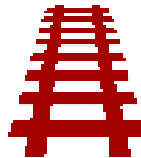


Università degli Studi di Padova

## RELAZIONE PROGETTO DI TECNOLOGIE WEB A.A.2023-2024

Iberu Trasporti



**Indirizzo del sito:**

[http:](http://tecweb.studenti.math.unipd.it/rfabbian/iberu/index.php)

[//tecweb.studenti.math.unipd.it/rfabbian/iberu/index.php](http://tecweb.studenti.math.unipd.it/rfabbian/iberu/index.php)

**Mail referente:**

riccardo.fabbian@studenti.unipd.it

**Utenti specifici:**

ID	Password
user@user.com	useruser
admin@admin.com	adminadmin

**Relatori:**

Carraro Alessandro	2000548
Chilese Elena	2008074
Fabbian Riccardo	2009110
Passarella Alessandro	2000557

# Contents

<b>1</b>	<b>Scopo del sito</b>	<b>3</b>
1.1	Analisi Utente . . . . .	3
1.2	Servizi offerti . . . . .	3
1.3	Possibili Ricerche . . . . .	3
<b>2</b>	<b>Sviluppo</b>	<b>5</b>
2.1	Struttura del sito . . . . .	5
2.2	HTML . . . . .	5
2.3	CSS . . . . .	6
2.3.1	Esempio visualizzazioni . . . . .	6
2.4	Database . . . . .	7
2.5	PHP . . . . .	8
2.5.1	Accesso al DB . . . . .	8
2.5.2	Metodi utilizzati . . . . .	8
2.5.3	Costruzione delle pagine . . . . .	9
2.5.4	Autenticazione e autorizzazione . . . . .	9
2.5.5	Controlli . . . . .	9
2.6	JavaScript . . . . .	10
<b>3</b>	<b>Accessibilità</b>	<b>11</b>
3.1	Separazione struttura, funzionalità e presentazione . . . . .	11
3.2	Navigazione . . . . .	11
3.3	Form . . . . .	12
3.4	Attenzioni particolari all'accessibilità . . . . .	12
<b>4</b>	<b>Verifica</b>	<b>13</b>
<b>5</b>	<b>Suddivisione del lavoro</b>	<b>14</b>
5.1	Web Design . . . . .	14
5.2	HTML e SEO . . . . .	14
5.3	CSS e JavaScript . . . . .	14
5.4	PHP . . . . .	15
5.5	Verifiche . . . . .	15
5.6	Conclusioni . . . . .	15
<b>6</b>	<b>Difficoltà incontrate</b>	<b>16</b>

# 1 Scopo del sito

Durante il seminario sull'accessibilità i componenti del gruppo sono rimasti colpiti dalla scarsa accessibilità dei siti di trasporto pubblico, in particolare dal sito web di Italo.

Inoltre il gruppo ha voluto evitare i soliti siti vetrina, i siti e-commerce e i siti su locali gastronomici.

Abbiamo, quindi, scelto come soggetto del progetto una società ferroviaria. Tutti i componenti hanno esperienza come pendolari e abbiamo cercato di creare un sito il più possibile cucito attorno alle nostre necessità e alla nostra esperienza.

Alla base del nostro sito c'è la necessità di reperire velocemente le informazioni, con particolare attenzione per la sezione delle comunicazioni urgenti, andando a limitare le sottopagine o i click necessari. Abbiamo comunque voluto completare il sito con notizie e offerte.

## 1.1 Analisi Utente

Alla luce di questa analisi, il sito è dedicato a tutte le persone che desiderano acquistare biglietti del treno in maniera accessibile e senza inciampare nelle difficoltà presenti in altri siti di trasporti.

## 1.2 Servizi offerti

In particolare, è presente:

- Una sezione per le comunicazioni pensata principalmente per informare tempestivamente di eventuali ritardi;
- Una sezione help dedicata all'assistenza del cliente;
- Una sezione offerte dedicata all'incentivare studenti, famiglie e anche anziani, che solitamente tendono a non spostarsi in treno;
- Pagina informativa dell'azienda e dei suoi principi;
- Effettuare la registrazione al sito, ottenendo i vantaggi dell'avere un account;
- Conseguente possibilità di eseguire il login;
- Possibilità di controllare i biglietti acquistati ancora validi;
- Tenere uno storico dei biglietti acquistati.

