



# ESU ADVISOR

## Relazione del Progetto

2025-06-18

**Sede** Università degli Studi di Padova  
**Facoltà** Informatica  
**Corso** Tecnologie Web  
**Anno** 2024/25

**Indirizzo web:** <http://tecweb.studenti.math.unipd.it/~mvasquez/ESU-ADVISOR/>

### A cura di

<b>Membro</b>	<b>Matricola</b>
Giacomo Loat	2077677
Giulio Bottacin	2042340
Malik Giafar Mohamed	2075543
Manuel Felipe Vasquez	2076425

**Credenziali utente:** user / user

**Email referente:** manulfelipe.vasquez@studenti.unipd.it

## Indice

1. Introduzione .....	4
1.1. Obiettivo del Progetto .....	4
1.2. Composizione del Team .....	4
1.3. Analisi degli Utenti Target .....	4
1.3.1. Caratteristiche Primarie degli Utenti .....	4
1.3.2. Possibili Ricerche sui Motori di Ricerca .....	4
1.4. Suddivisione del Lavoro .....	4
1.4.1. Componente di Contenuto (Giacomo Loat) .....	4
1.4.2. Componente di Presentazione (Giulio Bottacin) .....	4
1.4.3. Componente di Comportamento (Manuel Felipe Vasquez) .....	5
1.4.4. Struttura Database (Malik Giafar Mohamed) .....	5
1.5. Metodologia di Collaborazione .....	5
1.5.1. Strumenti e Workflow .....	5
2. Progettazione .....	5
2.1. Schema organizzativo .....	5
2.2. Schema a tre pannelli .....	5
2.3. Convenzioni interne .....	6
2.4. Layout .....	6
2.5. Funzionalità .....	7
2.5.1. Autenticazione .....	7
2.5.2. Visualizzazione piatti del giorno .....	7
2.5.3. Recensioni .....	7
2.5.4. Preferenze alimentari .....	7
2.5.5. Personalizzazione impostazioni .....	7
2.6. Note .....	7
3. Implementazione .....	8
3.1. Struttura delle Directory .....	8
3.2. Backend PHP .....	8
3.2.1. Pattern Architettuale MVC .....	8
3.2.1.1. Model Layer .....	8
3.2.1.2. View Layer .....	9
3.2.1.3. Controller Layer .....	9
3.2.2. Tecnica di Templating .....	9
3.3. Stili CSS .....	10
3.4. Scripts Javascript e Validazione dell'input .....	10
3.5. Gestione degli errori di navigazione .....	10
3.6. Database Design .....	10
3.6.1. Schema Relazionale Normalizzato .....	10
3.7. SEO e Performance .....	12
4. Accessibilità .....	12
4.1. Personalizzazione Accessibilità .....	12
4.2. Navigazione accessibile .....	12
5. Testing e Validazione .....	13
5.1. Ambiente di Sviluppo .....	13
5.2. Strumenti .....	13
5.2.1. Strumenti per L'accessibilità .....	13
5.2.2. Validazione HTML .....	13
5.2.3. Validazione CSS .....	13
5.2.4. Controllo qualità PHP .....	14

---

5.3. Browser Testing ..... 14

6. Istruzioni per l’installazione del progetto ..... 14

## 1. Introduzione

### 1.1. Obiettivo del Progetto

ESU-Advisor è una piattaforma web dedicata agli studenti universitari di Padova che permette di consultare i menu delle mense universitarie dell'ESU (Azienda Regionale per il Diritto allo Studio Universitario) e di lasciare recensioni sui piatti disponibili. Il progetto nasce dall'esigenza di fornire agli studenti uno strumento semplice e accessibile per valutare e condividere informazioni sui pasti offerti nelle diverse mense universitarie.

### 1.2. Composizione del Team

Il progetto è stato realizzato da un gruppo di quattro studenti con competenze complementari:

Nome	Cognome	Matricola	Competenze Principali
Giacomo	Loat	2077677	HTML5, Design UI/UX, Struttura Frontend
Giulio	Bottacin	2042340	CSS3, Responsive Design, Accessibilità
Malik	Giafar Mohamed	2075543	Database Design, SQL, Architettura Dati
Manuel Felipe	Vasquez	2076425	PHP, Backend Development

Table 1: Membri del team e relativi ruoli

### 1.3. Analisi degli Utenti Target

#### 1.3.1. Caratteristiche Primarie degli Utenti

Il pubblico di riferimento è costituito principalmente da studenti universitari di Padova, con un'età compresa tra 18 e 30 anni, che fanno un uso frequente di dispositivi mobili. Si è tenuto conto anche delle loro possibili esigenze alimentari specifiche (come allergie o intolleranze).

