



## Progetto per il corso di Tecnologie Web

Laurea Triennale in Informatica, Università degli studi di Padova  
Anno accademico 2020/2021

**Indirizzo sito web:**

<http://tecweb1920.studenti.math.unipd.it/fipinton>

**Email referente del gruppo** lorenzo.perinello@studenti.unipd.it



Utenti	Username	Password
Admin	admin	admin
User	user	user

Giro Silvia, 1201167  
Perinello Lorenzo, 1193553

Mantovani Niccolò, 1187325  
Pinton Filippo, 1187361



# Contents

<b>1</b>	<b>Analisi dei requisiti</b>	<b>2</b>
1.1	Caratteristiche dell'utenza . . . . .	2
1.2	Funzionalità . . . . .	2
<b>2</b>	<b>Progettazione</b>	<b>4</b>
2.1	Area pubblica . . . . .	4
2.2	Area privata Utente . . . . .	4
2.3	Area private Amministrazione . . . . .	5
2.4	Accessibilità . . . . .	6
<b>3</b>	<b>Implementazione</b>	<b>7</b>
3.1	HTML . . . . .	7
3.2	CSS . . . . .	7
3.2.1	CSS - schermi di piccole dimensioni . . . . .	7
3.2.2	CSS - stampa . . . . .	8
3.3	Javascript . . . . .	8
3.4	Database, SQL, querying . . . . .	11
3.5	PHP . . . . .	11
3.5.1	Costruzione delle pagine . . . . .	11
3.5.2	Autoload e Namespaces . . . . .	12
3.5.3	Paginazione . . . . .	12
3.5.4	Autenticazione . . . . .	12
3.5.5	Autorizzazione . . . . .	13
3.5.6	Controlli . . . . .	13
<b>4</b>	<b>Test</b>	<b>14</b>
4.1	Validazione codice . . . . .	14
4.2	Testing accessibilità . . . . .	14
<b>5</b>	<b>Organizzazione</b>	<b>15</b>

# Abstract

*Io viaggio non per andare da qualche parte, ma per andare.  
Viaggio per viaggiare. La gran cosa è muoversi, sentire più  
acutamente il prurito della nostra vita, scendere da questo  
letto di piume della civiltà e sentirsi sotto i piedi il granito  
del globo.*

---

— Robert Louis Stevenson

**FlyWeb** è un sito che si propone di agevolare l'utente nella scelta della meta per il suo prossimo viaggio.

L'utente verrà guidato nella selezione tramite una ricerca personalizzata, la quale gli consentirà di visualizzare sia le opzioni di viaggio più adatte alle sue preferenze, sia quelle ad esse strettamente correlate, il tutto suddiviso in diverse categorie.

Una volta terminato un viaggio è possibile per il cliente valutare il servizio offerto: questo potrà scrivere una recensione che sarà pubblicamente visualizzabile tra le informazioni dell'itinerario.

Le offerte di viaggio sono il frutto di una preselezione che avviene ad opera degli amministratori di FlyWeb, i quali hanno la possibilità sia di inserire nuove mete e modificare quelle presenti, sia di moderare le recensioni inserite dagli utenti.

# 1 Analisi dei requisiti

## 1.1 Caratteristiche dell'utenza

Flyweb è una piccola agenzia posta in un giovane quartiere di una grande metropoli occidentale; l'utenza che frequentava l'agenzia era prettamente locale e formata da giovani coppie e famiglie. L'emergenza coronavirus ha richiesto lo sviluppo di un piccolo portale web, ormai imprescindibile per lo sviluppo di qualsiasi attività commerciale, che possa accompagnare l'utenza ed informarla dei viaggi programmati per il futuro. Il sito web inoltre prevede una dashboard di amministrazione per la gestione delle attività essenziali del sito (aggiunta e modifica dei contenuti, moderazione degli utenti).

Le tipologie di utenti che si possono trovare all'interno del sito sono:

- **utente generico:** un utente generico che non ha effettuato il login oppure che non si è registrato;
- **utente loggato:** un utente che si è registrato e che ha effettuato il login;
- **utente admin:** un utente con un accesso privilegiato. L'admin gestisce il contenuto del sito web.