## 1.3 Possibili Ricerche

Il sito è stato oggetto di ottimizzazione attraverso l'incorporazione di keyword specifiche, tra cui "Trasporto ferroviario", "Prenotazioni", "Viaggi in treno", "Biglietti",

”Orari” ed ”Offerte”. L’ottimizzazione dei file è stata realizzata mediante la compressione e minificazione, contribuendo a migliorare la velocità di caricamento delle pagine. Un’analisi approfondita della semantica, responsività e accessibilità è stata eseguita per garantire una struttura di qualità.

L’utilizzo di queste keyword specifiche, combinato con le pratiche di ottimizzazione menzionate e una struttura di qualità, dovrebbe contribuire significativamente al miglioramento del ranking del sito all’interno dei motori di ricerca, come ad esempio Google. Infine, nelle pagine più pesanti, soprattutto nell’homepage e nella pagina di amministrazione (adminpage), il server risponde impostando un header per il cache-content con una durata di un giorno intero. Nonostante questa scelta possa sembrare eccessiva, è stata deliberatamente configurata a questo livello per renderla evidente durante l’analisi statica.

## 2 Sviluppo

Lo sviluppo del sito è iniziato con la creazione dei template per le singole pagine da parte di ogni componente, sia per la visualizzazione mobile che per la visualizzazione desktop. Successivamente, dopo un attento confronto, abbiamo definito i template su cui basarci e da essi abbiamo iniziato a sviluppare il codice HTML e CSS, utilizzando la strategia mobile-first.

### 2.1 Struttura del sito

Il sito realizzato si compone di sei pagine principali (Homepage, Offerte, Comunicazioni, Chi siamo, Help ed Account), due pagine specifiche (una per gli admin e una per gli utenti generici) a cui si ha accesso dopo aver fatto il login e altre pagine per la visualizzazione dei biglietti disponibili e per l'acquisto di quest'ultimi. Inizialmente si era pensato di creare anche una pagina per l'acquisto degli abbonamenti che però è stata successivamente eliminata a causa dell'eccessivo carico di lavoro che avrebbe comportato la gestione di quest'ulteriore funzione.

### 2.2 HTML

L'implementazione del sito web è stata svolta in HTML5, come da requisiti del progetto. Alcuni dei tag maggiormente utilizzati sono stati:

- **<nav>**  
Utilizzata in diverse posizioni all'interno del sito, ha il compito di guidare l'utente nella navigazione. Ha trovato implementazioni, agendo come barra di navigazione primaria, breadcrumb e barra di navigazione secondaria (ad esempio, nei layout a schede nelle pagine utente ed amministratore).
- **<header>**  
Utilizzato principalmente come contenitore per la barra di navigazione, in alcune pagine agisce anche come contenitore per l'intestazione di sezioni.
- **<breadcrumb>**  
Presente come barra di navigazione in linea in tutte le pagine, il suo ruolo è aiutare l'utente ad orientarsi all'interno del sito fornendo i link alle pagine precedenti.
- **<footer>**  
Funge da elemento terminale, utilizzato sia a piè pagina che come elemento finale in alcune sottosezioni.

## 2.3 CSS

Per la presentazione del sito, abbiamo implementato CSS3 con un layout di tipo puro, caratterizzato da una netta separazione tra la struttura e la formattazione. Va sottolineato che il design è responsive, garantendo un'adeguata visualizzazione e funzionalità su vari dispositivi, come desktop, laptop, tablet e smartphone.

Durante lo sviluppo, sono stati creati due file CSS distinti: `style.css` e `print.css`. `Style.css` incorpora media query interne che adattano il layout in base al dispositivo utilizzato. L'approccio mira a massimizzare la condivisione del codice tra le diverse media query, spostando il codice comune al di fuori di esse per ridurre lo sforzo computazionale su tutte le categorie di dispositivi.

D'altra parte, `print.css` si occupa esclusivamente dei layout per la stampa, semplificando i colori ed eliminando immagini e informazioni superflue.