#### 1.3.2. Possibili Ricerche sui Motori di Ricerca

- “mense universitarie Padova”;
- “menu mensa ESU oggi”;
- “recensioni cibo mense Padova”;
- “orari mense università Padova”;
- “piatti del giorno mense universitarie”.

### 1.4. Suddivisione del Lavoro

#### 1.4.1. Componente di Contenuto (Giacomo Loat)

- Sviluppo della struttura HTML5;
- Implementazione dei template delle pagine;
- Integrazione dei contenuti dinamici;
- Ottimizzazione per SEO.

#### 1.4.2. Componente di Presentazione (Giulio Bottacin)

- Implementazione dello stile CSS;
- Creazione delle componenti UI riutilizzabili;
- Implementazione del design responsive;
- Sviluppo delle feature di accessibilità (WCAG 2.0);
- Gestione dei temi chiaro/scuro;
- Supporto per utenti con dislessia.

### 1.4.3. Componente di Comportamento (Manuel Felipe Vasquez)

- Sviluppo della logica JavaScript client-side;
- Implementazione delle validazioni form;
- Architettura MVC backend in PHP;
- Gestione dell'autenticazione e autorizzazione;
- Sviluppo delle API REST.

### 1.4.4. Struttura Database (Malik Giafar Mohamed)

- Design dello schema del database normalizzato;
- Implementazione delle relazioni tra entità;
- Ottimizzazione delle query;
- Implementazione del sistema di gestione allergeni.

## 1.5. Metodologia di Collaborazione

### 1.5.1. Strumenti e Workflow

- **Git** per il controllo versione distribuito;
- **GitHub** per la gestione del repository e code review;
- **Docker** per lo sviluppo containerizzato senza prerequisiti di sistema;
- **Visual Studio Code** come IDE condiviso con Live Share.

## 2. Progettazione

### 2.1. Schema organizzativo

Poiché la struttura del sito web varia solo nel contenuto del tag `<main>`, mantenendo pressoché invariata la struttura composta da header e sidebar. Si può dire che lo schema organizzativo ambiguo scelto per il raggruppamento dei link alle pagine principali si applica a tutto il sito web.

Nel sito web infatti, sarà sempre presente una sidebar a sinistra, che contiene i link alle pagine principali del sito, tra le quali è possibile trovare:

- **Home:** pagina principale con i piatti del giorno e le recensioni;
- **Review:** pagina per la visualizzazione e l'inserimento di recensioni sui piatti;
- **Profilo:** pagina del profilo utente per visualizzare le informazioni personali e le recensioni;
- **Impostazioni:** pagina per la gestione di preferenze alimentari, stilistiche del sito, dello username e per la cancellazione dell'account.

### 2.2. Schema a tre pannelli

Per ridurre il fenomeno del disorientamento dell'utente, è stato scelto di adottare lo schema a tre pannelli, adattando il design del sito per rispondere a queste tre domande:

- **Dove sono?**
  - L'utente può orientarsi grazie all'evidenziazione della pagina corrente nella sidebar, oppure tramite la breadcrumb presente nell'header.
- **Dove posso andare?**
  - L'utente può navigare tra le pagine del sito tramite la sidebar, che mostra le sezioni principali del sito web.

Sono inoltre presenti dei link all'interno del contenuto del sito che portano alla visualizzazione dei piatti del giorno, rispondendo alla domanda ***Cosa posso fare?***.

- **Di cosa si tratta?**

- Ogni azione intraprendibile è descritta in modo chiaro e conciso, con l'eventuale uso di icone e testi esplicativi per facilitare la comprensione. Alcuni esempi sono i link "vedi le recensioni" presenti nelle schede dei piatti della homepage.

Il sito inoltre risponde alla domanda **Dove posso trovare altre informazioni?** mettendole a disposizione nella homepage o esplicitandole in dei link.

