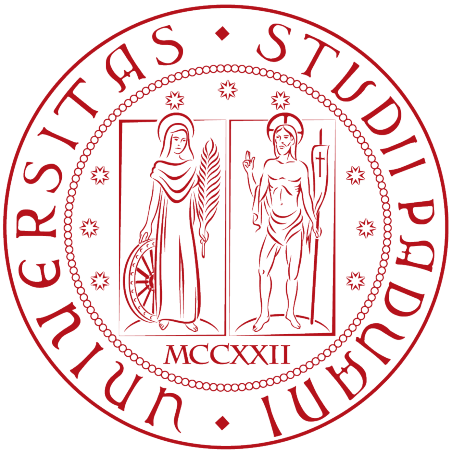
Università degli Studi di Padova

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea in Informatica



Realizzazione di backend e frontend di una webpage adibita al caricamento su cloud di immagini e video per concorso fotografico

Tesi di laurea

Relatore
Prof. Palazzi Claudio Enrico

Laureando
Nicola Cecchetto

Anno Accademico 2022-2023

Sommario

Questo documento è un resoconto del lavoro svolto durante l'esperienza di stage del laureando Nicola Cecchetto presso l'azienda Girolibero srl nel periodo che va dal 03/07/2023 al 01/09/2023 per una durata complessiva di 320 ore.

L'obiettivo era la realizzazione di una pagina web che permettesse a clienti Girolibero e non l'upload di foto e video delle proprie vacanze su cloud per un concorso fotografico, organizzato e gestito da Girolibero.

Il documento descrive in dettaglio organizzazione dello stage, fasi di analisi, progettazione, implementazione, verifica e validazione e tecnologie usate.

"An idiot admires complexity, a genius admires simplicity"

— T.A.D.

Ringraziamenti

Innanzitutto, vorrei esprimere la mia gratitudine al Prof. Palazzi Claudio Enrico, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura del lavoro.

Desidero ringraziare con affetto i miei genitori, i miei nonni e tutti gli altri parenti per avermi sostenuto e incoraggiato durante la mia carriera universitaria.

Voglio ringraziare i miei amici, vecchi e nuovi, per i momenti passati insieme e per quelli che ancora aspettano di essere vissuti.

Infine un ringraziamento lo voglio fare anche a Giulia. Possa questo essere d'esempio che con la calma si fa tutto.

Padova, Dicembre 2023 Nicola Cecchetto

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

Questo capitolo introduttivo raccoglie una descrizione dell'azienda ospitante e del progetto realizzato.

1.1 Descrizione dell'azienda

Girolibero è un'azienda che opera nel settore turistico dal 1998. Crescendo negli anni è diventata punto di riferimento internazionale nel settore delle vacanze attive e sostenibili. L'azienda offre itinerari organizzati con alloggi prenotati, trasporto bagagli a ogni tappa, mappe per seguire il percorso o un affidabile accompagnatore a fare strada al gruppo, assistenza 7 giorni su 7. Dagli uffici viene organizzata ogni partenza, con i servizi e la logistica, la selezione dei partner e dei fornitori internazionali, vengono dirette le attività di marketing e la gestione del sito web e-commerce, che si rivolge al mercato italiano e internazionale.

Il sito web ufficiale[1] è il portale principale dal quale è possibile sfogliare il catalogo dei viaggi offerti e effettuare dei preventivi. Questo sito è attualmente gestito da un team composto da personale marketing e figure *Information Technology (IT)*[g].

La sede principale (dove è stata effettuata l'esperienza di stage) è situata a Vicenza e ospita gli uffici manageriali, booking, marketing (che comprende anche IT) e un officina per le biciclette.

1.2 Descrizione del progetto

Lo scopo del progetto è la realizzazione di una pagina web che permetta a clienti Girolibero e non l'upload di foto e video per un contest fotografico. Queste foto e video devono essere caricati su un *Digital Asset Management (DAM)*[g](Cloudinary per la precisione, che verrà descritto più avanti nel documento), già usato dall'azienda per la gestione di tutti i file multimediali. Le foto e i video dovranno essere rinominati con una precisa nomenclatura in base al tipo di viaggio effettuato e verranno esaminati da una commissione interna dell'azienda che determinerà i vincitori e assegnerà loro dei buoni da utilizzare in uno dei viaggi Girolibero.

[1] *Sito web ufficiale di Girolibero.* URL: https://www.girolibero.it/.

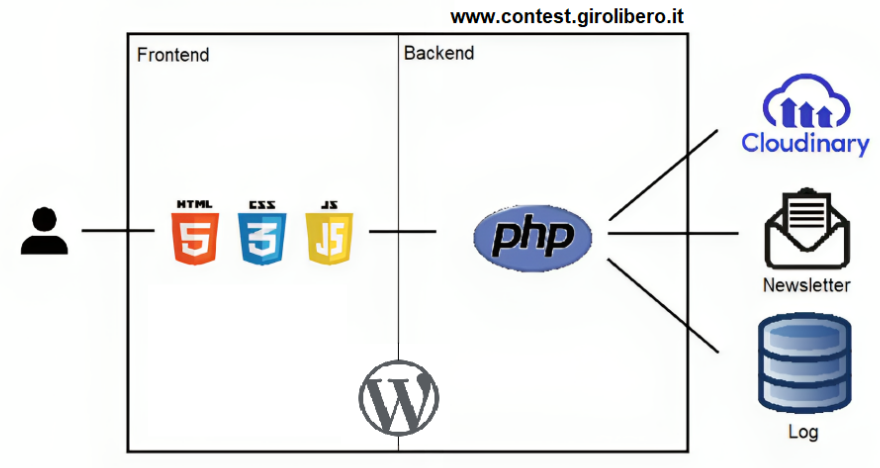


Figura 1.1: Struttura del progetto

Come accennato in precedenza ci sono due tipologie di utenti che possono utilizzare questa pagina:

- clienti Girolibero che dopo essere tornati da un viaggio ricevono un'email di bentornato con un invito a partecipare al concorso (tramite link creato appositamente per il cliente)
- non clienti (viaggiatori *Per Conto Mio (PCM)*[g]) che visitano la pagina del concorso[2]

In entrambi i casi la pagina è la stessa, ma il *form*[g]che appare dopo aver accettato il regolamento e la privacy policy è diverso. Nel caso del cliente, il form presenta solamente il campo di input dei file, mentre nel caso PCM si sono ulteriori campi da compilare per indicare i propri dati personali e informazioni sul viaggio effettuato. Dopo aver compilato e inviato il form e aver atteso del tempo di elaborazione e invio dei file se tutto è andato a buon fine viene presentata all'utente la scelta di chiudere la pagina (rendirizzandolo al sito principale) o di caricare le foto e video di un'altra vacanza. Se l'utente (cliente o meno) opta per questa seconda opzione viene presentato un nuovo form di compilazione di tipo PCM per l'upload di nuovi file per una nuova vacanza o viaggio. Questa operazione di caricamento secondario può essere ripetuta dall'utente un numero indefinito di volte.

L'invio del form non deve cambiare o ricaricare la pagina, quindi deve essere effettuato in maniera asincrona. L'operazione di invio deve includere, oltre all'upload su Cloudinary delle immagini e dei video rinominati correttamente, anche l'iscrizione alla *newsletter*[g]di Girolibero (se non già iscritto) e l'aggiunta delle informazioni dell'utente e di ciò che ha caricato in un *log*[g]accessibile al personale marketing dell'azienda. L'iscrizione alla newsletter prevede una richiesta effettuata verso uno script PHP già esistente che andrà ad aggiungere l'email dell'utente alla mailing list di Girolibero già

[2] *Pagina realizzata durante il progetto.* URL: https://contest.girolibero.it/.

esistente.

La pagina deve rispettare le scelte di design del sito principale, quindi struttura e stile della pagina devono essere coerenti e integrare quello che già esiste. In caso di nuove scelte di design da effettuare queste vanno decise insieme all'*User Experience (UX)*[g]designer che segue il progetto. Aspetto più importante è quello di mantenere quanto più possibile la coerenza con il menù presente nel sito principale.

Infine la pagina deve essere gestita tramite *Content Management System (CMS)*[g]Wordpress, già usato dall'azienda per facilitare la gestione dei contenuti delle pagine del sito principale da parte del personale marketing.

1.3 Organizzazione del testo

Il secondo capitolo descrive gli obiettivi fissati all'inizio dello stage, la suddivisione oraria, le varie figure interne all'azienda che hanno fornito supporto e le metodologie e tecnologie usate per poter lavorare e comunicare insieme.

Il terzo capitolo descrive l'intera fase di analisi dei requisiti, con i casi d'uso e i requisiti individuati.

Il quarto capitolo descrive le tecnologie utilizzate e approfondisce le fasi di progettazione e implementazione.

Il quinto capitolo descrive le fasi di verifica e validazione dei requisiti, con inoltre una sezione dedicata all'accessibilità.

Il sesto capitolo chiude la tesi con obiettivi raggiunti, conoscenze acquisite e considerazioni personali.

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento
- per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*[g]

Capitolo 2

Stage

Questo capitolo presenta obiettivi del progetto, suddivisione oraria, figure che hanno partecipato allo sviluppo del progetto e metodi e tecnologie per comunicare e lavorare con loro.

2.1 Obiettivi

Si farà riferimento ai requisiti secondo le seguenti notazioni:

- OB per i requisiti obbligatori, vincolanti in quanto obiettivo primario richiesto dal committente
- DE per i requisiti desiderabili, non vincolanti o strettamente necessari, ma dal riconoscibile valore aggiunto
- OP per i requisiti opzionali, rappresentanti valore aggiunto non strettamente competitivo

Le sigle precedentemente indicate saranno seguite da un numero, identificativo del requisito.

Gli obiettivi del progetto sono raccolti nella tabella 2.1.

Tabella 2.1: Tabella degli obiettivi

Codice	Descrizione
OB1	Realizzazione del *frontend*[g]
OB2	Realizzazione del *backend*[g]
OB3	Interazione con archivio *cloud*[g] Cloudinary
DE1	Predisposizione del backend per la gestione multilingua dei campi di testo
OP1	Implementazione tracciamento Google Tag Manager sulla pagina contest.girolibero.it per analisi digital marketing

Tabella 2.2: Tabella della suddivisione oraria

Fase introduttiva - 90h	
20h	Introduzione agli ambienti di sviluppo e linguaggi utilizzati
8h	Introduzione a Wordpress, alle tassonomie e ai campi personalizzati in uso
20h	Introduzione alla console Cloudinary per l'archiviazione, gestione e consegna al sito dei file multimediali
2h	Definizione dei requisiti dell'azienda per lo sviluppo del backend e del frontend di contest.girolibero.it
40h	Studio e test delle *Application Program Interface (API)*[g] di Cloudinary
Fase operativa - 150h	
50h	Realizzazione backend
50h	Realizzazione frontend
50h	Integrazione con Cloudinary
Fase di verifica, bugfix e ottimizzazione: 80h	
80h	Verifica, correzione di bug e di interfaccia, ottimizzazione del sistema di upload
Totale: 320h	

2.2 Suddivisione oraria

La tabella 2.2 riporta la suddivisione oraria del lavoro definita prima dell'inizio dello stage per il Piano di Lavoro.

2.3 Figure interne di riferimento e supporto

Di seguito vengono elencate tutte le figure che hanno partecipato allo sviluppo del progetto:

- Elena Riatti - Responsabile marketing (Tutor)
- Marco Matteazzi - Sviluppatore web (Co-tutor)
- Caterina Romio - Project manager e UX designer
- Simone Fontana - Sviluppatore web
- Antonio Giuliani - Sviluppatore web

Da notare è la presenza di due tutor aziendali. Questo perchè inizialmente era stata indicata solamente Elena Riatti come tutor aziendale, ma questo andava in conflitto con la necessità di avere come tutor una figura IT. Dopo aver chiesto il parere del responsabile stage, è stato deciso di aggiungere come co-tutor anche lo sviluppatore web Marco Matteazzi.

2.4 Metodi e tecnologie di comunicazione

Nonostante la possibilità di poter effettuare smart-working, l'intera esperienza di stage si è svolta interamente in presenza nella sede di Vicenza di Girolibero. Questo ha

permesso una comunicazione migliore con le figure presentate nella sezione precedente nelle varie fasi del progetto.

Il martedì di ogni settimana è stato il giorno dedicato alle riunioni del team, dove avevo la possibilità di descrivere l'andamento del progetto ed esprimere dubbi e incertezze, così da poter chiedere aiuto su come risolverli.

Detto questo, era comunque necessario avere delle metodologie di comunicazione remota per quei casi in cui qualcuno non fosse in azienda. Sono stati quindi utilizzate le seguenti tecnologie:

Slack

Applicazione di messaggistica istantanea pensato per la collaborazione aziendale. È stata utile per scambiare brevi messaggi e domande durante gli orari di lavoro.

Skype

Applicazione di messaggistica e chiamate. Usata appunto per chiamate in gruppo e non.

Google Meet

Servizio di chiamate e videochiamate offerto da Google. Anche questo è stato usato per chiamate di lavoro.

Google Docs

Editor di documenti online offerto da Google. Più persone possono visualizzare e modificare lo stesso documento, e questo è stato utile per fissare i punti chiave dell'intero progetto e tenere traccia di cosa è stato implementato.

Gmail

Servizio di email di Google. Utile per scambiare messaggi e domande di una certa lunghezza, anche in momenti in cui il destinatario non fosse reperibile.

2.5 Metodi e tecnologie di lavoro

Per lo sviluppo del progetto sono stati utilizzati:

Visual Studio Code

Editor di codice usato in ogni momento della progettazione e sviluppo. Grazie alle innumerevoli estensioni è in grado di supportare ogni linguaggio di programmazione e non solo.

XAMPP

Distribuzione Apache contenente MariaDB, PHP e Perl per lo sviluppo di siti web. Permette di creare un server locale sulla propria macchina per permettere allo sviluppatore di vedere in locale le modifiche alle quali sta lavorando.

Git

Sistema di controllo di versione utilizzato per coordinare il lavoro tra più sviluppatori.

Bitbucket

Servizio di web hosting per progetti che usano sistemi di controllo di versione già usato dal team di sviluppo dell'azienda.

Capitolo 3

Analisi dei requisiti

Questo capitolo tratta la fase di analisi dei requisiti del progetto, descrivendo gli attori, i casi d'uso e i requisiti individuati.

3.1 Descrizione casi d'uso

3.1.1 Attori

Gli attori individuati durante l'analisi dei requisiti sono i seguenti:

- Utente: attore che generalizza due diverse tipologie di utenti che possono visitare la pagina web:
  - Cliente (Girolibero): chi ritorna da un viaggio organizzato da Girolibero riceve un'email di bentornato. Dentro questa email andrà inserito un link alla pagina del contest con un apposito parametro utile ad identificare il cliente per poter ricavare le sue informazioni personali e quelle del viaggio appena effettuato
  - Non cliente (viaggiatore PCM): chi entra in www.contest.girolibero.it viene trattato come viaggiatore PCM e quindi dovrà inserire ulteriori informazioni su di sè e sul viaggio effettuato prima di poter caricare foto e video
- Personale marketing Girolibero: non interagisce direttamente con il sito ma deve poter modificare il regolamento ad inizio pagina e deve poter anche visualizzare una lista di persone che hanno inviato dei file

Oltre ai precedenti attori individuati, ne sono stati individuati degli altri. Questi non sono persone ma comunque entità terze rispetto alla pagina web che interagiscono con essa. Sono elencati di seguito:

- Cloudinary: riceve i file multimediali caricati dagli utenti. Invia sempre una risposta per informare dello stato dell'upload
- *Database Management System (DBMS)*[g]Wordpress: database gestito da Wordpress che conterrà una tabella di log con dentro le informazioni di tutti gli upload effettuati
- API Girolibero: script di varia utilità gestiti dal team di sviluppo Girolibero. A questa API viene effettuata la richiesta di iscrizione alla newsletter

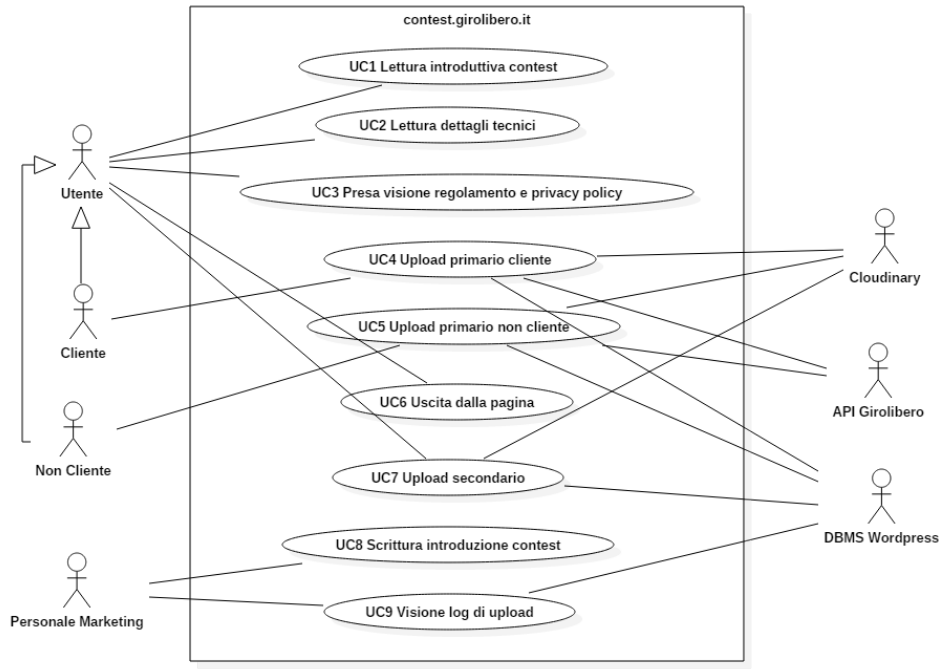


Figura 3.1: Scenario principale dei casi d'uso

3.1.2 Casi d'uso

Di seguito vengono riportati i casi d'uso individuati durante la fase di analisi aggiornato con le modifiche fatte anche durante la fase di progettazione e codifica. Ogni caso d'uso è composto da:

- un codice identificativo nella forma UCX, con X un numero progressivo che parte da 1 e può estendersi tramite sottopunti (esempio: 1.1, 1.2, 1.2.1 ...) nel caso di sottocasi correlati
- breve titolo descrittivo
- attori coinvolti nel caso d'uso
- precondizioni che devono essere soddisfatte per compiere il caso d'uso
- descrizione esaustiva del caso d'uso
- postcondizioni avvenute dopo il caso d'uso
- scenario principale che descrive in dettaglio i vari passaggi del caso d'uso
- opzionali scenari alternativi come casi di errore

UC1: Lettura introduttiva contest

Attori Principali: Utente.