Da notare che nel processo di sviluppo del codice CSS, abbiamo cercato di seguire la BEM convention per favorire la leggibilità e la manutenibilità del codice. Anche se l'applicazione di questa convenzione presenta alcune variazioni nelle denominazioni delle classi, soprattutto per rispettare specifiche restrizioni sulla presenza di tag HTML e composizioni di tag HTML e classi, è stata inizialmente adottata con l'obiettivo di facilitare il lavoro collaborativo. Tale convenzione non risulta applicata nel senso più stretto poiché spesso nella nomenclatura della classi si è rispettato più il format (`_` e `-`) rispetto alla semantica.

### 2.3.1 Esempio visualizzazioni

- Visualizzazione Desktop



- Visualizzazione Mobile



Il file di rendering utilizzato dal sito non è style.css ma style.min.css. Style.min.css è un file css minified, ovvero inline, in modo da guadagnare velocità durante il primo invio.

## 2.4 Database

Il database è scritto interamente in linguaggio SQL e, durante lo sviluppo del sito, è stato caricato sulla piattaforma phpMyAdmin.

Per facilitare l'accesso ad database abbiamo creato la classe DBAccess che si occupa di fare la connessione al database ed espone dei metodi di utilità per interagire con il DB stesso, soprattutto per l'aggiunta, la visualizzazione e l'eliminazione di tuple delle varie tabelle.

Il database serve per memorizzare le informazioni riguardanti i treni, le stazioni, le varie tratte, gli utenti, le comunicazioni e le offerte.

Le tabelle station, route\_station, route e route\_schedule servono per la memorizzazione delle tratte che Iberu Trasporti offre, degli orari di partenza dei treni per le singole tratte e sono utili per il calcolo dinamico della durata del viaggio che l'utente sceglierà, tale calcolo avviene direttamente nella pagina tickets.php tramite un'apposita query.

Le tabelle admin e user servono a memorizzare relativamente le credenziali di accesso per gli admin del sito (che verranno reindirizzati all'adminpage) e per gli utenti generici (che visualizzeranno la loro userpage personale).

La tabella tickets associa il singolo utente ai suoi biglietti acquistati e non ancora utilizzati, oltre a fungere da storico per i biglietti scaduti.

Infine vi sono le tabelle news e offers che contengono relativamente le comunicazioni e le offerte.

## 2.5 PHP

### 2.5.1 Accesso al DB

Per agevolare l'accesso al database è stato creato il file DBAccess.php al cui interno vi è la classe DBAccess che contiene le variabili di accesso utilizzate poi dal metodo openDBConnection il quale permette l'accesso al database sopra descritto. Per la chiusura della connessione abbiamo creato la funzione specifica closeDBConnectio().

### 2.5.2 Metodi utilizzati

La classe DBAccess fornisce inoltre vari metodi con utilità differenti e sono di seguito descritti:

- **I metodi check** svolgono la funzione di controllo.  
Nello specifico queste funzioni sono:
  - checkLogin(string \$email, string \$password) per controllare la correttezza dei dati inseriti (mail e password) per il login di un admin;
  - checkLoginAdmin(string \$email, string \$password) per controllare la correttezza dei dati inseriti (mail e password) per il login di un utente generico;
  - checkStazione(\$stazione) utilizzata per controllare se le stazioni inserite dall'utente sono effettivamente presenti nel nostro db.
- **I metodi add e delete** vengono utilizzati per l'aggiunta e l'eliminazione di tuple nel db.  
Nello specifico queste funzioni sono:
  - addData(string \$query) per l'aggiunta generica di una tupla;
  - addCommunication(\$data\_i, \$data\_f, \$titolo, \$contenuto) per l'aggiunta di una nuova comunicazione;
  - addOffer(\$classe, \$nome, \$titolo, \$contenuto, \$codice\_sconto, \$percentuale, \$data\_fine, \$minimo) per la creazione di una nuova offerta;
  - addUser(\$nome, \$cognome, \$email, \$data\_nascita, \$np) per la creazione di un nuovo utente;
  - deleteCommunication(\$id) per l'eliminazione di una specifica comunicazione;
  - deleteOffer(\$id) per l'eliminazione di una specifica offerta.
- **I metodi get e view** sono utili per la visualizzazione delle tuple delle varie tabelle.  
Nello specifico queste funzioni sono:
  - getData(string \$Tname) per la costruzione dinamica di una query di selezione;
  - getDataArray(string \$query) per la costruzione dinamica di una query di selezione e costruzione di un array di ritorno;