## 2.3. Convenzioni interne

Per garantire coerenza, usabilità e orientamento all'utente, sono state adottate delle convenzioni interne, che si riflettono in tutte le pagine dell'applicazione:

- **Stile bottoni:** sono stati definiti tre stili di bottone al fine di indicare a prima vista i loro ruoli:

- Bottone principale: sempre blu con testo bianco;
- Bottone secondario: testo blu su sfondo chiaro o scuro a seconda del tema e bordo blu;
- Bottoni di pericolo: sempre arancioni con testo bianco.

Tutti gli stili rispettano i corretti contrasti tra testo, corpo del pulsante e sfondo per garantire accessibilità.

- **Stile dei link:** lo stile dei link è stato mantenuto molto simile a quello standard ma la tonalità dei colori (blu per i link non visitati, viola per i link visitati) è stata adattata alla palette cromatica del sito per ottenere una buona resa estetica e corretti contrasti tra i colori;
- **Struttura a Card:** le varie sezioni delle pagine e i loro elementi sono illustrati con uno stile ripetuto, al fine di rendere più semplice e intuitiva la struttura secondo la quale i contenuti sono disposti all'interno del sito;
- **Navigazione adattiva:** la disposizione degli elementi di navigazione varia in base al dispositivo utilizzato. Su dispositivi mobili, il menu principale è posizionato in un footer fisso in basso, mentre i pulsanti di accesso, registrazione e logout sono collocati in alto a destra. Su desktop, invece, tutte le funzioni di navigazione sono raccolte in una sidebar laterale a sinistra, ottimizzando lo spazio e rendendo più intuitivo l'accesso alle diverse sezioni;
- **Assenza di link circolari:** nonostante spesso l'icona del titolo sia un link alla home, per evitare di creare un link circolare questa caratteristica non è presente. La home è comunque sempre raggiungibile tramite il nav;
- **Istruzioni di compilazione nei form:** dove necessario, i campi dei form sono preceduti da istruzioni di compilazione che spiegano cosa inserire e indicano i requisiti del campo;
- **Messaggi di errore nei form:** per ridurre la probabilità che un utente debba compilare e inviare un modulo più volte, i messaggi di errore vengono mostrati sotto i rispettivi campi già con l'evento "blur" (fatta eccezione per gli errori che devono essere verificati lato server).

## 2.4. Layout

In base a quanto dedotto nella sezione di analisi delle utenze, è stato deciso adottare un **approccio mobile-first** nella progettazione dell'interfaccia utente. Questo non significa soltanto aver adottato un **layout responsive**, ma significa che il design e lo sviluppo dell'applicazione sono stati inizialmente focalizzati sulle esigenze degli utenti che utilizzano dispositivi mobili, siccome questi saranno coloro che utilizzeranno maggiormente il sito web.

## 2.5. Funzionalità

Il sito web offre le seguenti funzionalità principali:

### 2.5.1. Autenticazione

Gli utenti possono registrarsi e accedere al sito per usufruire di funzionalità avanzate, come la possibilità di lasciare recensioni sui piatti e personalizzare le proprie preferenze alimentari.

### 2.5.2. Visualizzazione piatti del giorno

Gli utenti possono visualizzare i piatti del giorno delle mense universitarie, con dettagli su ingredienti, allergeni e recensioni degli utenti.

### 2.5.3. Recensioni

Gli utenti registrati possono lasciare recensioni sui piatti, valutandoli e fornendo commenti utili ad altri studenti.

### 2.5.4. Preferenze alimentari

Gli utenti possono impostare preferenze alimentari specifiche, come allergie o intolleranze, per ricevere avvisi sui piatti che potrebbero contenere ingredienti indesiderati.

### 2.5.5. Personalizzazione impostazioni

Nella pagina delle impostazioni, gli utenti sia autenticati che non possono personalizzare la propria esperienza di navigazione, in particolare, è possibile:

- Attivare la **modalità scura** per ridurre l'affaticamento visivo, se il dispositivo o il browser ha la preferenza per il tema scuro il sito apparirà in tale modo in automatico;
- Modificare le **dimensioni del testo** per una lettura più confortevole;
- Cambiare il font in uno specifico font per la **dislessia** per migliorare la leggibilità del testo.

Il font per la dislessia scelto è stato “OpenDyslexic”, un font open source progettato specificamente per facilitare la lettura dei contenuti testuali presenti nel sito.

Inoltre, sempre nella stessa pagina è possibile impostare una mensa preferita, che verrà mostrata subito al login senza necessità di selezionarla ogni volta. Sarà anche possibile impostare delle preferenze alimentari in modo che possano essere segnalati i piatti che contengono un insieme di allergeni specificati dagli utenti.