## 1.2 Funzionalità

Il sito dispone di diverse funzionalità, diverse per categoria di utenti:

- **utente generico:**
  - **ricerca del viaggio:** un utente generico può navigare all'interno del sito per visualizzare i viaggi proposti. La ricerca dei viaggi può essere effettuata tramite l'apposita casella di ricerca visualizzata nella pagina *Home* inserendo la parola chiave e la data di inizio e di fine viaggio. La data d'inizio rappresenta l'estremo inferiore e la data di fine l'estremo superiore, per cui, se presenti, vengono visualizzati i viaggi cui range sia all'interno delle date scelte durante la fase di ricerca. Il sito mira ad aiutare l'utente a ricercare un viaggio, per cui anche se alcuni dei campi citati prima (parola chiave e/o data) dovessero non essere inseriti, la ricerca comunque visualizzerà dei viaggi. Nella pagina relativa alla visualizzazione dei viaggi è presente, inoltre, una casella di ricerca con più campi quali *prezzo minimo*, *prezzo massimo*, *ordine di ricerca* e *tipologia di ricerca*;
  - **visualizzazione viaggi:** l'utente potrà visualizzare i dettagli di un viaggio, fra cui l'immagine, la descrizione, il prezzo e le recensioni relative ad esso lasciate dagli altri utenti;
  - **visualizzazione categorie di viaggi:** il sito può suggerire diverse categorie di viaggi, i quali sono raggruppati in viaggi *per lui*, viaggi *per lei*, viaggi di *coppia*, viaggi di *famiglia* e viaggi per l'*anniversario*.
  - **visualizzazione viaggi correlati:** all'interno del dettaglio di un viaggio, un utente può visualizzare una lista di *viaggi correlati* al viaggio in questione, accomunati da una o più categorie citate prima;
  - **registrazione e login:** l'utente può registrarsi o loggarsi, se registrato in precedenza.

- **utente loggato:**

Oltre alle funzionalità dell'utente generico, l'utente loggato può:

- **acquistare viaggi:** l'utente può aggiungere un viaggio nel carrello e effettuare il checkout;
- **modificare le proprie informazioni personali (dati e password);**
- **eliminare il proprio account;**
- **visualizzare gli ordini effettuati;**
- **lasciare una recensione ad un viaggio;**

- **admin:**

L'amministratore nella parte del sito dedicata può:

- **inserire un nuovo viaggio;**
- **modificare un viaggio esistente;**
- **eliminare un viaggio esistente;**
- **moderare le recensioni (approvarle o eliminarle);**

## 2 Progettazione

### 2.1 Area pubblica

Lo schema organizzativo adottato è quello a task.

- **Home:** la prima pagina che si visualizza quando si accede al sito. E' presente una searchbox, infatti lo scopo principale della pagina è cercare i vari viaggi presenti sul database. La searchbox permette di ricercare un itinerario per nome, per città o per Tag. E' possibile visualizzare tutti i viaggi presenti nel portale facendo una ricerca senza specificare nessun campo. Sotto la search box vengono visualizzati dei Tag di possibili idee di viaggio suggerite per l'utente (Per lei, per lui, coppia, famiglia, anniversario).
- **Pagina Risultati Ricerca:** Si viene indirizzati a questa pagina quando si effettua la ricerca dei viaggi dalla pagina Home. La pagina è quindi composta dai viaggi risultanti dalla query di ricerca.
- **Pagina dettaglio viaggio :** In questa pagina è possibile consultare tutte le informazioni relative al viaggio a cui siamo interessati. È inoltre presente il bottone "aggiungi al carrello" che serve per poter acquistare il viaggio. Inoltre in questa pagina si possono visualizzare le recensioni lasciate dagli utenti che hanno già usufruito del viaggio in passato e leggere un loro giudizio sull'esperienza. Sono presenti inoltre, dopo le informazioni e le recensioni, dei viaggi correlati.
- **Pagina About:** Descrizione dell'agenzia turistica e contatti di quest'ultima.

### 2.2 Area privata Utente

- **Pagina Profilo:** in questa pagina è possibile consultare le informazioni relative all'account con cui si è effettuato l'accesso.
- **Pagina Modifica Profilo:**
  - **Modifica Informazioni Profilo:** In questa pagina l'utente può modificare le sue informazioni personali.
  - **Modifica Password:** In questa pagina l'utente può modificare la sua password
- **Pagina Login:** Pagina che serve per effettuare il login al sito web se si è già registrati inserendo il proprio nome utente/email e la password.
- **Pagina Registrazione:** Se non si ha ancora un proprio account nel sito, in questa pagina è possibile effettuare la registrazione inserendo i propri dati personali, un nome utente con cui essere identificati univocamente ed una password con cui accedere alla propria area personale.
- **Pagina Carrello:** Quando viene aggiunto un determinato pacchetto viaggio al carrello è possibile trovarlo in questa pagina. La pagina visualizzerà quindi la lista degli elementi del carrello dell'utente e fornirà un subtotalo. E' possibile rimuovere dal carrello un viaggio premendo l'apposito bottone, il subtotalo verrà ricalcolato di conseguenza. Da questa pagina è possibile procedere al checkout.
- **Pagina Riepilogo Ordine:**