- `viewCommunication()` per la visualizzazione delle comunicazioni;
- `viewOffers()` per la visualizzazione delle offerte;

### 2.5.3 Costruzione delle pagine

Ciascuna pagina del sito web viene generata dinamicamente attraverso il seguente processo: all'interno di ogni file HTML, abbiamo implementato segnaposto con nomi specifici. Utilizzando PHP, successivamente sostituiamo tali segnaposto con frammenti di codice dinamicamente generati. Questa metodologia sfrutta due comandi chiave: *file\_get\_contents()*, che recupera il contenuto della pagina HTML, e successivamente *str\_replace()*, il quale effettua la sostituzione dei segnaposto con gli oggetti creati dinamicamente, avvalendosi dei dati presenti nel database.

### 2.5.4 Autenticazione e autorizzazione

Per gestire l'autenticazione nel sito, sfruttiamo le sessioni messe a disposizione da PHP. Una variabile di sessione è utilizzata per memorizzare l'ID dell'utente o dell'amministratore. Durante la procedura di logout, il file `logout.php` è chiamato per eliminare l'ID dalla sessione.

Per quanto riguarda la navigazione all'interno del sito, è permesso agli utenti di effettuare operazioni senza necessariamente eseguire il login. Tuttavia, al momento dell'acquisto di un biglietto, è obbligatorio, per gli utenti non autenticati, compilare tutti i campi del modulo di acquisto. Inoltre essi non avranno accesso allo storico dei biglietti precedentemente acquistati.

Va notato che anche se un utente è registrato ma non autenticato al momento l'acquisto, il biglietto verrà inviato via email utilizzando l'indirizzo fornito nel modulo di acquisto e non sarà visibile nel suo profilo personale. È importante sottolineare che, sebbene la funzionalità di invio mail sia menzionata, attualmente non è stata implementata.

### 2.5.5 Controlli

Sono stati implementati controlli backend con PHP per la validazione degli input. Questi controlli comprendono l'uso di espressioni regolari, logica di base per il controllo della lunghezza delle stringhe e confronti dinamici con il database. Tale approccio consente di garantire l'integrità e la sicurezza dei dati inseriti, proteggendo il sistema da possibili vulnerabilità e fornendo una user experience più affidabile.

## 2.6 JavaScript

La gestione delle funzionalità legate all'interazione dell'utente, come l'inserimento dei dati in un modulo, è principalmente gestita lato client attraverso funzioni JavaScript. In caso di inserimento di dati non validi, come ad esempio indirizzi email o numeri di carta di pagamento, vengono visualizzati messaggi di errore appropriati per guidare l'utente, o viene limitato il set di input consentito. Al fine di garantire l'accessibilità, nel caso in cui JavaScript non sia abilitato, viene eseguito un controllo lato server mediante PHP, con la conseguente visualizzazione degli errori individuati.

Ogni logica destinata a notificare l'utente durante l'inserimento di dati errati introduce dinamicamente nel DOM un paragrafo contenente il messaggio di errore corrispondente. Per interagire con il DOM, JavaScript utilizza gli ID degli elementi o specifici hook identificabili dalla presenza di "js-" antecedente alla classe.

Il file non contiene singole funzioni, ma racchiude, mediante strutture condizionali, il contenuto logico della pagina, controllando la presenza di determinati elementi all'interno della stessa. Come per CSS anche per Javascript servito non è main.js ma main.min.js.

## 3 Accessibilità

Il sito è basato sullo standard WCAG 2.2.

### 3.1 Separazione struttura, funzionalità e presentazione

La divisione del sito nelle sue tre componenti chiave - struttura, funzionalità e presentazione - ha apportato significativi vantaggi nell'ambito della gestione e dello sviluppo globale del progetto.

