

## Relazione Progetto Iris Fish

Iris Fish è il sito web di un ristorante di pesce che mostra le informazioni principali e il menù del ristorante, inoltre permette di prenotare un tavolo a seguito di una registrazione/login.

### FRONTEND

Il frontend è stato sviluppato in Angular in quanto facilita una gestione modulare e scalabile dell'applicazione attraverso un'architettura basata su componenti. Inoltre Angular permette di definire e gestire i percorsi di routing in modo semplice ed efficace.

Tra le componenti troviamo:

- home: rappresenta la schermata principale del sito e sono definite le informazioni generali del ristorante;
- specialita: schermata con il menù offerto;
- prenota: pagina accessibile solo a seguito di registrazione/login, permette di effettuare le prenotazioni;
- login: schermata accessibile dal link nella navbar Prenotazione;
- reg: schermata accessibile dalla pagina della login seguendo un link, permette di effettuare la registrazione al sito;
- registro-prenotazioni: schermata accessibile dalla pagina delle prenotazioni seguendo un link, mostra le prenotazioni effettuate e non ancora scadute.

Inoltre le parti grafiche comuni a tutte le schermate (logo, navbar e footer) sono state definite nell'appComponent.

### BACKEND

Il backend è stato sviluppato in SpringBoot combinato a Lombok per evitare la ripetitività di codice in quanto genera automaticamente i metodi comuni come getter, setter e costruttori attraverso le rispettive annotazioni, inoltre è stato utilizzato H2 Database per il salvataggio delle informazioni. Nel backend troviamo le entità con le rispettive repository che permettono il riempimento del database e il funzionamento delle API, tali entità sono:

- Account: rappresenta l'account di un utente;
- Prenotazioni: rappresenta la prenotazione di un tavolo;
- Token: rappresenta il token e il suo stato.

I servizi utilizzano i repository per accedere ai dati e tra essi troviamo:

- AccountServices: si occupa della creazione e del recupero degli account utente;
- AuthServices: si occupa dell'autenticazione degli account e della gestione dei token JWT;
- TokenServices: si occupa della generazione, validazione e gestione dei token JWT.
- PrenotazioneServices: si occupa della gestione delle prenotazioni.

I controller gestiscono le richieste HTTP e restituiscono le risposte al client. I controller implementati sono: AccountController e PrenotazioneController.

## AUTENTICAZIONE

Il sistema di autenticazione si basa su JWT (JSON Web Tokens) che consente un accesso sicuro e protetto.

Quando l'utente si registra i dati (password criptata) vengono salvati nel database e viene generato un token JWT con le informazioni necessarie all'autenticazione ed all'autorizzazione d'accesso dell'utente. Inoltre viene generato anche un token di refresh che permette il rinnovamento del token prima che scada.

La configurazione della sicurezza, definita nella classe SecurityConfig, stabilisce che gli endpoint di registrazione e login siano pubblici, mentre tutte le altre richieste richiedano l'autenticazione.

Attraverso il tokenInterceptor (intercettore HTTP) viene aggiunto automaticamente il token di autenticazione a tutte le richieste http, inoltre si occupa anche di gestire le scadenze dei token.