- **Scelta Metodo Pagamento:** E' possibile scegliere tra due metodi di pagamento: Paypal e Carta di credito/Debito. In base alla scelta effettuata si aprirà il form corrispondente.
- **Dati Pagamento:** In base al metodo di pagamento scelto si aprirà nella pagina un form diverso per l'inserimento dei dati di pagamento. Se la scelta è paypal si aprirà il form corrispondente dove immettere la mail relativa all'account di Paypal. Altrimenti si aprirà la form dove immettere i dati relativi alla carta con cui si effettuerà l'acquisto.
- **Dati Indirizzo Fatturazione:** In questa pagina in cui si viene indirizzati dopo l'immissione dei dati di pagamento è possibile inserire il proprio indirizzo di fatturazione.
- **Riepilogo Dati Ordine:** In questa pagina viene effettuato un riepilogo dell'acquisto: Viaggi acquistati, Metodo di pagamento usato e indirizzo di fatturazione. Da questa pagina è possibile completare l'effettivo acquisto del viaggio.
- **Pagina Ordini Effettuati:** Vengono qui visualizzati in ordine ascendente tutti gli ordini che un utente ha effettuato nel corso della sua permanenza sul sito web. Da questa pagina è possibile visualizzare il dettaglio di ogni ordine effettuato.
  - **Dettaglio Ordine:** Qui ci sono le informazioni relative al determinato acquisto effettuato: data, metodo di pagamento usato, l'indirizzo di fatturazione, totale, viaggi acquistati. Se il viaggio è già stato effettuato (ovvero dopo la data di fine viaggio) c'è la possibilità di lasciargli una recensione.
- **Pagina Inserimento Recensione:** A questa pagina si accede dal dettaglio Ordine e solamente se la data di ritorno del viaggio è antecedente al giorno odierno. Qui è possibile lasciare un commento dell'esperienza fatta, dando anche un giudizio complessivo da 1 a 5 (o da 0).
- **Pagina Recensioni lasciate:** in questa pagina l'utente può vedere tutte le recensioni che ha lasciato.
- **Footer:** il footer contiene una piccola mappa/menù di navigazione, informazioni di contatto sul sito, link ai canali social e le note legali. È presente in ogni pagina; nella parte di amministrazione il footer è stato ridotto e contiene solo le note legali.
- **Breadcrumb:** elemento che agevola l'orientamento dell'utente, contiene il percorso effettuato per arrivare in quella determinata pagina. È presente in ogni pagina (anche nel pannello di amministrazione).
- **"Torna su":** elemento che consente di ritornare all'inizio del body. È fissato in basso a destra e viene visualizzato solo dopo aver superato il confine del display.
- **Paginatore:** elemento che consente, ove necessario, di paginare gli elementi. In una pagina vengono visualizzati in tutto 9 elementi.

## 2.3 Area private Amministrazione

- **Home Amministrazione:** E' la pagina nella quale si viene indirizzati dopo aver fatto il login se si è abilitati come admin. Introduce la parte del sito web destinata esclusivamente agli amministratori ed è pensata per essere utilizzata da dispositivi Desktop.



- **Pagina Inserimento Viaggio:** In questa pagina l'amministratore ha la possibilità di inserire un nuovo viaggio immettendo nel form tutte le informazioni necessarie.
- **Pagina Modifica Viaggio:** Qui tramite la barra di ricerca posso individuare il viaggio/i viaggi da modificare. Individuato il viaggio/i viaggi posso eliminarlo dal database oppure modificarne le informazioni.
- **Pagina Modera Recensioni:** In questa pagina l'amministratore ha il compito di visionare le recensioni che sono state lasciate dagli utenti nei vari viaggi e di moderarle. Ha la possibilità di rifiutare, ad esempio nel caso di parole scurrili, oppure di approvarla. Una recensione approvata viene pubblicata nella pagina dei dettaglio di un viaggio.