In primo luogo, la separazione della struttura ha consentito di definire chiaramente l'ossatura del sito utilizzando HTML5, sfruttando tag semantici e funzionalità standard. Ciò ha semplificato le modifiche nelle fasi successive del progetto. L'impiego di CSS3 per la presentazione ha permesso di definire in modo distintivo e flessibile l'aspetto visuale del sito, garantendo uno stile coerente e agevolando eventuali aggiornamenti grafici. La separazione della presentazione ha inoltre semplificato le operazioni di manutenzione, consentendo modifiche al layout senza influire sulla logica sottostante.

La terza componente, relativa alle funzionalità, è stata implementata mediante JavaScript, apportando dinamicità e interattività al sito. Questo ha migliorato l'esperienza dell'utente senza richiedere modifiche sostanziali alla struttura o alla presentazione, evidenziando così la flessibilità e l'estensibilità del sistema. Infine, la continua verifica della conformità agli standard del W3C ha garantito un codice ben strutturato e in linea con le migliori pratiche, contribuendo a assicurare la compatibilità del sito con diversi browser e dispositivi.

### 3.2 Navigazione

Per la gestione della navigazione, abbiamo adottato la specifica WAI-ARIA sviluppata dal W3C al fine di garantire l'accessibilità a tutte le categorie di utenti.

Per migliorare la comprensione della struttura del sito, abbiamo implementato i breadcrumb, una lista ordinata di pagine separate da "»". L'ultimo elemento, che rappresenta la pagina corrente, non ha un link e tale ordinamento è corretto dal punto di vista semantico, indicando chiaramente l'ordine di navigazione per gli utenti.

Sono stati implementati alcuni link specifici per saltare determinati contenuti (ad esempio navigazione, faq e biglietti), consentendo agli utenti con gravi menomazioni visive di passare direttamente al successivo.

Per facilitare la navigazione tramite lo scroll, è stato introdotto un pulsante posizionato in basso a destra dello schermo durante lo scroll verso il basso. Se cliccato, questo pulsante esegue uno scroll verso l'alto, facilitando il ritorno all'inizio della pagina. Abbiamo adottato attributi HTML come "lang" per racchiudere parole in lingue diverse dall'italiano, agevolando la corretta lettura da parte degli screen reader. Nei breadcrumb, l'attributo "aria-current" è stato utilizzato per indicare la pagina corrente, mentre all'interno del tag "nav" compare "aria-label" con il valore "breadcrumb" o "website navigation", indicando che il contenuto di quella navigazione rappresenta la posizione dell'utente nel sito o la barra di navigazione principale.

### 3.3 Form

Il processo di validazione del form è gestito da uno script Javascript che, in modo dinamico, aggiunge un nuovo elemento al Document Object Model (DOM). Questo elemento appena introdotto è appositamente contrassegnato con l'attributo "aria-role="alert"", assegnandogli il ruolo di alert. Ciò ha l'effetto di catturare l'attenzione degli screen reader quando viene inserito nella pagina.

L'attributo "aria-role" con il valore "alert" svolge un ruolo fondamentale nell'assicurare che gli utenti che dipendono dagli screen reader siano prontamente informati riguardo agli avvisi correlati alla validazione del modulo. Questa pratica rafforza l'accessibilità generale del sito, garantendo una chiara comunicazione delle informazioni essenziali anche per coloro che utilizzano tecnologie assistive.

Vale la pena notare che eventuali errori inseriti persistono finché l'utente non ritorna al campo interessato per correggerli. In questo momento, lo script Javascript riesamina l'input e provvede a rimuovere o riproporre l'errore di conseguenza. Questa dinamica assicura un processo di correzione fluido e reattivo durante la validazione del modulo.

### 3.4 Attenzioni particolari all'accessibilità

Nel contesto dello sviluppo del sito, ci siamo confrontati frequentemente con elenchi ripetitivi di oggetti, come ad esempio le FAQ o i biglietti acquistabili. La lettura di tali elementi da parte degli screen reader ha evidenziato la necessità di migliorare l'esperienza degli utenti con disabilità visive.

