

Progetto Web: Gestione Ristoranti

Indice

1	Introduzione	2
2	Istruzioni per l'uso	2
3	Scelte architettureali	3
4	Scomposizione in servizi	3
5	Architettura generale del progetto	3
6	Limitazioni e problemi riscontrati	4

1 Introduzione

Descrizione del progetto

Il progetto “*Gestione Ristoranti*” è un’applicazione web sviluppata con l’obiettivo di creare una piattaforma semplice che consenta agli utenti di registrarsi, accedere e consultare una lista di ristoranti. Allo stesso tempo, offre ai ristoratori la possibilità di registrarsi e gestire le informazioni relative al proprio ristorante.

L’interfaccia è suddivisa in due principali categorie di utenti:

- Utenti visitatori, che possono esplorare i ristoranti presenti sulla piattaforma;
- Ristoratori, che hanno accesso ad una dashboard personale per gestire i propri dati.

Obiettivi del progetto

L’obiettivo iniziale era realizzare una piattaforma con le seguenti funzionalità:

- Registrazione e login sicuri per utenti e ristoratori;
- Gestione di un profilo ristoratore, con possibilità di inserire e modificare i dati del proprio ristorante;
- Visualizzazione da parte degli utenti dei ristoranti presenti nel sistema.

Una parte avanzata del progetto prevedeva l’integrazione di un sistema per la gestione dei menù e dei piatti, ma questa funzionalità non è stata implementata a causa di vincoli temporali e tecnici.

2 Istruzioni per l’uso

Requisiti

- XAMPP (PHP, MySQL)
- Un browser (es. Chrome, Firefox)

Installazione e avvio

1. Scaricare o clonare il progetto nella cartella `htdocs/` di XAMPP.
2. Importare il database su MySQL (es. `gestione_ristoranti_db.sql`).
3. Configurare i parametri di connessione al database nel file `config/db.php`.
4. Avviare Apache e MySQL da XAMPP.
5. Aprire il browser e navigare su `http://localhost/Gestione_Ristoranti/frontend/`.

Navigazione

- `register.html`: pagina di registrazione per nuovi utenti o ristoratori.
- `login.html`: login per accedere in base al ruolo scelto.
- `auth.html`: area riservata visibile dopo l’autenticazione, differenziata per ristoratori e utenti.

3 Scelte architetturali

Stack tecnologico

- Frontend: HTML, CSS, JavaScript (senza framework)
- Backend: PHP
- Database: MySQL

Motivazione delle scelte

Le tecnologie selezionate sono state scelte per semplicità di utilizzo e compatibilità con l'ambiente di sviluppo locale e perché molto utilizzate durante l'anno scolastico. PHP e MySQL offrono strumenti consolidati per lo sviluppo rapido di prototipi e applicazioni didattiche. La separazione tra frontend e backend è mantenuta a livello base, con richieste AJAX ai file PHP per comunicare con il database.

4 Scomposizione in servizi

Il backend è stato suddiviso in file distinti, ognuno con una responsabilità ben definita. Questo approccio ha facilitato la gestione del codice e la manutenzione durante lo sviluppo.

Servizi principali

Servizio	Descrizione
register.php	Gestisce la registrazione di utenti e ristoratori.
auth.php	Verifica le credenziali ed effettua il login, creando una sessione utente.
session.php	Verifica se un utente ha una sessione attiva.
logout.php	Termina la sessione dell'utente.

Ogni file PHP opera come un endpoint che riceve dati (principalmente in JSON) dal frontend e restituisce risposte in formato JSON, semplificando l'integrazione tra interfaccia e logica lato server.

5 Architettura generale del progetto

Struttura delle cartelle

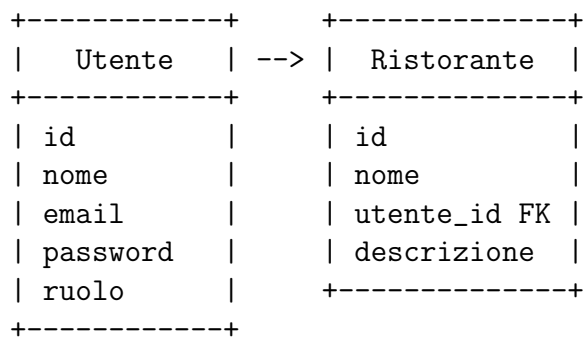
```
Gestione_Ristoranti/  
  
    backend/  
        api/  
            auth.php  
            register.php  
            session.php  
            logout.php  
        includes/  
            db.php  
            functions.php  
  
    frontend/  
        login.html
```

```

register.html
index.html
prenota.html
main.js
prenota.js
style.css

```

Schema logico semplificato (UML)



La tabella **utente** è collegata alla tabella **ristorante** tramite una chiave esterna. Solo gli utenti con ruolo ristoratore possono avere un ristorante associato.

6 Limitazioni e problemi riscontrati

Funzionalità non completate

- Gestione dei menù: inizialmente prevista come funzionalità, non è stata completata.
- Protezione avanzata: mancano controlli di sicurezza più avanzati (es. CSRF, validazione lato client estesa).

Problemi tecnici Durante lo sviluppo è stato riscontrato un errore frequente legato alla gestione delle risposte JSON. In particolare, la presenza di messaggi di errore o avvisi HTML nei file PHP causava un errore lato frontend nel parsing delle risposte JSON. Questo è stato risolto assicurandosi che tutti gli script PHP inviassero solo output JSON valido e che `session_start()` fosse posizionato all'inizio dei file che gestiscono le sessioni.

Conclusione

Il progetto rappresenta un primo passo verso la realizzazione di una piattaforma completa per la gestione di attività ristorative online. Pur non avendo raggiunto tutti gli obiettivi iniziali, offre una base solida e modulare, facilmente estendibile per funzionalità future.