## 2.4 Accessibilità

Abbiamo tenuto conto dell'accessibilità durante tutta la fase di progettazione e implementazione del progetto. In primis abbiamo tenuto una netta separazione tra struttura, comportamento e presentazione. Per aiutare l'orientamento dell'utente è stato aggiunto in ogni pagina l'elemento di navigazione Breadcrumb che indica il percorso fatto dall'utente per arrivare in quella pagina. Per gli utenti che usufruiscono del sito tramite screen reader sono stati posti degli aiuti alla navigazione per saltare determinati elementi ed andare direttamente ad altri contenuti (esempio: saltare il menù principale per andare direttamente al breadcrumb, tornare al top del searchbox, ritornare alla breadcrumb prima di procedere nel footer). È presente inoltre un bottone per facilitare ad ogni utente il ritorno all'inizio della pagina.

- **Immagini:** sono stati utilizzati alt vuoti per le immagini di sola presentazione, gli altri alt sono stati riempiti con un equivalente testuale. Le immagini inoltre hanno un background-color nel caso ci siano problemi (ad esempio di rete) cosicché non vengono visualizzati spazi vuoti ma colorati.
- **Link:** link circolari non sono presenti, link mantengono formattazione aderente a convenzioni esterne (ad eccezione dei link in breadcrumb: dal nostro punto di vista il loro ruolo, prevalentemente di orientamento, permette la loro non differenziazione tra visitato e non visitato). I link che hanno lo stesso contenuto testuale sono stati disambiguati con l'attributo title che li distingue.
- **Form:** sono stati raggruppate logicamente gli input in fieldset, utilizzati i placeholder e i tag required, l'attributo tabindex. In alcuni casi gli errori vengono visualizzati dopo il submit del form sopra ad esso (ad esempio se viene sbagliata la password nel login), in altri casi invece sono visualizzati sotto gli input in maniera dinamica.
- **Lingua:** il body ha attributo lang="it"; le parole non in italiano sono state segnalate attraverso l'attributo "lang=lingua". Inoltre nell'inserimento di una recensione è possibile scegliere la lingua in cui verrà scritta.
- **W-aria:** nonostante sia stato privilegiato l'utilizzo dei tag html5, alcuni elementi, in particolare per quelli dinamici (menù mobile dinamico, finestrella filtri searchbox ...), sono stati arricchiti con vari attributi w-aria per migliorare la fruizione del sito tramite tecnologie assistive.

## 3 Implementazione

In questa sezione viene descritto come è stato creato il sito, dividendo la realizzazione di quest'ultimo nei linguaggi utilizzati.

### 3.1 HTML

Le pagine del sito sono state realizzate utilizzando il linguaggio HTML5 piuttosto che XHTML. Tale decisione deriva dal fatto che il sito mira ad avere un'utenza prevalentemente giovane, quindi avente browser aggiornati, e soprattutto perchè HTML5, nell'imminente futuro, diventerà il linguaggio predefinito per la realizzazione di pagine web. Le principali funzionalità HTML5 che abbiamo sfruttato sono:

- **<input type="date">**: per l'inserimento delle date all'interno del sito. Per garantire l'accessibilità a tutti i browser non compatibili con tale tag, comunque, sono stati fatti dei controlli per notificare l'utente sul formato richiesto. Inoltre, nella componente di ricerca, sono stati aggiunti attributi `w-aria` per identificare lo screen-reader nel caso di campi mal formati;
- **<input type="month">**: per l'inserimento della scadenza della carta di credito durante il pagamento di un viaggio. Per garantire l'accessibilità a tutti i browser non compatibili con tale tag, comunque, sono stati fatti dei controlli per notificare l'utente sul formato richiesto.
- **<input type="email">**: per l'inserimento dell'email, dove viene richiesto;
- **<input type="number">**: per l'inserimento dei prezzi e codici per la carta di credito;
- **<nav>**: utilizzato per la creazione dei menù e degli elementi di navigazione in generale;
- **<footer>**: utilizzato per la realizzazione del footer;
- **<address>**: utilizzato per segnalare indirizzi;
- **<time>**: utilizzato per la segnalare eventuali orari;
- **required, placeholder, min, max, autocomplete**: vengono utilizzati questi nuovi attributi per aiutare l'utente nell'inserimento dei campi richiesti.

### 3.2 CSS

Per la parte di presentazione del sito è stato utilizzato CSS3. Il layout utilizzato è puro, con completa separazione tra struttura e formattazione, è inoltre responsive: per implementarlo abbiamo usufruito di flexbox.

#### 3.2.1 CSS - schermi di piccole dimensioni

Flexbox ha permesso di ridurre il codice ridefinito per gli schermi di minori dimensioni. Nonostante questo sono state definite alcune regole per adattare il layout che in ogni caso rimane coerente e coeso. Il cambiamento più importante riguarda il menù che è stato trasformato in un menù ad hamburger nella parte di sito visibile agli utenti. La parte di amministrazione, sebbene non sia pensata per l'utilizzo da mobile o schermi piccoli, si trasforma comunque in maniera elegante.