## 2.6. Note

Le immagini presenti nel sito sono state prese dagli smartphone dei membri del gruppo e dei loro conoscenti, mentre in alcuni casi, siccome non erano disponibili delle foto dove venisse mostrato il piatto singolo, abbiamo ricorso alla generazione delle foto con l'intelligenza artificiale.

Inoltre, per quanto riguarda i menu, la disposizione dei piatti nel sito non rispecchia quella delle mense reali.

## 3. Implementazione

### 3.1. Struttura delle Directory

Il progetto è organizzato con una netta separazione tra i file accessibili pubblicamente (public-html), che includono pagine PHP, stili, script e risorse, e il codice sorgente del backend (src), che contiene i file MVC e di configurazione, non esposti direttamente sul web.

```
ESU-ADVISOR/
├── docs/                                # Documentazione progetto
├── public-html/                         # File esposti pubblicamente accessibili
│   ├── styles/                         # Fogli di stile modulari
│   ├── scripts/                       # Script client-side
│   ├── images/                        # Risorse grafiche
│   ├── fonts/                         # Cartella per i font
│   ├── *.php                          # Pagine dell'applicazione
│   ├── .htaccess                      # File per la risoluzione delle path del server APACHE
│   ├── robots.txt                    # File per indicare ai crawler le pagine che possono visitare
│   └── index.php                      # Entry point applicazione
├── src/                                # File utilizzati lato backend non ottenibili dal webserver
│   ├── controllers/                   # Controller MVC
│   ├── models/                       # Model MVC
│   ├── templates/                    # File HTML puri contenenti la struttura della pagina
│   ├── views/                        # View MVC
│   ├── config.php                    # File per le variabili d'ambiente per l'accesso al DB
│   └── session_init.php              # File per le impostazioni della sessione di PHP
├── db.sql                             # Schema database e query di inserimento dati iniziali
└── README.md
```

### 3.2. Backend PHP

#### 3.2.1. Pattern Architettuale MVC

Il progetto implementa rigorosamente il pattern Model-View-Controller (MVC) per garantire modularità e manutenibilità, dove:

##### 3.2.1.1. Model Layer

Gestisce la logica dei dati e le interazioni con il database:

- **DatabaseModel**: gestione dell'accesso al database;
- **MenseModel**: informazioni sulle mense e orari;
- **MenuModel**: informazioni sui menu delle mense e i loro piatti;
- **PiattoModel**: catalogo piatti con categorie e allergeni;
- **PreferenzeUtenteModel**: personalizzazione esperienza utente;
- **RecensioneModel**: sistema di valutazioni e commenti;
- **UserModel**: gestione utenti e autenticazione.



### 3.2.1.2. View Layer

Si occupa della presentazione dei dati attraverso un sistema di template che separa il contenuto dalla formattazione:

- **BaseView**: classe di base da cui ogni view eredita le funzionalità;
- **ErrorView**: gestione della pagina di errore;
- **IndexView**: homepage con selezione mense;
- **LoginView** / **RegisterView**: autenticazione;
- **PiattoView**: dettaglio piatto e recensioni;
- **ProfileView**: dettaglio profilo;
- **ReviewView**: dettaglio di una recensione;
- **ReviewEditView**: dettaglio piatto e recensioni;
- **SettingsView**: gestione preferenze e accessibilità.

### 3.2.1.3. Controller Layer

Gestisce le richieste dell'utente, dialoga con il Model e seleziona la View appropriata da mostrare con i dati appropriati:

- **BaseController**: classe di base da cui ogni controller eredita le funzionalità;
- **ErrorController**: gestisce la richiesta errata;
- **IndexController**: Logica homepage e selezione mense;
- **LoginController** / **RegisterController**: autenticazione;
- **PiattoController**: gestione visualizzazione piatti;
- **ProfileController**: gestione informazioni utente;
- **ReviewController**: gestione recensioni;
- **ReviewEditController**: gestione modifiche recensioni;
- **SettingsController**: configurazione utente.