Precondizioni: La pagina web www.contest.girolibero.it è stata caricata sul browser

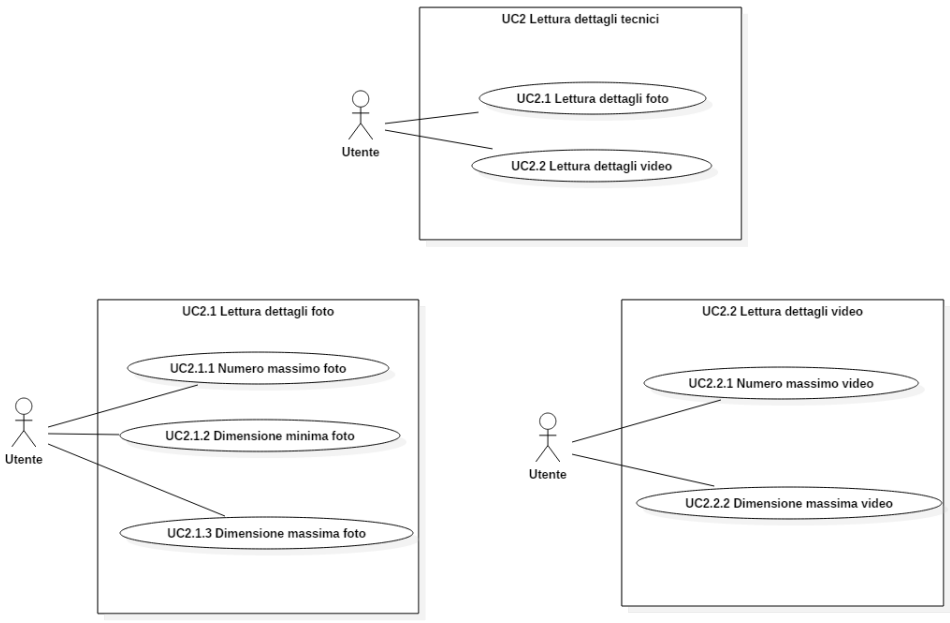


Figura 3.2: Diagrammi UC2

dell'utente.

Descrizione: L'utente vuole leggere nel *viewport*[g] della pagina una breve introduzione scritta dal personale marketing che spiega modalità, finalità e premi del concorso.

Postcondizioni: L'utente ha letto l'introduzione al concorso.

Scenario Principale:

- L'utente apre la pagina nel proprio browser e legge l'introduzione al concorso ad inizio pagina

UC2: Lettura dettagli tecnici

Attori Principali: Utente.

Precondizioni: La pagina web www.contest.girolibero.it è stata caricata sul browser dell'utente.

Descrizione: L'utente vuole leggere all'inizio della pagina una breve descrizione sui dettagli dei file che è possibile caricare.

Postcondizioni: L'utente ha letto la descrizione sui dettagli dei file che è possibile caricare.

Scenario Principale:

- L'utente apre la pagina nel proprio browser e legge la descrizione sui dettagli dei file che è possibile caricare

UC2.1: Lettura dettagli tecnici foto

Attori Principali: Utente.

Precondizioni: La pagina web www.contest.girolibero.it è stata caricata sul browser dell'utente.

Descrizione: L'utente vuole leggere nella pagina una breve descrizione sui dettagli delle foto che è possibile caricare.

Postcondizioni: L'utente ha letto la descrizione sui dettagli delle foto che è possibile caricare.

Scenario Principale:

- L'utente apre la pagina nel proprio browser e legge la descrizione sui dettagli delle foto che è possibile caricare

UC2.1.1: Numero massimo foto

Attori Principali: Utente.

Precondizioni: La pagina web www.contest.girolibero.it è stata caricata sul browser dell'utente.

Descrizione: L'utente vuole leggere il numero massimo di foto che è consentito caricare per un viaggio.

Postcondizioni: L'utente ha letto il numero massimo di foto che è consentito caricare.

Scenario Principale:

- L'utente legge il numero massimo di foto che è consentito caricare

UC2.1.2: Dimensione minima foto

Attori Principali: Utente.

Precondizioni: La pagina web www.contest.girolibero.it è stata caricata sul browser dell'utente.

Descrizione: L'utente vuole leggere la dimensione minima delle foto che è consentito caricare per un viaggio.

Postcondizioni: L'utente ha letto la dimensione minima delle foto che è consentito caricare.

Scenario Principale:

- L'utente legge la dimensione minima delle foto che è consentito caricare

UC2.1.3: Dimensione massima foto

Attori Principali: Utente.

Precondizioni: La pagina web www.contest.girolibero.it è stata caricata sul browser dell'utente.

Descrizione: L'utente vuole leggere la dimensione massima delle foto che è consentito caricare per un viaggio.

Postcondizioni: L'utente ha letto la dimensione massima delle foto che è consentito caricare.

Scenario Principale:

- L'utente legge la dimensione massima delle foto che è consentito caricare

UC2.2: Lettura dettagli tecnici video

Attori Principali: Utente.

Precondizioni: La pagina web www.contest.girolibero.it è stata caricata sul browser dell'utente.

Descrizione: L'utente vuole leggere nella pagina una breve descrizione sui dettagli dei video che è possibile caricare.

Postcondizioni: L'utente ha letto la descrizione sui dettagli dei video.

Scenario Principale:

- L'utente apre la pagina nel proprio browser e legge la descrizione sui dettagli dei video che è possibile caricare

UC2.2.1: Numero massimo video

Attori Principali: Utente.

Precondizioni: La pagina web www.contest.girolibero.it è stata caricata sul browser dell'utente.

Descrizione: L'utente vuole leggere il numero massimo di video che è consentito caricare per un viaggio.

Postcondizioni: L'utente ha letto il numero massimo di video che è consentito caricare.

Scenario Principale:

- L'utente legge il numero massimo dei video che è consentito caricare

UC2.2.2: Dimensione massima video

Attori Principali: Utente.

Precondizioni: La pagina web www.contest.girolibero.it è stata caricata sul browser dell'utente.

Descrizione: L'utente vuole leggere la dimensione massima dei video che è consentito caricare per un viaggio.

Postcondizioni: L'utente ha letto la dimensione massima dei video che è consentito caricare.

Scenario Principale:

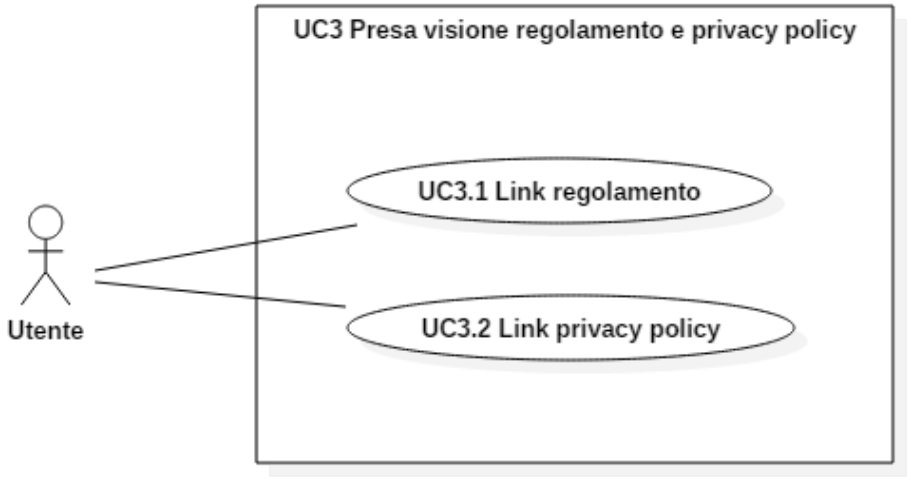


Figura 3.3: Diagramma UC3

- L'utente legge la dimensione massima dei video che è consentito caricare

UC3: Presa visione regolamento e privacy policy

Attori Principali: Utente.

Precondizioni: La pagina web www.contest.girolibero.it è stata caricata sul browser dell'utente.

Descrizione: L'utente vuole dare conferma di aver preso visione del regolamento del concorso e la privacy policy legata al trattamento dei dati inseriti nella pagina.

Postcondizioni: L'utente ha dato conferma di aver preso visione del regolamento e della privacy policy.

Scenario Principale:

- L'utente tramite una checkbox dà la propria conferma di aver preso visione del regolamento e della privacy policy
- L'utente preme un pulsante per proseguire nella pagina

Scenario Alternativo:

- L'utente senza aver dato la propria conferma di aver preso visione del regolamento e della privacy policy preme il pulsante per proseguire nella pagina
- Viene notificato all'utente che deve prendere visione del regolamento e della privacy policy prima di proseguire

UC3.1: Link regolamento

Attori Principali: Utente.

Precondizioni: La pagina web www.contest.girolibero.it è stata caricata sul browser dell'utente.

Descrizione: L'utente vuole visitare il link www.girolibero.it/contest-regolamento per la visione del regolamento del concorso.

Postcondizioni: L'utente ha visitato il link www.girolibero.it/contest-regolamento.

Scenario Principale:

- L'utente clicca il link e viene aperta nel browser una nuova pagina che contiene il regolamento completo del concorso

UC3.2: Link privacy policy

Attori Principali: Utente.

Precondizioni: La pagina web www.contest.girolibero.it è stata caricata sul browser dell'utente.

Descrizione: L'utente vuole visitare il link www.girolibero.it/privacy-policy per la visione della privacy policy.

Postcondizioni: L'utente ha visitato il link della privacy policy è presente nella pagina.

Scenario Principale:

- L'utente clicca il link e viene aperta nel browser una nuova pagina che contiene la privacy policy del concorso

UC4: Upload primario Cliente

Attori Principali: Cliente, Cloudinary, API Girolibero, DBMS Wordpress.

Precondizioni: Il cliente ha dato conferma di aver preso visione del regolamento e della privacy policy (UC3).

Descrizione: Il cliente vuole caricare i file del proprio viaggio Girolibero.

Postcondizioni: Il cliente ha caricato con successo i file del primo viaggio.

Scenario Principale:

- Una volta che il cliente conferma di aver preso visione del regolamento e della privacy policy, viene mostrato a schermo il form di inserimento dei file da caricare
- Il cliente sceglie i file e quando è pronto a inviare preme un pulsante
- I file vengono caricati su Cloudinary rinominati nel modo corretto
- Nel log di upload viene aggiunta l'informazione di upload
- Viene mandata la richiesta all'API Girolibero di iscrizione alla newsletter
- Terminate queste operazioni il form si disabilita (non può essere modificato dal cliente ma rimane visibile) e compare sotto di esso una conferma esplicita della riuscita dell'upload e due nuovi pulsanti per le operazioni disponibili (UC6 e UC7)

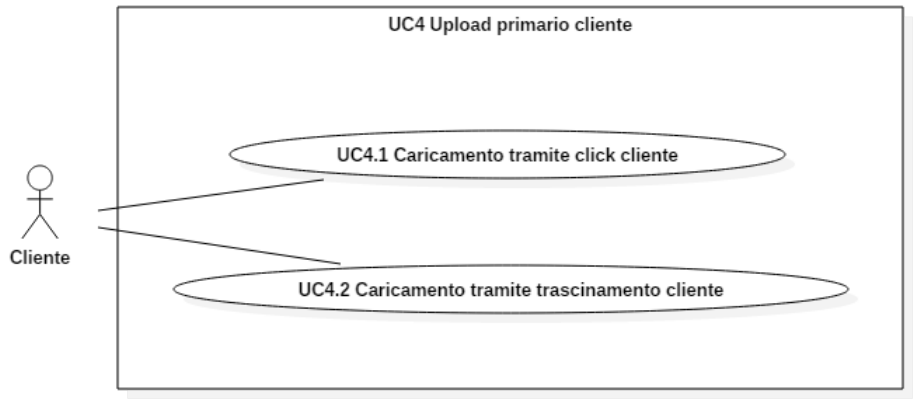


Figura 3.4: Diagramma UC4

Scenario Alternativo:

- Se l'upload dei file non riesce il server risponde con un errore e verrà visualizzata una notifica all'utente di riprovare in un secondo momento

UC4.1: Selezione file tramite click Cliente

Attori Principali: Cliente.

Precondizioni: Il cliente ha dato conferma di aver preso visione del regolamento e della privacy policy (UC3).

Descrizione: Il cliente vuole caricare i file attraverso un click che apre il gestore file del dispositivo.

Postcondizioni: Il cliente ha caricato con successo i file attraverso un click.

Scenario Principale:

- Il cliente clicca l'area di caricamento dei file e tramite il gestore file del dispositivo seleziona i file che intende caricare
- Selezionati i file, questi compariranno nell'area di caricamento e saranno pronti per essere caricati

Scenario Alternativo:

- Se il cliente seleziona dei file non validi (per dimensioni o tipo) questi non compariranno nell'area di caricamento e il cliente ne viene notificato

UC4.2: Selezione file tramite trascinamento Cliente

Attori Principali: Cliente.

Precondizioni: Il cliente ha dato conferma di aver preso visione del regolamento e della privacy policy (UC3).

Descrizione: Il cliente vuole caricare i file trascinandoli nell'apposito riquadro nella pagina.

Postcondizioni: Il cliente ha caricato con successo i file trascinandoli nell'apposito riquadro nella pagina.

Scenario Principale:

- Il cliente trascina i file che desidera caricare nell'area di caricamento (funzionalità esclusiva solo agli utenti desktop)
- Selezionati i file, questi compariranno nell'area di caricamento e saranno pronti per essere caricati

Scenario Alternativo:

- Se il cliente trascina dei file non validi (per dimensioni o tipo) questi non compariranno nell'area di caricamento e comparirà una notifica di avvertimento

UC5: Upload primario Non Cliente

Attori Principali: Non Cliente, Cloudinary, API Girolibero, DBMS Wordpress.

Precondizioni: Il non cliente ha dato conferma di aver preso visione del regolamento e della privacy policy (UC3).

Descrizione: Il non cliente vuole caricare i file del proprio viaggio indipendente.

Postcondizioni: Il non cliente ha caricato con successo i file del primo viaggio.

Scenario Principale:

- Una volta che il non cliente conferma di aver preso visione del regolamento e della privacy policy, viene mostrato a schermo il form di inserimento dei file da caricare
- Il non cliente inserisce le sue informazioni, quelle del viaggio, sceglie i file e quando è pronto a inviare preme un pulsante
- I file vengono caricati su Cloudinary rinominati nel modo corretto
- Nel log di upload viene aggiunta l'informazione di upload
- Viene mandata la richiesta all'API Girolibero di iscrizione alla newsletter
- Terminate queste operazioni il form si disabilita (non può essere modificato dal non cliente ma rimane visibile) e compare sotto di esso una conferma esplicita della riuscita dell'upload e due nuovi pulsanti per le operazioni disponibili (UC6 e UC7)

Scenario Alternativo:

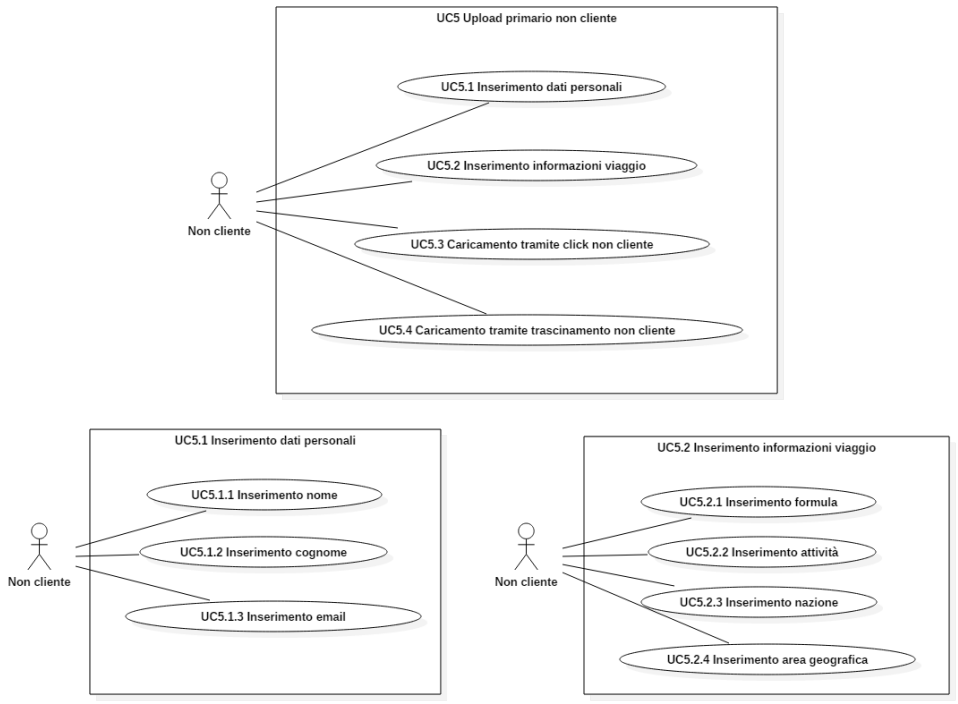


Figura 3.5: Diagrammi UC5

- Se le informazioni personali o quelle del viaggio non sono valide, verrà visualizzata una notifica
- Se l'upload dei file non riesce il server risponde con un errore e verrà visualizzata una notifica all'utente di riprovare in un secondo momento

UC5.1: Inserimento dati personali

Attori Principali: Non Cliente.

Precondizioni: Il non cliente ha dato conferma di aver preso visione del regolamento e della privacy policy (UC3).

Descrizione: Il non cliente vuole inserire i propri dati personali da associare ai file che vuole caricare.

Postcondizioni: Il non cliente ha inserito con successo i propri dati personali.

Scenario Principale:

- Nella procedura di caricamento è necessario che il non cliente inserisca i propri dati personali

UC5.1.1: Inserimento nome

Attori Principali: Non Cliente.

Precondizioni: Il non cliente ha dato conferma di aver preso visione del regolamento e della privacy policy (UC3).

Descrizione: Il non cliente vuole inserire in un apposito campo il proprio nome.

Postcondizioni: Il non cliente ha inserito con successo il proprio nome.

Scenario Principale:

- Nella procedura di caricamento è necessario che il non cliente inserisca il proprio nome in un campo di testo

UC5.1.2: Inserimento cognome

Attori Principali: Non Cliente.

Precondizioni: Il non cliente ha dato conferma di aver preso visione del regolamento e della privacy policy (UC3).

Descrizione: Il non cliente vuole inserire in un apposito campo il proprio cognome.

Postcondizioni: Il non cliente ha inserito con successo il proprio cognome.

Scenario Principale:

- Nella procedura di caricamento è necessario che il non cliente inserisca il proprio cognome in un campo di testo

UC5.1.3: Inserimento email

Attori Principali: Non Cliente.

Precondizioni: Il non cliente ha dato conferma di aver preso visione del regolamento e della privacy policy (UC3).