Figure 1: Layout "standard".



Figure 2: Layout "mobile".



Figure 3: Formattazione di stampa.

### 3.2.2 CSS - stampa

Per la formattazione di stampa sono state tolte le immagini di presentazione e lasciate quelle utili a descrivere i viaggi. Sono stati tolti menù e minimizzato di più possibile il footer, la breadcrumb è stata lasciata volutamente come riferimento. I colori definiti sono tutti tra #fff e #000 (bianco, nero, scala di grigi). Il font utilizzato per la stampa è un font con grazie (serif). Il css per la stampa utilizza misure assolute ed è ottimizzato per le seguenti pagine: ordini, dettaglio ordine, carrello, dettaglio del viaggio, lista viaggi.

### 3.3 Javascript

La maggior parte di funzionalità che richiedono l'interazione con l'utente, come per esempio l'inserimento dei campi di una form, vengono controllate lato cliente tramite funzioni **javascript**, visualizzando all'utente il relativo messaggio d'errore nel caso qualche campo inserito non sia valido. Inoltre, per garantire l'accessibilità, se il javascript non dovessere essere abilitato, viene fatto un controllo lato server tramite **php**, e visualizzato il relativo errore. Ogni funzione, per notificare l'utente durante l'inserimento di un campo sbagliato, rende visibile

un paragrafo nascosto contenente il messaggio d'errore. Si è deciso di agire in questo modo per separare maggiormente codice HTML e codice Javascript, infatti viene solo cambiata una proprietà di stile di un paragrafo e modificato il suo contenuto.

Di seguito le funzione javascript utilizzate:

- **validationData()**: controlla che la data di inizio durante l'inserimento di un viaggio da parte dell'amministratore sia antecedente alla data di fine;
- **validationPrz()**: controlla la validità del prezzo (prezzo maggiore uguale a 0) durante l'inserimento di un viaggio da parte dell'amministratore;
- **checkCartaCredito()**: controlla che il codice della carta di credito inserito durante la fase di pagamento abbia da 13 a 16 cifre numeriche;
- **checkScadenzaCarta()**: controlla se la data inserita è scaduta e lo notifica all'utente se necessario;
- **checkCVV()**: controlla che il CVV della carta di credito inserito durante la fase di pagamento sia esattamente formato da 3 cifre numeriche;
- **checkIndirizzi(id,Element)**: controlla che l'input inserito in *via*, *provincia* e *comune* durante l'inserimento dei dati di fatturazione abbia almeno delle lettere o numeri. Inoltre verifica che l'input abbia solamente caratteri da A-Z, a-z, 0-9, \_ e il carattere spazio. I campi *id* e *Element* identificano il campo da dove prendere il valore da controllare e il campo dove inserire il messaggio d'errore da visualizzare, rispettivamente;
- **checkCAP()**: controlla che il CAP inserito durante l'inserimento dei dati di fatturazione abbia esattamente 5 cifre numeriche;
- **validationTipoNome(id,Element)**: controlla che i dati inseriti in alcune form che richiedono un nominativo, o un dato simile ad un nominativo, contengano da 2 a 30 caratteri totali fra A-Z, a-z, lettere accentate e il carattere spazio. Inoltre il campo deve contenere almeno delle lettere. I campi *id* e *Element* identificano il campo da dove prendere il valore da controllare e il campo dove inserire il messaggio d'errore da visualizzare, rispettivamente. Tale controllo viene utilizzato durante la registrazione di un utente, la modifica delle informazioni personali di un utente e nell'intestatarario della carta di credito durante il checkout;
- **validationUsername()**: controlla che l'username inserito dall'utente sia composto da almeno 4 e massimo 15 caratteri. Inoltre sono permessi i caratteri come #, -, \_, @ e l'username deve contenere almeno delle lettere;
- **validationEmail()**: controlla che l'email inserita da parte dell'utente sia in un formato standard;
- **validationPassword()**: controlla che la password inserita abbia almeno un carattere maiuscolo, uno minuscolo, un numero e sia lunga almeno 8 caratteri;
- **validationPasswordRepeat()**: controlla che la password inserita nel campo di *ripetizione password* corrisponda alla password precedentemente inserita;