### 3.2.2. Tecnica di Templating

Per separare la struttura dal comportamento abbiamo fatto uso del tag `<template>` fornito da HTML per evitare di inserire snippet di PHP all'interno della struttura del documento. Per ogni tag viene aggiunto un id univoco con il formato **nome-contenuto-template**. Poi per inserire i dati relativi si è fatto uso della funzione `replaceTemplateContent($dom, $templateId, $newContent)` presente nel file `Utils.php`, dove:

- **\$dom** contiene il testo dell'intero file html parsabile contenente i tag `<template>`;
- **\$templateId** l'id del tag `<template>` che si vuole sostituire;
- **\$newContent** il contenuto in formato HTML da inserire.

Per effettuare il parsing del file e ottenere la posizione del tag è stata utilizzata la classe **DOMDocument** di PHP, che fornisce una serie di funzioni per gestire e manipolare i file HTML. Nel nostro caso specifico utilizziamo la classe per controllare che il contenuto inserito sia formattato correttamente con caratteri utf-8 e che rispetti la sintassi HTML.

Questa funzione viene utilizzata spesso per gestire tutte le varie componenti del sito che richiedono dati provenienti dal DB o che richiedono di essere processate, ad esempio viene utilizzata per modularizzare la struttura della pagina, in particolare ogni View eredita dalla classe **BaseView** che espone il metodo **render()** che di default inserisce le componenti **header**, **footer** e **sidebar** in ogni file html che viene richiesto affinché esistano i tag template corrispondenti. Questo permette di separare la struttura di componenti ripetute in file unici favorendo la modularità del sito e permettendo a più componenti del gruppo di lavorare sul progetto evitando il più possibile di avere conflitti nelle modifiche.

### 3.3. Stili CSS

Per gestire gli stili che abbiamo applicato alla pagina sono state definite delle variabili globali per rendere coerenti le proprietà dei vari elementi, come colore dei bottoni, il testo, background di elementi ripetuti, spaziature, font, etc..., inoltre vi è presente la variazione delle stesse variabili per il tema scuro che verranno applicate in base ad una classe che verrà aggiunta al nodo root `<html>` per favorire l'uso delle stesse classi per entrambi i temi

### 3.4. Scripts Javascript e Validazione dell'input

Sono stati utilizzati script javascript per gestire il comportamento del sito, in particolare:

- **Validazione dell'input dei form lato client:** non strettamente necessaria in quanto vi è l'equivalente validazione a lato server per motivi di sicurezza, ma aiuta l'utente ad inserire correttamente i dati richiesti prima di ritrovarsi con una pagina d'errore;
- **Funzionalità di ricerca dei piatti nel menù:** per favorire la ricerca di alimenti o allergeni tra i piatti della mensa attualmente selezionata in maniera veloce, vi è implementata una barra di ricerca in cima alla sezione del menu del giorno, che filtrerà i piatti presenti nascondendo quelli che non corrispondono ai termini di ricerca;
- **Suggerimenti piatti durante la recensione:** per facilitare il processo di recensione dei piatti, una volta selezionata la mensa, al momento della scrittura del nome del piatto, verrà fornita una lista dei piatti di cui la mensa dispone al fine di evitare problemi di spelling del nome del piatto. Questa scelta è preferibile rispetto ad avere soltanto una lista, come nel caso della selezione della mensa, per rendere più semplice e veloce l'inserimento del nome del piatto qualora il menù della mensa contenesse numerose opzioni.

Il sito web inoltre può funzionare correttamente pur non avendo abilitato javascript nel browser. In tal caso verrà mostrato un avviso che inviterà l'utente ad abilitarlo.

### 3.5. Gestione degli errori di navigazione

Nel caso si volesse fare accesso a link che puntano a risorse inesistenti or malformati, come ad esempio `http://<server>/index.php/wrong/path` o `http://<server>/wrong-filename.php`, sono state definite delle regole nel file `public_html/.htaccess` per gestire queste condizioni. In particolare se un utente volesse visitare una pagina che non esiste o che non può visitare senza un account (ad esempio `review.php` e `profile.php`), essi verrebbero reindirizzati alla pagina `error.php` dove in base alle condizioni del reindirizzamento verranno indicate informazioni utili all'utente per comprendere cosa è appena successo e ad indicargli cosa dovrebbe fare, nell'esempio corrente gli indicherà che "È necessario effettuare l'accesso per visualizzare questa pagina", con sotto i pulsanti per effettuare l'accesso o la registrazione se l'utente non ha già un account.

Queste regole sono state scritte per supportare deployment del sito web sia nel caso in cui sia presente nel root del server (`http://<server>/index.php`) o che si trovi in una sottocartella, come nel caso del server universitario (`http://<server>/<name>/index.php`).