A seguito di un'attenta analisi, abbiamo optato per l'introduzione di un'apposita funzionalità. Simile alla logica applicata nella barra di navigazione, abbiamo implementato un pulsante che agevola gli utenti nello skip delle informazioni non necessarie, consentendo un rapido accesso alle FAQ successive o ai biglietti successivi, preceduti da brevi introduzioni al contenuto.

In particolare, per le FAQ, l'introduzione fornisce una categorizzazione visibile al resto degli utenti attraverso la suddivisione cromatica, facilitando così la navigazione anche per gli utenti privi di disabilità visive. Per i biglietti, il passaggio avviene dopo la lettura di informazioni cruciali quali stazione di partenza, orario di partenza, durata totale, stazione di arrivo e orario di arrivo.

## 4 Verifica

La fase di verifica è stata svolta sia manualmente, che tramite software di validazione. Innanzitutto si è andato a verificare il corretto funzionamento delle singole pagine di visualizzazione per poi passare alla ricerca dei biglietti dopo aver inserito i dati richiesti dal form. Qui inizialmente il problema è stato la creazione di una query adeguata in grado di trovare gli orari sia della tratta in quanto tale, sia della tratta se essa rappresenta una sottotratta di una tratta più estesa.

L'accessibilità rappresenta la capacità di un sito di essere utilizzabile ... sia dai diversi dispositivi. Il testing accessibilità è stato svolto in tutte le fasi, sia di progettazione che di sviluppo e infine di revisione, ed in particolare:

- è stata testata manualmente la navigazione del sito tramite tab;
- è stato testato il contrasto dei colori tramite [colora11y.com](http://colora11y.com);
- sono stati aggiunti gli alt alle immagini nell'adminpage che verranno poi utilizzate come sfondo delle offerte;
- è stato testato l'uso sia su desktop che da mobile;

Per la validazione del sito, sono stati impiegati diversi strumenti, tra cui il W3C HTML5 Validator, Total Validator e il plugin WAVE per Chrome.

Il W3C HTML5 Online Validator è stato utilizzato nelle prime fasi di sviluppo HTML per una verifica preliminare, focalizzandosi sulla rilevazione di eventuali errori semantici o strutturali.

Successivamente, al termine dello sviluppo delle pagine PHP dinamiche, è stato eseguito un controllo più approfondito utilizzando Total Validator. Questo strumento ha permesso di individuare e correggere piccoli errori e omissioni, nonché di aggiungere attributi ARIA supplementari per migliorare l'accessibilità complessiva delle pagine. Gli eventuali errori rimasti, principalmente riconducibili a verifica ortografica, sono stati esclusi dalla considerazione.

Infine, il plugin WAVE di Chrome è stato impiegato per un controllo esteso dei colori utilizzati nelle diverse pagine, con l'obiettivo di massimizzare il contrasto e garantire un'esperienza di navigazione accessibile.

Al termine delle verifiche di validazione, è stata eseguita un'analisi delle performance del sito tramite il tool Lighthouse integrato di Chrome. I risultati complessivi sono positivi, ottenendo sempre il punteggio massimo per aspetti quali accessibilità, SEO e best practices. Riguardo alle performance in generale, il sito, essendo ottimizzato per dispositivi mobili, tende a presentare prestazioni superiori nella visualizzazione su schermi più piccoli, ad eccezione di alcune pagine quali home e sezione amministrativa. Si è riscontrato però che le performance decadono per via dell'utilizzo di font importati dinamicamente a runtime che comportano un eccessivo Cumulative Layout Shift della pagina. È possibile ovviare a tale problema integrando i font (file con estensione .ttf) presenti all'interno della cartella css/fonts. Tale azione non è stata eseguita ma solo testata per via di possibili problemi di compatibilità con dispositivi datati.

## 5 Suddivisione del lavoro

Il gruppo ha inizialmente affrontato la prima fase di progettazione insieme, tramite incontri in presenza dopo le lezioni oppure incontri remoti tramite Discord.

In questa prima fase tutti i componenti sono stati coinvolti nella scelta del soggetto del sito, l'idea generale di selezionare un sito di trasporti ha subito convinto tutti; inizialmente ci siamo confrontati per dare o un tema piu' fantasioso, trasporto con navicelle nello spazio, o un tema piu' concreto, tema effettivamente scelto.