- **validationDataNascita()**: controlla che l'utente, durante la fase di registrazione o di modifica del profilo, abbia almeno 14 anni. Tale funzione utilizza una funzione di supporto per il calcolo dell'età, chiamata **getAge(data\_nascita)**;
- **titoloEDescrizioneRecensione(id,Element)**: controlla che il titolo o la descrizione, durante l'inserimento di una recensione, abbia almeno delle lettere e che contenga solamente caratteri come A-Z, a-z, 0-9, punteggiatura, \_ e il carattere spazio. I campi *id* e *Element* identificano il campo da dove prendere il valore da controllare e il campo dove inserire il messaggio d'errore da visualizzare, rispettivamente;
- **hamb()**: funzione che gestisce il menù ad hamburger che compare negli schermi al di sotto di 868px. La funzione, attraverso `classList.add` e `classList.remove`, aggiunge/rimuove delle classi le quali determinano la formattazione del menù. Per capire se è già visualizzato il menù controlliamo la presenza della classe "hide-menumobile" nel componente con il comando `.classList.contains("hide-menumobile")`;
- **filters()**: funzione che gestisce la visualizzazione dei filtri nel searchbox. Il funzionamento è analogo a quello della funzione `hamb()`;
- **scrollFunction()**: funzione che visualizza il tasto *Torna su* dopo che si è superato un certo valore di scroll;
- **ConfermaEliminazione()**: funzione che mostra un pop-up di conferma durante l'eliminazione di un viaggio o di una recensione da parte dell'amministratore. Si è deciso di utilizzare un pop-up perchè tale azione avviene molto spesso e mostrare una seconda pagina di riepilogo delle informazioni avrebbe allungato di molto la procedura;
- **conferma()**: funzione che mostra un pop-up di conferma durante l'inserimento di un viaggio da parte dell'amministratore. Si è deciso di utilizzare un pop-up perchè tale azione avviene molto spesso e mostrare una seconda pagina di riepilogo delle informazioni avrebbe allungato di molto la procedura;
- **validationSearchDate()**: funzione che controlla se le date inserite durante la ricerca di un viaggio da parte di un utente generico siano valide (l'amministratore non ha questo controllo perchè ha la possibilità di modificare anche viaggi già terminati). Nello specifico controllano che:
  - la data di inizio sia successiva alla data odierna;
  - la data di fine sia successiva alla data odierna;
  - la data di fine sia successiva alla data d'inizio.

Se queste condizioni non vengono rispettate, viene notificato l'utente tramite un messaggio di validazione personalizzato dell' HTML5;

- **validationSearchPrice()**: funzione che controlla se i prezzi inseriti durante la ricerca di un viaggio da parte di un amministratore o di un utente generico siano validi. Nello specifico controlla che il prezzo massimo sia maggiore del prezzo minimo. Se queste condizioni non vengono rispettate, viene notificato l'utente tramite un messaggio di validazione personalizzato dell' HTML5.

### 3.4 Database, SQL, querying

Il database serve per mantenere le informazioni riguardanti gli utenti, i carrelli, gli ordini, i viaggi, le recensioni e i tag associati ai viaggi.

La tabella Utente identifica un utente iscritto al sito web. Un utente può essere un normale Utente o un Admin, tale distinzione avviene attraverso il campo *admin*. La chiave primaria della tabella è un id univoco, sono inoltre unique anche gli attributi username ed email per permettere l'accesso al sito utilizzando indistintamente l'email e l'username.

Tutte le tabelle hanno un identificatore univoco autoincrementale.

La tabella Carrello presenta come chiave esterna l'id dell'utente a cui è associato il carrello e l'id del viaggio in esso contenuto: è così possibile, nel caso l'utente volesse, inserire più volte lo stesso viaggio nel carrello.

Nella tabella Ordine la chiave esterna è ID\_Utente e associa l'ordine ad un determinato utente. Per associare un viaggio ad un ordine utilizziamo una tabella intermedia OrdineViaggio. La tabella Recensioni e RecensioniViaggi hanno una struttura analoga.

La tabella Tag identifica le istanze tramite chiave primaria ID\_Tag, un tag è associato ad un viaggio tramite le chiavi esterne presenti nella tabella TagViaggio.

Per ogni tabella presente nel db è stata creata una classe in PHP in modo da poter manipolare in maniera semplice gli elementi del Database, come se fossero degli oggetti. Tali classi sono chiamate *modelli*. Molti modelli usano un particolare costruttore che permette di sopperire all'impossibilità di definire in PHP più costruttori di una classe con un numero di parametri diversi.

Per facilitare l'accesso ad db abbiamo creato la classe *Database* che si occupa di fare la connessione al database ed espone dei metodi di utilità per interagire con il DB, tra cui un metodo per poter effettuare le query. Le query sono tutte eseguite tramite prepare statement, per prevenire problemi di sicurezza.