### 3.6. Database Design

#### 3.6.1. Schema Relazionale Normalizzato

Lo schema è normalizzato fino alla Terza Forma Normale (3NF) per minimizzare la ridondanza dei dati e prevenire anomalie di inserimento, aggiornamento e cancellazione.

Tabella	Scopo	Relazioni
mensa	Informazioni mense	1:N con orarioapertura, menu
piatto	Catalogo piatti	1:N con recensione, menu, piatto_foto
utente	Gestione utenti	1:N con recensione, 1:1 con preferenze_utente
orarioapertura	Orari mense	-
menu	Associazione piatti-mense	N:N tra piatto e mensa
recensione	Sistema valutazioni	N:1 con utente, piatto
piatto_foto	Associazione piatti-foto	-
piatto_allergeni	Gestione allergeni	N:N tra piatto e allergeni
preferenze_utente	Personalizzazione	1:1 con utente
allergeni_utente	Personalizzazione	1:1 con utente

Table 2: Tabelle principali del database

Lo schema si articola attorno a tre entità fondamentali: le **mense**, i **piatti** e gli **utenti**.

- **mensa:** questa tabella contiene le informazioni anagrafiche di ogni mensa universitaria. Il nome della mensa funge da chiave primaria e identificatore univoco. Gli altri campi includono l'indirizzo, il numero di telefono e un link esterno a Google Maps per la localizzazione geografica;
- **piatto:** memorizza il catalogo di tutti i piatti che possono essere serviti. Ogni piatto è identificato univocamente dal suo nome (chiave primaria) e appartiene a una categoria predefinita ("Primo", "Secondo", "Contorno"). Contiene inoltre una descrizione testuale;
- **utente:** gestisce i dati degli utenti registrati alla piattaforma. Ogni utente ha un id numerico auto-incrementale come chiave primaria e uno username univoco. La tabella memorizza anche la password (che verrà sottoposta ad hashing prima dell'inserimento) e la data di nascita.

Le entità principali sono collegate tra loro e arricchite di dettagli attraverso una serie di tabelle satellite.

- **orarioapertura:** definisce gli orari di apertura settimanali per ciascuna mensa, creando una relazione uno-a-molti con la tabella mensa. Un vincolo di CHECK assicura che il giornoSettimana sia un valore compreso tra 1 e 7;
- **menu:** si tratta di una tabella di collegamento che implementa una relazione molti-a-molti tra piatto e mensa, specificando quali piatti sono disponibili in quali mense;
- **recensione:** è la tabella centrale per le interazioni degli utenti. Stabilisce una relazione tra un utente, un piatto e una mensa. Un utente può lasciare una sola recensione per un dato piatto in una specifica mensa. La recensione include un voto (da 1 a 5), una descrizione testuale opzionale e la data di inserimento. La chiave esterna composta (piatto, mensa) garantisce che una recensione possa essere lasciata solo per un piatto effettivamente presente nel menu di quella mensa.

Infine, sono presenti tabelle per gestire informazioni aggiuntive sui piatti e le preferenze degli utenti.

- **piatto\_foto** e **piatto\_allergeni:** queste due tabelle aggiungono dettagli ai piatti. La prima gestisce l'associazione uno-a-molti tra un piatto e le sue foto (salvando il percorso all'immagine nel db). La seconda definisce una relazione molti-a-molti per associare a ogni piatto uno o più allergene da una lista predefinita conforme alla normativa europea;
- **preferenze\_utente** e **allergeni\_utente:** queste tabelle sono dedicate alla personalizzazione dell'esperienza utente. preferenze\_utente ha una relazione uno-a-uno con utente e memorizza impostazioni di accessibilità come la dimensione\_testo, l'uso del font per la dislessia (modifica\_font), il modifica\_tema visivo e la mensa\_preferita. allergeni\_utente, invece, permette agli utenti di registrare i propri allergeni personali in una relazione molti-a-molti, per ricevere avvisi mirati.

