{
    width: 95%;
    margin: 2%;
    text-align: center;
  }

  .settings{
    float:none;
    margin: 2%;
  }

  .logout{
    width: 40%;
    margin: 2%;
  }

  .yourTests{
    float: none;
    width: 40%;
    margin: 2%;
  }
}

```

```
}

.welcomeMessage{
  width: 75%;
  margin: 2%;
}

.btn-outline-success{
  float: none;
  width: 40%;
  margin: 2%;
}

.btn-outline-danger{
  float: none;
  width: 40%;
  margin: 2%;
}

.descriptionTitle{
  padding: 0.5%;
  padding-top: 3%;
  font-size:1.5em;
  font-weight: 500;
}

.description{
  padding: 1%;
  margin: 1%;
  font-weight: 400;
  font-size:1em;
}

.card-body{
  padding:1%;
}

.card-title{
  font-size: 0.9em;
  font-weight: 350;
}

.card-body p{
  font-size: 0.8em;
  font-weight: 250;
}

.settingForm{
```

```

padding: 2%;
width: 98%;
margin: 2%;
}

.input-group.mb-3{
margin: 2%;
width: 95%;
float: none;
}

.input-group-text.link{
width: 60%;
display: initial;
/*da attivare quando ci sarà https
overflow:hidden;
white-space:nowrap;
text-overflow:ellipsis;
/**/
overflow:scroll;
}

.settingForm.ref .input-group-text.title{
width:40%;
}

.input-group.mb-3 select{
width: 60%;
}

.form-select{
margin: 2%;
margin-bottom: 1rem;
width: 95%;
float: none;
}

.input-group.mb-3.notes{
margin: 2%;
width: 95%;
float: none;
}

.btn-primary.btn-lg.m-3{
width: 90%;
margin: 5%;
}

```

```

table{
    font-size: 0.8em;
    font-weight: 350;
}

.tableHeader{
    font-size: 1em;
}

td{
    overflow:hidden;
    white-space:nowrap;
    text-overflow:ellipsis;
    text-align: left;
    padding-left:1%;
    padding-top: 1%;
    padding-bottom: 1%;
    max-width: 1em;
}
}

```

test.css

```

.btn{
    margin: 4%;
    margin-bottom: 0%;
    padding: 2%;
    padding-left: 3%;
    padding-right: 3%;
    border: none;
}

form {
    padding: 3%;
    background: #DDD;
    border: 2px solid #000;
    width: 50%;
    margin: 30px auto;
    border-radius: 10px;
    text-align: center;
    display: none;
}

.window{
    padding: 4%;
    background: #DDD;
    border: 2px solid #000;
    width: 30%;
}

```

```

margin: 10% auto;
border-radius: 10px;
text-align: center;
}

.window .btn-success{
margin-top: 20%;
padding: 5%;
}

.alert{
margin: 40%;
margin-top: 0px;
margin-bottom: 0px;
display: none;
text-align: center;
}

.p-3.border.bg-light{
height: 99%;
border-radius: 20px;
}

.btn-primary.btn-lg.m-3{
color: white;
font-weight: lighter;
text-align: center;
cursor: pointer;
background-color:darkred;
border: 0;
}

.btn-primary.btn-lg.m-3:hover{
background-color:#50000A;
}

@media all and (max-width: 900px) {
.window{
padding: 5%;
width: 80%;
margin: 10% auto;
}

.window .btn-success{
margin: 2%;
margin-top: 5%;
padding: 3% 5% 3%;
}
}

```

```

form {
  padding: 5%;
  width: 90%;
  margin: 10% auto;
}

.alert{
  margin: 20%;
  margin-top: 0px;
  margin-bottom: 0px;
}
}

```

testPreview.js

```

var context = new AudioContext();

// minimum initial variation
var varFreq = freq;           // frequency of the variable
var stdFreq = freq;          // frequency of the standard

var startingDelta = delta;

var stdDur = dur;             // duration of the standard
var varDur = dur;            // duration of the variable

var stdAmp = amp;            // intensity of the standard
var varAmp = amp;            // intensity of the variable

switch (type) {
  case "amplitude":
    varAmp = amp + startingDelta;
    break;
  case "frequency":
    varFreq = freq + startingDelta;
    break;
  case "duration":
    varDur = dur + startingDelta;
    break;
  case "nduration":

```

```

        varDur = dur + startingDelta;
        break;
    }

    var stdMod = mod; // onset and offset duration of ramp of
    the standard
    var varMod = mod; // onset and offset duration of ramp of
    the variable

    var swap = -1; // position of variable sound

    var maxDur = 15; // durata massima rumore
    var betweenRampDur = 0.03 // durata rampa nel gap

    var channels = 2; // durata massima rumore
    var frameCount = context.sampleRate * maxDur; // durata rampa nel gap
    var myArrayBuffer = context.createBuffer(channels, frameCount,
    context.sampleRate); // numero canali di uscita
    for (let channel = 0; channel < channels; channel++) { // imposto una durata
    massima del rumore di n (15) secondi
        let nowBuffering = myArrayBuffer.getChannelData(channel);
        for (let i = 0; i < frameCount; i++) {
            nowBuffering[i] = Math.random() * 2 - 1;
        }
    }
}

//funzione per generare il suono variabile
function playVarSound(time) {
    var volume1 = context.createGain(); //volume
    volume1.gain.setValueAtTime(0, context.currentTime + time); //imposto
    volume iniziale a 0
    volume1.connect(context.destination); //collego all'uscita audio
    volume1.gain.setTargetAtTime((10 ** (parseInt(varAmp) / 20)),
    context.currentTime + time, varMod / 1000); //implemento onset ramp
    volume1.gain.setTargetAtTime(0, context.currentTime + time + (varDur /
    1000) - 3 * (varMod / 1000), varMod / 1000); //implemento offset ramp
    (l'espressione "3 * (Mod/1000)" serve a garantire un raggiungimento del target
    del 95% circa, guardare documentazione di setTargetAtTime, in particolare la
    tabella del timeconstant)

    oscillator = context.createOscillator();//Creiamo il primo oscillatore
    oscillator.connect(volume1); //Colleghiamo l'oscillatore al
    oscillator.frequency.value = varFreq; //frequency
    oscillator.type = "sine"; // tipo di onda

    oscillator.start(context.currentTime + time); //Facciamo partire
    l'oscillatore

```



```

    oscillator.stop(context.currentTime + time + (varDur / 1000)); //Fermiamo
l'oscillatore dopo dur secondi
}

//funzione per generare il suono standard
function playStdSound(time) {
    var volume2 = context.createGain(); //volume
    volume2.gain.setValueAtTime(0, context.currentTime + time); //imposto
volume iniziale a 0
    volume2.connect(context.destination); //collego all'uscita audio
    volume2.gain.setTargetAtTime((10 ** (parseInt(stdAmp) / 20)),
context.currentTime + time, stdMod / 1000); //implemento onset ramp
    volume2.gain.setTargetAtTime(0, context.currentTime + time + (stdDur /
1000) - 3 * (stdMod / 1000), stdMod / 1000); //implemento offset ramp
(l'espressione "3 * (Mod/1000)" serve a garantire un raggiungimento del target
del 95% circa, guardare documentazione di setTargetAtTime, in particolare la
tabella del timeconstant)

    oscillator = context.createOscillator();//Creiamo il secondo oscillatore
    oscillator.connect(volume2); //Colleghiamo l'oscillatore al
    oscillator.frequency.value = stdFreq; //frequency
    oscillator.type = "sine"; //tipo di onda

    oscillator.start(context.currentTime + time); //Facciamo partire
l'oscillatore
    oscillator.stop(context.currentTime + time + (stdDur / 1000));//Fermiamo
l'oscillatore dopo dur secondi
}

//funzione per generare il rumore variabile con gap
function playVarNoiseGap(time) {
    let start1 = 0; // punto di
partenza della prima metà del rumore
    let end1 = ((varDur / 1000) / 2) - ((delta / 1000) / 2); // punto di
stop della prima metà del rumore
    let start2 = ((varDur / 1000) / 2) + ((delta / 1000) / 2); // punto di
partenza della seconda metà del rumore
    let end2 = varDur / 1000; // punto di
stop della seconda metà del rumore

    var volume1 = context.createGain(); // creo volume
    volume1.gain.setValueAtTime(0, context.currentTime); // imposto volume
iniziale a 0
    volume1.connect(context.destination); // connetto il volume all'uscita
    volume1.gain.setTargetAtTime((10 ** (parseInt(amp) / 20)),
context.currentTime + time + start1, mod / 1000); // eseguo rampa onset
iniziale (inizia da start1 e dura mod ms)

```

```

    volume1.gain.setTargetAtTime(0, context.currentTime + time + end1 - 3 *
(betweenRampDur / 1000), betweenRampDur); // eseguo rampa offset del gap
(inizia a end1 e dura betweenRampDur s)
    volume1.gain.setTargetAtTime((10 ** (parseInt(amp) / 20)),
context.currentTime + time + start2, betweenRampDur); // eseguo rampa onset
del gap (inizia a start2 e dura betweenRampDur s)
    volume1.gain.setTargetAtTime(0, context.currentTime + time + end2 - 3 *
(mod / 1000), mod / 1000); // eseguo rampa offset finale (inizia da end2 e
dura mod ms)

    source = context.createBufferSource(); // creo sorgente
    source.buffer = myArrayBuffer; // collego i buffer
    source.connect(volume1); // connetto la sorgente al volume
    source.start(context.currentTime + time + start1); // riproduco il rumore
    source.stop(context.currentTime + time + end2); // fermo il rumore
}

//funzione per generare il rumore variabile (no gap)
function playVarNoise(time) {
    var volume2 = context.createGain(); // creo volume
    volume2.gain.setValueAtTime(0, context.currentTime); // imposto volume
iniziale a 0
    volume2.connect(context.destination); // connetto il volume all'uscita
    volume2.gain.setTargetAtTime((10 ** (parseInt(amp) / 20)),
context.currentTime + time, varMod / 1000); // eseguo rampa onset iniziale
    volume2.gain.setTargetAtTime(0, context.currentTime + time + (varDur /
1000) - 3 * (varMod / 1000), varMod / 1000); // eseguo rampa offset finale

    source = context.createBufferSource(); // creo sorgente
    source.buffer = myArrayBuffer; // collego i buffer
    source.connect(volume2); // connetto la sorgente al volume
    source.start(context.currentTime + time); // riproduco il rumore
    source.stop(context.currentTime + time + (varDur / 1000)); // fermo il
rumore
}

//funzione per generare il rumore standard
function playStdNoise(time) {
    var volume2 = context.createGain(); // creo volume
    volume2.gain.setValueAtTime(0, context.currentTime); // imposto volume
iniziale a 0
    volume2.connect(context.destination); // connetto il volume all'uscita
    volume2.gain.setTargetAtTime((10 ** (parseInt(amp) / 20)),
context.currentTime + time, stdMod / 1000); // eseguo rampa onset iniziale
    volume2.gain.setTargetAtTime(0, context.currentTime + time + (stdDur /
1000) - 3 * (stdMod / 1000), mod / 1000); // eseguo rampa offset finale

    source = context.createBufferSource(); // creo sorgente

```

```

source.buffer = myArrayBuffer; // collego i buffer
source.connect(volume2); // connetto la sorgente al volume
source.start(context.currentTime + time); // riproduco il rumore
source.stop(context.currentTime + time + (stdDur / 1000)); // fermo il
rumore
}

//funzione per randomizzare l'output
function random() {
  for (var j = 1; j <= nAFC; j++)
    document.getElementById("button" + j).disabled = true;
  document.getElementById("playTest").disabled = true;
  document.getElementById("alert").style.visibility = "hidden";

  var rand = Math.floor(Math.random() * nAFC); // the variable sound will be
the rand-th sound played

  for (var j = 0; j < nAFC; j++) {
    if (type == "amplitude" || type == "frequency") {
      if (j == rand) {
        playVarSound((j * (dur / 1000)) + j * (ISI / 1000));
      } else {
        playStdSound((j * (dur / 1000)) + j * (ISI / 1000));
      }
    } else if (type == "duration") {
      if (j == rand)
        playVarSound((j * (stdDur / 1000)) + j * (ISI / 1000));
      else if (j < rand)
        playStdSound((j * (stdDur / 1000)) + j * (ISI / 1000));
      else if (j > rand)
        playStdSound(((j - 1) * (stdDur / 1000)) + (varDur / 1000) + j
* (ISI / 1000));
    } else if (type == "gap") {
      if (j == rand)
        playVarNoiseGap((j * (dur / 1000)) + j * (ISI / 1000));
      else
        playStdNoise((j * (dur / 1000)) + j * (ISI / 1000));
    } else if (type == "nduration") {
      if (j == rand)
        playVarNoise((j * (stdDur / 1000)) + j * (ISI / 1000));
      else if (j < rand)
        playStdNoise((j * (stdDur / 1000)) + j * (ISI / 1000));
      else if (j > rand)
        playStdNoise(((j - 1) * (stdDur / 1000)) + (varDur / 1000) + j
* (ISI / 1000));
    }
  }
}
}

```

```

swap = rand + 1;

//after playing the sound, the response buttons are reactivated
if (type != "gap" && type != "nduration") {
    oscillator.onended = () => { //quando l'oscillatore sta suonando il
programma non si ferma, quindi serve questo per riattivare i pulsanti solo
quando finisce
        for (var j = 1; j <= nAFC; j++)
            document.getElementById("button" + j).disabled = false;
        document.getElementById("playTest").disabled = false;
    }
} else {
    source.onended = () => { //quando l'oscillatore sta suonando il
programma non si ferma, quindi serve questo per riattivare i pulsanti solo
quando finisce
        for (var j = 1; j <= nAFC; j++)
            document.getElementById("button" + j).disabled = false;
        document.getElementById("playTest").disabled = false;
    }
}
}

function select(button) {
    for (var j = 1; j <= nAFC; j++)
        document.getElementById("button" + j).disabled = true;
    let element = document.getElementById("alert")
    if (button == swap) {
        element.style.visibility = "visible";
        element.innerText = "Correct!"
        element.classList.remove("alert-danger");
        element.classList.add("alert-success");
    } else {
        element.style.visibility = "visible";
        element.innerText = "Wrong!"
        element.classList.remove("alert-success");
        element.classList.add("alert-danger");
    }
}
}

```

soundsFrequency.js

```

//contesto e dichiarazione variabili da cambiare durante il test,
probabilmente andranno tolte molte variabili globali da qui una volta
terminato l'algoritmo
var context= new AudioContext();