Descrizione: Il non cliente vuole inserire in un apposito campo la propria email.

Postcondizioni: Il non cliente ha inserito con successo la propria email.

Scenario Principale:

- Nella procedura di caricamento è necessario che il non cliente inserisca la propria email in un campo di testo

UC5.2: Inserimento informazioni viaggio

Attori Principali: Non Cliente.

Precondizioni: Il non cliente ha dato conferma di aver preso visione del regolamento e della privacy policy (UC3).

Descrizione: Il non cliente vuole inserire le informazioni del viaggio da associare ai file che vuole caricare.

Postcondizioni: Il non cliente ha inserito con successo le informazioni del viaggio.

Scenario Principale:

- Nella procedura di caricamento è necessario che il non cliente inserisca le informazioni del viaggio effettuato

UC5.2.1: Inserimento formula

Attori Principali: Non Cliente.

Precondizioni: Il non cliente ha dato conferma di aver preso visione del regolamento e della privacy policy (UC3).

Descrizione: Il non cliente vuole inserire in un apposito campo la formula del viaggio.

Postcondizioni: Il non cliente ha inserito con successo la formula del viaggio.

Scenario Principale:

- Nella procedura di caricamento è necessario che il non cliente inserisca la formula del viaggio, selezionandola tra le varie opzioni proposte in un menù a tendina

UC5.2.2: Inserimento attività

Attori Principali: Non Cliente.

Precondizioni: Il non cliente ha dato conferma di aver preso visione del regolamento e della privacy policy (UC3).

Descrizione: Il non cliente vuole inserire in un apposito campo l'attività svolta durante il viaggio.

Postcondizioni: Il non cliente ha inserito con successo l'attività svolta.

Scenario Principale:

- Nella procedura di caricamento è necessario che il non cliente inserisca l'attività del viaggio, selezionandola tra le varie opzioni proposte in un menù a tendina

UC5.2.3: Inserimento nazione

Attori Principali: Non Cliente.

Precondizioni: Il non cliente ha dato conferma di aver preso visione del regolamento e della privacy policy (UC3).

Descrizione: Il non cliente vuole inserire in un apposito campo la nazione che ha visitato durante il suo viaggio.

Postcondizioni: Il non cliente ha inserito con successo la nazione visitata.

Scenario Principale:

- Nella procedura di caricamento è necessario che il non cliente inserisca la nazione visitata, selezionandola tra le varie opzioni proposte in un menù a tendina
- Data la lunghezza della lista delle opzioni, l'utente può anche cercare la nazione tramite casella di testo

UC5.2.4: Inserimento area geografica

Attori Principali: Non Cliente.

Precondizioni: Il non cliente ha dato conferma di aver preso visione del regolamento

e della privacy policy (UC3).

Descrizione: Il non cliente vuole inserire in un apposito campo opzionale l'area geografica che ha visitato durante il suo viaggio.

Postcondizioni: Il non cliente ha inserito con successo l'area geografica visitata.

Scenario Principale:

- Nella procedura di caricamento il non cliente può opzionalmente inserire l'area geografica del viaggio in un campo di testo

UC5.3: Selezione file tramite click Non Cliente

Attori Principali: Non Cliente.

Precondizioni: Il non cliente ha dato conferma di aver preso visione del regolamento e della privacy policy (UC3).

Descrizione: Il non cliente vuole caricare i file attraverso un click che apre il gestore file del dispositivo.

Postcondizioni: Il non cliente ha caricato con successo i file attraverso un click.

Scenario Principale:

- Il non cliente clicca l'area di caricamento dei file e tramite il gestore file del dispositivo seleziona i file che intende caricare
- Selezionati i file, questi compariranno nell'area di caricamento e saranno pronti per essere caricati

Scenario Alternativo:

- Se il non cliente seleziona dei file non validi (per dimensioni o tipo) questi non compariranno nell'area di caricamento e comparirà una notifica di avvertimento

UC5.4: Selezione file tramite trascinamento Non Cliente

Attori Principali: Non Cliente.

Precondizioni: Il non cliente ha dato conferma di aver preso visione del regolamento e della privacy policy (UC3).

Descrizione: Il non cliente vuole caricare i file trascinandoli nell'apposito riquadro nella pagina.

Postcondizioni: Il non cliente ha caricato con successo i file trascinandoli nell'apposito riquadro nella pagina.

Scenario Principale:

- Il non cliente trascina i file che desidera caricare nell'area di caricamento (funzionalità esclusiva solo agli utenti desktop)
- Selezionati i file, questi compariranno nell'area di caricamento e saranno pronti per essere caricati

Scenario Alternativo:

- Se il non cliente trascina dei file non validi (per dimensioni o tipo) questi non compariranno nell'area di caricamento e comparirà una notifica di avvertimento

UC6: Uscita dalla pagina

Attori Principali: Utente.

Precondizioni: L'utente ha caricato con successo i file del primo viaggio (UC4, UC5).

Descrizione: L'utente dopo aver caricato almeno una vacanza vuole essere reindirizzato a www.girolibero.it.

Postcondizioni: L'utente è stato reindirizzato a www.girolibero.it.

Scenario Principale:

- L'utente preme il pulsante che lo reindirizzerà a www.girolibero.it, chiudendo la pagina attuale

UC7: Upload secondario

Attori Principali: Utente, Cloudinary, DBMS Wordpress.

Precondizioni: L'utente ha caricato con successo i file del primo viaggio (UC4, UC5).

Descrizione: L'utente vuole caricare i file di un altro viaggio compiuto in modo indipendente.

Postcondizioni: L'utente ha caricato con successo i file del viaggio.

Scenario Principale:

- Dopo aver caricato la prima volta i file di un viaggio, l'utente preme il pulsante per caricare i file di un altro viaggio
- L'utente inserisce le informazioni del viaggio, sceglie i file e quando è pronto a inviare preme un pulsante
- I file vengono caricati su Cloudinary rinominati nel modo corretto
- Nel log di upload viene aggiunta l'informazione di upload
- Terminate queste operazioni il form si disabilita (non può essere modificato dal'utente ma rimane visibile) e compare sotto di esso una conferma esplicita della riuscita dell'upload e due nuovi pulsanti per le operazioni disponibili (UC6 e UC7)

Scenario Alternativo:

- Se le informazioni del viaggio non sono valide, verrà visualizzata una notifica

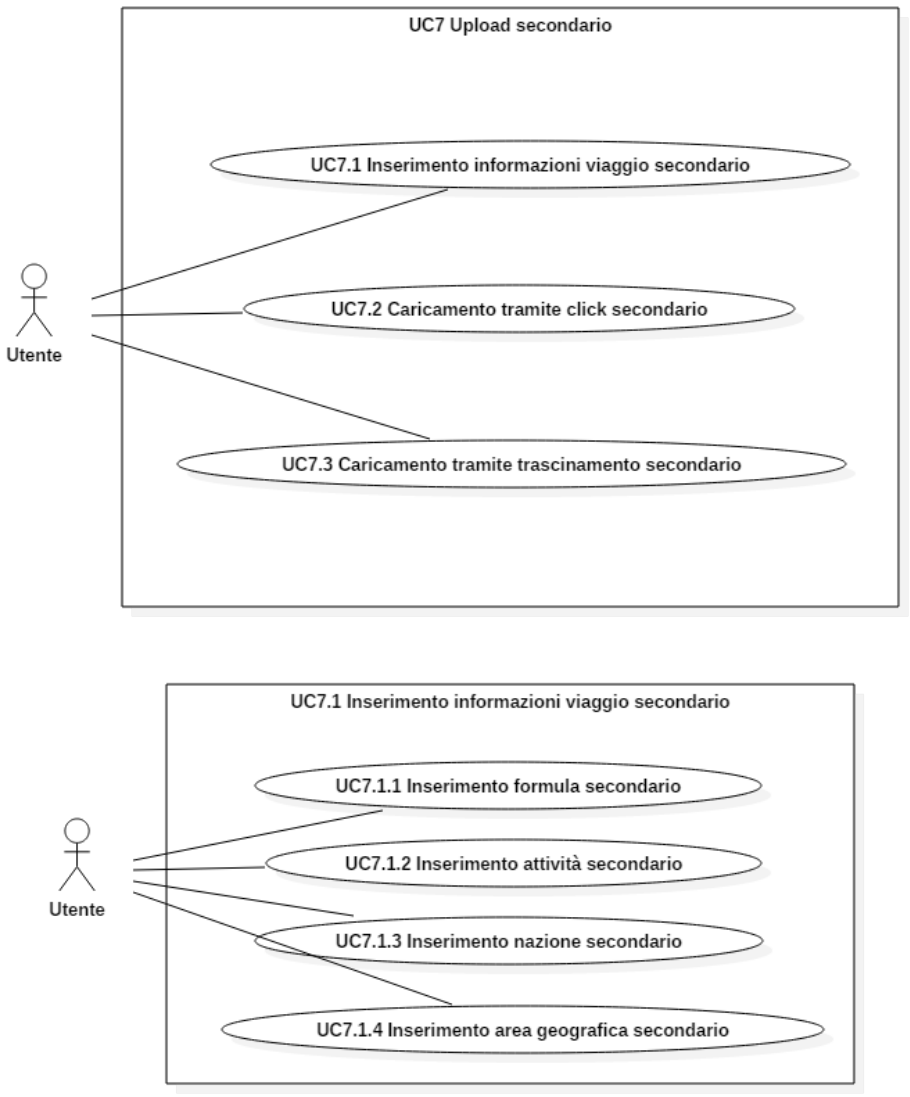


Figura 3.6: Diagrammi UC7

- Se l'upload dei file non riesce il server risponde con un errore e verrà visualizzata una notifica all'utente di riprovare in un secondo momento

UC7.1: Inserimento informazioni viaggio secondario

Attori Principali: Utente.

Precondizioni: L'utente ha caricato con successo i file del primo viaggio (UC4, UC5).

Descrizione: L'utente vuole inserire le informazioni del viaggio da associare ai file che vuole caricare.

Postcondizioni: L'utente ha inserito con successo le informazioni del viaggio.

Scenario Principale:

- Nella procedura di caricamento è necessario che l'utente inserisca le informazioni del viaggio effettuato

UC7.1.1: Inserimento formula secondario

Attori Principali: Utente.

Precondizioni: L'utente ha caricato con successo i file del primo viaggio (UC4, UC5).

Descrizione: L'utente vuole inserire in un apposito campo la formula del viaggio.

Postcondizioni: L'utente ha inserito con successo la formula del viaggio.

Scenario Principale:

- Nella procedura di caricamento è necessario che l'utente inserisca la formula del viaggio, selezionandola tra le varie opzioni proposte in un menù a tendina

UC7.1.2: Inserimento attività secondario

Attori Principali: Utente.

Precondizioni: L'utente ha caricato con successo i file del primo viaggio (UC4, UC5).

Descrizione: L'utente vuole inserire in un apposito campo l'attività svolta durante il viaggio.

Postcondizioni: L'utente ha inserito con successo l'attività svolta.

Scenario Principale:

- Nella procedura di caricamento è necessario che l'utente inserisca l'attività del viaggio, selezionandola tra le varie opzioni proposte in un menù a tendina

UC7.1.3: Inserimento nazione secondario

Attori Principali: Utente.

Precondizioni: L'utente ha caricato con successo i file del primo viaggio (UC4, UC5).

Descrizione: L'utente vuole inserire in un apposito campo la nazione che ha visitato durante il suo viaggio.

Postcondizioni: L'utente ha inserito con successo la nazione visitata.

Scenario Principale:

- Nella procedura di caricamento è necessario che l'utente inserisca la nazione visitata, selezionandola tra le varie opzioni proposte in un menù a tendina
- Data la lunghezza della lista delle opzioni, l'utente può anche cercare la nazione tramite casella di testo

UC7.1.4: Inserimento area geografica secondario

Attori Principali: Utente.

Precondizioni: L'utente ha caricato con successo i file del primo viaggio (UC4, UC5).

Descrizione: L'utente vuole inserire in un apposito campo opzionale l'area geografica che ha visitato durante il suo viaggio.

Postcondizioni: L'utente ha inserito con successo l'area geografica visitata.

Scenario Principale:

- Nella procedura di caricamento l'utente può opzionalmente inserire l'area geografica del viaggio in un campo di testo

UC7.2: Selezione file tramite click secondario

Attori Principali: Utente.

Precondizioni: L'utente ha caricato con successo i file del primo viaggio (UC4, UC5).

Descrizione: L'utente vuole caricare i file attraverso un click che apre il gestore file del dispositivo.

Postcondizioni: L'utente ha caricato con successo i file attraverso un click.

Scenario Principale:

- L'utente clicca l'area di caricamento dei file e tramite il gestore file del dispositivo seleziona i file che intende caricare
- Selezionati i file, questi compariranno nell'area di caricamento e saranno pronti per essere caricati

Scenario Alternativo:

- Se l'utente seleziona dei file non validi (per dimensioni o tipo) questi non compariranno nell'area di caricamento e comparirà una notifica di avvertimento

UC7.3: Selezione file tramite trascinamento secondario

Attori Principali: Utente.

Precondizioni: L'utente ha dato conferma di aver preso visione del regolamento e della privacy policy (UC3).

Descrizione: L'utente vuole caricare i file trascinandoli nell'apposito riquadro nella pagina.

Postcondizioni: L'utente ha caricato con successo i file trascinandoli nell'apposito riquadro nella pagina.

Scenario Principale:

- L'utente trascina i file che desidera caricare nell'area di caricamento (funzionalità esclusiva solo agli utenti desktop)
- Selezionati i file, questi compariranno nell'area di caricamento e saranno pronti per essere caricati

Scenario Alternativo:

- Se l'utente trascina dei file non validi (per dimensioni o tipo) questi non compariranno nell'area di caricamento e comparirà una notifica di avvertimento

UC8: Scrittura introduzione contest

Attori Principali: Personale Marketing.

Precondizioni: Il sito web www.girolibero.it è attivo sul server.

Descrizione: Il personale marketing Girolibero vuole scrivere o modificare l'introduzione all'inizio della pagina.

Postcondizioni: Il personale marketing Girolibero ha scritto o modificato con successo l'introduzione all'inizio della pagina.

Scenario Principale:

- Il personale marketing Girolibero scrive o aggiorna da backend l'introduzione all'inizio della pagina

UC9: Visione log di upload

Attori Principali: Personale Marketing, DBMS Wordpress.

Precondizioni: Il sito web www.girolibero.it è attivo sul server.

Descrizione: Il personale marketing Girolibero vuole visualizzare una lista di tutte le persone che hanno effettuato l'upload di file.

Postcondizioni: Il personale marketing Girolibero ha visualizzato la lista di tutte le persone che hanno effettuato l'upload di file.

Scenario Principale:

- Il personale marketing Girolibero visualizza la lista di tutte le persone che hanno effettuato l'upload di file da backend

3.2 Requisiti

Di seguito vengono riportati i requisiti individuati tramite i casi d'uso elencati nella sezione precedente e riunioni con tutto il personale che ha seguito il progetto.
I requisiti individuati si dividono in:

- funzionali
- qualitativi
- di vincolo

Ogni requisito inoltre viene classificato tramite un livello di importanza, che può essere:

- obbligatorio
- desiderabile
- opzionale

Tabella 3.1: Tabella del tracciamento dei requisti funzionali

Requisito	Descrizione	Classificazione	Fonte
RF1	La pagina deve contenere nel viewport un'introduzione scritta redatta dal personale marketing per il concorso	Obbligatorio	UC1
RF2	La pagina deve contenere una breve descrizione sui dettagli dei file che è possibile caricare	Obbligatorio	UC2
RF2.1	La descrizione sui dettagli dei file deve contenere una descrizione sui dettagli delle foto che è possibile caricare	Obbligatorio	UC2.1
RF2.1.1	La descrizione sui dettagli delle foto deve contenere il numero massimo delle foto che è consentito caricare per un viaggio	Obbligatorio	UC2.1.1
RF2.1.2	La descrizione sui dettagli delle foto deve contenere la dimensione minima delle foto che è consentito caricare per un viaggio	Obbligatorio	UC2.1.2
RF2.1.3	La descrizione sui dettagli delle foto deve contenere la dimensione massima delle foto che è consentito caricare per un viaggio	Obbligatorio	UC2.1.3
RF2.2	La descrizione sui dettagli dei file deve contenere una descrizione sui dettagli dei video che è possibile caricare	Obbligatorio	UC2.2

Requisito	Descrizione	Classificazione	Fonte
RF2.2.1	La descrizione sui dettagli dei video deve contenere il numero massimo dei video che è consentito caricare per un viaggio	Obbligatorio	UC2.2.1
RF2.2.2	La descrizione sui dettagli dei video deve contenere la dimensione massima dei video che è consentito caricare per un viaggio	Obbligatorio	UC2.2.2
RF3	La pagina deve richiedere all'utente di aver preso visione del regolamento e della privacy policy	Obbligatorio	UC3
RF3.1	La pagina deve avere un link alla pagina www.girolibero.it/contest-regolamento	Obbligatorio	UC3.1
RF3.2	La pagina deve avere un link alla pagina www.girolibero.it/privacy-policy	Obbligatorio	UC3.2
RF4	La pagina deve permettere ad un cliente di caricare i file del proprio viaggio Girolibero	Obbligatorio	UC4
RF4.1	Il cliente deve poter selezionare i file da caricare tramite un click che apre il gestore file del dispositivo	Obbligatorio	UC4.1
RF4.2	Il cliente deve poter selezionare i file da caricare trascinandoli nell'apposito riquadro di caricamento nella pagina	Obbligatorio	UC4.2
RF5	La pagina deve permettere ad un non cliente di caricare i file del proprio viaggio indipendente	Obbligatorio	UC5
RF5.1	Il non cliente deve poter inserire i propri dati personali da associare ai suoi file	Obbligatorio	UC5.1
RF5.1.1	Il non cliente deve poter inserire il suo nome	Obbligatorio	UC5.1.1
RF5.1.2	Il non cliente deve poter inserire il suo cognome	Obbligatorio	UC5.1.2
RF5.1.3	Il non cliente deve poter inserire la sua email	Obbligatorio	UC5.1.3
RF5.2	Il non cliente deve poter inserire le informazioni del viaggio da associare ai suoi file	Obbligatorio	UC5.2
RF5.2.1	Il non cliente deve poter inserire la formula del viaggio	Obbligatorio	UC5.2.1
RF5.2.2	Il non cliente deve poter inserire l'attività svolta	Obbligatorio	UC5.2.2