Normalmente viene creato solamente un oggetto *Database*, e tale oggetto viene istanziato esclusivamente nella classe *BaseController*, da cui derivano tutti gli altri controller che sono delle classi in cui vengono manipolati i dati. Le query possono avvenire esclusivamente attraverso l'uso di un controller.

### 3.5 PHP

#### 3.5.1 Costruzione delle pagine

Tutte le pagine del sito sono costruite dinamicamente tramite la composizione di elementi chiamati *componenti*.

L'idea alla base del sistema a componenti che abbiamo creato è quella di creare dei pezzi di HTML che fossero facilmente riutilizzabili, senza doversi necessariamente preoccupare del modo in cui tali componenti vengono creati o funzionano. Ogni componente è composto da un template che contiene il codice HTML del componente, e da una classe PHP che si occupa di idratare il template. Alla base di questo sistema c'è la classe *Template* che si occupa di prelevare il template HTML del componente e di esporre dei metodi di utilità che permettono l'eventuale idratazione del template, come ad esempio *replaceTag(string, string): string*

che si occupa di fare il replace di tutte le occorrenze di un tag (solitamente invalido) presente all'interno del template caricato, con una stringa arbitraria (solitamente un altro componente); oppure *replaceValue(string, string)*: *string* che si occupa, invece, di fare il replace di tutte le occorrenze di un segnaposto-di-valore (indicato *%segnaposto-di-valore%*) presente all'interno del template caricato, con una stringa arbitraria.

La classe *template* espone inoltre il metodo principale, ovvero il metodo *render()*: *string* che, attraverso l'override del magic method *\_\_toString()*: *string* permette di trasformare gli oggetti *Template* in stringhe contenenti il codice HTML caricato dal template ed eventualmente idratato.

Per facilitare la creazione di componenti e' stata creata la classe *BaseComponent* che estende la classe *Template*, imponendo delle convenzioni circa il posizionamento dei file all'interno dell'albero delle directory del sito, e aggiungendo alcuni metodi di utilita', che permettono ad esempio il caricamento di alcuni parametri direttamente dalla richiesta effettuata al server.

In questo modo, sebbene inizialmente il lavoro per il design e la creazione di un componente sia superiore rispetto alla scrittura "*hardcoded*", con il passare del tempo, il riutilizzo di componenti gia' scritte, funzionanti permette di accelerare notevolmente lo sviluppo.

### 3.5.2 Autoload e Namespaces

Per rendere trasparente l'importazione delle classi utilizzate all'interno degli script PHP che costruiscono le pagine abbiamo creato uno script php che si occupa di importarle automaticamente. Tale script si chiama *autoload*. A tale scopo abbiamo anche organizzato le classi in namespaces, in modo da organizzarle logicamente in gruppi omogenei e facilitarne l'utilizzo e l'importazione.

### 3.5.3 Paginazione

Alcune pagine del nostro sito rischiavano di essere di dimensioni eccessivamente grandi, come ad esempio la pagina che mostra i risultati di una ricerca. Per questo motivo abbiamo deciso di visualizzare, nelle pagine che potrebbero contenere un numero molto alto di elementi, i risultati paginati. In questo modo l'utente e' in grado di visualizzare un numero adeguato di risultati per pagina, rendendo agevole la consultazione e ridotte le dimensioni della pagina. L'utente, tramite un selettore posto in fondo alla pagina e' in grado di muoversi tra le pagine disponibili. In ogni momento dal paginatore sono sempre raggiungibili (se esistono) la prima pagina disponibile, l'ultima pagina disponibile, la pagina precedente e la pagina successiva a quella corrente.

### 3.5.4 Autenticazione

L'autenticazione è gestita attraverso il sistema delle sessioni, fornito nativamente da PHP. Quando un utente effettua il login viene salvata una variabile di sessione *logged\_in=true*, e, nel caso in cui l'utente loggato sia un amministratore, viene salvata anche una variabile di sessione *admin=true*. Quando l'utente esegue il logout, la sessione viene resettata attraverso le funzioni *session\_unset()* e *session\_destroy()*, che eliminano tutte le informazioni di sessione salvate.