// minimum initial variation

```

```

var varFreq = freq + delta;           // frequency of the variable
var stdFreq = freq;                   // frequency of the standard
var startingDelta = delta;

var stdDur = dur;                      // duration of the standard
var varDur = dur;                      // duration of the variable

var stdAmp = amp;                      // intensity of the standard
var varAmp = amp;                      // intensity of the variable

var stdMod = mod;                      // onset and offset duration of ramp of
the standard
var varMod = mod;                      // onset and offset duration of ramp of
the variable

var swap = -1;                         // position of variable sound
var correctAnsw = 0;                   // number of correct answers

var currentFactor = factor;            // first or second factor, depending on
the number of reversals

// array and variables for data storage
var history = [];                      // will have the answers ('1' if right, '0' if
wrong)
var reversalsPositions = [];           // will have the position of the i-th reversal
in the history array
var i = 0;                              // next index of the array
var countRev = 0;                       // count of reversals
var results = [[], [], [], [], [], [], [], []]; // block, trial, delta,
variable value, variable position, pressed button, correct answer?, reversals
var score = 0                           // final score
var positiveStrike = -1;                 // -1 = unsetted, 0 = negative strike, 1 =
positive strike
var result = "";                         // final results that will be saved on the db

var timestamp = 0;                       // timestamp of the starting of the test
var pressedButton;

//funzione per generare il primo suono
function playVar(time){
  console.log("delta: "+(varAmp-stdAmp));
  console.log("factor: "+currentFactor);

  var volume1 = context.createGain();     //volume
  volume1.gain.setValueAtTime(0, context.currentTime + time); //imposto
volume iniziale a 0
  volume1.connect(context.destination);   //collego all'uscita audio

```

```

    volume1.gain.setTargetAtTime((10**(parseInt(varAmp)/20)),
context.currentTime + time, varMod / 1000); //implemento onset ramp
    volume1.gain.setTargetAtTime(0, context.currentTime + time + (varDur/1000)
- 3 * (varMod/1000), varMod / 1000); //implemento offset ramp (l'espressione
"3 * (Mod/1000)" serve a garantire un raggiungimento del target del 95% circa,
guardare documentazione di setTargetAtTime, in particolare la tabella del
timeconstant)

    oscillator = context.createOscillator();//Creiamo il primo oscillatore
    oscillator.connect(volume1); //Colleghiamo l'oscillatore al
    oscillator.frequency.value = varFreq; //frequency
    oscillator.type = "sine"; // tipo di onda

    oscillator.start(context.currentTime + time); //Facciamo partire
l'oscillatore
    oscillator.stop(context.currentTime + time + (varDur/1000));//Fermiamo
l'oscillatore dopo 1 secondo
}

//funzione per generare il secondo suono
function playStd(time){
    var volume2 = context.createGain(); //volume
    volume2.gain.setValueAtTime(0, context.currentTime + time); //imposto
volume iniziale a 0
    volume2.connect(context.destination); //collego all'uscita audio
    volume2.gain.setTargetAtTime((10**(parseInt(stdAmp)/20)),
context.currentTime + time, stdMod / 1000); //implemento onset ramp
    volume2.gain.setTargetAtTime(0, context.currentTime + time + (stdDur/1000)
- 3 * (stdMod/1000), stdMod / 1000); //implemento offset ramp (l'espressione
"3 * (Mod/1000)" serve a garantire un raggiungimento del target del 95% circa,
guardare documentazione di setTargetAtTime, in particolare la tabella del
timeconstant)

    oscillator = context.createOscillator();//Creiamo il secondo oscillatore
    oscillator.connect(volume2); //Colleghiamo l'oscillatore al
    oscillator.frequency.value = stdFreq; //frequency
    oscillator.type = "sine"; //tipo di onda

    oscillator.start(context.currentTime + time); //Facciamo partire
l'oscillatore
    oscillator.stop(context.currentTime + time + (stdDur/1000));//Fermiamo
l'oscillatore dopo 1 secondo
}

//funzione per randomizzare l'output
function random(){
    var rand = Math.floor(Math.random() * nAFC);// the variable sound will be
the rand-th sound played

```

```

for(var j=0;j<nAFC;j++){
    if(j==rand)
        playVar((j*(dur/1000)) + j*(ISI/1000));
    else
        playStd((j*(dur/1000)) + j*(ISI/1000));
}

swap = rand+1;

//after playing the sound, the response buttons are reactivated
oscillator.onended = () => { //quando l'oscillatore sta suonando il
programma non si ferma, quindi serve questo per riattivare i pulsanti solo
quando finisce
    for(var j=1;j<=nAFC;j++)
        document.getElementById("button"+j).disabled = false;
}
}

function saveResults(){
    //save new data
    results[0][i] = currentBlock;           // block
    results[1][i] = i+1;                    // trial
    results[2][i] = parseFloat(parseInt((varFreq-stdFreq)*1000)/1000); //
approximated delta
    results[3][i] = parseFloat(parseInt(varFreq*1000)/1000);           //
approximated variable value
    results[4][i] = swap;           // variable position
    results[5][i] = pressedButton; // pressed button
    results[6][i] = pressedButton==swap? 1:0; // is the answer correct? 1-
>yes, 0->no
}

//funzione per implementare l'algoritmo SimpleUpDown
function select(button){
    pressedButton = button;
    saveResults();

    switch(algorithm){
        case 'SimpleUpDown':
            nDOWNNoneUP(1);
            break;
        case 'TwoDownOneUp':
            nDOWNNoneUP(2);
            break;
        case 'ThreeDownOneUp':
            nDOWNNoneUP(3);
            break;
    }
}

```

```

        default:
            nDOWNNoneUP(2);
            break;
    }

    results[7][i] = countRev; // reversals counter is updated in nDOWNNoneUP()
function and saved after it

    //increment counter
    i++;

    //use the second factor from now
    if(countRev == reversals)
        currentFactor = secondFactor;

    //end of the test
    if(countRev == reversals+secondReversals){
        //format datas as a csv file
        //format:
block;trials;delta;variableValue;variablePosition;button;correct;reversals;";
        for(var j = 0; j < i; j++){
            result += results[0][j] + ";" + results[1][j] + ";" +
results[2][j] + ";" + results[3][j] + ";"
            result += results[4][j] + ";" + results[5][j] + ";" +
results[6][j] + ";" + results[7][j] + ",";
        }

        //calculate score
        for(var j = countRev - reversalThreshold; j<countRev; j++){
            deltaBefore = results[2][reversalsPositions[j]-1]; //delta before
the reversal
            deltaAfter = results[2][reversalsPositions[j]]; //delta after the
reversal
            score += (deltaBefore + deltaAfter)/2; //average delta of the
reversal
        }
        score /= reversalThreshold; //average deltas of every reversal
        score = parseFloat(parseInt(score*100)/100); //approximate to 2
decimal digits

        //format description as a csv file
        //prima tutti i nomi, poi tutti i dati
        var description =
"&amp;="+amp+"&freq="+freq+"&dur="+dur+"&modu="+mod+/"&phase="+phase+/"&block
s="+blocks+"&delta="+startingDelta+"&nAFC="+nAFC+"&ISI="+ISI+"&ITI="+ITI;
        description +=
"&fact="+factor+"&secFact="+secondFactor+"&rev="+reversals+"&secRev="+secondRe

```



```

versals+"&threshold="+reversalThreshold+"&alg="+algorithm + "&sampleRate=" +
context.sampleRate;

    //pass the datas to the php file
    location.href="php/saveData.php?result="+result+"&timestamp="+timestam
p+"&type=freq"+description+"&currentBlock="+currentBlock+"&score="+score+"&sav
eSettings="+saveSettings;
}
//if the test is not ended
else{

    // disable the response buttons until the new sounds are heard
    for(var j=1;j<=nAFC;j++)
        document.getElementById("button"+j).disabled = true;

    //randomize and play the next sounds
    window.setTimeout("random()", ITI); //next sounds after
interTrialInterval ms
}
}

document.addEventListener('keypress', function keypress(event){
    if(!document.getElementById("button1").disabled){
        if ((event.code >= 'Digit1' && event.code <= 'Digit'+nAFC) ||
(event.code >= 'Numpad1' && event.code <= 'Numpad'+nAFC)) {
            select(event.key)
            console.log('You pressed '+ event.key + ' button');
        }
    }
});

//funzione per implementare l'algorithmo nD1U
function nDOWNNoneUP(n){
    delta = varFreq-stdFreq;

    if(pressedButton == swap){ //correct answer
        history[i] = 1;
        correctAnsw += 1;
        if(correctAnsw == n){ //if there are n consecutive correct answers
            varFreq = stdFreq + (delta/currentFactor);
            correctAnsw = 0;
            if(positiveStrike == 0){
                //there was a reversal
                reversalsPositions[countRev] = i-(n-1); //save the position of
that reversal
                countRev++;
            }
            positiveStrike = 1;

```

```

    }
    if(feedback){
        document.getElementById("correct").style.display="inherit";
        document.getElementById("wrong").style.display="none";
        window.setTimeout("timer()", 500);
    }

}

}else{ //wrong answer
    history[i] = 0;
    correctAnsw = 0;
    varFreq = stdFreq + (delta*currentFactor);

    if(positiveStrike == 1){
        //there was a reversal
        reversalsPositions[countRev] = i;//save the position of that
reversal
        countRev++;
    }
    positiveStrike = 0;

    if(feedback){
        document.getElementById("correct").style.display="none";
        document.getElementById("wrong").style.display="inherit";
        window.setTimeout("timer()", 500);
    }
}
}

//starting function
function start(){
    document.getElementById("StartingWindow").style.display="none"; //starting
window becomes invisible
    document.getElementById("PlayForm").style.display="inherit"; //test
interface becomes visible

    // take the timestamp when the test starts
    var currentdate = new Date();
    timestamp = currentdate.getFullYear()+"-"+(currentdate.getMonth()+1)+"-
"+currentdate.getDate()+"
"+currentdate.getHours()+":"+currentdate.getMinutes()+":"+currentdate.getSeco
nds();

    window.setTimeout("random()", ITI); //test starts after interTrialInterval
ms
}

function timer(){

```

```

document.getElementById("wrong").style.display="none";
document.getElementById("correct").style.display="none";
}

```

soundsDuration.js

```

//contesto e dichiarazione variabili da cambiare durante il test,
//probabilmente andranno tolte molte variabili globali da qui una volta
//terminato l'algoritmo
var context= new AudioContext();

// minimum initial variation
var varFreq = freq;           // frequency of the variable
var stdFreq = freq;          // frequency of the standard
var startingDelta = delta;

var stdDur = dur;             // duration of the standard
var varDur = dur+delta;       // duration of the variable

var stdAmp = amp;             // intensity of the variable
var varAmp = amp;             // intensity of the standard

var stdMod = mod;             // onset and offset duration of ramp of
// the standard
var varMod = mod;             // onset and offset duration of ramp of
// the variable

var swap = -1;                // position of variable sound
var correctAnsw = 0;          // number of correct answers

var currentFactor = factor;    // first or second factor, depending on
// the number of reversals

// array and variables for data storage
var history = [];             // will have the answers ('1' if right, '0' if
// wrong)
var reversalsPositions = [];   // will have the position of the i-th reversal
// in the history array
var i = 0;                    // next index of the array
var countRev = 0;              // count of reversals
var results = [[], [], [], [], [], [], [], []]; // block, trial, delta,
// variable value, variable position, pressed button, correct answer?, reversals
var score = 0                  // final score
var positiveStrike = -1;       // -1 = unsetted, 0 = negative strike, 1 =
// positive strike
var result = "";               // final results that will be saved on the db
var timestamp = 0;             // timestamp of the starting of the test

```

```

var pressedButton;

//funzione per generare il primo suono
function playVar(time){
  console.log("delta: "+(varAmp-stdAmp));
  console.log("factor: "+currentFactor);

  var volume1 = context.createGain(); //volume
  volume1.gain.setValueAtTime(0, context.currentTime + time); //imposto
  volume iniziale a 0
  volume1.connect(context.destination); //collego all'uscita audio
  volume1.gain.setTargetAtTime((10**(parseInt(varAmp)/20)),
context.currentTime + time, varMod / 1000); //implemento onset ramp
  volume1.gain.setTargetAtTime(0, context.currentTime + time + (varDur/1000)
- 3 * (varMod/1000), varMod / 1000); //implemento offset ramp (l'espressione
"3 * (Mod/1000)" serve a garantire un raggiungimento del target del 95% circa,
guardare documentazione di setTargetAtTime, in particolare la tabella del
timeconstant)

  oscillator = context.createOscillator();//Creiamo il primo oscillatore
  oscillator.connect(volume1); //Colleghiamo l'oscillatore al
  oscillator.frequency.value = varFreq; //frequency
  oscillator.type = "sine"; // tipo di onda

  oscillator.start(context.currentTime + time); //Facciamo partire
  l'oscillatore
  oscillator.stop(context.currentTime + time + (varDur/1000));//Fermiamo
  l'oscillatore dopo 1 secondo
}

//funzione per generare il secondo suono
function playStd(time){
  var volume2 = context.createGain(); //volume
  volume2.gain.setValueAtTime(0, context.currentTime + time); //imposto
  volume iniziale a 0
  volume2.connect(context.destination); //collego all'uscita audio
  volume2.gain.setTargetAtTime((10**(parseInt(stdAmp)/20)),
context.currentTime + time, stdMod / 1000); //implemento onset ramp
  volume2.gain.setTargetAtTime(0, context.currentTime + time + (stdDur/1000)
- 3 * (stdMod/1000), stdMod / 1000); //implemento offset ramp (l'espressione
"3 * (Mod/1000)" serve a garantire un raggiungimento del target del 95% circa,
guardare documentazione di setTargetAtTime, in particolare la tabella del
timeconstant)

  oscillator = context.createOscillator();//Creiamo il secondo oscillatore
  oscillator.connect(volume2); //Colleghiamo l'oscillatore al
  oscillator.frequency.value = stdFreq; //frequency
  oscillator.type = "sine"; //tipo di onda

```

```

    oscillator.start(context.currentTime + time);          //Facciamo partire
l'oscillatore
    oscillator.stop(context.currentTime + time + (stdDur/1000)); //Fermiamo
l'oscillatore dopo 1 secondo
}

//funzione per randomizzare l'output
function random(){
    var rand = Math.floor(Math.random() * nAFC); // the variable sound will be
the rand-th sound played

    for(var j=0;j<nAFC;j++){
        if(j==rand)
            playVar((j*(stdDur/1000)) + j*(ISI/1000));
        else if(j<rand)
            playStd((j*(stdDur/1000)) + j*(ISI/1000));
        else if(j>rand)
            playStd(((j-1)*(stdDur/1000)) + (varDur/1000) + j*(ISI/1000));
    }

    swap = rand+1;