Requisito	Descrizione	Classificazione	Fonte
RF5.2.3	Il non cliente deve poter inserire la nazione visitata	Obbligatorio	UC5.2.3
RF5.2.4	Opzionalmente il non cliente deve poter inserire l'area geografica visitata	Obbligatorio	UC5.2.4
RF5.3	Il non cliente deve poter selezionare i file da caricare tramite un click che apre il gestore file del dispositivo	Obbligatorio	UC5.3
RF5.4	Il non cliente deve poter selezionare i file da caricare trascinandoli nell'apposito riquadro di caricamento nella pagina	Obbligatorio	UC5.4
RF6	Caricati con successo dei file almeno una volta, l'utente deve poter uscire dalla pagina ed essere reindirizzato a www.girolibero.it	Obbligatorio	UC6
RF7	Caricati con successo dei file almeno una volta, l'utente deve poter caricare file di un altro viaggio compiuto in modo indipendente	Obbligatorio	UC7
RF7.1	L'utente deve poter inserire le informazioni del viaggio da associare ai suoi file	Obbligatorio	UC7.1
RF7.1.1	L'utente deve poter inserire la formula del viaggio	Obbligatorio	UC7.1.1
RF7.1.2	L'utente deve poter inserire l'attività svolta	Obbligatorio	UC7.1.2
RF7.1.3	L'utente deve poter inserire la nazione visitata	Obbligatorio	UC7.1.3
RF7.1.4	Opzionalmente l'utente deve poter inserire l'area geografica visitata	Obbligatorio	UC7.1.4
RF7.2	L'utente deve poter selezionare i file da caricare tramite un click che apre il gestore file del dispositivo	Obbligatorio	UC7.2
RF7.3	L'utente deve poter selezionare i file da caricare trascinandoli nell'apposito riquadro di caricamento nella pagina	Obbligatorio	UC7.3
RF8	Il personale marketing deve poter scrivere o modificare l'introduzione della pagina	Obbligatorio	UC8
RF9	Il personale marketing deve poter visualizzare una lista di tutte le persone che hanno effettuato l'upload di file	Obbligatorio	UC9

Requisito	Descrizione	Classificazione	Fonte
RFE.1	L'utente va notificato quando prova a continuare nella pagina senza aver preso visione del regolamento e della privacy policy	Obbligatorio	UC3
RFE.2	Il non cliente deve essere notificato quando inserisce informazioni personali non valide	Obbligatorio	UC5
RFE.3	L'utente deve essere notificato quando inserisce dati del viaggio non validi	Obbligatorio	UC5, UC7
RFE.4	L'utente deve essere notificato quando seleziona file non validi per il caricamento	Obbligatorio	UC4.1, UC4.2, UC5.3, UC5.4, UC7.2, UC7.3
RFE.5	L'utente deve essere notificato di eventuali problemi lato server durante il caricamento	Obbligatorio	UC4, UC5, UC7
RF10	Deve essere predisposto il backend per la gestione multilingua dei campi testo	Desiderabile	Riunione

Tabella 3.2: Tabella del tracciamento dei requisiti qualitativi

Requisito	Descrizione	Classificazione	Fonte
RQ1	La pagina deve mantenere le scelte di design del sito girolibero.it	Obbligatorio	Riunione
RQ2	Per scelte di design non presenti nel sito girolibero.it, ci si deve affidare all'UX designer	Obbligatorio	Riunione
RQ3	La pagina deve funzionare sui principali browser presenti su Windows e macOS	Desiderabile	Riunione
RQ3.1	La pagina deve funzionare sull'ultima versione di Google Chrome	Desiderabile	Riunione
RQ3.2	La pagina deve funzionare sull'ultima versione di Firefox	Desiderabile	Riunione
RQ3.3	La pagina deve funzionare sull'ultima versione di Safari	Desiderabile	Riunione
RQ4	Devono essere eseguiti dei test per il backend	Desiderabile	Riunione
RQ5	Devono essere eseguiti dei test per l'interfaccia web	Desiderabile	Riunione

Tabella 3.3: Tabella del tracciamento dei requisiti di vincolo

Requisito	Descrizione	Classificazione	Fonte
RV1	Deve essere usato Wordpress come CMS della pagina	Obbligatorio	Riunione
RV2	Il backend deve essere sviluppato in PHP	Obbligatorio	Riunione
RV3	Il frontend deve essere sviluppato in Twig/Timber	Obbligatorio	Riunione
RV4	Gli script frontend devono essere sviluppati in Javascript/JQuery	Obbligatorio	Riunione
RV5	I fogli stile devono essere sviluppati in Sass	Obbligatorio	Riunione
RV6	Per quanto possibile, devono essere usati i fogli stile già scritti per girolibero.it	Desiderabile	Riunione
RV7	Deve essere implementato il menù presente in girolibero.it	Desiderabile	Riunione

Capitolo 4

Progettazione e implementazione

4.1 Tecnologie principali

Questa sezione raccoglie e descrive le tecnologie principali usate per il progetto. Tutte queste tecnologie sono state scelte durante la fase di analisi del progetto perchè si è voluto ricercare quanta più somiglianza con l'attuale codifica del sito girolibero.it per una futura integrazione (non scopo del progetto).

Un possibile svantaggio di questa scelta è che si è limitati nello scegliere le tecnologie più adatte per questo singolo progetto. Un vantaggio è che essendo tutte tecnologie già usate dal team di sviluppo sono in grado di garantire aiuto e assistenza per eventuali problemi riscontrati durante lo sviluppo.

4.1.1 Cloudinary

Cloudinary è una piattaforma di gestione multimediale basata su cloud che offre servizi per caricare, archiviare, ottimizzare e distribuire contenuti multimediali su siti web e applicazioni.

Girolibero usa questa piattaforma per archiviare tutti i contenuti multimediali del sito girolibero.it. Questo include immagini di sfondo, foto e video per caroselli e immagini informative dei viaggi.

Accedendo tramite credenziali ad un account associato all'azienda è possibile visualizzare una dashboard con tutte le informazioni di utilizzo del DAM e sfogliare la "media library" (figura 4.1), ossia una raccolta di tutti i file caricati, con la possibilità di filtrare per tipo, dimensioni e data di caricamento.

Ogni asset (file multimediale) caricato su Cloudinary è identificato da un *public id*[g] univoco. Inoltre si ha la possibilità di associare dei tag ad ogni asset per una migliore categorizzazione. Il public id e i tag verranno utilizzati nel progetto durante il caricamento dei file.

Sito web di Cloudinary. URL: https://cloudinary.com/

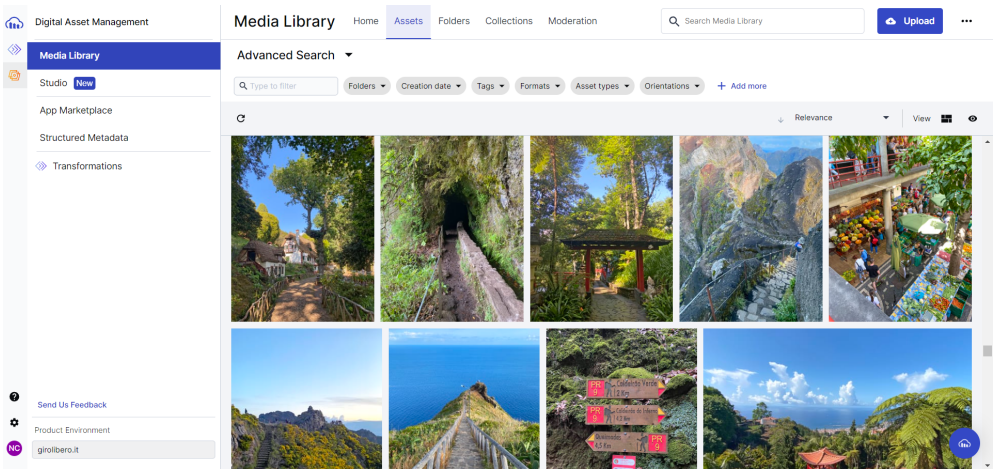


Figura 4.1: Visualizzazione della "media library" di Cloudinary

4.1.2 Wordpress

Wordpress è il CMS più popolare per la creazione e gestione di siti web. È usato dal 63% dei siti che usano un CMS, ed è alla base del 43.1% di tutti i siti web[1]. La sua popolarità è molto probabilmente data dal fatto che il software è interamente *open source*[g]con licenza *GNU General Public License (GPLv2)*[g]. Per poterlo usare infatti basta solamente scaricare una delle versioni presenti nel sito web ufficiale wordpress.org ed installarla nel proprio web server. Esistono due prodotti distinti associati al nome Wordpress, ma solo uno di questi è rilevante in questa sezione:

- Wordpress.com offre un servizio a pagamento basato sul software Wordpress, più semplice da usare ma con maggiori limitazioni. Questa è la soluzione ideale per chi vuole aprire un sito web senza conoscenze di web development e senza dover gestire un host web
- Wordpress.org o Wordpress Self-Hosted è il software Wordpress gratuito e open source che è possibile scaricare e installare nel proprio host web. Sono ovviamente necessarie conoscenze di web development

Ogni sito web Wordpress è composto da quattro parti:

- i file del "core", che sono tutti quei file che compongono la base per far funzionare Wordpress
- il database (generalmente MySQL) per archiviare informazioni cruciali come articoli, pagine, commenti, impostazioni e metadati. La struttura del database è definita nei file del core di WordPress e può essere modificata tramite il sistema di gestione del database
- i temi (o template), che danno struttura e stile alle pagine del sito. I temi quindi gestiscono come verranno visualizzati i contenuti nel frontend

[1] *Usage statistics and market share of WordPress.* URL: https://w3techs.com/technologies/details/cm-wordpress.

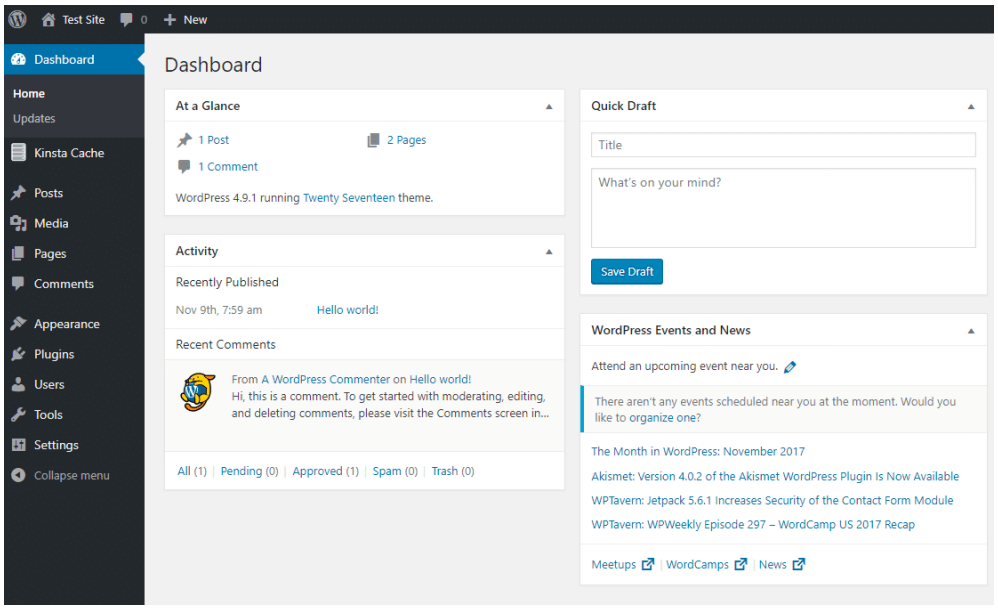


Figura 4.2: Console dell'amministratore di Wordpress

- i plugin, che aggiungono funzionalità extra a Wordpress. Sono sviluppati da terze parti e possono essere facilmente installati e attivati

Dopo aver effettuato l'installazione, è possibile accedere alla console di amministrazione (figura 4.2) tramite credenziali scelte durante l'installazione. Per accedere alla console basta digitare sulla barra degli indirizzi di un browser http://sito.com/wp-admin. Wordpress usa due termini specifici per identificare due tipi di pagine web di un sito:

- post, ossia pagine di articoli di blog o comunque qualsiasi tipologia di pagina che viene ripetuta nel tempo
- pagine, ossia vere e proprie pagine statiche del sito

Dalla console di amministrazione è possibile creare entrambe le cose. I contenuti andranno visualizzati nel sito seguendo la struttura e lo stile del tema installato o sviluppato. Il compito dello sviluppatore web quando usa Wordpress è proprio quello di creare il tema che verrà utilizzato per tutto il sito. I contenuti verranno creati da altri utenti che non devono per forza possedere competenze di sviluppo web.

Sito web di Wordpress.org. URL: https://wordpress.org/

4.1.3 PHP

PHP è un linguaggio di scripting ampiamente utilizzato per lo sviluppo web. È un linguaggio di scripting lato server, il che significa che il suo codice viene eseguito sul server web prima di inviare il risultato al browser del cliente. Questo consente di generare dinamicamente pagine web basate su dati, input dell'utente e altro.

```
{% if users|length > 0 %}
    <ul>
        {% for user in users %}
            <li>{{ user.username|e }}</li>
        {% endfor %}
    </ul>
{% endif %}
```

Figura 4.3: Esempio di utilizzo delle strutture if e for in Twig

Ha una sintassi simile ad altri linguaggi di programmazione, permette di manipolare codice HTML e per questo ogni script PHP inizia con il tag <?php e finisce con ?>.

PHP supporta variabili non tipizzate per la memorizzazione di valori e offre le tipiche strutture di controllo di flusso che possono essere trovate negli altri linguaggi di programmazione.

È possibile definire e utilizzare funzioni personalizzate in PHP per organizzare il codice in blocchi riutilizzabili. PHP fornisce anche molte funzioni predefinite per compiti comuni come la manipolazione di stringhe, la gestione di array, la connessione al database e altro.

PHP permette il dialogo con il server web per gestire richieste e risposte HTTP e l'invio di dati e file dal browser.

PHP è alla base del CMS Wordpress descritto nella sezione precedente e quindi scelta obbligatoria come linguaggio di backend.

Sito web di PHP. URL: https://www.php.net/

4.1.4 Twig / Timber

Twig è un motore di template il cui scopo principale è separare il codice PHP dalla logica di markup HTML in applicazioni web, consentendo una gestione più pulita e più organizzata.

Un tipico file Twig è un semplice file di testo con all'interno tag HTML che vanno a formare la struttura della pagina. Il contenuto di questa pagina invece viene gestito da pezzi di codice in sintassi Twig (contenuti in { } per definire variabili e {% %} per il controllo di flusso, figura 4.3).

Twig supporta la definizione di blocchi ed ereditarietà tra template. Per esempio, è

possibile creare un template di base per poi estendere o modificare dei blocchi al suo interno in template figli.

Per poter usare Twig dentro ad un sito Wordpress è necessario installare il plugin Timber dalla console di amministrazione di Wordpress.

Sito web di Twig. URL: https://twig.symfony.com/

4.1.5 Javascript / jQuery

JavaScript è un linguaggio di scripting lato client principalmente utilizzato per aggiungere interattività e dinamicità alle pagine web. Il codice JavaScript viene eseguito direttamente sul browser e non sul server come nel caso di PHP.
Oltre ad avere una sintassi semplice da imparare perchè simile a quella di altri linguaggi, permette la manipolazione del *Document Object Model (DOM)*[g], la gestione degli eventi e le richieste HTTP asincrone *Asynchronous JavaScript and XML (AJAX)*[g].

jQuery è invece una libreria tra le più popolari per JavaScript, che semplifica notevolmente la scrittura di codice JavaScript, in particolare le operazioni sul DOM.

Documentazione di JavaScript. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript

Sito web di jQuery. URL: https://jquery.com/

4.1.6 CSS / Sass

CSS, o Cascading Style Sheets, è un linguaggio di stile utilizzato per definire la presentazione e la formattazione di documenti HTML e XML. In altre parole, CSS determina come un sito web appare e viene visualizzato nel browser.

Sass (Syntactically Awesome Stylesheets) è un'estensione di CSS che aggiunge funzionalità avanzate al linguaggio, rendendolo più potente e flessibile per la scrittura di fogli di stile.
Tra queste funzionalità troviamo:

- Variabili: Sass permette di definire variabili per memorizzare valori ripetuti, come colori o dimensioni, e utilizzarli in tutto il foglio di stile. Questo rende più facile apportare modifiche globali al design
- Nesting: Sass supporta il nesting, il che significa che è possibile annidare selettori CSS all'interno di altri selettori. Questo aiuta a organizzare il codice in una struttura più chiara e gerarchica
- Mixin: I mixin sono blocchi di codice riutilizzabili che possono essere inclusi all'interno di altri selettori. Ciò facilita l'applicazione di regole di stile comuni a più selettori
- Ereditarietà: Sass consente di ereditare le proprietà da un selettore padre a uno o più selettori figli. Questo promuove la riutilizzabilità del codice

- Partials e importazioni: Sass consente di dividere il codice in parti più piccole chiamate "partials" e importarle nei fogli di stile principali. Ciò favorisce la modularità e la manutenibilità
- Operazioni matematiche: Sass supporta operazioni matematiche direttamente nei fogli di stile, il che è utile per calcolare dimensioni dinamiche
- Variabili dinamiche: Sass permette l'uso di variabili dinamiche che possono essere calcolate in base ad altre variabili o valori

I fogli stile Sass non sono direttamente leggibili dai browser ed è quindi necessario compilarli per creare dei fogli stile in formato CSS.

Documentazione di CSS. URL: https://developer.mozilla.org/en-US/docs/Web/CSS

Sito web di Sass. URL: https://sass-lang.com/

4.2 Scelte progettuali

4.2.1 Come rinominare i file

La modalità di rinominazione dei file caricati dagli utenti è stato uno dei primi argomenti affrontati in questa esperienza di stage. Il comportamento di default di Cloudinary quando viene caricato un file con un public id già esistente nell'archivio è semplicemente quello di sovrascrivere il vecchio file con quello nuovo. Questo implica che bisogna trovare una nomenclatura univoca per ogni file caricato. Inoltre il public id deve essere composto in modo da essere quanto più informativo possibile sul viaggio compiuto per motivi di filtraggio.

Ogni file caricato da un cliente Girolibero avrà la seguente nomenclatura:
codweb-nazione-titolo-da-(a)-attività-formula-durata-filename

- *codweb*[g], il codice alfanumerico di 5 cifre con il quale viene identificato un viaggio Girolibero
- nazione, la nazione dove si è svolto il viaggio
- titolo, il titolo del viaggio assegnato da Girolibero
- da/a, città di partenza e città di arrivo del viaggio (nel caso di viaggi "ad anello" è presente solo la città di partenza)
- *attività*[g], l'attività svolta durante il viaggio (in bici, in bici e barca, trekking, tour)
- *formula*[g], l'organizzazione del viaggio (in gruppo, indipendente, per famiglie)
- durata, durata del viaggio in numero di giorni seguito da "gg"
- filename, nome del file caricato

Ad ogni file caricato dall'utente vengono aggiunti dei tag contententi la stringa "contest", l'anno di upload (per esempio "2023"), il mese del viaggio (per esempio "settembre"), il nome e il cognome.