### 3.7. SEO e Performance

Al fine di ottimizzare il posizionamento del sito nei motori di ricerca, sono state implementate le seguenti strategie:

- **Definizione di Meta tags:** sono stati creati meta tag specifici per ogni pagina, inclusi titolo e descrizione per ogni pagina del sito web;
- **Sitemap XML:** è stata creata una sitemap per facilitare l'indicizzazione da parte dei motori di ricerca, che include tutte le pagine principali del sito. La sitemap viene generata automaticamente dallo script `sitemap.php` presente nella cartella `public_html/`, e viene aggiornata includendo ogni pagina del sito web, al momento dell'accesso;
- **URLs SEO-friendly:** ogni URL del sito web è stato progettato per essere descrittivo e contenere parole chiave pertinenti, evitando di utilizzare parametri complessi o identificatori numerici. Ad esempio, l'URL per la pagina di recensioni di un piatto specifico è strutturato come `http://<server>/review.php?piatto=<nome-piatto>&mensa=<nome-mensa>`, dove `<nome-piatto>` e `<nome-mensa>` sono i nomi dei piatti e delle mense rispettivamente.

## 4. Accessibilità

### 4.1. Personalizzazione Accessibilità

Le impostazioni di accessibilità possono essere gestite sia dagli utenti registrati che non. Esse verranno salvate all'interno della sessione di PHP, e nel caso tali fossero salvate per un utente registrato, una volta effettuato l'accesso esse verranno applicate automaticamente. Le impostazioni disponibili sono:

- **Dimensioni testo:** piccolo, medio, grande;
- **Font per dislessia:** OpenDyslexic font selezionabile dalle impostazioni;
- **Temi:** chiaro, scuro, sistema (prende l'impostazione dal tema del browser), questo viene caricato prima che l'utente possa visualizzare la pagina per evitare flashing.

### 4.2. Navigazione accessibile

Vengono riportati qui sotto altri aspetti che rendono questo sito accessibile a tutte le tipologie di utenti:

- **Navigazione da Tastiera:** il sito è interamente utilizzabile tramite la tastiera. Gli elementi non interagibili della pagina (come la mensa e il nome del piatto durante la modifica di una recensione) vengono nascosti. Inoltre sono presenti dei link di salto rapido ("Vai al contenuto" e "Vai alla navigazione"), per facilitare la navigazione e migliorare l'esperienza per chi utilizza uno screen reader;
- **Supporto per gli screen reader:** sono state utilizzate classi CSS per migliorare l'accessibilità del sito per gli utenti che necessitano l'uso di uno screen reader, dove tali applicano descrizioni e informazioni utili soprattutto riguardo alle immagini dei piatti presenti nel sito e le stelle di valutazione nella pagina per la review, dove ad ogni stella vi è indicata il numero e il significato della valutazione;
- **Tabelle accessibili:** ogni tabella del sito web è stata resa accessibile tramite:
  - **Attributi di scoping:** Ogni tabella ha un'intestazione chiara e descrittiva per ogni colonna, che aiuta gli utenti a comprendere il contenuto della tabella;
  - **Tag di accessibilità:** lo scopo principale delle tabelle nel sito è quello di mostrare gli orari di apertura delle mense di Padova, quindi sono stati utilizzati i tag `<abbr>` per abbreviare i nomi dei giorni della settimana, ad esempio "Lun" per "Lunedì", "Mar" per "Martedì", etc. Questo aiuta gli utenti a comprendere rapidamente il significato delle abbreviazioni. Inoltre, è stato utilizzato il tag `<time>` per indicare gli orari di apertura e chiusura delle mense, in modo che gli screen reader possano leggere correttamente le informazioni temporali;
  - **Descrizione accessibile e caption:** ogni tabella ha una caption che descrive il suo contenuto e scopo, migliorando l'accessibilità per gli utenti di screen reader. È presente inoltre l'attributo `aria-`

describedby per fornire una descrizione aggiuntiva della tabella, che viene letta dagli screen reader per fornire ulteriori informazioni sul contenuto della tabella.

- **Assenza di Link Circolari:** non sono presenti link circolari o che portano a pagine senza contenuto utile, per evitare confusione e disorientamento dell'utente;
- **Form accessibili:** i form di inserimento dati sono stati progettati per essere accessibili, con etichette chiare e descrittive per ogni campo. Sono stati utilizzati attributi ARIA per migliorare l'accessibilità dei campi di input e dei pulsanti di invio;
- **Utilizzo di breadcrumbs:** è stata implementata una breadcrumb che mostra il percorso di navigazione dell'utente all'interno del sito, facilitando l'orientamento e la comprensione della struttura delle informazioni;
- **Contrasto e Colori:** i colori utilizzati per il tema chiaro/scuro sono stati scelti al fine di garantire un adeguato rapporto di contrasto tra i vari elementi presenti nel sito al fine di rispettare almeno le normative dello standard **WCAG 2.1 AA**.