La scelta e' avvenuta considerando la maggior reperibilita' di immagini e di informazioni, una maggior semplicita' nell'immaginare il sito e una maggior serietà con l'ambiente universitario di valutazione.

### 5.1 Web Design

Il primo compito del gruppo e' stato quello di immaginare il design del sito.

Ogni componente del gruppo ha proposto un suo design, andando ad illustrare le varie pagine necessarie, il relativo contenuto ed il relativo stile grafico.

Successivamente sono state scelte o unite le idee migliore ed e' stata creata una prima bozza grafica del sito, condivisa ed approvata da tutti.

### 5.2 HTML e SEO

Definito il design del sito si e' iniziato il lavoro sulla struttura del sito, quindi tramite repository GitHub condiviso abbiamo cominciato a programmare le pagine .html.

Si e' deciso di suddividere equamente il numero di pagine fra i componenti, due pagine a testa, e di darci una scadeza di una settimana per completare il lavoro. Alla fine della settimana l'effettiva programmazione in HTML e' stata svolta dai componenti Carraro Alessandro e Fabbian Riccardo.

La prima versione di struttura del sito era completa, ma col passare del tempo, si e' evoluta per conformita' fra le pagine e per adeguarsi alle nostre esigenze con gli altri linguaggi necessari per il sito (CSS, JS e PHP).

### 5.3 CSS e JavaScript

La fase successiva e' stata l'implementazione dei linguaggi CSS e JavaScript, per garantire accessibilita' e esperienza utente.

Il design, pagina per pagina, e' stato valutato dai componenti Carraro Alessandro e Fabbian Riccardo. La gran parte del codice CSS e' stato scritto da Riccardo, vista la pregressa esperienza, e controllato da entrambi i due componenti. Passarella Alessandro ha contribuito fornendo le immagine del sito ed offrendo consigli estetici.

La parte di esperienza utente, in particolare la dinamicita' del sito, e' stata svolta completamente da Fabbian Riccardo, sia la programmazione in JavaScript e che la sua verifica.

## **5.4 PHP**

Successivamente si è passati a PHP e all'integrazione di un database nel sito. Il lavoro in PHP è stato svolto da Chilese Elena e Fabbian Riccardo, andando entrambi a lavorare sia lato server che lato client.

Il database è stato creato da Fabbian Riccardo, migliorato e popolato da Chilese Elena stessa con l'aiuto di Carraro Alessandro.

## **5.5 Verifiche**

Infine abbiamo dedicato un paio di giorni alle verifiche del sito: prove sul server di Laboratorio e controlli tramite validatori online. La parte di verifica sul server di laboratorio è stata svolta da Fabbian Riccardo e Carraro Alessandro.

Le verifiche riguardanti il funzionamento dei form, delle espressioni regolari e del corretto inserimento/cancellazione dei dati nel database oltre al controllo generale dell'utilizzo del sito sono stati svolti da Chilese Elena.

## **5.6 Conclusioni**

Il gruppo si è trovato, per un serie di impegni e di incompatibilità di orari universitari e personali, a lavorare per la maggior parte in modalità asincrona.

Questa situazione ha portato ad una scarsa collaborazione ed ad una suddivisione del lavoro per linguaggi, andando a confrontarci sempre solo fra alcune componenti del gruppo, per poi comunicare a tutto il gruppo il progresso svolto.

## 6 Difficoltà incontrate

Durante il processo di sviluppo, le sfide più significative sono emerse nella creazione dinamica degli elementi HTML mediante PHP, focalizzandoci in particolare sulla generazione e ordinamento dinamico dei biglietti. L'obiettivo era organizzare tali biglietti in ordine temporale e attribuire loro id per agevolare la creazione di una catena navigabile mediante la funzionalità di skip.

La realizzazione del codice Javascript ha rappresentato un impegno notevole, principalmente in termini di tempo, ma in termini di complessità non ha presentato sfide eccessive. Analogamente, la scrittura del codice HTML si è dimostrata relativamente agevole. Tuttavia, è emerso che il CSS ha richiesto una quantità di tempo significativa oltre le aspettative, con un periodo di sviluppo che si è esteso su diverse settimane.