Ogni file caricato da un non cliente avrà la seguente nomenclatura:
PCM-nazione-(area geografica)-attività-formula-filename

- PCM, codice per identificare viaggi non di Girolibero svolti in modo autonomo
- nazione, la nazione dove si è svolto il viaggio
- area geografica, l'area geografica, regione o città dove si è svolto il viaggio (opzionale)
- attività, l'attività svolta durante il viaggio (in bici, in bici e barca, trekking, tour)
- formula, l'organizzazione del viaggio (in gruppo, indipendente, per famiglie)
- filename, nome del file caricato

Ad ogni file caricato dall'utente vengono aggiunti dei tag contententi la stringa "contest", l'anno di upload (per esempio "2023"), il nome e il cognome.

4.2.2 Upload su Cloudinary

La seconda questione da affrontare è stata la decisione di quale metodo di upload utilizzare per caricare i file su Cloudinary. Compiendo una analisi preliminare, il mio tutor ha scritto all'assistenza Cloudinary un'email per chiedere quale fosse la migliore opzione per consentire a degli utenti il caricamento di foto e video da browser verso l'archivio Cloudinary. Come risposta è stato indicato un widget di upload sviluppato da Cloudinary.

L'upload widget (figura 4.4) di Cloudinary è un'interfaccia utente che permette agli utenti di un sito web di caricare file da molte fonti (non solo file locali del dispositivo). Il widget richiede poche linee di codice per essere integrato e elimina il bisogno di sviluppare soluzioni personalizzate per l'upload.

Le fonti da cui è possibile scegliere i file sono:

- gestione file del dispositivo
- URL remoto
- fotocamera del dispositivo
- cerca immagini di Google
- social network come Instagram
- siti web di immagini stock come GettyImages

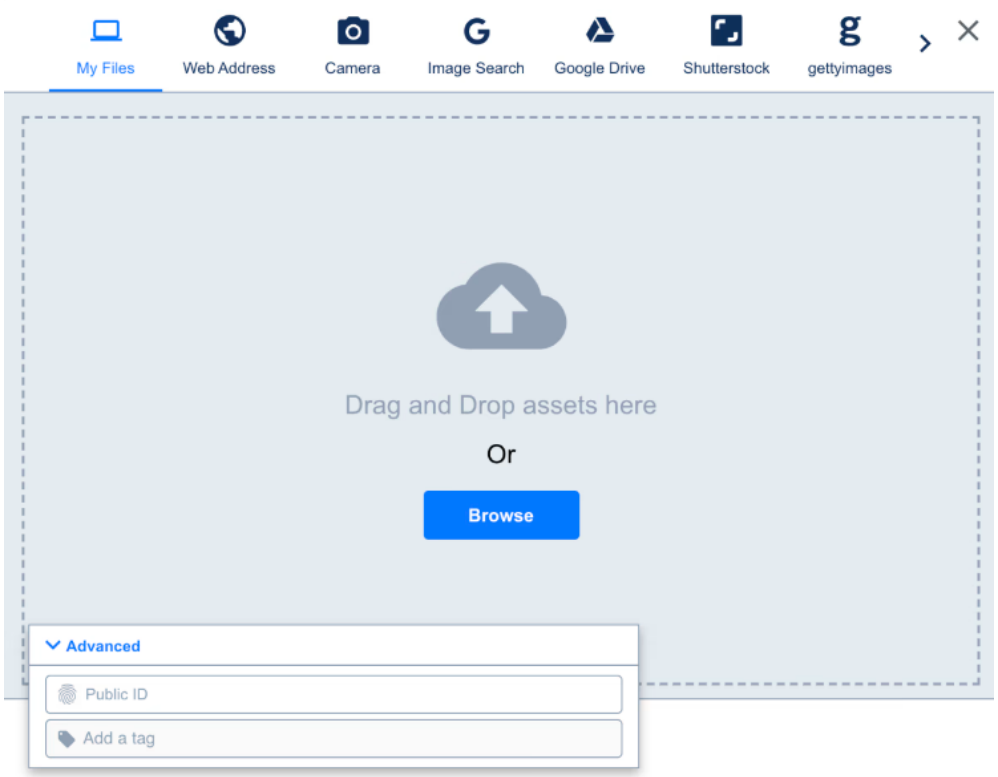


Figura 4.4: Widget di upload di Cloudinary

Il widget è responsive, supporta il trascinamento dei file e gestisce gli errori durante il caricamento. Purtroppo lo stile è fisso e non personalizzabile e l'upload non passa per il web server, impedendo così validazioni aggiuntive prima del caricamento.

Una seconda opzione è stata trovata durante lo studio di Cloudinary e la sua documentazione completa. Cloudinary offre una serie di *Software Development Kit (SDK)*[g]per i principali linguaggi e framework odierni che permettono agli sviluppatori di creare soluzioni personalizzate per l'upload nell'archivio di file e la loro manipolazione. Esistono SDK per Node.js, Python, PHP, Java, Ruby, .NET, GO, Dart, React, Vue.js, Angular, JavaScript, jQuery, Swift, Android, Kotlin e Flutter. È stato deciso di utilizzare l'SDK di PHP e di sviluppare la logica di upload nel backend.

Documentazione SDK Cloudinary per PHP. URL: `https://cloudinary.com/documentation/php_integration`

Per effettuare l'upload di un file in PHP bisogna seguire i seguenti passi:

- creare un oggetto Cloudinary, che va inizializzato fornendo "cloud name", "api key" e "api secret". Queste informazioni sono disponibili nella dashboard di Cloudinary dopo aver effettuato l'accesso con un account aziendale
- utilizzando l'oggetto appena creato, ricavare da esso un'istanza dell'upload API
- tramite metodo "upload" dell'upload API, caricare il file passando anche vari parametri (come public id e tag associati ad esso)

Messi a confronto, questi due metodi di upload sono uno l'opposto dell'altro. Il primo è pronto all'uso ma con personalizzazione quasi nulla e troppe funzionalità che non si vogliono dare all'utente, mentre il secondo richiede più lavoro da parte dello sviluppatore ma permette la creazione di una soluzione completamente ad hoc sia per quanto riguarda comportamento che per stile e presentazione. Per questo progetto è stata scelta la seconda opzione proprio per questi motivi.

4.2.3 Selezione dei file da caricare

Una volta deciso il metodo di upload, è stato creato un proof of concept con interfaccia utente minimale. Subito è risultato evidente come il tag HTML di default per l'input di file non fosse in grado di soddisfare le esigenze del progetto. Infatti esso non è in grado di gestire il caricamento di file tramite trascinamento. Inoltre non è personalizzabile dal punto di vista dello stile.

Scartata questa opzione, si è andati alla ricerca di qualche soluzione adatta al problema. L'obiettivo era una metodologia di input di file da browser che:

- gestisse caricamento sia da gestore file del dispositivo che con trascinamento
- fosse personalizzabile per quanto riguarda lo stile
- fosse ben documentato e testato

Si sono presentate due opzioni:

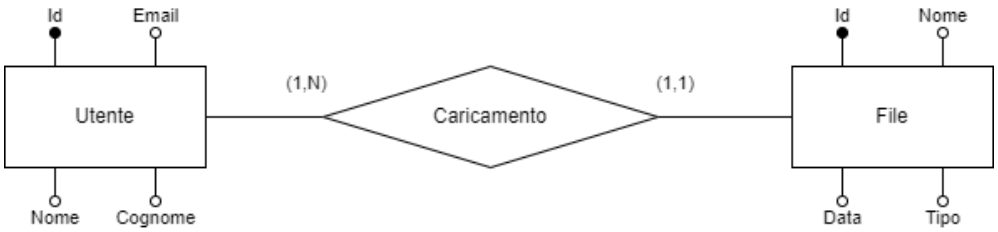


Figura 4.5: Diagramma ER

- HTML Drag and Drop API, un insieme di interfacce, metodi ed eventi forniti dai browser moderni che interagiscono con il DOM di una pagina e permettono il trascinamento di elementi (anche file se adattato)
- Dropzone.js, una libreria JavaScript open source che rimpiazza il classico tag di input file, completamente personalizzabile e che gestisce direttamente l'upload al server dei file selezionati

La scelta finale è ricaduta sulla libreria Dropzone.js, perchè documentata a fondo e ampiamente usata e testata. Per poterla usare è necessario scaricare i file della libreria e includerli nel progetto. I dettagli implementativi verranno trattati in una prossima sezione.

Documentazione HTML Drag and Drop API. URL: https://developer.mozilla.org/en-US/docs/Web/API/HTML_Drag_and_Drop_API

Sito web di Dropzone.js. URL: https://www.dropzone.dev/

4.2.4 Log di caricamento

Per quanto riguarda il log di upload, la soluzione trovata è semplicemente di andare a salvare le informazioni in tabelle MySQL create all'interno del database di Wordpress.

Quello che si vuole andare a salvare per ogni upload di file è:

- email dell'utente che ha effettuato l'upload
- nome dell'utente
- cognome dell'utente
- nome del file caricato
- data di caricamento
- tipologia di caricamento (Cliente o PCM)

Ovviamente per salvare questi dati vanno usate due tabelle, una con le informazioni dell'utente e una con le informazioni delle foto.

Questo è lo schema relazionale ricavato dal diagramma 4.5:

Utente(id, email, nome, cognome)
File(id, utente, nome, data, tipo)
File.utente -> Utente.id

Per quanto riguarda la visione di questi dati, era stata espressa la necessità del personale marketing di voler visualizzare questi dati tramite un foglio di calcolo. Per farlo è necessario usare una query che riunisca queste due tabelle in un unico risultato tabellare, esportabile poi in formato csv. La query in questione è:

SELECT Utente.email, Utente.nome, Utente.cognome, File.nome, File.data, File.tipo
FROM Utente JOIN File ON Utente.id = File.utente

4.2.5 API Girolibero

Il team di sviluppo di Girolibero ha realizzato nel tempo una propria API per scopi interni. Questa API è usata principalmente per ottenere risultati di query dal database principale e per lo scambio di dati tra le varie applicazioni interne. Questo progetto ha fatto uso di alcune richieste all'API Girolibero. Di seguito vengono elencati tutti gli usi e le modalità:

- ottenimento dati di viaggio del cliente da preventivo: nel caso del cliente, la pagina viene aperta tramite un link inviato per email. Questo link contiene un parametro denominato "id", che contiene una stringa cifrata. Questa stringa una volta decifrata permette di ottenere l'id del preventivo che identifica il viaggio compiuto dal cliente. Tramite una richiesta API con questo id, è possibile ottenere un JSON con le informazioni del viaggio, tra cui il codweb, il nome, il cognome e l'email del cliente e l'anno e il mese di viaggio.
- ottenimento informazioni di viaggio da codweb: con il codweb ottenuto in precedenza è possibile effettuare un'altra richiesta all'API, che ritorna un JSON con tutte le informazioni del viaggio, tra cui quelle che interessano per comporre il public id, che sono nazione, titolo del viaggio, da/a, attività, formula e durata.
- ottenimento del menù (spiegato nella prossima sezione del documento)
- iscrizione alla newsletter: anche questo processo avviene tramite API, mandando una richiesta passando email, nome, cognome e lingua dell'utente che si vuole iscrivere alla newsletter

4.2.6 Menù

La volontà di avere un menù che rispecchiasse quanto più possibile quello di girolibero.it è nata in un secondo momento e non durante le fasi iniziali del progetto. Il problema aveva due aspetti fondamentali:

- certe funzionalità del menù (come la ricerca tramite campo di testo) richiedono codice JavaScript e richieste GET che devono essere gestite nel modo giusto
- il menù contiene anche link che continuano a cambiare (come i riferimenti a viaggi in certe nazioni o legati da un certo tema)

La soluzione ideata è stata quella di predisporre l'API a preparare su richiesta blocchi di codice (contenenti le varie parti del menù) e effettuare delle chiamate all'inizializzazione

della pagina per ottenerli e comporre la pagina nel modo corretto. Questo permette di avere sempre una versione aggiornata del menù. Per quanto riguarda lo stile è semplicemente bastato importare i fogli stile rilevanti, mentre per gli script JavaScript sono state necessarie modifiche più importanti al codice.

La predisposizione dell'API e l'import dei fogli di stile e file JavaScript corretti sono stati compiuti dal tutor Marco Matteazzi in coordinazione con il laureando.

4.2.7 Crittografia API

Il passaggio dell'id del preventivo tramite parametro dell'URL è ovviamente un punto delicato, perchè non si vuole dare la possibilità ad un malintenzionato di inserire un qualunque codice a caso.

La soluzione ideata come descritto nella sezione precedente è stata di criptare questo id, in modo che se anche un malintenzionato dovesse provare ad inserire qualcosa al suo posto, questa stringa deve passare per un processo di decodifica e validazione, che se non superato non permette l'accesso alla pagina.

In coordinazione con Simone Fontana, è stata decisa la procedura di criptazione da seguire per inserire nelle email di bentornato che i clienti ricevono dopo il viaggio il link dal quale accedere alla pagina dedicata. Decisi una secret key e un secret iv, la procedura prevede di:

- accodare alla stringa "contest-preventivo-id-" l'id che si vuole criptare
- effettuare l'hash della secret key usando l'algoritmo sha256
- effettuare l'hash del secret iv usando l'algoritmo sha256 e prendere i primi 16 byte
- criptare la stringa usando i componenti descritti qui sopra con l'algoritmo AES-256-CBC
- codificare il risultato in base64

4.2.8 Generazione dei form

In una normale pagina web, se non si vuole rendere tutto accessibile all'utente appena la pagina viene aperta la soluzione ideale è di inserire comunque questa parte nella struttura della pagina, ma di renderla non visibile tramite fogli di stile.

La scelta di design di lasciare caricare i file di quanti viaggi l'utente voglia e soprattutto il fatto di voler lasciare visibili ma non più interagibili i form per l'inserimento dei file non si sposa bene con la tecnica descritta nel paragrafo precedente, perchè non possiamo sapere a priori il numero di viaggi (e quindi di form) di cui l'utente vuole caricare file. Questo ha portato all'unica soluzione possibile: generare dinamicamente (tramite JavaScript) i form man mano che l'utente prosegue nella pagina.

Quando l'utente attraverso un pulsante dà la propria conferma di aver preso visione del regolamento e della privacy policy, parte una funzione che genera il primo form di caricamento (che può essere per clienti o non clienti). Una volta terminato con

successo il primo caricamento di file, il form viene bloccato e all'utente viene proposto di lasciare la pagina (con reindirizzamento a girolibero.it) oppure di caricare i file di un'altro viaggio. Se l'utente sceglie questa seconda opzione verrà generato un nuovo form subito sotto (sempre di tipo PCM). Questo processo viene ripetuto idealmente fino a quando l'utente è soddisfatto.

4.2.9 Integrazione WPML

WPML è un plugin di Wordpress che permette la gestione di siti multilingua. Per i testi statici (ricavati dalle pagine e post Wordpress) la traduzione è semplice perchè direttamente dalla console dell'amministratore Wordpress è possibile scrivere le traduzioni. Per quanto riguarda le stringhe scritte nei vari file di template bisogna procedere in modo diverso, andando a usare una speciale dicitura per indicare ogni stringa traducibile. Questa dicitura è __('stringa in lingua originale', 'gruppo di stringhe di appartenenza'). Il plugin scannerizza tutti i file e cerca queste diciture, per poi offrire in console di amministratore una scheda dove poter effettuare la traduzione di ciascuna di queste stringhe.

Sito web di WPML. URL: https://wpml.org/

Il problema si pone quando la pagina web genera stringhe a frontend (come nel caso dei form della pagina) perchè WPML va a controllare solo i file di template. La soluzione è preparare queste stringhe comunque a backend nei template, e trovare qualche modo per comunicarle al frontend.

Il metodo scelto è stato quello di usare un div nascosto nella pagina identificabile con id "data" e usare i suoi attributi data-*. Questo speciale tipo di attributi HTML permettono di definire degli attributi custom ad ogni tag e assegnare loro dei valori. Questi attributi sono poi accessibili da JavaScript dal DOM.

Questo metodo ovviamente non permette lo scambio di dati tra backend e frontend in modo sicuro, perchè chiunque può usare gli strumenti di sviluppatore del browser e guardare il contenuto di questo tag nascosto.

4.2.10 Validazione frontend campi di input ed errori

In questa sezione vengono descritti in dettaglio i comportamenti della pagina a fronte di input non corretti dell'utente o risposte negative dal server durante il caricamento.

Presa visione del regolamento e della privacy policy

La pagina una volta inizializzata presenta il breve testo introduttivo, una descrizione sui file che è possibile caricare, una checkbox per dare conferma di aver preso visione di regolamento e privacy policy del concorso (consultabili tramite link) e un bottone con scritto "Seleziona i tuoi file" da premere per continuare nella pagina. Se l'utente preme questo bottone senza aver premuto la checkbox la label di quest'ultima viene evidenziata in rosso e la pagina non subisce modifiche. Se invece preme prima la checkbox e dopo il bottone, alla pagina viene aggiunto in basso il primo form da compilare.