    //after playing the sound, the response buttons are reactivated
    oscillator.onended = () => { //quando l'oscillatore sta suonando il
programma non si ferma, quindi serve questo per riattivare i pulsanti solo
quando finisce
        for(var j=1;j<=nAFC;j++){
            document.getElementById("button"+j).disabled = false;
        }
    }
}

function saveResults(){
    //save new data
    results[0][i] = currentBlock;          // block
    results[1][i] = i+1;                   // trial
    results[2][i] = parseFloat(parseInt((varDur-stdDur)*1000)/1000); //
approximated delta
    results[3][i] = parseFloat(parseInt(varDur*1000)/1000); //
approximated variable value
    results[4][i] = swap;                  // variable position
    results[5][i] = pressedButton;        // pressed button
    results[6][i] = pressedButton==swap? 1:0; // is the answer correct? 1-
>yes, 0->no
}

//funzione per implementare l'algoritmo SimpleUpDown
function select(button){

```

```

pressedButton = button;
saveResults();

switch(algorithm){
  case 'SimpleUpDown':
    nDOWNNoneUP(1);
    break;
  case 'TwoDownOneUp':
    nDOWNNoneUP(2);
    break;
  case 'ThreeDownOneUp':
    nDOWNNoneUP(3);
    break;
  default:
    nDOWNNoneUP(2);
    break;
}

results[7][i] = countRev; // reversals counter is updated in nDOWNNoneUP()
function and saved after it

//increment counter
i++;

//use the second factor from now
if(countRev == reversals)
  currentFactor = secondFactor;

//end of the test
if(countRev == reversals+secondReversals){
  //format datas as a csv file
  //format:
block;trials;delta;variableValue;variablePosition;button;correct;reversals;";
  for(var j = 0; j < i; j++){
    result += results[0][j] + ";" + results[1][j] + ";" +
results[2][j] + ";" + results[3][j] + ";"
    result += results[4][j] + ";" + results[5][j] + ";" +
results[6][j] + ";" + results[7][j] + ",";
  }

  //calculate score
  for(var j = countRev - reversalThreshold; j<countRev; j++){
    deltaBefore = results[2][reversalsPositions[j]-1]; //delta before
the reversal
    deltaAfter = results[2][reversalsPositions[j]]; //delta after the
reversal
    score += (deltaBefore + deltaAfter)/2; //average delta of the
reversal

```

```

    }
    score /= reversalThreshold; //average deltas of every reversal
    score = parseFloat(parseInt(score*100)/100); //approximate to 2
decimal digits

    //format description as a csv file
    //prima tutti i nomi, poi tutti i dati
    var description =
"&amp;=" + amp + "&freq=" + freq + "&dur=" + dur + "&modu=" + mod + /*"&phase=" + phase + */"&block
s=" + blocks + "&delta=" + startingDelta + "&nAFC=" + nAFC + "&ISI=" + ISI + "&ITI=" + ITI;
    description +=
"&fact=" + factor + "&secFact=" + secondFactor + "&rev=" + reversals + "&secRev=" + secondRe
versals + "&threshold=" + reversalThreshold + "&alg=" + algorithm + "&sampleRate=" +
context.sampleRate;

    //pass the datas to the php file
    location.href="php/saveData.php?result="+result+"&timestamp="+timestam
p+"&type=dur"+description+"&currentBlock="+currentBlock+"&score="+score+"&save
Settings="+saveSettings;
    }
    //if the test is not ended
    else{

        // disable the response buttons until the new sounds are heard
        for(var j=1;j<=nAFC;j++)
            document.getElementById("button"+j).disabled = true;

        //randomize and play the next sounds
        window.setTimeout("random()", ITI); //next sounds after
interTrialInterval ms
    }
}

document.addEventListener('keypress', function keypress(event){
    if(!document.getElementById("button1").disabled){
        if ((event.code >= 'Digit1' && event.code <= 'Digit'+nAFC) ||
(event.code >= 'Numpad1' && event.code <= 'Numpad'+nAFC)) {
            select(event.key)
            console.log('You pressed ' + event.key + ' button');
        }
    }
});

//funzione per implementare l'algoritmo nD1U
function nDOWNNoneUP(n){
    delta = varDur-stdDur;

    if(pressedButton == swap){ //correct answer

```

```

    history[i] = 1;
    correctAnsw += 1;
    if(correctAnsw == n){ //if there are n consecutive correct answers
        varDur = stdDur + (delta/currentFactor);
        correctAnsw = 0;
        if(positiveStrike == 0){
            //there was a reversal
            reversalsPositions[countRev] = i-(n-1); //save the position of
that reversal
            countRev++;
        }
        positiveStrike = 1;
    }
    if(feedback){
        document.getElementById("correct").style.display="inherit";
        document.getElementById("wrong").style.display="none";
        window.setTimeout("timer()", 500);
    }

}
else{ //wrong answer
    history[i] = 0;
    correctAnsw = 0;
    varDur = stdDur + (delta*currentFactor);

    if(positiveStrike == 1){
        //there was a reversal
        reversalsPositions[countRev] = i; //save the position of that
reversal
        countRev++;
    }
    positiveStrike = 0;

    if(feedback){
        document.getElementById("correct").style.display="none";
        document.getElementById("wrong").style.display="inherit";
        window.setTimeout("timer()", 500);
    }
}
}

//starting function
function start(){
    document.getElementById("StartingWindow").style.display="none"; //starting
window becomes invisible
    document.getElementById("PlayForm").style.display="inherit"; //test
interface becomes visible

    // take the timestamp when the test starts

```



```

    var currentdate = new Date();
    timestamp = currentdate.getFullYear()+"-"+(currentdate.getMonth()+1)+"-
"+currentdate.getDate()+"
"+currentdate.getHours()+":"+currentdate.getMinutes()+":"+currentdate.getSecon
ds();

    window.setTimeout("random()", ITI); //test starts after interTrialInterval
ms
}

function timer(){
    document.getElementById("wrong").style.display="none";
    document.getElementById("correct").style.display="none";
}

```

soundsAmplitude.js

```

//contesto e dichiarazione variabili da cambiare durante il test,
probabilmente andranno tolte molte variabili globali da qui una volta
terminato l'algoritmo
var context = new AudioContext();

// minimum initial variation
var varFreq = freq;           // frequency of the variable
var stdFreq = freq;          // frequency of the standard
var startingDelta = delta;

var stdDur = dur;            // duration of the standard
var varDur = dur;            // duration of the variable

var stdAmp = amp;            // intensity of the standard
var varAmp = amp+delta;      // intensity of the variable

var stdMod = mod;            // onset and offset duration of ramp of
the standard
var varMod = mod;            // onset and offset duration of ramp of
the variable

var swap = -1;                // position of variable sound
var correctAnsw = 0;          // number of correct answers

var currentFactor = factor;   // first or second factor, depending on
the number of reversals

// array and variables for data storage
var history = [];             // will have the answers ('1' if right, '0' if
wrong)

```

```

var reversalsPositions = []; // will have the position of the i-th reversal
in the history array
var i = 0; // next index of the array
var countRev = 0; // count of reversals
var results = [[], [], [], [], [], [], [], []]; // block, trial, delta,
variable value, variable position, pressed button, correct answer?, reversals
var score = 0 // final score
var positiveStrike = -1; // -1 = unsetted, 0 = negative strike, 1 =
positive strike
var result = ""; // final results that will be saved on the db

var timestamp = 0; // timestamp of the starting of the test
var pressedButton;

//funzione per generare il primo suono
function playVar(time){
  console.log("delta: "+(varAmp-stdAmp));
  console.log("factor: "+currentFactor);

  var volume1 = context.createGain(); //volume
  volume1.gain.setValueAtTime(0, context.currentTime + time); //imposto
volume iniziale a 0
  volume1.connect(context.destination); //collego all'uscita audio
  volume1.gain.setTargetAtTime((10*(parseInt(varAmp)/20)),
context.currentTime + time, varMod / 1000); //implemento onset ramp
  volume1.gain.setTargetAtTime(0, context.currentTime + time + (varDur/1000)
- 3 * (varMod/1000), varMod / 1000); //implemento offset ramp (l'espressione
"3 * (Mod/1000)" serve a garantire un raggiungimento del target del 95% circa,
guardare documentazione di setTargetAtTime, in particolare la tabella del
timeconstant)

  oscillator = context.createOscillator();//Creiamo il primo oscillatore
  oscillator.connect(volume1); //Colleghiamo l'oscillatore al
  oscillator.frequency.value = varFreq; //frequency
  oscillator.type = "sine"; // tipo di onda

  oscillator.start(context.currentTime + time); //Facciamo partire
l'oscillatore
  oscillator.stop(context.currentTime + time + (varDur/1000));//Fermiamo
l'oscillatore dopo 1 secondo
}

//funzione per generare il secondo suono
function playStd(time){
  var volume2 = context.createGain(); //volume
  volume2.gain.setValueAtTime(0, context.currentTime + time); //imposto
volume iniziale a 0
  volume2.connect(context.destination); //collego all'uscita audio

```

```

    volume2.gain.setTargetAtTime((10**((parseInt(stdAmp)/20)),
context.currentTime + time, stdMod / 1000); //implemento onset ramp
    volume2.gain.setTargetAtTime(0, context.currentTime + time + (stdDur/1000)
- 3 * (stdMod/1000), stdMod / 1000); //implemento offset ramp (l'espressione
"3 * (Mod/1000)" serve a garantire un raggiungimento del target del 95% circa,
guardare documentazione di setTargetAtTime, in particolare la tabella del
timeconstant)

    oscillator = context.createOscillator();//Creiamo il secondo oscillatore
    oscillator.connect(volume2);           //Colleghiamo l'oscillatore al
    oscillator.frequency.value = stdFreq;  //frequency
    oscillator.type = "sine";              //tipo di onda

    oscillator.start(context.currentTime + time); //Facciamo partire
l'oscillatore
    oscillator.stop(context.currentTime + time + (stdDur/1000)); //Fermiamo
l'oscillatore dopo 1 secondo
}

//funzione per randomizzare l'output
function random(){
    var rand = Math.floor(Math.random() * nAFC); // the variable sound will be
the rand-th sound played

    for(var j=0;j<nAFC;j++){
        if(j==rand)
            playVar((j*(dur/1000)) + j*(ISI/1000));
        else
            playStd((j*(dur/1000)) + j*(ISI/1000));
    }

    swap = rand+1;

    //after playing the sound, the response buttons are reactivated
    oscillator.onended = () => { //quando l'oscillatore sta suonando il
programma non si ferma, quindi serve questo per riattivare i pulsanti solo
quando finisce
        for(var j=1;j<=nAFC;j++)
            document.getElementById("button"+j).disabled = false;
    }
}

function saveResults(){
    //save new data
    results[0][i] = currentBlock; // block
    results[1][i] = i+1; // trial
    results[2][i] = parseFloat(parseInt((varAmp-stdAmp)*1000)/1000); //
approximated delta
}

```

```

    results[3][i] = parseFloat(parseInt(varAmp*1000)/1000);           //
    approximated variable value
    results[4][i] = swap;                                           // variable position
    results[5][i] = pressedButton;                                  // pressed button
    results[6][i] = pressedButton==swap? 1:0;                       // is the answer correct? 1-
    >yes, 0->no
}

//funzione per implementare l'algoritmo SimpleUpDown
function select(button){
    pressedButton = button;
    saveResults();

    switch(algorithm){
        case 'SimpleUpDown':
            nDOWNNoneUP(1);
            break;
        case 'TwoDownOneUp':
            nDOWNNoneUP(2);
            break;
        case 'ThreeDownOneUp':
            nDOWNNoneUP(3);
            break;
        default:
            nDOWNNoneUP(2);
            break;
    }

    results[7][i] = countRev; // reversals counter is updated in nDOWNNoneUP()
    function and saved after it

    //increment counter
    i++;

    //use the second factor from now
    if(countRev == reversals)
        currentFactor = secondFactor;

    //end of the test
    if(countRev == reversals+secondReversals){
        //format datas as a csv file
        //format:
        block;trials;delta;variableValue;variablePosition;button;correct;reversals;";
        for(var j = 0; j < i; j++){
            result += results[0][j] + ";" + results[1][j] + ";" +
results[2][j] + ";" + results[3][j] + ";"
            result += results[4][j] + ";" + results[5][j] + ";" +
results[6][j] + ";" + results[7][j] + ",";

```

```

    }

    //calculate score
    for(var j = countRev - reversalThreshold; j<countRev; j++){
        deltaBefore = results[2][reversalsPositions[j]-1]; //delta before
the reversal
        deltaAfter = results[2][reversalsPositions[j]]; //delta after the
reversal
        score += (deltaBefore + deltaAfter)/2; //average delta of the
reversal
    }
    score /= reversalThreshold; //average deltas of every reversal
    score = parseFloat(parseInt(score*100)/100); //approximate to 2
decimal digits

    //format description as a csv file
    //prima tutti i nomi, poi tutti i dati
    var description =
"&amp;" + amp + "&freq=" + freq + "&dur=" + dur + "&modu=" + mod + "/" + "&phase=" + phase + "/" + "&block
s=" + blocks + "&delta=" + startingDelta + "&nAFC=" + nAFC + "&ISI=" + ISI + "&ITI=" + ITI;
    description +=
"&fact=" + factor + "&secFact=" + secondFactor + "&rev=" + reversals + "&secRev=" + secondRe
versals + "&threshold=" + reversalThreshold + "&alg=" + algorithm + "&sampleRate=" +
context.sampleRate;

    //pass the datas to the php file
    location.href = "php/saveData.php?result=" + result + "&timestamp=" + timestam
p + "&type=" + amp + description + "&currentBlock=" + currentBlock + "&score=" + score + "&save
Settings=" + saveSettings;
    }
    //if the test is not ended
    else{

        // disable the response buttons until the new sounds are heard
        for(var j=1; j<=nAFC; j++)
            document.getElementById("button"+j).disabled = true;

        //randomize and play the next sounds
        window.setTimeout("random()", ITI); //next sounds after
interTrialInterval ms
    }
}

document.addEventListener('keypress', function keypress(event){
    if(!document.getElementById("button1").disabled){
        if ((event.code >= 'Digit1' && event.code <= 'Digit'+nAFC) ||
(event.code >= 'Numpad1' && event.code <= 'Numpad'+nAFC)) {
            select(event.key)
        }
    }
}

```

```

        console.log('You pressed ' + event.key + ' button');
    }
}
});