## 5. Testing e Validazione

Il sito durante la fase di sviluppo e collaudo finale è stato sottoposto a numerosi test che ci hanno permesso di valutare alcune decisioni prese e trovare parti che necessitavano correzioni o migliorie.

### 5.1. Ambiente di Sviluppo

Per avere un ambiente di sviluppo unico tra i membri del gruppo è stato utilizzato **Docker** per eseguire il codice sviluppato in un ambiente il più simile a quello del server del laboratorio, in particolare:

- Apache con PHP 8.2 (stessa del server universitario, diversamente da quanto inizialmente scritto nella consegna del progetto detta a lezione, ossia 8.1);
- MariaDB 10.6.7;
- phpmyadmin come strumento di utility per visualizzare il database.

Periodicamente durante il periodo di sviluppo del progetto il gruppo ha provato il sito anche sul server universitario per confermare la sua corretta funzionalità

### 5.2. Strumenti

Qui sono riportati vari strumenti utilizzati al fine di convalidare sia la correttezza del codice utilizzato che la sua accessibilità:

#### 5.2.1. Strumenti per L'accessibilità

- **WAVE** (Web Accessibility Evaluation Tool);
- **axe-core** accessibility checker;
- **Screen reader testing** con NVDA;
- **Keyboard navigation testing** completo;
- **Total Validator** per individuare possibili problemi su ogni pagina del sito;
- **Lighthouse** per convalidare la performance, accessibilità, best practice e SEO;
- **Suite d'accessibilità di Chrome & Firefox** per assicurarsi che i browser riconoscessero le nostre misure al fine di assicurarsi l'utilizzo del sito da qualsiasi categoria d'utente.

#### 5.2.2. Validazione HTML

- **W3C Markup Validator:** 0 errori, 0 warning;
- **HTML5 semantic validation:** Markup semanticamente corretto;
- **XML compliance:** sintassi XML valida.

#### 5.2.3. Validazione CSS

- **W3C CSS Validator:** CSS3 valido;
- **Cross-browser compatibility:** Chrome, Firefox, Safari, Edge;

- **Mobile compatibility:** Android, iOS.

#### 5.2.4. Controllo qualità PHP

- **PSR-12** coding standards compliance;
- **Static analysis** con PHPStan;
- **Security scan** con RIPS;
- **Performance profiling** con Xdebug.

### 5.3. Browser Testing

Il sito è stato testato sui seguenti browser/dispositivi:

- Chrome 120+ (desktop/mobile);
- Firefox 121+ (desktop/mobile);
- Safari 17+ (desktop/mobile);
- Edge 120+ (desktop).

## 6. Istruzioni per l'installazione del progetto

Per installare il progetto in locale in un ambiente simile a quello del laboratorio (**Apache** con **PHP 8.2** e **MariaDB**):

1. Copiare la cartella **public\_html** nella corrispettiva cartella dove Apache andrà a prendere i file da servire;
2. Copiare la cartella **src** nella stessa directory della cartella **public\_html** o equivalente, come mostrato in **Section 3.1**;
3. Eseguire lo script **db.sql** su **MariaDB**;
4. Per configurare l'accesso al database ci sono 2 modi:
  - Nel proprio ambiente dove vi è il server apache configurare le seguenti variabili d'ambiente;
  - Andare su `src/config.php` e modificare i valore nelle linee 2-5 rimpiazzando i **getenv("DB\_\*")** con le stringhe dei rispettivi valori.

Tali variabili sono:

- **DB\_HOST**: indica l'indirizzo del server in cui si trova il server MariaDB, nel nostro caso `localhost`;
  - **DB\_NAME**: indica il nome del database di MariaDB;
  - **DB\_USER**: indica l'username dell'account per accedere a MariaDB;
  - **DB\_PASS**: indica la password associata all'username specificato precedentemente.
5. In base alla configurazione del server Apache (come ad esempio il caso del laboratorio) se il sito viene servito su una path diversa da `"/` bisogna andare su **public\_html/.htaccess** e cambiare la seconda riga dove vi indica la path che prenderà in considerazione per gestire le regole per il robots.txt, errori 404 e 500 (nel nostro caso è **RewriteBase /mvasquez**).