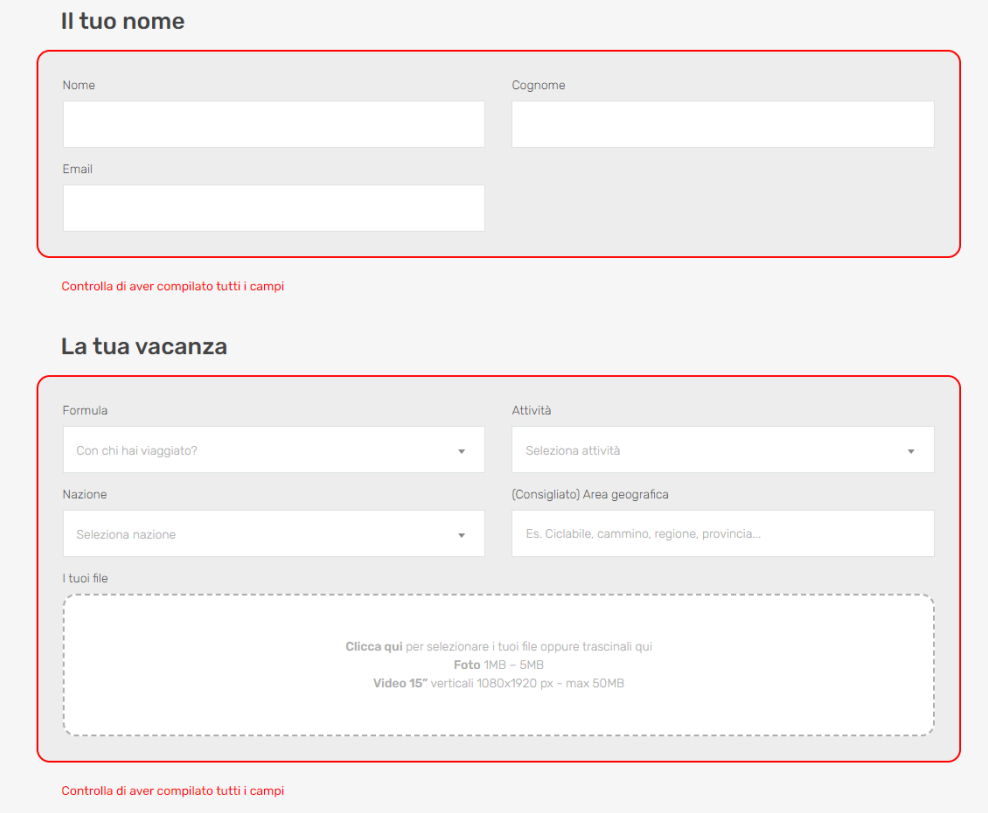


Figura 4.6: Messaggio di errore nella pagina per non clienti

Validazione form

Per inviare il form l'utente deve premere un bottone con scritto "Carica adesso" posizionato alla fine di tale form. Premere questo bottone fa partire una funzione che controlla che tutti i campi siano stati compilati e che ci sia almeno un file selezionato nella Dropzone. Se qualche campo non è stato compilato, il form viene evidenziato in rosso, un messaggio con scritto "Controlla di aver compilato tutti i campi" appare sotto al form e la pagina (se necessario) scrolla verso il primo campo non compilato che trova. Non appena tutti i campi sono stati compilati, l'evidenziatura rossa e il messaggio scompaiono, indicando che tutti i problemi sono stati risolti ed è possibile inviare il form. Nel caso del non cliente, viene considerato come form a sè lo spazio dedicato all'inserimento dei dati personali. Se qualcosa non è valido lì verrà evidenziato in rosso tale spazio.

Errore del server

Nello sfortunato caso in cui qualcosa dovesse andare storto durante il caricamento, in sovrimpressione sulla pagina apparirà un modal di notifica (figura 4.7). Questo ha lo scopo di allertare l'utente che c'è stato un errore non per colpa sua e invitarlo a riprovare in un secondo momento.

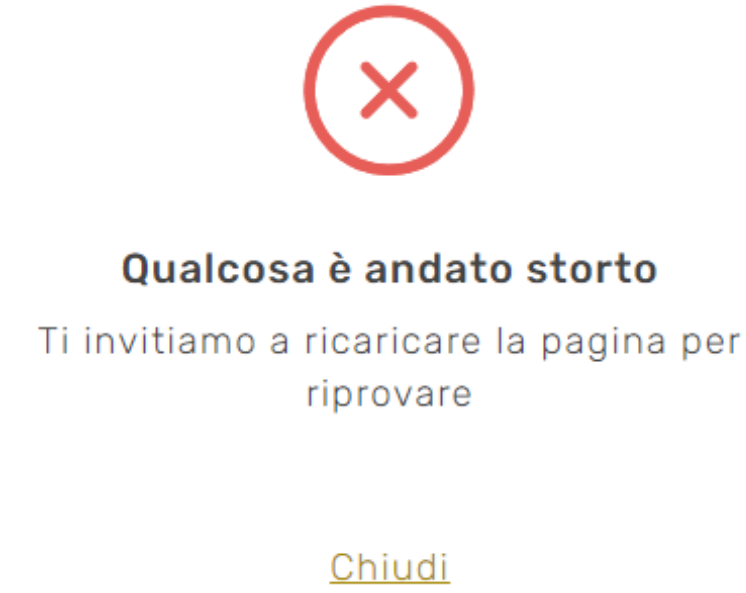


Figura 4.7: Modal di notifica in caso il server rispondesse con un errore

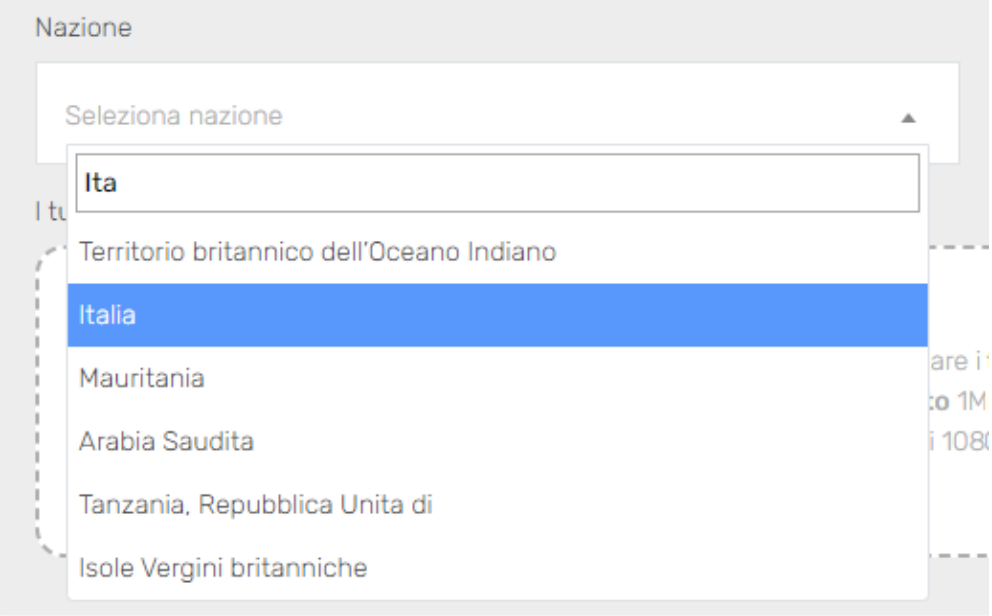


Figura 4.8: Menù a tendina con il campo di ricerca creato dalla libreria Select2

4.2.11 Campo di ricerca per il menù a tendina delle nazioni

Data la grandezza della lista di tutte le nazioni esistenti al mondo, si è voluto dare un modo per poter cercare al suo interno la nazione desiderata. Questo è già possibile di default sui dispositivi desktop (infatti basta selezionare il menù a tendina e iniziare a digitare su tastiera, l'opzione corrispondente verrà evidenziata) ma sui dispositivi mobile questo è impossibile. L'unico modo per offrire questa funzionalità è stato l'uso di una libreria JavaScript, ossia Select2, che con poche righe di codice permette la creazione di menù a tendina con l'opzione di ricerca.

Purtroppo i menù a tendina generati da questa libreria e quelli generati di default dai browser non sono stilisticamente uguali e questo rendeva il menù a tendina per le nazioni e quelli per l'attività e la formula completamente diversi tra loro. Visto che questo problema non era risolvibile semplicemente con regole di stile, si è deciso di utilizzare la libreria Select2 anche per questi due menù (senza opzione di ricerca viste le poche opzioni che entrambi offrono).

Sito web di Select2. URL: https://select2.org/

4.3 Struttura del progetto

Di seguito viene riportata la struttura della directory principale del progetto:

- assets/: cartella che contiene i file JavaScript e i fogli stile scss. Contiene inoltre una serie di file importati dal sito principale che vengono usati per i font e per certi caratteri speciali (utilizzati nel menù e quindi inclusi nel progetto)

- lib/: cartella contenente le varie librerie utilizzate nel progetto importate a mano (Dropzone, Select2)
- templates/: cartella di tutti i template Twig del progetto
- tests/: cartella dei test
- formClientiAJAX.php: file che gestisce le richieste AJAX dalle pagine per clienti
- formContestInit.php: file che inizializza la pagina usando i template Twig
- formNonClientiAJAX.php: file che gestice le richieste AJAX delle pagine per non clienti
- myCloudinary.php: definizione dei parametri e delle funzioni per usare gli strumenti offerti dall'SDK di Cloudinary
- myUtilities.php: definizione di funzioni usate negli altri file PHP

4.3.1 Templates Twig

Tutti i template Twig sono contenuti nella cartella templates del progetto. I file all'interno di questa cartella sono organizzati in questo modo:

- components/
  - macros.twig
- pages/
  - formpage.twig
- partials/
  - form-base.twig
  - html-header.twig
- base.twig

macros.twig

Questo file contiene le macro usate nei vari file Twig. Una macro si definisce tramite un blocco di codice HTML e Twig, che deve iniziare con {% macro nome(params)%} e finire con {% endmacro %}. Come si può intuire, alle macro di Twig è possibile dare dei parametri da usare all'interno della definizione.

Per usare le macro in un file Twig basta importare all'inizio questo file scrivendo {% import 'percorso file' as macros %}. Le macro saranno quindi accessibili dall'oggetto macros.

base.twig

Questo file contiene la struttura principale della pagina, quindi head (che viene preso dal file html-header.twig tramite inclusione), body e footer. Il body contiene una serie di blocchi vuoti, che andranno riempiti tramite estensione nel file formpage.twig

html-header.twig

Questo è un file di default configurato dal plugin Timber che contiene un header configurato per funzionare con Wordpress.

form-base.twig

Qui vengono definiti la checkbox per la presa della visione del regolamento e della privacy policy, il pulsante per continuare nella pagina, il div contenitore di tutti i form che verranno generati e tramite un blocco Twig di selezione ({% if clienti == false %}) si mette nella pagina l'area dedicata all'inserimento dei dati personali del non cliente in caso appunto la pagina sia stata aperta in modalità PCM.

formpage.twig

Questo file va ad estendere base.twig tramite dicitura {% extends 'base.twig' %}. Questo vuol dire che formpage.twig usa base.twig come base, andando però a sostituire tutti i blocchi che hanno lo stesso nome con il proprio contenuto. Per esempio, in base.twig è definito il blocco {% block content %} che però è vuoto. Anche form-base.twig ha un blocco con lo stesso nome, e il suo contenuto andrà a sostituire quello di base.twig per costruire la pagina.

In questo file vengono infatti predisposti il blocco di contenuto principale (che contiene la sezione iniziale di regolamento e informazioni prese da Wordpress, i contenuti del file form-base.twig, il modal per la notifica di errori da parte del server e le varie stringhe WPML usate nei form generati dinamicamente) e altri blocchi che contengono tutti i vari componenti del menù presi da API Girolibero.

4.3.2 Inizializzazione della pagina

L'inizializzazione della pagina è gestita dal file formContestInit.php presente nella directory principale del progetto. Compito di questo file è preparare l'oggetto context, ossia un parametro da dare alla funzione render di Timber, che insieme al template si occupa di renderizzare la pagina nel browser. Questo oggetto è accessibile all'interno dei file twig e può essere usato per inviare informazioni e dati.

Prima di tutto, con la funzione isset si guarda se il parametro id è presente nella variabile globale GET. Se lo è allora vuol dire che chi sta tentando di aprire la pagina è un cliente, e devono essere compiuti dei passaggi ulteriori. Vengono elencati qui di seguito:

- viene decriptato questo id
- si effettua la richiesta API per ottenere i dati del cliente e del viaggio dal preventivo. Il risultato è un JSON con dentro codweb, nome, cognome, email, mese e anno.
- si effettua un'altra richiesta API per ottenere le informazioni del viaggio dal codweb appena ottenuto. Il risultato è un JSON con dentro nazione, titolo del viaggio, da/a, attività, formula e durata
- si costruisce la prima parte del public id (senza nome del file) da associare ai file inviati dal primo form disponibile al cliente

- il public id appena creato viene inserito nell'oggetto context insieme ad altre informazioni utili che servono al template per generare la pagina

Oltre a questi passaggi compiuti nel caso del cliente, in ogni caso viene inserito nell'oggetto context l'attributo booleano cliente, che ha valore true se la pagina è stata aperta da un cliente e false nel caso contrario. Questo attributo serve nei file twig per capire cosa inserire nella pagina.

Vengono inseriti nel context anche i vari componenti del menù e le etichette del menù a tendina delle nazioni (in lingua italiana, ma la funzione che restituisce l'array di nazioni è stata predisposta anche per le lingue inglese e tedesco).

La funzione render di Timber richiede come parametri context e templates. Quest'ultimo indica il template da utilizzare per renderizzare la pagina. Inoltre dalla console di amministratore di Wordpress è possibile associare una pagina ad un template. Questo fa in modo che all'interno dell'attributo post del context venga posto il contenuto scritto su Wordpress (in questo caso l'introduzione e le informazioni sui file che è possibile caricare). Effettuando delle modifiche su Wordpress queste vengono applicate in automatico anche alla pagina.

4.3.3 Logica frontend

Tutte le funzioni elencate di seguito sono contenute all'interno del file myFunctions.js e compongono la logica di frontend per la validazione dei campi input e la gestione dei click dei vari pulsanti della pagina.

documentInit

Funzione che viene eseguita quando il document scatena l'evento ready. Per prima cosa si vanno a recuperare tutti gli attributi data-* del div identificato con id "data". Questi attributi contengono tutte le stringhe WPML per la parte dinamica della pagina più altre informazioni utili, come un valore booleano che indica se la pagina è stata aperta da un cliente o meno. Nel caso in cui la pagina sia stata aperta proprio da un cliente, si va a modificare il titolo della pagina, con un "Ciao -nome-" e un sottotitolo che riprende il titolo del viaggio Girolibero compiuto dal cliente. Per finire viene assegnata al pulsante posto subito dopo la checkbox della presa visione di regolamento e privacy policy la funzione che andrà eseguita quando viene premuto.

newUpload, newUploadPrimary e newUploadSecondary

Queste tre funzioni hanno il compito di generare un nuovo form di upload. Nello specifico, newUpload è la funzione principale, mentre newUploadPrimary e newUploadSecondary sono delle funzioni wrapper, che preparano in modi diversi la funzione newUpload e vengono assegnate al click di diversi pulsanti: newUploadPrimary viene assegnata esclusivamente al bottone posto dopo la checkbox della presa visione, mentre newUploadSecondary viene assegnata ad ogni pulsante "Carica ancora" posto dopo un form che è stato caricato con successo. newUpload richiede come parametro il tipo di pagina caricata dall'utente, quindi se cliente o PCM. newUploadPrimary chiama la funzione newUpload passando il valore booleano ricevuto nella funzione di inizializzazione ricavato dagli attributi data-*. newUploadSecondary chiama la funzione newUpload

```
window.onbeforeunload = function() { // alert se chiude la pagina durante il caricamento
    return '';
}
```

Figura 4.9: Override dell'evento onbeforeunload

sempre in modalità PCM.

newUpload come prima cosa va a prendere il div identificato dall'id "formContainer" che è il div che conterrà tutti i form generati dalla pagina. Ricava quindi il numero di figli che questo contenitore ha (partendo ovviamente da 0) e lo usa come indice per andare a generare i vari componenti del form, ognuno con il proprio id univoco all'interno della pagina. Per esempio, il primo form avrà id "form0", la Dropzone al suo interno "myDropzone0" e così via per tutti gli altri componenti, compresi i campi di input e il pulsante per inviare il form. Per ultima cosa chiama la funzione setUpDropzone, che verrà spiegata nella prossima sezione.

setUpDropzone

setUpDropzone usa la libreria Dropzone.js per rendere funzionante il caricamento dei file. Al div identificato con "myDropzoneN" (N numero progressivo descritto precedentemente) va ad applicare il costruttore di Dropzone, con le opportune configurazioni (che verranno descritte in dettaglio in una prossima sezione).

validateForm

Questa funzione viene chiamata ogni volta che viene premuto il pulsante di caricamento del form. Come si può intuire dal nome, ha il compito di assicurare che ogni campo del form da inviare sia valido. Questo viene fatto attraverso altre funzioni che non verranno descritte in questo documento. Se tutto è ok dà il via a Dropzone che incomincerà l'upload dei file. Inoltre toglie il pulsante di caricamento del form e al suo posto mette un messaggio che dice all'utente di attendere la fine del caricamento. Se per qualche motivo l'utente dovesse chiudere la pagina, il browser lancerebbe un alert di default che impedisce la chiusura immediata della pagina. Questo è possibile grazie a questa istruzione (figura 4.1). Quello che fa è andare a definire un comportamento all'evento onbeforeunload (scatenato ovviamente quando l'utente tenta di chiudere la pagina) tramite una funzione che non ritorna niente. Facendo questo il browser attiva il suo comportamento predefinito, ossia l'alert di default appena descritto. Purtroppo non è possibile modificare il messaggio di questo alert, ma è stata la soluzione preferibile piuttosto che non avere nessun ostacolo alla chiusura della pagina mentre si effettua un caricamento.

4.3.4 Dropzone e AJAX

Per configurare correttamente una Dropzone, bisogna creare un oggetto Dropzone tramite il costruttore dedicato. Il costruttore ha due parametri: l'id del contenitore della Dropzone e un oggetto di configurazione. Questo oggetto di configurazione presenta molte opzioni, documentate nella pagina Github della libreria[2]. Dropzone

[2] *Pagina Github di Dropzone.js con tutte le opzioni.* URL: https://github.com/dropzone/dropzone/blob/main/src/options.js.

gestisce in automatico l'invio dei file (uno alla volta) tramite AJAX, quindi queste opzioni servono anche per gestire gli eventi del caricamento e la gestione della risposta. Di seguito vengono descritte le opzioni configurate per le Dropzone del sito:

- url: l'action AJAX (ossia il percorso del file php che gestisce la richiesta)
- autoProcessQueue: booleano che indica se Dropzone deve inviare i file non appena vengono selezionati o trascinati dentro all'area di caricamento. Settato a false
- paramName: il nome con il quale viene identificato il file all'interno della variabile globale FILES di PHP
- addRemoveLinks: booleano che indica se aggiungere per ogni file un pulsante per rimuovere tale file dall'area di caricamento. Settato a true
- dictDefaultMessage: testo da visualizzare all'interno dell'area di caricamento quando non sono ancora stati selezionati dei file. Impostato tramite un valore ricavato dagli attributi data-*
- dictRemoveFile: testo da visualizzare nel pulsante per rimuovere i file dall'area di caricamento. È stato possibile inserire un tag svg così da visualizzare un'icona che rappresenta una X

Oltre a queste opzioni, è possibile definire una funzione chiamata init che permette al suo interno la gestione di vari eventi. Sono elencati di seguito quelli gestiti e come sono stati gestiti:

- addedfiles: evento che parte quando viene aggiunto un file nell'area di caricamento. Qui viene effettuata la validazione frontend dei file. Se un file non è valido questo non viene aggiunto all'area di caricamento e l'utente ne è notificato tramite un messaggio.
- sending: evento che parte prima di inviare un file tramite richiesta AJAX in modalità POST. Ai file vengono aggiunti i dati personali e le informazioni del viaggio nel caso del non cliente, oppure i dati personali del cliente ricavati dal preventivo e il public id parziale preparato in formContestInit.php.
- success: evento che parte quando un file viene caricato correttamente (il server risponde senza errori). Gestito a scopo di logging
- error: evento che parte quando un file non viene caricato (il server risponde con un errore). Gestito a scopo di logging
- complete: evento che parte quando un file ha finito (sia success o error). Se tutti i file sono stati caricati con successo allora questa funzione disabilita il form corrente (per impedire interazioni dell'utente) e mostra il pulsante "Chiudi" che reindirizza a girolibero.it e il pulsante "Carica ancora" che apre un nuovo form

4.3.5 Cloudinary, DB, Newsletter

Le richieste AJAX inviate da Dropzone vengono gestite dai file formClientiAJAX.php e formNonClientiAJAX.php.
Per prima cosa vengono presi i dati dalla variabile globale POST, che sono per il cliente:

- il public id parziale (manca il nome del file)
- i tags del file
- il nome del cliente
- il cognome del cliente
- l'email del cliente

Nome, cognome e email vengono controllati e se non sono validi il server ritornerà errore. Nome e cognome vengono controllati tramite espressione regolare, mentre l'email viene controllata usando la funzione PHP is_email.