//funzione per implementare l'algoritmo nD1U
function nDOWNoneUP(n){
    delta = varAmp-stdAmp;

    if(pressedButton == swap){ //correct answer
        history[i] = 1;
        correctAnsw += 1;
        if(correctAnsw == n){ //if there are n consecutive correct answers

            varAmp = stdAmp + (delta/currentFactor);
            correctAnsw = 0;

            if(positiveStrike == 0){
                //there was a reversal
                reversalsPositions[countRev] = i-(n-1); //save the position of
that reversal
                countRev++;
            }
            positiveStrike = 1;
        }
        if(feedback){
            document.getElementById("correct").style.display="inherit";
            document.getElementById("wrong").style.display="none";
            window.setTimeout("timer()", 500);
        }
    }

    }else{ //wrong answer
        history[i] = 0;
        correctAnsw = 0;
        if(stdAmp + (delta*currentFactor)<=0)// varAmp can't be more than 0
            varAmp = stdAmp + (delta*currentFactor);

        if(positiveStrike == 1){
            //there was a reversal
            reversalsPositions[countRev] = i; //save the position of that
reversal
            countRev++;
        }
        positiveStrike = 0;

        if(feedback){
            document.getElementById("correct").style.display="none";
            document.getElementById("wrong").style.display="inherit";

```

```

        window.setTimeout("timer()", 500);
    }
}

//starting function
function start(){
    document.getElementById("StartingWindow").style.display="none"; //starting
window becomes invisible
    document.getElementById("PlayForm").style.display="inherit"; //test
interface becomes visible

    // take the timestamp when the test starts
    var currentdate = new Date();
    timestamp = currentdate.getFullYear()+"-"+(currentdate.getMonth()+1)+"-
"+currentdate.getDate()+"
"+currentdate.getHours()+":"+currentdate.getMinutes()+":"+currentdate.getSecon
ds();

    window.setTimeout("random()", ITI); //test starts after interTrialInterval
ms
}

function timer(){
    document.getElementById("wrong").style.display="none";
    document.getElementById("correct").style.display="none";
}

```

noiseGap.js

```

//contesto e dichiarazione variabili da cambiare durante il test,
probabilmente andranno tolte molte variabili globali da qui una volta
terminato l'algoritmo
var context = new AudioContext();

// minimum initial variation
var startingDelta = delta;

var stdDur = dur; // duration of the standard
var varDur = dur; // duration of the variable

var stdAmp = amp; // intensity of the variable
var varAmp = amp; // intensity of the standard

var stdMod = mod; // onset and offset duration of ramp of
the standard
var varMod = mod; // onset and offset duration of ramp of
the variable

```

```

var swap = -1; // position of variable sound
var correctAnsw = 0; // number of correct answers

var currentFactor = factor; // first or second factor, depending on
the number of reversals

// array and variables for data storage
var history = []; // will have the answers ('1' if right, '0' if
wrong)
var reversalsPositions = []; // will have the position of the i-th reversal
in the history array
var i = 0; // next index of the array
var countRev = 0; // count of reversals
var results = [[], [], [], [], [], [], [], []]; // block, trial, delta,
variable value, variable position, pressed button, correct answer?, reversals
var score = 0 // final score
var positiveStrike = -1; // -1 = unsetted, 0 = negative strike, 1 =
positive strike
var result = ""; // final results that will be saved on the db

var timestamp = 0; // timestamp of the starting of the test
var pressedButton;

var maxDur = 15; // durata massima rumore
var betweenRampDur = 0.03 // durata rampa nel gap

var channels = 2; // numero canali di uscita
var frameCount = context.sampleRate * maxDur; // imposto una duarata massima
del rumore di n (15) secondi
var noiseBuffer = context.createBuffer(channels, frameCount,
context.sampleRate); // creo un nuovo buffer
for (let channel = 0; channel < channels; channel++) { // riempio il buffer
con rumore [-1,+1]
    let nowBuffering = noiseBuffer.getChannelData(channel);
    for (let i = 0; i < frameCount; i++) {
        nowBuffering[i] = Math.random() * 2 - 1;
    }
}

//funzione per generare il rumore con il gap
function playVar(time) {
    console.log("delta: " + delta);
    console.log("factor: " + currentFactor);

    let start1 = 0; // punto
di partenza della prima metà del rumore

```



```

    let end1 = ((varDur / 1000) / 2) - ((delta / 1000) / 2); // punto
di stop della prima metà del rumore
    let start2 = ((varDur / 1000) / 2) + ((delta / 1000) / 2); // punto
di partenza della seconda metà del rumore
    let end2 = varDur / 1000; // punto
di stop della seconda metà del rumore

    var volume1 = context.createGain(); // creo volume
    volume1.gain.setValueAtTime(0, context.currentTime); // imposto volume
iniziale a 0
    volume1.connect(context.destination); // connetto il volume all'uscita
    volume1.gain.setTargetAtTime((10 ** (parseInt(amp) / 20)),
context.currentTime + time + start1, mod / 1000); // eseguo rampa onset
iniziale (inizia da start1 e dura mod ms)
    volume1.gain.setTargetAtTime(0, context.currentTime + time + end1 - 3 *
(betweenRampDur / 1000), betweenRampDur); // eseguo rampa offset del gap
(inizia a end1 e dura betweenRampDur s)
    volume1.gain.setTargetAtTime((10 ** (parseInt(amp) / 20)),
context.currentTime + time + start2, betweenRampDur); // eseguo rampa onset
del gap (inizia a start2 e dura betweenRampDur s)
    volume1.gain.setTargetAtTime(0, context.currentTime + time + end2 - 3 *
(mod / 1000), mod / 1000); // eseguo rampa offset finale (inizia da end2 e
dura mod ms)

    source = context.createBufferSource(); // creo sorgente
    source.buffer = noiseBuffer; // collego i buffer
    source.connect(volume1); // connetto la sorgente al volume
    source.start(context.currentTime + time + start1); // riproduco il rumore
    source.stop(context.currentTime + time + end2); // fermo il rumore
}

//funzione per generare il rumore senza gap
function playStd(time) {
    var volume2 = context.createGain(); // creo volume
    volume2.gain.setValueAtTime(0, context.currentTime); // imposto volume
iniziale a 0
    volume2.connect(context.destination); // connetto il volume all'uscita
    volume2.gain.setTargetAtTime((10 ** (parseInt(amp) / 20)),
context.currentTime + time, stdMod / 1000); // eseguo rampa onset iniziale
    volume2.gain.setTargetAtTime(0, context.currentTime + time + (stdDur /
1000) - 3 * (stdMod / 1000), mod / 1000); // eseguo rampa offset finale

    source = context.createBufferSource(); // creo sorgente
    source.buffer = noiseBuffer; // collego i buffer
    source.connect(volume2); // connetto la sorgente al volume
    source.start(context.currentTime + time); // riproduco il rumore
    source.stop(context.currentTime + time + (stdDur / 1000)); // fermo il
rumore

```

```

}

//funzione per randomizzare l'output
function random() {
    var rand = Math.floor(Math.random() * nAFC);    // the variable sound will
    be the rand-th sound played

    for (var j = 0; j < nAFC; j++) {
        if (j == rand)
            playVar((j * (stdDur / 1000)) + j * (ISI / 1000));
        else if (j < rand)
            playStd((j * (stdDur / 1000)) + j * (ISI / 1000));
        else if (j > rand)
            playStd(((j - 1) * (stdDur / 1000)) + (varDur / 1000) + j * (ISI /
1000));
    }

    swap = rand + 1;

    //after playing the sound, the response buttons are reactivated
    source.onended = () => { //quando l'oscillatore sta suonando il programma
non si ferma, quindi serve questo per riattivare i pulsanti solo quando
finisce
        for (var j = 1; j <= nAFC; j++)
            document.getElementById("button" + j).disabled = false;
    }
}

function saveResults() {
    //save new data
    results[0][i] = currentBlock;           // block
    results[1][i] = i + 1;                 // trial
    results[2][i] = parseFloat(parseInt(delta * 1000) / 1000); //
approximated delta
    results[3][i] = parseFloat(parseInt(delta * 1000) / 1000); //
approximated variable value
    results[4][i] = swap;                 // variable position
    results[5][i] = pressedButton;        // pressed button
    results[6][i] = pressedButton == swap ? 1 : 0; // is the answer correct?
1->yes, 0->no
}

//funzione per implementare l'algoritmo SimpleUpDown
function select(button) {
    pressedButton = button;
    saveResults();

    switch (algorithm) {

```

```

    case 'SimpleUpDown':
        nDOWNNoneUP(1);
        break;
    case 'TwoDownOneUp':
        nDOWNNoneUP(2);
        break;
    case 'ThreeDownOneUp':
        nDOWNNoneUP(3);
        break;
    default:
        nDOWNNoneUP(2);
        break;
}

results[7][i] = countRev; // reversals counter is updated in nDOWNNoneUP()
function and saved after it

//increment counter
i++;

//use the second factor from now
if (countRev == reversals)
    currentFactor = secondFactor;

//end of the test
if (countRev == reversals + secondReversals) {
    //format datas as a csv file
    //format:
block;trials;delta;variableValue;variablePosition;button;correct;reversals;";
    for (var j = 0; j < i; j++) {
        result += results[0][j] + ";" + results[1][j] + ";" +
results[2][j] + ";" + results[3][j] + ";"
        result += results[4][j] + ";" + results[5][j] + ";" +
results[6][j] + ";" + results[7][j] + ",";
    }

    //calculate score
    for (var j = countRev - reversalThreshold; j < countRev; j++) {
        deltaBefore = results[2][reversalsPositions[j] - 1]; //delta
before the reversal
        deltaAfter = results[2][reversalsPositions[j]]; //delta after the
reversal
        score += (deltaBefore + deltaAfter) / 2; //average delta of the
reversal
    }
    score /= reversalThreshold; //average deltas of every reversal
    score = parseFloat(parseInt(score * 100) / 100); //approximate to 2
decimal digits

```

```

        //format description as a csv file
        //prima tutti i nomi, poi tutti i dati
        var description = "&amp;=" + amp + "&freq=" + freq + "&dur=" + dur +
"&modu=" + mod + /*"&phase="+phase+*/"&blocks=" + blocks + "&delta=" +
startingDelta + "&nAFC=" + nAFC + "&ISI=" + ISI + "&ITI=" + ITI;
        description += "&fact=" + factor + "&secFact=" + secondFactor +
"&rev=" + reversals + "&secRev=" + secondReversals + "&threshold=" +
reversalThreshold + "&alg=" + algorithm + "&sampleRate=" + context.sampleRate;

        //pass the datas to the php file
        location.href = "php/saveData.php?result=" + result + "&timestamp=" +
timestamp + "&type=gap" + description + "&currentBlock=" + currentBlock +
"&score=" + score + "&saveSettings=" + saveSettings;
    }
    //if the test is not ended
    else {

        // disable the response buttons until the new sounds are heard
        for (var j = 1; j <= nAFC; j++)
            document.getElementById("button" + j).disabled = true;

        //randomize and play the next sounds
        window.setTimeout("random()", ITI); //next sounds after
interTrialInterval ms
    }
}

document.addEventListener('keypress', function keypress(event) {
    if (!document.getElementById("button1").disabled) {
        if ((event.code >= 'Digit1' && event.code <= 'Digit' + nAFC) ||
(event.code >= 'Numpad1' && event.code <= 'Numpad' + nAFC)) {
            select(event.key)
            console.log('You pressed ' + event.key + ' button');
        }
    }
});

//funzione per implementare l'algoritmo nD1U
function nDOWNoneUP(n) {
    if (pressedButton == swap) { //correct answer
        history[i] = 1;
        correctAnsw += 1;
        if (correctAnsw == n) { //if there are n consecutive correct answers
            delta = delta / currentFactor;
            correctAnsw = 0;
            if (positiveStrike == 0) {
                //there was a reversal
            }
        }
    }
}

```

```

        reversalsPositions[countRev] = i - (n - 1); //save the position
of that reversal
        countRev++;
    }
    positiveStrike = 1;
}
if (feedback) {
    document.getElementById("correct").style.display = "inherit";
    document.getElementById("wrong").style.display = "none";
    window.setTimeout("timer()", 500);
}

} else { //wrong answer
    history[i] = 0;
    correctAnsw = 0;
    delta = delta * currentFactor;

    if (positiveStrike == 1) {
        //there was a reversal
        reversalsPositions[countRev] = i; //save the position of that
reversal
        countRev++;
    }
    positiveStrike = 0;

    if (feedback) {
        document.getElementById("correct").style.display = "none";
        document.getElementById("wrong").style.display = "inherit";
        window.setTimeout("timer()", 500);
    }
}
}

//starting function
function start() {
    document.getElementById("StartingWindow").style.display = "none";
//starting window becomes invisible
    document.getElementById("PlayForm").style.display = "inherit"; //test
interface becomes visible

    // take the timestamp when the test starts
    var currentdate = new Date();
    timestamp = currentdate.getFullYear() + "-" + (currentdate.getMonth() + 1)
+ "-" + currentdate.getDate() + " " + currentdate.getHours() + ":" +
currentdate.getMinutes() + ":" + currentdate.getSeconds();

    window.setTimeout("random()", ITI); //test starts after interTrialInterval
ms

```

```

}

function timer() {
  document.getElementById("wrong").style.display = "none";
  document.getElementById("correct").style.display = "none";
}

```

noisesDuration.js

```

//contesto e dichiarazione variabili da cambiare durante il test,
//probabilmente andranno tolte molte variabili globali da qui una volta
//terminato l'algoritmo
var context = new AudioContext();

// minimum initial variation
var varFreq = freq;           // frequency of the variable
var stdFreq = freq;          // frequency of the standard
var startingDelta = delta;

var stdDur = dur;            // duration of the standard
var varDur = dur + delta;    // duration of the variable

var stdAmp = amp;           // intensity of the variable
var varAmp = amp;           // intensity of the standard

var stdMod = mod;           // onset and offset duration of ramp of
// the standard
var varMod = mod;           // onset and offset duration of ramp of
// the variable

var swap = -1;              // position of variable sound
var correctAnsw = 0;        // number of correct answers

var currentFactor = factor;  // first or second factor, depending on
// the number of reversals

// array and variables for data storage
var history = [];           // will have the answers ('1' if right, '0' if
// wrong)
var reversalsPositions = []; // will have the position of the i-th reversal
// in the history array
var i = 0;                  // next index of the array
var countRev = 0;           // count of reversals
var results = [[], [], [], [], [], [], [], []]; // block, trial, delta,
// variable value, variable position, pressed button, correct answer?, reversals
var score = 0               // final score
var positiveStrike = -1;    // -1 = unsetted, 0 = negative strike, 1 =
// positive strike

```