### 3.5.5 Autorizzazione

L'accesso ad alcune pagine e' ristretto ad alcune categorie di utenti, per questo motivo e' stato necessario trovare un modo per limitare l'accesso a tali pagine. Per fare cio' abbiamo creato un controller chiamato *RouteController* che espone dei metodi statici che controllano le autorizzazioni dell'utente che sta richiedendo la pagina e se l'utente e' autorizzato servono la pagina, altrimenti forzano il redirect verso una pagina accessibile all'utente. All'inizio di ogni pagina e' quindi necessario definire il tipo di pagina di cui si tratta attraverso i metodi statici sopra accennati. I tipi di pagina messi a disposizione sono:

- *Protected*: rappresenta le pagine accessibili solamente agli amministratori;
- *Unprotected*: rappresenta le pagine accessibili solamente agli utenti autenticati non amministratori;
- *Logged*: rappresenta le pagine accessibili a tutti gli utenti autenticati;
- *Unlogged*: rappresenta le pagine accessibili a tutti gli utenti.

Nel caso in cui non si sia autenticati e si richieda una pagina che richiede autenticazione si viene rediretti alla pagina di login, e, una volta effettuato il login, si viene rediretti alla pagina che era stata richiesta precedentemente, in questo modo e' possibile continuare il flusso di operazioni che si stava svolgendo.

### 3.5.6 Controlli

Sono stati implementati con PHP controlli backend sugli input.





Figure 4: Esempio color testing manuale.



Figure 5: Esempio color contrast checking automatico.

## 4 Test

### 4.1 Validazione codice

Sono stati utilizzati i seguenti strumenti per la validazione dei codici

- HTML5: W3C Markup Validation Service
- CSS3: W3C CSS Validation Service

### 4.2 Testing accessibilità

L'accessibilità è stata testata sia con strumenti automatici che manualmente, di seguito riportiamo i tools utilizzati:

- screen reader utilizzati per valutare manualmente l'accessibilità:
  - VoiceOver (screen reader integrato su dispositivi Apple)
  - Screen Reader (plugin per Chrome sviluppato da Google)
- tool per simulare diverse capacità visive e di apprendimento:
  - Silktide (plugin per Google Chrome)
- tool per valutare il contrasto dei colori:
  - Color contrast accessibility validator (color.a11y.com)
  - Wave (plugin per Google Chrome)

Il sito è stato inoltre testato sui principali browser (Chrome, Firefox, Safari, Edge), e su vari dispositivi Desktop e Mobile (sia iOS che Android).

## 5 Organizzazione

Il gruppo ha dapprima effettuato la fase di analisi e progettazione, per poi passare alla definizione degli incarichi da svolgere in autonomia. I compiti sono stati assegnati tramite l'utilizzo del software Jira, per garantire un pieno controllo su ciò che è stato fatto e sulle future implementazioni. Le mansioni sono state suddivise per categoria (HTML,CSS,PHP...), e ogni categoria è stata trattata per la maggior parte da una singola persona del team, così da garantire una fase di sviluppo più rapida e in grado di sfruttare al meglio le competenze di ognuno. Come ci si può aspettare, vi sono state delle eccezioni che hanno garantito un equo apprendimento della materia da parte di ogni componente. La relazione è stata stesa da tutti i componenti.

Per quanto concerne i singoli incarichi:

- **Filippo Pinton:**

- Definizione e organizzazione struttura del codice;
- Classi base per componenti e controller, autoload;
- Componenti e controller per la paginazione;
- Registrazione, autenticazione e autorizzazione utenti;
- Componenti e controller riguardanti la ricerca, i risultati della ricerca e i dettagli dei viaggi;
- Revisione e approvazione generale, fix di integrazione

- **Silvia Giro:**

- Componenti (HTML, PHP) delle parti relative al profilo utente;
- Componenti (HTML, PHP) delle parti relative al carrello;
- Componenti (HTML, PHP) delle parti relative alla procedura di acquisto;
- Componenti (HTML, PHP) delle parti relative agli ordini;
- Database;
- Validazione codice;
- Testing accessibilità;

- **Lorenzo Perinello:**

- CSS;
- Componenti (HTML, PHP) della parte di amministrazione;
- Funzioni javascript hamb() e filters();
- Validazione codice;
- Testing accessibilità;
- Database.

- **Niccolò Mantovani** ha realizzato:

- Alcune funzioni Javascript;

- Le componenti per la visualizzazione delle recensioni relative ad un viaggio e quelle specifiche di un utente (HTML e PHP);
- Componenti per la visualizzazione dei viaggi correlati delle categorie suggerite, ossia i **tag** (HTML e PHP);
- Componenti relative ai menù (HTML e PHP);
- Componente footer visualizzato dall'utente generico (HTML e PHP);
- Pagina ***About Us*** (HTML e PHP);
- Pagina ***404*** (HTML e PHP);
- Componente per la realizzazione di un messaggio di response (HTML e PHP).