Per il non cliente vengono invece presi i dati:

- nome
- cognome
- email
- nazione
- area
- attività
- formula

Una volta validati, questi dati vengono utilizzati per comporre il public id.
In entrambi i casi viene preso poi il file da caricare dalla variabile globale FILES. Di questo file si controllano estensione e peso (anche se è superfluo visto che sono stati già controllati a frontend). Se per qualche motivo il file non dovesse essere valido il server ritornerà errore.

Si procede quindi con l'upload del file su Cloudinary e l'inserimento dell'informazione di caricamento nel log. La stessa funzione che si occupa del caricamento nel database si occupa anche dell'invio all'API Girolibero della richiesta per iscrivere l'utente alla newsletter. Questo però viene fatto solamente quando l'email non viene trovata all'interno del log, questo per effettuare l'iscrizione una volta sola.

La risposta di Cloudinary e l'esito del database vengono salvati in un oggetto con due rispettivi attributi. Questo oggetto chiamato response viene codificato in formato JSON e inviato come risposta alla Dropzone che ha inviato il file.

4.3.6 Stile

Tutti i file inerenti allo stile sono contenuti in assests/scss. Il file principale è style.scss, che non contiene regole di stile ma tutti gli import degli altri file scss. Questo infatti è il file di partenza per l'operazione di compilazione, che produce come risultato il file main.css che il browser utilizzerà per renderizzare la pagina nel modo corretto.

Visto che si voleva ottenere un di discreto livello di congruenza tra il sito girolibero.it e la pagina contest.girolibero.it, la quasi totalità dei file presenti in questa cartella

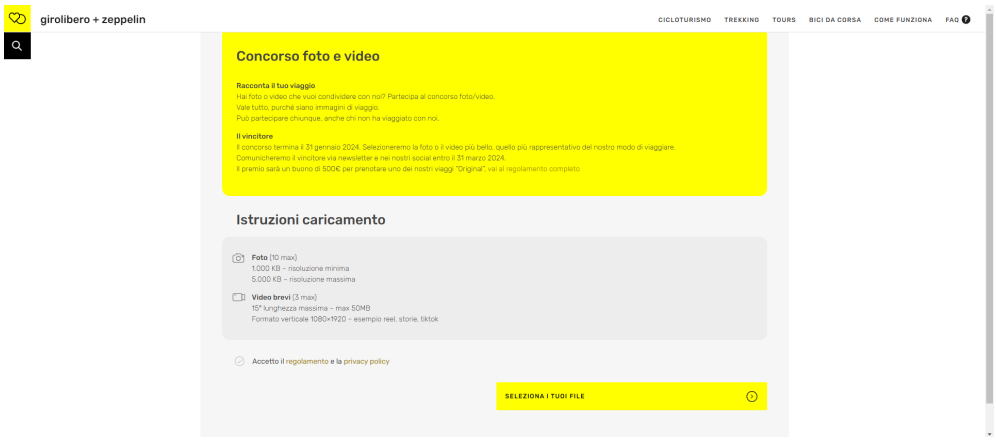


Figura 4.10: Schermata principale appena la pagina viene aperta

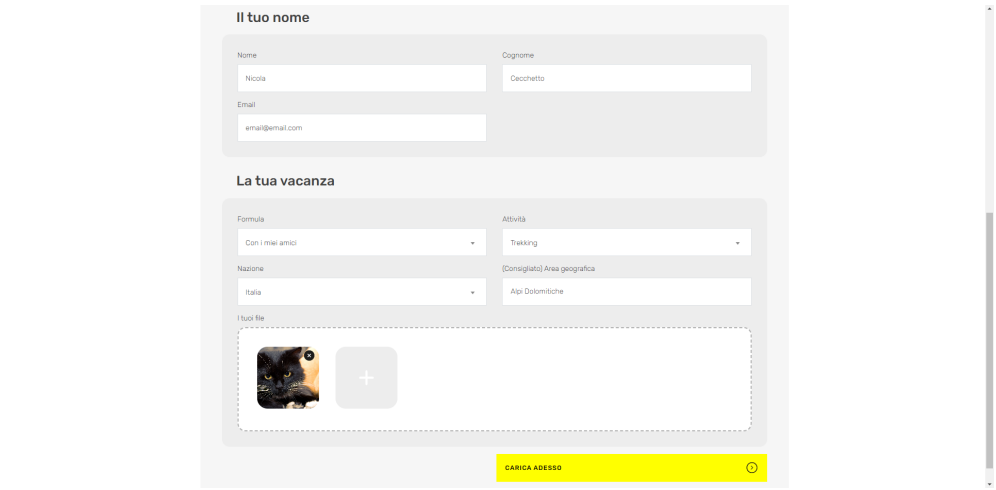


Figura 4.11: Form di caricamento per non clienti

sono gli stessi usati per generare lo stile del sito principale. Questo include variabili, tipografia, caratteri speciali e mixins. L'unico file creato da zero è il file contest.scss, che ovviamente contiene regole specifiche per la pagina.

Il design della pagina è stato realizzato in cooperazione con l'UX designer Caterina Romio grazie al software Figma, facendo riferimento allo stile del preventivatore online di viaggi del sito girolibero.it.

La pagina è responsive, pensata sia per utilizzo da desktop che da mobile. L'unico dettaglio degno di nota è che sopra i 992px i campi di inserimento delle informazioni personali e dei dati del viaggio sono disposti su due colonne, mentre al di sotto di questa soglia sono posti uno sotto l'altro.

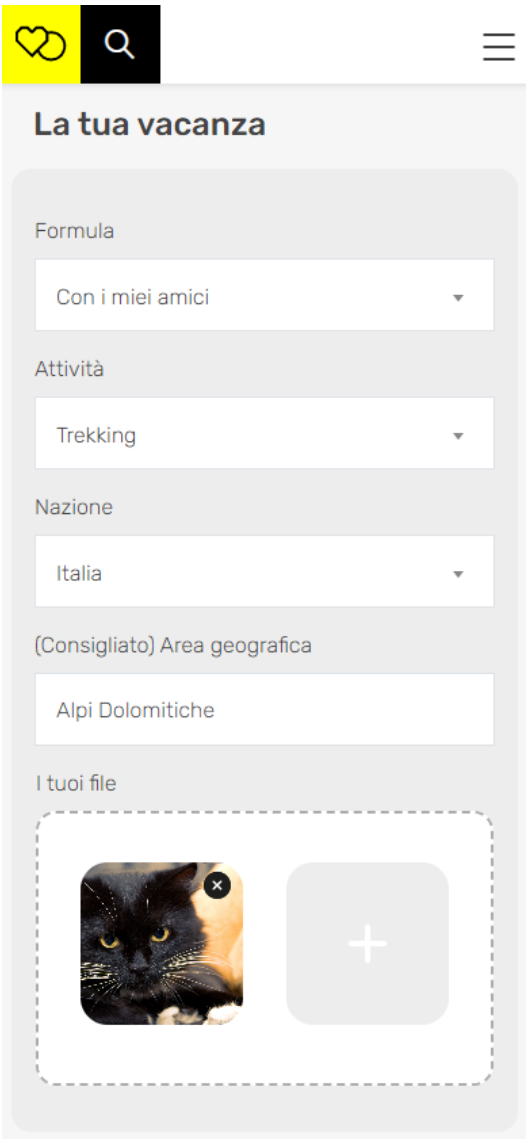


Figura 4.12: Visualizzazione mobile del form

Capitolo 5

Verifica, validazione e accessibilità

5.1 Test di unità (Backend)

I test di unità sul backend PHP sono stati eseguiti usando PHPUnit. PHPUnit è un framework di testing di unità per PHP ed è usato per verificare il corretto funzionamento di funzioni, metodi di classe e classi stesse.

Sito web di PHPUnit. URL: https://phpunit.de/

I test sono stati eseguiti sulle funzioni contenute nel file myUtilities.php, che raccoglie appunto tutte le varie funzioni di utilità usate nel backend. I test devono essere definiti tramite metodi di una classe (nel caso del progetto la classe myUtilitiesTest) che deve estendere la classe TestCase. Ognuno di questi metodi deve essere definito come una funzione pubblica e che ritorna il tipo void. La nomenclatura dei test deve essere descrittiva, iniziare con la parola "test" e utilizzare il camel case.

Tutti i metodi di test sono stati scritti secondo il pattern Arrange Act Assert. Questo pattern si basa su tre passaggi, uno per ogni parola che compone il nome del pattern:

- Arrange: si preparano le variabili e tutto ciò che può servire al test
- Act: si esegue il metodo target, che deve ritornare un risultato
- Assert: si confrontano risultato atteso con risultato ottenuto

Ciascun test di unità è identificato da un codice univoco, creato con la seguente notazione:

TUX

dove:

- TU: Test d'unità
- X: numero progressivo

Tabella 5.1: Tabella del tracciamento dei test di unità

Test	Nome	Descrizione
TU1	Id Preventivo Can Be Decrypted	Verifica della corretta decriptazione di un id preventivo
TU2	DecryptId Returns Empty String On Empty String	Verifica del corretto comportamento della funzione decryptId se si prova a decriptare una stringa vuota
TU3	GetApi Throws Value Error Exception On Empty Url	Verifica del corretto comportamento della funzione getApi se si prova ad inserire un Url vuoto
TU4	GetApi Returns Null If Url Is Wrong	Verifica del corretto comportamento della funzione getApi se si passa un Url sbagliato
TU5	GetMese returns Gennaio on Input 1	Verifica che la funzione getMese ritorni "Gennaio" dato il parametro "1"
TU6	GetMese returns Dicembre on Input 12	Verifica che la funzione getMese ritorni "Dicembre" dato il parametro "12"
TU7	GetMese Returns Null On Input 0	Verifica che la funzione getMese ritorni null dato il parametro "0"
TU8	GetMese returns Null On Input 13	Verifica che la funzione getMese ritorni null dato il parametro "13"
TU9	Pulisci Works	Verifica del corretto comportamento della funzione Pulisci
TU10	Public Id Struttura Lineare	Verifica del corretto comportamento della funzione getPublicIdCliente quando presentata con un viaggio che ha struttura lineare
TU11	Public Id Struttura Non Lineare	Verifica del corretto comportamento della funzione getPublicIdCliente quando presentata con un viaggio che ha struttura non lineare
TU12	Public Id Cicloturismo	Verifica del corretto comportamento della funzione getPublicIdCliente quando presentata con un viaggio di cicloturismo
TU13	Public Id Famiglie	Verifica del corretto comportamento della funzione getPublicIdCliente quando presentata con un viaggio per famiglie
TU14	Valida File Filetype Not Supported	Verifica del corretto comportamento della funzione validaFile quando presentato con un file con tipo non valido
TU15	Valida File Image Too Big	Verifica del corretto comportamento della funzione validaFile quando presentato con un'immagine troppo grande
TU16	Valida File Image Too Small	Verifica del corretto comportamento della funzione validaFile quando presentato con un'immagine troppo piccola
TU17	Valida File Video Too Big	Verifica del corretto comportamento della funzione validaFile quando presentato con un video troppo grande
TU18	Get Country List Wrong Country	Verifica del corretto comportamento della funzione getCountryList quando si richiede una lingua che non è supportata

Test	Nome	Descrizione
TU19	Preventivo Can't Be Null	Verifica del corretto comportamento della funzione checkValidPreventivo se si passa come parametro null
TU20	Preventivo Can't Have Letters	Verifica del corretto comportamento della funzione checkValidPreventivo se si passa una stringa con lettere al suo interno
TU21	Preventivo Only Numbers	Verifica del corretto comportamento della funzione checkValidPreventivo se si passa una stringa con solo numeri
TU22	Nome Can't Be Null	Verifica del corretto comportamento della funzione checkValidNome se si passa una stringa vuota
TU23	Nome Can't Have Numbers	Verifica del corretto comportamento della funzione checkValidNome se si passa una stringa contenente numeri
TU24	Nome Only Letters	Verifica del corretto comportamento della funzione checkValidNome se si passa una stringa contenente solo lettere
TU25	PCM Public Id	Verifica del corretto comportamento della funzione getPublicIdPCM
TU26	Get File Extension	Verifica del corretto comportamento della funzione getFileExtension
TU27	Get File Name	Verifica del corretto comportamento della funzione getFileName

5.2 Test dell'interfaccia (Frontend)

Il team di sviluppo di Girolibero si avvale del personale marketing per verificare il corretto funzionamento delle pagine. Alcuni comportamenti sono testabili sono in questo modo, ma in alcuni casi un sistema di test automatico risparmierebbe molto tempo. Per questo una volta iniziato ad affrontare il problema dei test dell'interfaccia il team di sviluppo ha invitato allo studio e all'uso di un framework per l'automatizzazione di questi test. Il progetto è servito come banco di prova per capire se questa tecnologia potesse essere utile al team di sviluppo.

Il framework in questione è Robot Framework, che grazie alla libreria SeleniumLibrary è in grado di eseguire test automatizzati per verificare il comportamento di applicazioni web.
In ogni file .robot è possibile definire:

- Setting: configurazioni iniziali del file. In questa sezione viene inclusa la libreria SeleniumLibrary
- Variables: definizioni di variabili da poter usare nei test cases
- Keywords: blocchi di azioni da compiere in successione. I blocchi possono essere richiamati nei test cases solamente attraverso il loro nome per eseguire l'intero blocco
- Test Cases: definizione dei test cases

Ogni test case è composto da una frase descrittiva e da una serie di azioni che vengono eseguite in successione. Le azioni che è possibile eseguire sono molteplici e sono raccolte nella documentazione ufficiale della libreria. Di seguito ne vengono elencate e spiegate alcune usate nel progetto:

- Open Browser: dato un url e un browser (Chrome, Firefox, etc...) apre la pagina indicata nel browser scelto
- Click Element: clicca un elemento della pagina tramite un localizzatore
- Wait Until Element Is Visible: blocca l'esecuzione del test fino a che l'elemento definito dal localizzatore non è visibile
- Wait For Condition: aspetta fino a che una riga di codice JavaScript non ritorna vero
- Input Text: dato un localizzatore di un input di testo e una stringa, scrive la stringa all'interno dell'input
- Select From List By Value: dato un localizzatore di un menù a tendina e il testo di un'opzione, seleziona quell'opzione
- Choose File: dato un localizzatore di un input di file e il percorso di un file, carica quel file
- Location Should Be: controlla che l'url attuale sia quello fornito

Sito web di Robot Framework. URL: https://robotframework.org/

Documentazione della libreria SeleniumLibrary per Robot Framework. URL: https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html

Sono stati creati tre diversi file di testing:

- clienteBasic.robot, per testare upload primario di un cliente e secondario
- nonClienteBasic.robot, per testare errori di validazione e upload primario di un non cliente
- testMenu.robot, per testare il menù

Di seguito vengono descritti elencati i test di interfaccia eseguiti, ognuno identificato con un codice creato con la seguente notazione:

TIX

dove:

- TU: Test d'interfaccia
- X: numero progressivo

clienteBasic.robot

Tabella 5.2: Tabella del tracciamento dei test d'interfaccia del file clienteBasic.robot

Test	Nome	Descrizione
TI1	Clicking "Carica adesso" without completing "La tua vacanza" is not allowed and shows an error	Verifica del corretto funzionamento della validazione del form per clienti
TI2	Inserting a valid image will show the image in the Dropzone	Verifica il corretto funzionamento del form primario per clienti
TI3	Clicking "Seleziona i tuoi file" uploads the files and after some time shows "Grazie per aver partecipato"	Verifica del corretto comportamento dopo aver inviato il form principale per clienti
TI4	Clicking "Nuova vacanza" opens second form	Verifica del corretto comportamento del pulsante "Nuova vacanza"
TI5	Upload File in second form	Verifica del corretto comportamento del form secondario
TI6	Clicking "Chiudi" will redirect to girolibero.it	Verifica del corretto comportamento del pulsante "Chiudi"

nonClienteBasic.robot

Tabella 5.3: Tabella del tracciamento dei test d'interfaccia del file nonClienteBasic.robot

Test	Nome	Descrizione
TI7	Clicking "Seleziona i tuoi file" without accepting privacy shows an error	Verifica del corretto funzionamento della validazione della checkbox per la presa visione del regolamento e della privacy policy
TI8	Clicking "Seleziona i tuoi file" after accepting privacy is allowed and shows the form	Verifica del corretto funzionamento della validazione della checkbox per la presa visione del regolamento e della privacy policy
TI9	Clicking "Carica adesso" without completing neither "Il tuo nome" and "La tua vacanza" is not allowed and shows two errors	Verifica del corretto funzionamento della validazione del form
TI10	Filling "Il tuo nome" turns the error off for this form	Verifica del corretto funzionamento della validazione dei campi "Il tuo nome"
TI11	Filling "La tua vacanza" turns the error off for this form	Verifica del corretto funzionamento della validazione dei campi "La tua vacanza"
TI12	Clicking "Seleziona i tuoi file" uploads the files and after some time shows "Grazie per aver partecipato"	Verifica del corretto comportamento dopo aver inviato il form principale per non clienti

testMenu.robot

Tabella 5.4: Tabella del tracciamento dei test d'interfaccia del file testMenu.robot

Test	Nome	Descrizione
TI13	Website logo redirects to girolibero.it	Verifica del corretto funzionamento quando si clicca sul logo
TI14	In search overlay the search functionality works	Verifica del corretto funzionamento della funzionalità cerca
TI15	Cicloturismo overlay works	Verifica del corretto funzionamento dell'overlay Cicloturismo
TI16	Trekking overlay works	Verifica del corretto funzionamento dell'overlay Trekking
TI17	Tours overlay works	Verifica del corretto funzionamento dell'overlay Tours
TI18	Bici Da Corsa overlay works	Verifica del corretto funzionamento dell'overlay Bici Da Corsa
TI19	Come Funziona overlay works	Verifica del corretto funzionamento dell'overlay Come Funziona
TI20	Faq search works	Verifica del corretto funzionamento della funzionalità Faq

5.3 Validazione dei requisiti

La copertura dei requisiti è riassunta in questa tabella:

Tabella 5.5: Tabella della copertura dei requisiti

Requisito	Soddisfatto
RF1	Sì
RF2	Sì
RF2.1	Sì
RF2.1.1	Sì
RF2.1.2	Sì
RF2.1.3	Sì
RF2.2	Sì
RF2.2.1	Sì
RF2.2.2	Sì
RF3	Sì
RF3.1	Sì
RF3.2	Sì
RF4	Sì
RF4.1	Sì
RF4.2	Sì
RF5	Sì
RF5.1	Sì
RF5.1.1	Sì
RF5.1.2	Sì
RF5.1.3	Sì
RF5.2	Sì

Requisito	Soddisfatto
RF5.2.3	Sì
RF5.2.4	Sì
RF5.3	Sì
RF5.4	Sì
RF6	Sì
RF7	Sì
RF7.1	Sì
RF7.1.1	Sì
RF7.1.2	Sì
RF7.1.3	Sì
RF7.1.4	Sì
RF7.2	Sì
RF7.3	Sì
RF8	Sì
RF9	Sì
RF10	Sì
RFE.1	Sì
RFE.2	Sì
RFE.3	Sì
RFE.4	Sì
RFE.5	Sì
RQ1	Sì
RQ2	Sì
RQ3	Sì
RQ3.1	Sì
RQ3.2	Sì
RQ3.3	Sì
RQ4	Sì
RQ5	Sì
RV1	Sì
RV2	Sì
RV3	Sì
RV4	Sì
RV5	Sì
RV6	Sì
RV7	Sì

5.4 Accessibilità

Quando si parla di accessibilità nel contesto di un sito web si intende la sua capacità di offrire informazioni o servizi a quante più categorie di utenti possibili. Questo può includere utenti che soffrono di disabilità (uditiva, cognitiva, neurologica, fisica, visiva) come anche utenti che non soffrono di disabilità (schermo di grandezza diversa, modalità di input diverse, limitazioni situazionali dovute all'ambiente circostante, connessione lenta)[1].