```

var result = ""; // final results that will be saved on the db

var timestamp = 0; // timestamp of the starting of the test
var pressedButton;

var maxDur = 15; // durata massima rumore

var channels = 2; // numero canali di uscita
var frameCount = context.sampleRate * maxDur; // imposto una duarata massima
del rumore di n (15) secondi
var noiseBuffer = context.createBuffer(channels, frameCount,
context.sampleRate); // creo un nuovo buffer
for (let channel = 0; channel < channels; channel++) { // riempio il buffer
con rumore [-1,+1]
    let nowBuffering = noiseBuffer.getChannelData(channel);
    for (let i = 0; i < frameCount; i++) {
        nowBuffering[i] = Math.random() * 2 - 1;
    }
}

//funzione per generare il primo suono
function playVar(time) {
    console.log("delta: " + (varDur - stdDur));
    console.log("factor: " + currentFactor);

    var volume1 = context.createGain(); // creo volume
    volume1.gain.setValueAtTime(0, context.currentTime); // imposto volume
iniziale a 0
    volume1.connect(context.destination); // connetto il volume all'uscita
    volume1.gain.setTargetAtTime((10 ** (parseInt(varAmp) / 20)),
context.currentTime + time, varMod / 1000); // eseguo rampa onset iniziale
    volume1.gain.setTargetAtTime(0, context.currentTime + time + (varDur /
1000) - 3 * (varMod / 1000), varMod / 1000); // eseguo rampa offset finale

    source = context.createBufferSource(); // creo sorgente
    source.buffer = noiseBuffer; // collego i buffer
    source.connect(volume1); // connetto la sorgente al volume
    source.start(context.currentTime + time); // riproduco il rumore
    source.stop(context.currentTime + time + (varDur / 1000)); // fermo il
rumore
}

//funzione per generare il secondo suono
function playStd(time) {
    var volume2 = context.createGain(); // creo volume
    volume2.gain.setValueAtTime(0, context.currentTime); // imposto volume
iniziale a 0
    volume2.connect(context.destination); // connetto il volume all'uscita

```

```

    volume2.gain.setTargetAtTime((10 ** (parseInt(stdAmp) / 20)),
context.currentTime + time, stdMod / 1000); // eseguo rampa onset iniziale
    volume2.gain.setTargetAtTime(0, context.currentTime + time + (stdDur /
1000) - 3 * (stdMod / 1000), stdMod / 1000); // eseguo rampa offset finale

    source = context.createBufferSource(); // creo sorgente
    source.buffer = noiseBuffer; // collego i buffer
    source.connect(volume2); // connetto la sorgente al volume
    source.start(context.currentTime + time); // riproduco il rumore
    source.stop(context.currentTime + time + (stdDur / 1000)); // fermo il
rumore
}

//funzione per randomizzare l'output
function random() {
    var rand = Math.floor(Math.random() * nAFC); // the variable sound will be
the rand-th sound played

    for (var j = 0; j < nAFC; j++) {
        if (j == rand)
            playVar((j * (stdDur / 1000)) + j * (ISI / 1000));
        else if (j < rand)
            playStd((j * (stdDur / 1000)) + j * (ISI / 1000));
        else if (j > rand)
            playStd(((j - 1) * (stdDur / 1000)) + (varDur / 1000) + j * (ISI /
1000));
    }

    swap = rand + 1;

    //after playing the sound, the response buttons are reactivated
    source.onended = () => { //quando l'oscillatore sta suonando il programma
non si ferma, quindi serve questo per riattivare i pulsanti solo quando
finisce
        for (var j = 1; j <= nAFC; j++)
            document.getElementById("button" + j).disabled = false;
    }
}

function saveResults() {
    //save new data
    results[0][i] = currentBlock; // block
    results[1][i] = i + 1; // trial
    results[2][i] = parseFloat(parseInt((varDur - stdDur) * 1000) / 1000); //
approximated delta
    results[3][i] = parseFloat(parseInt(varDur * 1000) / 1000); //
approximated variable value
    results[4][i] = swap; // variable position
}

```



```

    results[5][i] = pressedButton;           // pressed button
    results[6][i] = pressedButton == swap ? 1 : 0; // is the answer correct?
1->yes, 0->no
}

//funzione per implementare l'algoritmo SimpleUpDown
function select(button) {
    pressedButton = button;
    saveResults();

    switch (algorithm) {
        case 'SimpleUpDown':
            nDOWNNoneUP(1);
            break;
        case 'TwoDownOneUp':
            nDOWNNoneUP(2);
            break;
        case 'ThreeDownOneUp':
            nDOWNNoneUP(3);
            break;
        default:
            nDOWNNoneUP(2);
            break;
    }

    results[7][i] = countRev; // reversals counter is updated in nDOWNNoneUP()
function and saved after it

    //increment counter
    i++;

    //use the second factor from now
    if (countRev == reversals)
        currentFactor = secondFactor;

    //end of the test
    if (countRev == reversals + secondReversals) {
        //format datas as a csv file
        //format:
block;trials;delta;variableValue;variablePosition;button;correct;reversals;";
        for (var j = 0; j < i; j++) {
            result += results[0][j] + ";" + results[1][j] + ";" +
results[2][j] + ";" + results[3][j] + ";"
            result += results[4][j] + ";" + results[5][j] + ";" +
results[6][j] + ";" + results[7][j] + ",";
        }

        //calculate score

```

```

        for (var j = countRev - reversalThreshold; j < countRev; j++) {
            deltaBefore = results[2][reversalsPositions[j] - 1]; //delta
before the reversal
            deltaAfter = results[2][reversalsPositions[j]]; //delta after the
reversal
            score += (deltaBefore + deltaAfter) / 2; //average delta of the
reversal
        }
        score /= reversalThreshold; //average deltas of every reversal
        score = parseFloat(parseInt(score * 100) / 100); //approximate to 2
decimal digits

        //format description as a csv file
        //prima tutti i nomi, poi tutti i dati
        var description = "&amp;" + amp + "&freq=" + freq + "&dur=" + dur +
"&modu=" + mod + /*"&phase="+phase+*/"&blocks=" + blocks + "&delta=" +
startingDelta + "&nAFC=" + nAFC + "&ISI=" + ISI + "&ITI=" + ITI;
        description += "&fact=" + factor + "&secFact=" + secondFactor +
"&rev=" + reversals + "&secRev=" + secondReversals + "&threshold=" +
reversalThreshold + "&alg=" + algorithm + "&sampleRate=" + context.sampleRate;

        //pass the datas to the php file
        location.href = "php/saveData.php?result=" + result + "&timestamp=" +
timestamp + "&type=ndur" + description + "&currentBlock=" + currentBlock +
"&score=" + score + "&saveSettings=" + saveSettings;
    }
    //if the test is not ended
    else {

        // disable the response buttons until the new sounds are heard
        for (var j = 1; j <= nAFC; j++)
            document.getElementById("button" + j).disabled = true;

        //randomize and play the next sounds
        window.setTimeout("random()", ITI); //next sounds after
interTrialInterval ms
    }
}

document.addEventListener('keypress', function keypress(event) {
    if (!document.getElementById("button1").disabled) {
        if ((event.code >= 'Digit1' && event.code <= 'Digit' + nAFC) ||
(event.code >= 'Numpad1' && event.code <= 'Numpad' + nAFC)) {
            select(event.key)
            console.log('You pressed ' + event.key + ' button');
        }
    }
});

```

```

//funzione per implementare l'algoritmo nD1U
function nDOWNNoneUP(n) {
    delta = varDur - stdDur;

    if (pressedButton == swap) { //correct answer
        history[i] = 1;
        correctAnsw += 1;
        if (correctAnsw == n) { //if there are n consecutive correct answers
            varDur = stdDur + (delta / currentFactor);
            correctAnsw = 0;
            if (positiveStrike == 0) {
                //there was a reversal
                reversalsPositions[countRev] = i - (n - 1); //save the position
of that reversal
                countRev++;
            }
            positiveStrike = 1;
        }
        if (feedback) {
            document.getElementById("correct").style.display = "inherit";
            document.getElementById("wrong").style.display = "none";
            window.setTimeout("timer()", 500);
        }
    } else { //wrong answer
        history[i] = 0;
        correctAnsw = 0;
        varDur = stdDur + (delta * currentFactor);

        if (positiveStrike == 1) {
            //there was a reversal
            reversalsPositions[countRev] = i; //save the position of that
reversal
            countRev++;
        }
        positiveStrike = 0;

        if (feedback) {
            document.getElementById("correct").style.display = "none";
            document.getElementById("wrong").style.display = "inherit";
            window.setTimeout("timer()", 500);
        }
    }
}

//starting function
function start() {

```

```

document.getElementById("StartingWindow").style.display = "none";
//starting window becomes invisible
document.getElementById("PlayForm").style.display = "inherit"; //test
interface becomes visible

// take the timestamp when the test starts
var currentdate = new Date();
timestamp = currentdate.getFullYear() + "-" + (currentdate.getMonth() + 1)
+ "-" + currentdate.getDate() + " " + currentdate.getHours() + ":" +
currentdate.getMinutes() + ":" + currentdate.getSeconds();

window.setTimeout("random()", ITI); //test starts after interTrialInterval
ms
}

function timer() {
document.getElementById("wrong").style.display = "none";
document.getElementById("correct").style.display = "none";
}

```

changePsw.php

```

<?php
try{
    session_start();
    include "config.php";

    //sql injections handling
    $elements = ['oldPsw', 'newPsw'];
    $characters = ['"', '\\', chr(0)];
    $specialCharacters = false;
    foreach($elements as $elem){
        $_POST[$elem] = str_replace('"', "'", $_POST[$elem]);
        foreach($characters as $char)
            $specialCharacters |= is_numeric(strpos($_POST[$elem],
$char));
    }

    if($specialCharacters)
        header("Location: ../userSettings.php?&err=0");
    else{
        // connessione al database
        $conn = new mysqli($host, $user, $password, $dbname);
        if ($conn->connect_errno)
            throw new Exception('DB connection failed');
        mysqli_set_charset($conn, "utf8");
        // query
        $oldPsw = $_POST['oldPsw'];

```

```

        $newPsw = $_POST['newPsw'];
        $sql = "SELECT password , email FROM account WHERE Username
= '$_SESSION['usr'].'' AND password=SHA2('$oldPsw', 256)";
        $result=$conn->query($sql);
        $row = $result->fetch_assoc();
        $psw = $row['password'];
        $email =$row['email'];

        if($result->num_rows>0){

                $sql ="UPDATE account SET password = SHA2('$newPsw',
256) WHERE username= '$_SESSION['usr'].''";
                $conn->query($sql);
                header('Location: ../userSettings.php?err=3');
        }
        else
                header('Location: ../userSettings.php?err=2');
        mail($email,'Password changing','you have correctly changed the
password');
    }
}catch(Exception $e){
    header("Location: ../index.php?err=db");
}
?>

```

Download.php

```

<?php
    try{
        include "config.php";
        session_start();

        if(!isset($_GET['format']) || ($_GET['format']!="complete" &&
$_GET['format']!="reduced"))
            header("Location: ../index.php");
        else{
            //apro la connessione con la sessione e col db
            $conn = new mysqli($host, $user, $password, $dbname);
            if ($conn->connect_errno)
                throw new Exception('DB connection failed');
            mysqli_set_charset($conn, "utf8");

            //prendo i dati del guest
            if(!isset($_SESSION['idGuestTest']) &&
!isset($_SESSION['idGuest'])){
                header("Location: ../index.php?err=2");
            }else{

```

```

        if(isset($_SESSION['idGuest']))
            $id = $_SESSION['idGuest'];
        else if(isset($_SESSION['idGuestTest']))
            $id = $_SESSION['idGuestTest'];

        if(isset($_SESSION['name']))
            $sql = "SELECT name, surname, age, gender FROM guest WHERE
ID='$id'";
        else
            $sql = "SELECT name, surname, gender, date FROM guest
INNER JOIN account ON account.Guest_ID = guest.ID WHERE ID='$id'";
        $result = $conn->query($sql);
        $row = $result->fetch_assoc();

        //se il test è stato fatto dal guest dell'account loggato, la
sua età viene calcolata dalla data di nascita
        if(!isset($_SESSION['name']))
            $age = date_diff(date_create($row['date']),
date_create('now'))->y;
        else
            $age = $row['age'];

        //creo e apro il file csv
        if($_GET['format']=="complete")
            $path = "results.csv";
        else
            $path = "reducedResults.csv";
        $txt = fopen($path, "w") or die("Unable to open file!");
        fwrite($txt, chr(0xEF).chr(0xBB).chr(0xBF)); //utf8 encoding

        //scrivo il nome delle colonne
        $line = "Name;Surname;Age;Gender;Test Type;Timestamp;Sample
Rate;Amplitude;Frequency;Duration;n. of blocks;nAFC;ISI;";
        $line .= "ITI;First factor;First reversals;Second
factor;Second reversals;reversal threshold;algorithm;";
        if($_GET['format']=="complete")
            $line .= "block;trials;delta;variable;Variable
Position;Pressed button;correct?;reversals\n";
        else
            $line .= "block;threshold\n";

        fwrite($txt, $line);

        //valore della prima parte (quella fissa che va ripetuta)
        $firstValues =
$row["name"].".".$row["surname"].".".$age.".".$row["gender"].".".$_SESSION["ty
pe"].".".$_SESSION["time"].".".$_SESSION["sampleRate"].".".$_SESSION["amp"].";
";

```

```

        $firstValues .=
$_SESSION["freq"].";".$_SESSION["dur"].";".$_SESSION["blocks"].";".$_SESSION["
nAFC"].";".$_SESSION["ISI"].";".$_SESSION["ITI"].";";
        $firstValues .=
$_SESSION["fact"].";".$_SESSION["rev"].";".$_SESSION["secFact"].";".$_SESSION["
secRev"].";".$_SESSION["thr"].";".$_SESSION["alg"];

        if($_GET['format']=="complete"){
            //parte variabile e scrittura su file
            $results = explode(",", $_SESSION["results"]);
            //results sarà nella forma
["b11;tr1;del1;var1;varpos1;but1;cor1;rev1", "b12;tr2;...", ...]
            for($i = 0;$i<count($results)-1;$i++){
                fwrite($txt, $firstValues.";");//scrivo i valori fissi
                fwrite($txt, $results[$i]);//scrivo i valori variabili
                fwrite($txt, "\n");//vado all'altra linea
            }
        }else{
            $results = explode(";", $_SESSION["score"]);
            for($i = 0;$i<count($results);$i++){
                fwrite($txt, $firstValues.";");//scrivo i valori fissi
                fwrite($txt, ($i+1).";");//scrivo il blocco
                fwrite($txt, $results[$i]);//scrivo lo score del
blocco

                fwrite($txt, "\n");//vado all'altra linea
            }
        }

        fclose($txt);
        /*scrittura su file (per disattivare togliere uno slash da
questo commento)
        header('Content-Description: File Transfer');
        header('Content-Disposition: attachment;
filename='.basename($path));
        header('Expires: 0');
        header('Cache-Control: must-revalidate');
        header('Pragma: public');
        header('Content-Length: ' . filesize($path));
        header("Content-Type: text/plain");
        readfile($path);
        /**/
        unlink($path);//elimino il file dal server
    }
}
}catch(Exception $e){
    header("Location: ../index.php?err=db");
}
?>

```

downloadAll.php