[1] *Introduction to Web Accessibility*. URL: https://www.w3.org/WAI/fundamentals/accessibility-intro/.

Negli anni l'accessibilità è divenuto un tema di grande importanza, anche grazie alla divulgazione, ai concorsi e alle legislazioni che istituzioni, associazioni ed università hanno promosso.

Sebbene l'accessibilità della pagina non fosse un requisito posto dall'azienda, degli accorgimenti possono essere fatti da chi progetta/sviluppa un sito senza dover fare troppo lavoro in più. Gli accorgimenti nel caso del progetto riguardano scelte di design (prese col consenso dell'UX designer) come disposizione degli elementi della pagina e stile di alcuni elementi grafici.

Detto questo, la pagina rimane migliorabile sotto molti punti di vista, e questa sezione è adibita all'analisi della pagina per un eventuale futuro aggiornamento.

5.4.1 Link, bottoni e altre interazioni

La pagina presenta dei link che rimandano al regolamento e alla privacy policy del concorso fotografico. Questi link (che aprono una nuova scheda nel browser) sono distinguibili dal testo normale solo tramite un diverso colore. Questa (come altre descritte in questa sezione) è una scelta di design fatta per rispecchiare quanto più possibile il design del sito girolibero.it.

Per i link, è buona prassi:

- utilizzare un colore univoco
- sottolineare il link
- utilizzare un'altro colore per far capire all'utente che ha già visitato quella pagina

I bottoni della pagina sono grandi e gialli, e questo li rende ben visibili. Questi bottoni servono per procedere nella pagina, e quindi è giusto che catturino l'attenzione dell'utente e che siano di facile utilizzo.

L'unica checkbox del sito (quella della presa visione del regolamento e della privacy policy) purtroppo è la nota dolente della pagina, in quanto non è un elemento accessibile tramite tastiera. Questo perchè non è una vera checkbox ma un tag svg e una label. Questo costrutto viene da girolibero.it e non è possibile rimpiazzarlo senza modificare il design della pagina (e quindi distaccarsi da quello del sito originale). Il mio consiglio al team di sviluppo è di rimpiazzarlo completamente a favore di una soluzione più accessibile.

5.4.2 Form

Anche per il form ho fatto riferimento al design del sito originale. I campi di input sono grandi e attirano l'attenzione. Inoltre quando sono selezionati vengono contornati di colore giallo.

L'unica parte di design che è originale è l'area di caricamento dei file Dropzone. Questa zona riempie in larghezza tutta la pagina in modo da essere ben evidente. È inoltre

contornata da una linea tratteggiata, che è come di solito si segnala un'area dove è possibile trascinare file.

In caso di input non validi, i bordi del form si colorano di rosso e un testo compare sotto di esso, in modo da rendere evidente che è richiesto un cambiamento prima di procedere con l'invio.
L'aspetto che è possibile migliorare è il contrasto dei suggerimenti all'interno dei campi del form, come verrà descritto nella prossima sezione.

5.4.3 Colori

I colori del sito sono questi:

- #FFFFFF (bianco) per lo sfondo della pagina e per i campi di input dei form
- #444444 (grigio scuro) per i testi
- #FFFF00 (giallo) per lo sfondo dell'introduzione e i bottoni
- #a68703 (oro) per i link
- #F6F6F6 (grigio chiaro) per lo sfondo principale
- #EDEDED (grigio) per lo sfondo dei blocchi
- #ADADAD (grigio) per i suggerimenti all'interno dei campi del form
- #FF0000 (rosso) per i testi di avvertimento del form e per il contorno dei blocchi in caso di errore

Tutti questi colori e i loro usi derivano dal design di girolibero.it. Di seguito vengono analizzati i contrasti:

- testo grigio scuro e sfondo giallo: 9.07:1 (passa WCAG AA e WCAG AAA)
- link oro e sfondo giallo: 3.21:1 (non passa WCAG AA)
- testo grigio scuro e sfondo grigio chiaro: 9.01:1 (passa WCAG AA e WCAG AAA)
- link oro e sfondo grigio chiaro: 3.19:1 (non passa WCAG AA)
- testo grigio scuro e sfondo grigio: 8.31:1 (passa WCAG AA e WCAG AAA)
- testo rosso e sfondo grigio chiaro: 3.69:1 (non passa WCAG AA)
- testo grigio dei suggerimenti e sfondo bianco: 2.24:1 (non passa WCAG AA)

I consigli che posso dare per ottenere risultati migliori nei test di contrasto è cambiare il colore dei link e degli avvertimenti. Per i link conviene utilizzare il colore #876E0D che è un oro leggermente più scuro, che permette di avere un contrasto di 4.58:1 con uno sfondo giallo e di 4.55:1 con uno sfondo grigio chiaro (entrambi validi per WCAG AA). Per gli avvertimenti basta usare una tonalità di rosso più scura come #E00000 per avere un contrasto con lo sfondo grigio chiaro di 4.66:1 (valido per WCAG AA).

Capitolo 6

Conclusioni

6.1 Raggiungimento degli obiettivi

Gli obiettivi prefissati all'inizio dell'esperienza di stage sono riassunti nella seguente tabella insieme agli esiti di tali obiettivi:

Tabella 6.1: Tabella del raggiungimento degli obiettivi

Codice	Descrizione	Esito
OB1	Realizzazione del frontend	Obiettivo raggiunto con struttura, stile e comportamento adeguatamente realizzati
OB2	Realizzazione del backend	Obiettivo raggiunto tramite setup di Wordpress, realizzazione dei template Twig e corretta gestione delle richieste AJAX
OB3	Interazione con archivio cloud Cloudinary	Obiettivo raggiunto con la realizzazione della procedura di upload dei file
DE1	Predisposizione del backend per la gestione multilingua dei campi di testo	Obiettivo raggiunto con l'installazione del plugin WPML di Wordpress
OP1	Implementazione tracciamento Google Tag Manager sulla pagina contest.girolibero.it per analisi digital marketing	Obiettivo non raggiunto

L'obiettivo opzionale OP1 era stato proposto in caso il tempo preventivato per il progetto fosse risultato più del dovuto, ma così non è stato.

6.2 Conoscenze acquisite

Il progetto è stato utile per potermi approcciare a nuove tecnologie e conoscenze senza però sovraccaricarmi di cose nuove da imparare perchè non tutto era nuovo per me.

Lo studio della piattaforma Cloudinary mi ha permesso di avere a che fare per la prima volta con un servizio di archiviazione cloud. Anche se in futuro non dovessi usare Cloudinary, la capacità di creare una soluzione ad hoc per poter interagire con un servizio cloud (tramite l'aiuto di documentazione ufficiale) rimarrà utile in futuro per qualunque altro tipo di servizio online.

L'uso di Wordpress è stata la mia prima esperienza con un CMS. Questi strumenti permettono a sviluppatori web e creatori di contenuti di poter lavorare allo stesso sito web in modo però quasi indipendente. L'esperienza è stata veramente illuminante e ha aperto nella mia mente numerose idee e progetti che sono realizzabili con questo tipo di strumenti. Sono sicuro che incontrerò ancora Wordpress nella mia vita lavorativa ma credo di aver acquisito le capacità per poter lavorare anche con altri CMS.

Twig come motore di template per la creazione di pagine HTML è uno strumento per separare il codice PHP dal markup HTML. La sua sintassi non è complicata e semplifica la scrittura delle pagine di un sito. Interagire e lavorare con una sintassi nuova è una sfida che ogni sviluppatore affronta periodicamente a causa di nuovi linguaggi di programmazione o aggiornamenti di vecchi linguaggi. Questa esperienza mi ha dimostrato che un buon programmatore deve essere sempre in grado di imparare cose nuove, tramite l'aiuto di documentazione o esempi.

Sass si è rivelata una utile estensione di CSS. Ho sempre ritenuto quest'ultimo un linguaggio potente ma confusionario in certi aspetti. Sass offre funzionalità aggiuntive che migliorano l'esperienza di creazione di fogli stile per pagine web. Non ho ancora appreso a pieno le potenzialità di questo linguaggio ma immagino che questa non sarà la mia ultima interazione con esso e avrò modo di approfondire.

6.3 Valutazione personale

Questa è stata la mia seconda esperienza di stage nel campo IT. La prima è stata effettuata tra il quarto e il quinto anno di scuola secondaria di secondo grado presso un'azienda che sviluppava software per banche. Comparando queste mie due esperienze lavorative posso dire di essere migliorato in molti aspetti, sia per quanto riguarda le mie conoscenze della materia informatica sia per quanto riguarda le mie competenze trasversali. Inoltre mi ha permesso di vedere come mondi totalmente separati (settore bancario e settore turistico) abbiano a volte problemi completamente opposti e a volte gli stessi esatti problemi. Se al settore bancario interessano più validazione dei dati ed efficienza, per quello turistico contano più comunicazione col cliente e design.Un problema comune invece ai due mondi è la traduzione delle proprie applicazioni o siti web in più lingue. Questo è sicuramente un aspetto su cui uno sviluppatore web che vuole entrare nel mondo del lavoro in Italia o in Europa deve ragionare e studiare attentamente.

Il progetto ha consolidato le cose che ho avuto modo di imparare nei corsi della laurea triennale (tra i più rilevanti citerei Tecnologie Web e Ingegneria del Software) ma mi ha permesso anche di entrare in contatto con concetti e tecnologie con cui non avevo mai interagito prima di questa esperienza. Le conoscenze che già possedevo in campo di sviluppo web si sono rivelate di grande aiuto per progettare ed implementare la

pagina web, ma anche per utilizzare le nuove tecnologie con cui sono entrato in contatto.

Lavorare all'interno di un team già in funzione e con le proprie metodologie di lavoro e comunicazione mi ha permesso di mettere in relazione ciò che ho imparato con quello che accade in un contesto reale e anche di poter applicare quanto studiato. A posteriori della mia carriera universitaria avrei preferito spendere più tempo per approfondire i software di versionamento del codice, visto che sono degli strumenti ampiamente utilizzati in ogni realtà IT, dalle più grandi alle più piccole.

In conclusione, l'esperienza vissuta all'interno del team di sviluppo Girolibero è stata notevolemente positiva ed ha consolidato la mia voglia di intraprendere un percorso nel mondo dello sviluppo web.

Acronimi e abbreviazioni

AJAX Asynchronous JavaScript and XML. 35, 51, 68

API Application Program Interface. 5, 8, 14, 16, 39–42, 48, 52, 68

CMS Content Management System. 3, 30, 32, 34, 68

DAM Digital Asset Management. 1, 31, 68

DBMS Database Management System. 8, 68

DOM Document Object Model. 35, 40, 43, 68

GPLv2 GNU General Public License v2. 32, 69

IT Information Technology. 1, 5, 65, 66, 69

PCM Per Conto Mio. 2, 8, 37, 43, 48–50, 69

SDK Software Development Kit. 39, 47, 69

UX User Experience. 3, 29, 53, 62, 69

Glossario

AJAX tecniche di sviluppo web utilizzate per la creazione di siti web asincroni, ossia che hanno la capacità di mandare e ricevere dati senza il bisogno di bloccare l'intera pagina web. 67

API in informatica con il termine *Application Programming Interface API* (ing. interfaccia di programmazione di un'applicazione) si indica ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per l'espletamento di un determinato compito all'interno di un certo programma. La finalità è ottenere un'astrazione, di solito tra l'hardware e il programmatore o tra software a basso e quello ad alto livello semplificando così il lavoro di programmazione. 67

Attività l'attività svolta durante un viaggio Girolibero (in bici, in bici e barca, trekking, tour). 36, 37, 41, 46, 48, 52

Backend in un sito web, tutta la parte di logica che risiede sul server. 4, 5, 25, 29, 34, 39, 43, 55, 64

Cloud server a cui si accede tramite internet più software e database che vengono eseguiti su tali server. 4, 31, 64

CMS è uno strumento software, installato su un server web, il cui compito è facilitare la gestione dei contenuti di siti web. 67

Codweb il codice alfanumerico di 5 cifre con il quale viene identificato un viaggio Girolibero. 36, 41, 48

DAM un *Digital Asset Management* è un sistema software centralizzato usato dalle aziende per immagazzinare, gestire, trovare e condividere file multimediali e documenti. 67

DBMS sistema software per la creazione, la manipolazione e l'interrogazione efficiente di database e ospitato su architettura hardware dedicata oppure su semplice computer. 67

DOM Interfaccia di programmazione per pagine web. Rappresenta una pagina tramite dei nodi e oggetti, in modo che un programma possa interagire con la pagina e modificare struttura, contenuti e stile. 67

Form in una pagina web, un *form* permette ad un utente di inserire dati per essere mandati ad un server che li andrà a processare. 2, 14, 16, 21, 42–44, 48–51, 59, 62, 63

Formula l'organizzazione di un viaggio Girolibero (in gruppo, indipendente, per famiglie). 36, 37, 41, 46, 48, 52

Frontend tutto quello che in un sito web ha a che fare con l'interfaccia con la quale l'utente interagisce. 4, 5, 30, 32, 43, 49, 51, 52, 64

GPLv2 licenza per software libero stesa originariamente nel 1989 da Richard Stallman. Qualsiasi software sotto questa licenza deve rimanere libera anche nelle sue versioni successive. 67

IT con *Information Technology* si indica un insieme di materie che comprende sistemi informatici, software, linguaggi di programmazione e di archiviazione e manipolazione di dati e informazioni. 67

Log file dove un software registra eventi legati a determinate operazioni. 2, 8, 14, 16, 21, 40, 52

Newsletter aggiornamento informativo periodico che un'azienda, un ente (pubblico o privato), un'associazione o un gruppo di lavoro, invia a un determinato target, come utenti, clienti o membri, aggiornandoli sulle proprie attività. 2, 8, 14, 16, 41, 52

Open source si indica un software distribuito sotto i termini di una licenza open source, che ne concede lo studio, l'utilizzo, la modifica e la redistribuzione. 32, 40

PCM nel contesto del concorso fotografico, un viaggiatore *Per Conto Mio* è un utente della pagina web che carica foto e video di un viaggio non Girolibero e quindi indipendente. 67

Public Id stringa univoca che serve per identificare un file all'interno di Cloudinary. 31, 36, 39, 41, 48, 49, 51, 52

SDK un insieme di strumenti per lo sviluppo e la documentazione di software. 67

UX come un utente interagisce con un sistema software. 67

Viewport Regione visibile della pagina web sullo schermo di un desktop o mobile. Buona prassi mettere in questa parte della pagina web le informazioni più importanti e le interazioni principali. 10, 26

Bibliografia

Siti web consultati

Documentazione della libreria SeleniumLibrary per Robot Framework. URL: https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html (cit. a p. 58).

Documentazione di CSS. URL: https://developer.mozilla.org/en-US/docs/Web/CSS (cit. a p. 36).

Documentazione di JavaScript. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript (cit. a p. 35).

Documentazione HTML Drag and Drop API. URL: https://developer.mozilla.org/en-US/docs/Web/API/HTML_Drag_and_Drop_API (cit. a p. 40).

Documentazione SDK Cloudinary per PHP. URL: https://cloudinary.com/documentation/php_integration (cit. a p. 39).

Introduction to Web Accessibility. URL: https://www.w3.org/WAI/fundamentals/accessibility-intro/ (cit. a p. 61).

Pagina Github di Dropzone.js con tutte le opzioni. URL: https://github.com/dropzone/dropzone/blob/main/src/options.js (cit. a p. 50).

Pagina realizzata durante il progetto. URL: https://contest.girolibero.it/ (cit. a p. 2).

Sito web di Cloudinary. URL: https://cloudinary.com/ (cit. a p. 31).

Sito web di Dropzone.js. URL: https://www.dropzone.dev/ (cit. a p. 40).

Sito web di jQuery. URL: https://jquery.com/ (cit. a p. 35).

Sito web di PHP. URL: https://www.php.net/ (cit. a p. 34).

Sito web di PHPUnit. URL: https://phpunit.de/ (cit. a p. 55).

Sito web di Robot Framework. URL: https://robotframework.org/ (cit. a p. 58).

Sito web di Sass. URL: https://sass-lang.com/ (cit. a p. 36).

Sito web di Select2. URL: https://select2.org/ (cit. a p. 46).

Sito web di Twig. URL: https://twig.symfony.com/ (cit. a p. 35).

Sito web di Wordpress.org. URL: https://wordpress.org/ (cit. a p. 33).

Sito web di WPML. URL: https://wpml.org/ (cit. a p. 43).

Sito web ufficiale di Girolibero. URL: https://www.girolibero.it/ (cit. a p. 1).

Usage statistics and market share of WordPress. URL: https://w3techs.com/technologies/details/cm-wordpress (cit. a p. 32).