```
<?php
function writeResults($txt, $firstValues, $results){
    //results sarà nella forma
    ["b11;tr1;del1;var1;varpos1;but1;cor1;rev1", "b12;tr2;...", ...]
    for($i = 0;$i<count($results)-1;$i++){
        fwrite($txt, $firstValues.";");
        fwrite($txt, $results[$i]);
        fwrite($txt, "\n");//vado all'altra linea
    }
}

try{
    //apro la connessione con la sessione e col db
    include "config.php";
    session_start();

    $conn = new mysqli($host, $user, $password, $dbname);
    if ($conn->connect_errno)
        throw new Exception('DB connection failed');
    mysqli_set_charset($conn, "utf8");

    //prendo i dati del guest
    $usr = $_SESSION['usr'];
    $id = $_SESSION['idGuest'];

    //controllo di sicurezza
    $sql = "SELECT Type FROM account WHERE Guest_ID='$id' AND
Username='$usr'";
    $result=$conn->query($sql);
    $row=$result->fetch_assoc();
    if($row['Type'] == 1){

        //creo e apro il file csv
        $path = "allResults.csv";
        $txt = fopen($path, "w") or die("Unable to open file!");
        fwrite($txt, chr(0xEF).chr(0xBB).chr(0xBF)); //utf8 encoding

        //scrivo il nome delle colonne
        $line = "Name;Surname;Age;Gender;Test Count;Test
Type;Timestamp;Sample Rate;Amplitude;Frequency;Duration;n. of blocks;";
        $line .= "nAFC;ISI;ITI;First factor;First reversals;Second
factor;Second reversals;reversal threshold;algorithm;";
        $line .= "block;trials;delta;variable;Variable Position;Pressed
button;correct?;reversals\n";
    }
}
```



```

        fwrite($txt, $line);

        //metto i dati dei guest collegati
        $sql = "SELECT guest.Name as name, guest.Surname as surname,
guest.Age as age, guest.Gender as gender,
                test.Test_count as count, test.Type as type,
test.Timestamp as time, test.Amplitude as amp,
                test.Frequency as freq, test.Duration as dur,
test.SampleRate as sampleRate, test.blocks as blocks, test.nAFC as nafc,
                test.ISI as isi, test.ITI as iti, test.Factor as fact,
test.Reversal as rev, test.SecFactor as secfact,
                test.SecReversal as secrev, test.Threshold as thr,
test.Algorithm as alg, test.Result as results,
                account.Date as date

        FROM guest
        INNER JOIN test ON guest.ID=test.Guest_ID
        LEFT JOIN account ON guest.ID=account.Guest_ID;";
        $result = $conn->query($sql);

        while($row = $result->fetch_assoc()){
            if($row['date']!="")
                $age = strval(date_diff(date_create($row['date']),
date_create('now'))->y);
            else if(strval($row['age'])!='0')
                $age = strval($row['age']);
            else
                $age = "";

            //valore della prima parte (quella fissa che va ripetuta)
            $firstValues =
$row["name"].".".$row["surname"].".".$age.".".$row["gender"].".".$row["count"]
.".".$row["type"].".".$row["time"].".".$row["sampleRate"].".";
            $firstValues .=
$row["amp"].".".$row["freq"].".".$row["dur"].".".$row["blocks"].".".$row["nafc"]
.".".$row["isi"].".".$row["iti"].".";
            $firstValues .=
$row["fact"].".".$row["rev"].".".$row["secfact"].".".$row["secrev"].".".$row["
thr"].".".$row["alg"];

            //parte variabile e scrittura su file
            $results = explode(",", $row["results"]);
            writeResults($txt, $firstValues, $results);
        }

        fclose($txt);

```

```

        /*scrittura su file (per disattivare togliere uno slash da questo
commento)
        header('Content-Description: File Transfer');
        header('Content-Disposition: attachment;
filename='.basename($path));
        header('Expires: 0');
        header('Cache-Control: must-revalidate');
        header('Pragma: public');
        header('Content-Length: ' . filesize($path));
        header("Content-Type: text/plain");
        readfile($path);
        /**/
        unlink($path);//elimino il file dal server
    }
    else{//tentativo di accesso senza permessi: scrivo nel file log
        date_default_timezone_set('Europe/Rome');
        $date = date('Y/m/d h:i:s a', time());

        $txt = fopen("files/log.txt", "a") or die("Unable to open file!");

        fwrite($txt, "Attempt to access downloadAll.php without permission
- timestamp: ".$date);
        if(isset($_SESSION['usr']))
            fwrite($txt, " - username: ".$_SESSION['usr']);
        fwrite($txt, "\n");

        fclose($txt);
        header("Location: ../index.php?err=1");
    }
}
}catch(Exception $e){
    header("Location: ../index.php?err=db");
}
?>

```

DownloadYours.php

```

<?php
function addMine($conn, $txt, $usr){
    //prendo i dati dei test collegati al guest dell'account
    $sql = "SELECT guest.Name as name, guest.Surname as surname,
guest.Gender as gender,
            test.Test_count as count, test.Type as type, test.Timestamp as
time, test.Amplitude as amp, test.Frequency as freq, test.Duration as dur,
test.SampleRate as sampleRate,
            test.blocks as blocks, test.nAFC as nafc, test.ISI as isi,
test.ITI as iti, test.Factor as fact, test.Reversal as rev,
            test.SecFactor as secfact, test.SecReversal as secrev,
test.Threshold as thr, test.Algorithm as alg, test.Result as results,

```

```

        account.date as date

        FROM account
        INNER JOIN guest ON account.Guest_ID=guest.ID
        INNER JOIN test ON guest.ID=test.Guest_ID

        WHERE account.username='$usr';
$result = $conn->query($sql);

//parte variabile e scrittura su file
while($row = $result->fetch_assoc()){
    //valore della prima parte (quella fissa che va ripetuta)
    $age = date_diff(date_create($row['date']), date_create('now'))-
>y;

    $firstValues =
$row["name"].".".$row["surname"].".".$age.".".$row["gender"].".".$row["count"]
.".".$row["type"].".".$row["time"].".".$row["sampleRate"].".";
    $firstValues .=
$row["amp"].".".$row["freq"].".".$row["dur"].".".$row["blocks"].".".$row["nafc
"].".".$row["isi"].".".$row["iti"].".";
    $firstValues .=
$_SESSION["fact"].".".$_SESSION["rev"].".".$_SESSION["secFact"].".".$_SESSION[
"secRev"].".".$_SESSION["thr"].".".$_SESSION["alg"];

    $results = explode(",", $row["results"]);
    writeResults($txt, $firstValues, $results);
}
}

function writeResults($txt, $firstValues, $results){
    //results sarà nella forma
["b11;tr1;del1;var1;varpos1;but1;cor1;rev1", "b12;tr2;...", ...]
    for($i = 0;$i<count($results)-1;$i++){
        fwrite($txt, $firstValues.";");
        fwrite($txt, $results[$i]);
        fwrite($txt, "\n");//vado all'altra linea
    }
}

try{
    //apro la connessione con la sessione e col db
    include "config.php";
    session_start();

    $conn = new mysqli($host, $user, $password, $dbname);
    if ($conn->connect_errno)
        throw new Exception('DB connection failed');
    mysqli_set_charset($conn, "utf8");
}

```

```

//prendo i dati del guest
$usr = $_SESSION['usr'];
$id = $_SESSION['idGuest'];

//creo e apro il file csv
$path = "yourResults.csv";
$txt = fopen($path, "w") or die("Unable to open file!");
fwrite($txt, chr(0xEF).chr(0xBB).chr(0xBF)); //utf8 encoding

//scrivo il nome delle colonne
$line = "Name;Surname;Age;Gender;Test Count;Test Type;Timestamp;Sample
Rate;Amplitude;Frequency;Duration;n. of blocks;";
$line .= "nAFC;ISI;ITI;First factor;First reversals;Second
factor;Second reversals;reversal threshold;algorithm;";
$line .= "block;trials;delta;variable;Variable Position;Pressed
button;correct?;reversals\n";

fwrite($txt, $line);

//metto i dati dei test dell'utente, se vanno messi
if(isset($_GET['all']) && $_GET['all']==1)
    addMine($conn, $txt, $usr);

//metto i dati dei guest collegati
$sql = "SELECT guest.Name as name, guest.Surname as surname, guest.Age
as age, guest.Gender as gender,
        test.Test_count as count, test.Type as type, test.Timestamp as
time, test.Amplitude as amp, test.Frequency as freq,
        test.Duration as dur, test.SampleRate as sampleRate,
test.blocks as blocks, test.nAFC as nafc, test.ISI as isi, test.ITI as iti,
test.Factor as fact, test.Reversal as rev,
        test.SecFactor as secfact, test.SecReversal as secrev,
test.Threshold as thr, test.Algorithm as alg, test.Result as results

        FROM account
        INNER JOIN guest ON account.Username=guest.fk_guest
        INNER JOIN test ON guest.ID=test.Guest_ID

        WHERE account.username='$usr'";
$result = $conn->query($sql);

while($row = $result->fetch_assoc()){
    //valore della prima parte (quella fissa che va ripetuta)
    $firstValues =
$row["name"].".".$row["surname"].".".$row["age"].".".$row["gender"].".".$row["
count"].".".$row["type"].".";

```

```

        $firstValues .=
$row["time"].".".$row["sampleRate"].".".$row["amp"].".".$row["freq"].".".$row[
"dur"].".".$row["blocks"].".".$row["nafc"].".".$row["isi"].".".$row["iti"]."."
;
        $firstValues .=
$row["fact"].".".$row["rev"].".".$row["secfact"].".".$row["secrev"].".".$row["
thr"].".".$row["alg"];

        //parte variabile e scrittura su file
        $results = explode(",", $row["results"]);
        writeResults($txt, $firstValues, $results);
    }

    fclose($txt);
    /**scrittura su file (per disattivare togliere uno slash da questo
commento)
    header('Content-Description: File Transfer');
    header('Content-Disposition: attachment; filename='.basename($path));
    header('Expires: 0');
    header('Cache-Control: must-revalidate');
    header('Pragma: public');
    header('Content-Length: ' . filesize($path));
    header("Content-Type: text/plain");
    readfile($path);
    /**/
    unlink($path);**/elimino il file dal server
}catch(Exception $e){
    header("Location: ../index.php?err=db");
}
?>

```

Log.php

```

<?php
    try{
        include "config.php";
        //apro la sessione per comunicare con le altre pagine del sito
        session_start();

        //sql injections handling
        $elements = ['usr', 'psw'];
        $characters = ['"', '\\', chr(0)];
        $specialCharacters = false;
        foreach($elements as $elem){
            $_POST[$elem] = str_replace("'", "", $_POST[$elem]);
            foreach($characters as $char)
                $specialCharacters |= is_numeric(strpos($_POST[$elem],
$char));

```

```

    }

    if($specialCharacters)
        header("Location: ../login.php?&err=0");
    else{
        //apro la connessione con il db
        $conn = new mysqli($host, $user, $password, $dbname);

        //controllo se è andata a buon fine
        if ($conn->connect_errno)
            throw new Exception('DB connection failed');

        //uso codifica utf8 per comunicare col db
        mysqli_set_charset($conn, "utf8");

        //recupero username e password dal form di registrazione
        $usr = $_POST['usr'];
        $psw = $_POST['psw'];

        //controllo se esiste
        $sql = "SELECT Guest_ID FROM account WHERE Username='$usr' AND
Password=SHA2('$psw', 256)";

        $result=$conn->query($sql);

        if($result->num_rows>0){
            $row=$result->fetch_assoc();

            //faccio sapere alle altre pagine quale utente è loggato
            $_SESSION['usr'] = $usr;
            $_SESSION['idGuest'] = $row['Guest_ID'];

            $conn->close();
            header('Location: ../index.php');
        }else{
            $conn->close();
            header('Location: ../login.php?err=1');
        }
    }
}
}catch(Exception $e){
    header("Location: ../index.php?err=db");
}
?>

```

Logout.php

```

<?php
    session_start();

```

```

unset($_SESSION['usr']);
unset($_SESSION['idGuest']);

header("Location: ../index.php");
?>

```

newReferral.php

```

<?php
try{
    session_start();
    include "config.php";

    $conn = new mysqli($host, $user, $password, $dbname);
    if ($conn->connect_errno)
        throw new Exception('DB connection failed');
    mysqli_set_charset($conn, "utf8");

    $sql = "SELECT Referral FROM account WHERE
Username='".$_$_SESSION['usr']."'";
    $result = $conn->query($sql);
    $row = $result->fetch_assoc();
    $ref = $row['referral'];

    $newRef = base64_encode($_SESSION['usr'].rand(-9999, 9999));
    while($newRef == $ref)
        $newRef = base64_encode($_SESSION['usr'].rand(-9999, 9999));

    $sql = "UPDATE account SET Referral='$newRef' WHERE
Username='".$_$_SESSION['usr']."'";
    $conn->query($sql);

    header("Location: ../userSettings.php");
}catch(Exception $e){
    header("Location: ../index.php?err=db");
}
?>

```

newUsername.php

```

<?php
try{
    include "config.php";
    session_start();

    //sql injections handling
    $characters = ['"', '\\', chr(0)];

```

```

$specialCharacters = false;
$_POST['username'] = str_replace("'", "", $_POST['username']);
foreach($characters as $char)
    $specialCharacters |= is_numeric(strpos($_POST['username'],
$char));

if($specialCharacters)
    header("Location: ../userSettings.php?&err=0");
else{
    $conn = new mysqli($host, $user, $password, $dbname);
    if ($conn->connect_errno)
        throw new Exception('DB connection failed');
    mysqli_set_charset($conn, "utf8");

    //prendo i dati del guest
    $usr = $_SESSION['usr'];
    $id = $_SESSION['idGuest'];

    //controllo di sicurezza
    $sql = "SELECT Type FROM account WHERE Guest_ID='$id' AND
Username='$usr'";
    $result=$conn->query($sql);
    $row=$result->fetch_assoc();
    if($row['Type'] == 1){

        $sql = "UPDATE account SET Type='1' WHERE
Username='".$_POST['username']."'";
        $conn->query($sql);
    }

    header("Location: ../userSettings.php");
}
}catch(Exception $e){
    header("Location: ../index.php?err=db");
}
?>

```

personalInfoValidation.php

```

<?php
try{
    include "config.php";
    //accesso alla sessione
    session_start();
    unset($_SESSION['idGuestTest']); //se c'erano stati altri guest
temporanei, li elimino per evitare collisioni
    unset($_SESSION['name']); //se è settato dopo questa pagina, allora è
stato creato un nuovo guest

```



```

unset($_SESSION['test']); //se è settato dopo questa pagina, allora è
stato usato un referral

if(isset($_POST["ref"]))
    $ref = "&ref=".$_POST["ref"];
else
    $ref = "";

if(isset($_GET["test"]))
    $type = "&test=".$_GET["test"];
else
    $type = "";

//sql injections handling
$elements = ['name', 'surname', 'notes', 'ref'];
$characters = ['"', '\\', chr(0)];
$specialCharacters = false;
foreach($elements as $elem){
    $_POST[$elem] = str_replace('"', "'", $_POST[$elem]);
    foreach($characters as $char)
        $specialCharacters |= is_numeric(strpos($_POST[$elem],
$char));
}
$specialCharacters |= (!is_numeric($_POST["age"]) &&
$_POST["age"]!="");

if($specialCharacters){
    header("Location: ../demographicData.php?".$type.$ref."&err=0");
}else{
    //connessione al db
    $conn = new mysqli($host, $user, $password, $dbname);

    if ($conn->connect_errno)
        throw new Exception('DB connection failed');

    mysqli_set_charset($conn, "utf8");

    // vedo se andranno salvati i dati del test
    $checkSave = 0;
    if(isset($_POST["checkSave"]))
        $checkSave = 1;
    $_SESSION["checkSave"] = $checkSave;

    if($checkSave==0){
        header("Location: ../soundSettings.php?".$type);
    }else{
        //scrivo la query di creazione del guest

```



```

    }else{ //c'è il referral
        $_SESSION["ref"] = $_POST["ref"];

        $refSQL = "SELECT Username, fk_GuestTest, fk_TestCount
FROM account WHERE Referral='{$_SESSION["ref"]}';";
        $result = $conn->query($refSQL);
        $row = $result->fetch_assoc();

        $_SESSION['test'] = array(
            "guest" => $row['fk_GuestTest'],
            "count" => $row['fk_TestCount']
        );

        $sql .= "".$row['Username'].");SELECT
LAST_INSERT_ID() as id;";
    }

    $conn->multi_query($sql);
    $conn->next_result();
    $result = $conn->store_result();
    $row = $result->fetch_assoc();

    $id = $row['id'];
    $_SESSION['idGuestTest']=$id;
    if(isset($_SESSION['test']))
        header("Location: ../info.php");
    else
        header("Location: ../soundSettings.php?".$type);
}
else{ //è stato fatto il log in
    if($_POST["name"]=="" && $_POST['ref']=="){//log in ma
niente nome e niente referral, il test va collegato all'account che ha fatto
il log in

        $_SESSION['idGuestTest'] = $_SESSION['idGuest'];
        header("Location: ../soundSettings.php?".$type);
    }else if($_POST["name"]!="" && $_POST['ref']=="){//log in
e nome ma niente referral, va creato un nuovo guest e va collegato all'account
che ha fatto il log in

        $_SESSION["name"] = $_POST["name"];

        $sql .= "".$_SESSION['usr'].");SELECT
LAST_INSERT_ID() as id;";

        $conn->multi_query($sql);
        $conn->next_result();
        $result = $conn->store_result();
        $row = $result->fetch_assoc();

```

```

$id = $row['id'];
$_SESSION['idGuestTest']=$id;

$refSQL = "SELECT fk_GuestTest, fk_TestCount FROM
account WHERE Username='{$_SESSION['usr']}'";
$result = $conn->query($refSQL);
$row = $result->fetch_assoc();

$_SESSION['test'] = array(
    "guest" => $row['fk_GuestTest'],
    "count" => $row['fk_TestCount']
);

if(isset($_SESSION['test']))
    header("Location: ../info.php");
else
    header("Location: ../soundSettings.php?".$type);
}else if($_POST["name"]=="" && $_POST['ref']!=""){//log in
e referral ma niente nome, va lanciato un errore (nome obbligatorio col
referral)

    header("Location:
../demographicData.php?".$type.$ref."&err=2");
}else if($_POST["name"]!="" && $_POST['ref']!=""){//log
in, referral e nome, va creato un nuovo guest e va collegato all'account del
referral

    $_SESSION["name"] = $_POST["name"];

    $_SESSION["ref"] = $_POST["ref"];

    $refSQL = "SELECT Username FROM account WHERE
Referral='{$_SESSION["ref"]}'";
    $result = $conn->query($refSQL);
    $row = $result->fetch_assoc(); // dopo aver fatto la
query controllo se il risultato é nullo, se lo é, il referral non é valido
    if (!isset($row['Username'])) {

        header("Location:
../demographicData.php?".$type.$ref."&err=3");
    }
    else{
        $sql .= "".$row['Username'].'';SELECT
LAST_INSERT_ID() as id";

        $conn->multi_query($sql);
        $conn->next_result();
        $result = $conn->store_result();
        $row = $result->fetch_assoc();

```



```

if($specialCharacters)
    header("Location: ../register.php?&err=0");
else{
    //apro la connessione con il db
    $conn = new mysqli($host, $user, $password, $dbname);

    //controllo se è andata a buon fine
    if ($conn->connect_errno)
        throw new Exception('DB connection failed');

    //uso codifica utf8 per comunicare col db
    mysqli_set_charset($conn, "utf8");

    //recupero username e password dal form di registrazione
    $usr = $_POST['usr'];

    //controllo se esiste già
    $sql = "SELECT * FROM account WHERE Username='$usr'";
    $result=$conn->query($sql);
    if($result->num_rows>0)
        header('Location: ../register.php?err=1'); //errore 1: lo
        definisco come errore di username già esistente

    //se non esiste eseguo la registrazione
    else{
        //prendo tutti i dati
        $psw = $_POST['psw'];
        $name = $_POST['name'];
        $surname = $_POST['surname'];
        $date = $_POST['date'];
        $gender = strtoupper($_POST['gender']);
        $notes = $_POST['notes'];
        $email = $_POST['email']; // permetto la creazione di più
        utenti username diversi ma con la stessa email

        //inizio a creare la query inserendo i valori non NULL
        $sql = "INSERT INTO guest (Name";
        $sqlVal = " VALUES ('$name'";

        if($surname != ""){
            $sql .= ",Surname";
            $sqlVal .= ",'$surname'";
        }

        if($gender != "NULL"){
            $sql .= ",Gender";
            $sqlVal .= ",'$gender'";
        }
    }
}

```

```

        if($notes != ""){
            $sql .= ",Notes";
            $sqlVal .= ",'$notes'";
        }

        $sql .= ")";
        $sqlVal .= ");SELECT LAST_INSERT_ID() as id;";

        //creo il guest
        $sql .= $sqlVal;

        $conn->multi_query($sql);
        $conn->next_result();
        $result = $conn->store_result();
        $row = $result->fetch_assoc();
        $id = $row['id'];

        //creo e collego l'account, salvo l'hash della password con
sha2-256, tipo di account 0 (base)
        $sql = "INSERT INTO account VALUES ('$usr', SHA2('$psw', 256)
";

        if($date != "")
            $sql .= ",'$date' ";
        else
            $sql .= ",NULL ";

        $sql .= ",'$id', '0', '".base64_encode($usr)."', NULL, NULL,
'$email'");";
        $conn->query($sql);

        //faccio sapere alle altre pagine quale utente è loggato
        $_SESSION['usr'] = $usr;
        $_SESSION['idGuest'] = $id;

        $conn->close();

        header('Location: ../index.php');
    }
}
}catch(Exception $e){
    header("Location: ../index.php?err=db");
}
?>

```

saveData.php

```
<?php
try{
    include "config.php";
    session_start();

    if(isset($_GET['result']) && isset($_GET['timestamp']) &&
isset($_GET['type'])
        && isset($_GET['amp']) && isset($_GET['freq']) &&
isset($_GET['dur']) && isset($_GET['modu']) && isset($_GET['blocks']) &&
isset($_GET['delta'])
        && isset($_GET['nAFC']) && isset($_GET['ITI']) &&
isset($_GET['ISI']) && isset($_GET['fact']) && isset($_GET['secFact']) &&
isset($_GET['rev'])
        && isset($_GET['secRev'])&& isset($_GET['threshold']) &&
isset($_GET['alg']) && isset($_GET['score'])
        && isset($_GET['saveSettings']) && isset($_GET['currentBlock']))){

    //trova il tipo
    $type = "";
    if($_GET['type'] == "freq")
        $type = "PURE_TONE_FREQUENCY";
    else if($_GET['type'] == "amp")
        $type = "PURE_TONE_INTENSITY";
    else if($_GET['type'] == "dur")
        $type = "PURE_TONE_DURATION";
    else if($_GET['type'] == "gap")
        $type = "WHITE_NOISE_GAP";
    else if($_GET['type'] == "ndur")
        $type = "WHITE_NOISE_DURATION";

    if(isset($_SESSION["score"]))
        $_SESSION["score"] .= ";" . $_GET['score'];
    else
        $_SESSION["score"] = $_GET['score'];

    if(isset($_SESSION["results"]))
        $_SESSION["results"] .= $_GET['result'];
    else
        $_SESSION["results"] = $_GET['result'];

    $_SESSION["time"] = $_GET['timestamp'];
    $_SESSION["type"] = $type;
    $_SESSION["amp"] = $_GET['amp'];
    $_SESSION["freq"] = $_GET['freq'];
    $_SESSION["dur"] = $_GET['dur'];
    $_SESSION["modu"] = $_GET['modu'];
```



```

$_SESSION["blocks"] = $_GET['blocks'];
//$_SESSION["delta"] = $_GET['delta'];
$_SESSION["nAFC"] = $_GET['nAFC'];
$_SESSION["ITI"] = $_GET['ITI'];
$_SESSION["ISI"] = $_GET['ISI'];
$_SESSION["fact"] = $_GET['fact'];
$_SESSION["secFact"] = $_GET['secFact'];
$_SESSION["rev"] = $_GET['rev'];
$_SESSION["secRev"] = $_GET['secRev'];
$_SESSION["thr"] = $_GET['threshold'];
$_SESSION["alg"] = $_GET['alg'];
$_SESSION["currentBlock"] = $_GET['currentBlock'];
$_SESSION["sampleRate"] = $_GET['sampleRate'];

if($_GET['currentBlock']<$_GET['blocks']){
    header("Location: ../results.php?continue=1");
}else{
    //apro la connessione con il db
    $conn = new mysqli($host, $user, $password, $dbname);

    //controllo se è andata a buon fine
    if ($conn->connect_errno)
        throw new Exception('DB connection failed');

    //uso codifica utf8 per comunicare col db
    mysqli_set_charset($conn, "utf8");

    //save the test, if it must be saved
    if($_SESSION["checkSave"]){
        if(!isset($_SESSION['idGuestTest'])){
            header("Location: ../index.php?err=2");
        }else{
            //trovo l'id a cui associare il test
            $id = $_SESSION['idGuestTest'];

            //trova il numero di test effettuati fin'ora
            $sql = "SELECT Max(Test_count) as count FROM test
WHERE Guest_ID='$id'";
            $result = $conn->query($sql);
            $row = $result->fetch_assoc();

            /*il test corrente è il numero di test già effettuati
+ 1
            $count = $row['count']+1;*/

            //inserisci i dati del nuovo test
            $sql = "UPDATE test SET Result = '{$_GET['result']}',
`Timestamp`='{$_GET['timestamp']}', `Sample Rate`='{$_GET['sampleRate']}'";

```

```

        /* ('$id', '$count', '{$_GET['timestamp']}', '$type',
";
        $sql .= "'{$_GET['amp']}', '{$_GET['freq']}',
'{$_GET['dur']}', '{$_GET['modu']}', '{$_GET['blocks']}', '{$_GET['delta']}',
";
        $sql .= "'{$_GET['nAFC']}', '{$_GET['ITI']}',
'{$_GET['ISI']}', '{$_GET['fact']}', '{$_GET['rev']}', ";
        $sql .= "'{$_GET['secFact']}', '{$_GET['secRev']}',
'{$_GET['threshold']}', '{$_GET['alg']}', '{$_GET['result']}',
'{$_GET['sampleRate']}'");
        */
        echo $sql;
        $conn->query($sql);

        // if($_GET['saveSettings']){
        //     $sql = "UPDATE account SET fk_guestTest = '$id',
fk_testCount = '$count' WHERE username = '{$_SESSION['usr']}' ";
        //     $conn->query($sql);
        // }
    }
}

if(!$SESSION["checkSave"] && $_GET['saveSettings']){
    header("Location: ../results.php?continue=0&err=1");
}else{
    header("Location: ../results.php?continue=0");
}
}
}else
    header("Location: ../index.php?err=2");
}catch(Exception $e){
    header("Location: ../index.php?err=db");
}
?>

```

saveSettings.php

```

<?php
try{
    session_start();
    include "config.php";

    //sql injections handling
    $elements = ['usr', 'email', "name", "surname", "notes"];
    $characters = ['"', '\\', chr(0)];
    $specialCharacters = false;
    foreach($elements as $elem){
        $_POST[$elem] = str_replace("'", "''", $_POST[$elem]);
    }
}

```

```

        foreach($characters as $char)
            $specialCharacters |= is_numeric(strpos($_POST[$elem],
$char));
    }

    if($specialCharacters)
        header("Location: ../userSettings.php?&err=0");
    else{
        $conn = new mysqli($host, $user, $password, $dbname);
        if ($conn->connect_errno)
            throw new Exception('DB connection failed');
        mysqli_set_charset($conn, "utf8");

        $sql = "SELECT username, date, email, name, surname, notes, gender
            FROM account INNER JOIN guest ON account.Guest_ID = guest.ID
            WHERE username='{$_SESSION['usr']}'";
        $result = $conn->query($sql);
        $row = $result->fetch_assoc();

        $somethingChanged = false;
        $oldUsr = $_SESSION['usr'];
        $sql = "UPDATE account SET ";

        if($_POST['usr']!=$row['username']){
            $userControl = "SELECT username FROM account WHERE
username='{$_POST['usr']}'";
            $result = $conn->query($sql);
            if($result->num_rows!=0){
                header("Location: ../userSettings.php?&err=1");
            }else{
                $sql.="username = '{$_POST['usr']}', ";
                $_SESSION['usr'] = $_POST['usr'];
                $somethingChanged = true;
            }
        }

        if($_POST['email']!=$row['email']){
            $sql.="email = '{$_POST['email']}', ";
            $somethingChanged = true;
        }

        if($_POST['date']!=$row['date']){
            $sql.="date = '{$_POST['date']}', ";
            $somethingChanged = true;
        }

        $sql = substr($sql, 0, -2);
        $sql .= "WHERE username='$oldUsr'";
    }

```

```

        if($somethingChanged)
            $conn->query($sql);

        $somethingChanged = false;
        $sql = "UPDATE guest SET ";

        if($_POST['name']!=$row['name']){
            $sql.="name = '{$_POST['name']}' , ";
            $somethingChanged = true;
        }

        if($_POST['surname']!=$row['surname']){
            $sql.="surname = '{$_POST['surname']}' , ";
            $somethingChanged = true;
        }

        if($_POST['notes']!=$row['notes']){
            $sql.="notes = '{$_POST['notes']}' , ";
            $somethingChanged = true;
        }

        if($_POST['gender']!=$row['gender']){
            $sql.="gender = '{$_POST['gender']}' , ";
            $somethingChanged = true;
        }

        $sql = substr($sql, 0, -2);
        $sql .= "WHERE ID='{$_SESSION['idGuest']}'";

        if($somethingChanged)
            $conn->query($sql);

        header("Location: ../userSettings.php");
    }
}catch(Exception $e){
    header("Location: ../index.php?err=db");
}
?>

```

soundSettingsValidation.php

```

<?php

session_start();
include "config.php";
if(isset($_SESSION['test'])){

```

```

try{
    $conn = new mysqli($host, $user, $password, $dbname);
    if ($conn->connect_errno)
        throw new Exception('DB connection failed');
    mysqli_set_charset($conn, "utf8");

    $sql = "SELECT Type, Amplitude as amp, Frequency as freq, Duration
as dur, Modulation as modu, blocks, Delta, nAFC,
        ISI, ITI, Factor as fact, Reversal as rev, SecFactor as
secfact, SecReversal as secrev,
        Threshold as thr, Algorithm as alg

        FROM test

        WHERE Guest_ID='{$_SESSION['test']['guest']}' AND
Test_count='{$_SESSION['test']['count']}'";
    $result = $conn->query($sql);
    $row=$result->fetch_assoc();

    if($row['Type']== 'PURE_TONE_INTENSITY')
        $type="amp";
    else if($row['Type']== 'PURE_TONE_FREQUENCY')
        $type="freq";
    else if($row['Type']== 'PURE_TONE_DURATION')
        $type="dur";
    else if($row['Type']== 'WHITE_NOISE_GAP')
        $type="gap";
    else if($row['Type']== 'WHITE_NOISE_DURATION')
        $type="ndur";

    $checkFb = 0;
    if(isset($_POST["checkFb"]))
        $checkFb = 1;

    $_SESSION["amplitude"] = $row["amp"];
    $_SESSION["frequency"] = $row["freq"];
    $_SESSION["duration"] = $row["dur"];
    $_SESSION["modulation"] = $row["modu"];
    $_SESSION["blocks"] = $row["blocks"];
    $_SESSION["nAFC"] = $row["nAFC"];
    $_SESSION["ITI"] = $row["ITI"];
    $_SESSION["ISI"] = $row["ISI"];
    $_SESSION["delta"] = $row["Delta"];
    $_SESSION["checkFb"] = $checkFb;
    $_SESSION["saveSettings"] = 0;
    $_SESSION["factor"] = $row["fact"];
    $_SESSION["secFactor"] = $row["secfact"];
    $_SESSION["reversals"] = $row["rev"];

```

```

        $_SESSION["secReversals"] = $row["secrev"];
        $_SESSION["threshold"] = $row["thr"];
        $_SESSION["algorithm"] = $row["alg"];
        unset($_SESSION['score']);
        unset($_SESSION['currentBlock']);
        unset($_SESSION['results']);

        header("Location: ../{$type}test.php");
    }catch(Exception $e){
        header("Location: ../index.php?err=db");
    }
}
}else{

    $sound_irreg_exp = "/^([a-zA-Z])+$/";
    //controlli su amplitudine
    if (($_POST["amplitude"]== "") || ($_POST["amplitude"]== "undefined"))
        header("Location:
../soundSettings.php?test={$_GET['test']}&err=amp1");

        else if(!is_numeric($_POST["amplitude"]))
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=amp2");

            else if(intval($_POST["amplitude"])>0)
                header("Location:
../soundSettings.php?test={$_GET['test']}&err=amp3");

                //controlli su frequency
                else if ((($_POST["frequency"]== "") || ($_POST["frequency"]==
"undefined")))
                    header("Location:
../soundSettings.php?test={$_GET['test']}&err=freq1");

                    else if (!is_numeric($_POST["frequency"]) ||
intval($_POST["frequency"])<0)
                        header("Location:
../soundSettings.php?test={$_GET['test']}&err=freq2");

                        //Controlli su duration
                        else if ((($_POST["duration"]== "") || ($_POST["duration"]==
"undefined")))
                            header("Location:
../soundSettings.php?test={$_GET['test']}&err=dur1");

                            else if (!is_numeric($_POST["duration"]) || $_POST["duration"]<0)
                                header("Location:
../soundSettings.php?test={$_GET['test']}&err=dur2");

```

```

        else if (($_POST["modulation"]== "") || ($_POST["modulation"]==
"undefined"))
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=mod1");

        else if (!is_numeric($_POST["modulation"]) || $_POST["modulation"]<0)
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=mod2");

        /*controlli su phase
        else if (($_POST["phase"]== "") || ($_POST["phase"]== "undefined"))
            header("Location:
soundSettings.php?test={$_GET['test']}&err=phase1");

        else if ($_POST["phase"]<0)
            header("Location:
soundSettings.php?test={$_GET['test']}&err=phase2");
        */
        //controlli su number of blocks
        else if (($_POST["blocks"]== "") || ($_POST["blocks"]== "undefined"))
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=numblock1");

        else if (!is_numeric($_POST["blocks"]) || $_POST["blocks"]<0)
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=numblock2");

        //controlli su delta
        else if (($_POST["delta"]== "") || ($_POST["delta"]== "undefined"))
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=delta1");

        else if (!is_numeric($_POST["delta"]) || $_POST["delta"]<0)
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=delta2");

        else if ($_GET['test']=="amp" &&
$_POST["amplitude"]+$_POST["delta"]>0)
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=delta3");

        //controlli su ITI
        else if (($_POST["ITI"]== "") || ($_POST["ITI"]== "undefined"))
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=ITI1");

        else if (!is_numeric($_POST["ITI"]) || $_POST["ITI"]<1000)

```

```

        header("Location:
../soundSettings.php?test={$_GET['test']}&err=ITI2");

        //controlli su ISI
        else if (($_POST["ISI"]== "") || ($_POST["ISI"]== "undefined"))
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=ISI1");

        else if (!is_numeric($_POST["ISI"]) || $_POST["ISI"]<0)
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=ISI2");

        //controlli su nAFC
        else if (($_POST["nAFC"]== "") || ($_POST["nAFC"]== "undefined"))
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=nAFC1");

        else if (!is_numeric($_POST["nAFC"]) || $_POST["nAFC"]<2)
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=nAFC2");

        else if ($_POST["nAFC"]>9)
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=nAFC3");

        //controlli sul factor
        else if (($_POST["factor"]== "") || ($_POST["factor"]== "undefined"))
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=factor1");

        else if (!is_numeric($_POST["factor"]) || $_POST["factor"]<
$_POST["secFactor"])
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=factor2");

        //controlli sul factor
        else if (($_POST["secFactor"]== "") || ($_POST["secFactor"]==
"undefined"))
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=secFactor1");

        else if (!is_numeric($_POST["secFactor"]) || $_POST["secFactor"]< 1)
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=secFactor2");

        //controlli su starting rev

```



```

        else if (($_POST["reversals"]== "") || ($_POST["reversals"]==
"undefined"))
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=rev1");

        else if (!is_numeric($_POST["reversals"]) || $_POST["reversals"]<0)
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=rev2");

        // controllli su secreversal
        else if (($_POST["secReversals"]== "") || ($_POST["secReversals"]==
"undefined"))
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=secRev1");

        else if (!is_numeric($_POST["secReversals"]) ||
$_POST["secReversals"]<0)
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=secRev2");

        //controllli su revTh
        else if (($_POST["threshold"]== "") || ($_POST["threshold"]==
"undefined"))
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=threshold1");

        else if (!is_numeric($_POST["threshold"]) || $_POST["threshold"]<0)
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=threshold2");

        else if
($_POST["threshold"]>$_POST["reversals"]+$_POST["secReversals"])
            header("Location:
../soundSettings.php?test={$_GET['test']}&err=threshold3");

    else{
        $checkFb = 0;
        if(isset($_POST["checkFb"]))
            $checkFb = 1;

        $checkSave = 0;
        if(isset($_POST["saveSettings"]))
            $checkSave = 1;

        $_SESSION["amplitude"] = $_POST["amplitude"];
        $_SESSION["frequency"] = $_POST["frequency"];
        $_SESSION["duration"] = $_POST["duration"];
        $_SESSION["modulation"] = $_POST["modulation"];
    }

```

```

//$_SESSION["phase"] = $_POST["phase"];
$_SESSION["blocks"] = $_POST["blocks"];
$_SESSION["nAFC"] = $_POST["nAFC"];
$_SESSION["ITI"] = $_POST["ITI"];
$_SESSION["ISI"] = $_POST["ISI"];
$_SESSION["delta"] = $_POST["delta"];
$_SESSION["checkFb"] = $checkFb;
$_SESSION["saveSettings"] = $checkSave;
$_SESSION["factor"] = $_POST["factor"];
$_SESSION["secFactor"] = $_POST["secFactor"];
$_SESSION["reversals"] = $_POST["reversals"];
$_SESSION["secReversals"] = $_POST["secReversals"];
$_SESSION["threshold"] = $_POST["threshold"];
$_SESSION["algorithm"] = $_POST["algorithm"];
unset($_SESSION['score']);
unset($_SESSION['currentBlock']);
unset($_SESSION['results']);

try{
$conn = new mysqli($host, $user, $password, $dbname);
if ($conn->connect_errno)
    throw new Exception('DB connection failed');
mysqli_set_charset($conn, "utf8");

$id = $_SESSION['idGuestTest'];
$type = "";
if($_GET['test'] == "freq")
    $type = "PURE_TONE_FREQUENCY";
else if($_GET['test'] == "amp")
    $type = "PURE_TONE_INTENSITY";
else if($_GET['test'] == "dur")
    $type = "PURE_TONE_DURATION";
else if($_GET['test'] == "gap")
    $type = "WHITE_NOISE_GAP";
else if($_GET['test'] == "ndur")
    $type = "WHITE_NOISE_DURATION";

$sql = "SELECT Max(Test_count) as count FROM test WHERE
Guest_ID='$id'";

$result = $conn->query($sql);
$row = $result->fetch_assoc();

//il test corrente è il numero di test già effettuati
+ 1
$count = $row['count']+1;

```

```

        $sql = "INSERT INTO test VALUES ('$id', '$count',
current_timestamp(), '$type', ";
        $sql .= "'{'$_POST['amplitude']}'",
'{'$_POST['frequency']}'', '{$_POST['duration']}'', '{$_POST['modulation']}'',
'{'$_POST['blocks']}'', '{$_POST['delta']}'', ";
        $sql .= "'{'$_POST['nAFC']}'", '{$_POST['ITI']}'',
'{'$_POST['ISI']}'', '{$_POST['factor']}'', '{$_POST['reversals']}'', ";
        $sql .= "'{'$_POST['secFactor']}'",
'{'$_POST['secReversals']}'', '{$_POST['threshold']}'', '{$_POST['algorithm']}'',
'', '0', '$checkFb)";

        $conn->query($sql);
        if($checkSave){

            $sql = "UPDATE account SET fk_guestTest = '$id', fk_testCount
= '$count' WHERE username = '{$_SESSION['usr']}' ";
            $conn->query($sql);
        }
        header("Location: ../{$_GET['test']}test.php");

        }catch(Exception $e){
            echo'sono nel catch';
            header("Location: ../index.php?err=db");
        }

    }
}
?>

```

updateSettings.php

```

<?php
    session_start();
    $_SESSION['updatingSavedSettings']=true;
    header("Location: ../soundSettings.php?test=".$_GET['test']);
?>

```

updateSavedSettings.php

```

<?php
    try{
        include "config.php";
        session_start();

        //apro la connessione con il db

```

```

$conn = new mysqli($host, $user, $password, $dbname);

//controllo se è andata a buon fine
if ($conn->connect_errno)
    throw new Exception('DB connection failed');

//uso codifica utf8 per comunicare col db
mysqli_set_charset($conn, "utf8");

$id = $_SESSION['idGuest'];

//trova il numero di test effettuati fin'ora
$sql = "SELECT Max(Test_count) as count FROM test WHERE
Guest_ID='$id'";
$result = $conn->query($sql);
$row = $result->fetch_assoc();

//il test corrente è il numero di test già effettuati + 1
$count = $row['count']+1;

$checkFb = 0;
if(isset($_POST["checkFb"]))
    $checkFb = 1;

//inserisci i dati del nuovo test
$sql = "INSERT INTO test VALUES ('$id', '$count', current_timestamp(),
'".$_GET['test'].', ";
    $sql .= "'".$_POST["amplitude"].', "'.$_POST["frequency"].',
'".$_POST["duration"].', ";
    $sql .= "'".$_POST["modulation"].', "'.$_POST['blocks'].',
'".$_POST['delta'].', ";
    $sql .= "'".$_POST['nAFC'].', "'.$_POST['ITI'].', "'.$_POST['ISI'].',
'".$_POST['factor'].', "'.$_POST['reversals'].', ";
    $sql .= "'".$_POST['secFactor'].', "'.$_POST['secReversals'].',
'".$_POST['threshold'].', "'.$_POST['algorithm'].', ', '0', '$checkFb)";
    $conn->query($sql);

    $sql = "UPDATE account SET fk_guestTest = '$id', fk_testCount =
'$count' WHERE username = '".$_SESSION['usr']."' ";
    $conn->query($sql);

    header("Location: ../userSettings.php?err=4");
}catch(Exception $e){
    header("Location: ../index.php?err=db");
}
?>

```