

SMARTFOOD

Descrizione del progetto

1. Obiettivo del progetto

Il progetto SmartFood è una applicazione web pensata per la valorizzazione dei prodotti alimentari locali.

L'idea principale è quella di permettere agli utenti di consultare un catalogo di prodotti, mentre i produttori possono inserire e gestire i prodotti che offrono.

L'applicazione è stata sviluppata come web app e può essere utilizzata anche da dispositivi mobile tramite browser. Il progetto è stato realizzato utilizzando le tecnologie viste durante il corso.

2. Funzionalità principali

Le principali funzionalità implementate sono:

- visualizzazione di una lista di prodotti con informazioni come nome, descrizione, metodo di produzione e certificazioni;
- registrazione e login degli utenti;
- gestione di diversi ruoli utente (USER, PRODUCER e ADMIN);
- possibilità di inserire nuovi prodotti solo per gli utenti autorizzati;
- protezione delle operazioni più importanti.

L'utente non autenticato può solo consultare i prodotti, mentre le operazioni di inserimento sono riservate ai produttori.

3. Architettura dell'applicazione

L'applicazione è stata sviluppata seguendo il modello Single Page Application.

Il frontend è realizzato in React e gestisce l'interfaccia utente e il routing lato client.

Il backend è realizzato in Node.js con Express e fornisce delle API REST utilizzate dal frontend.

Frontend e backend comunicano tramite richieste HTTP e scambio di dati in formato JSON. Questo approccio permette di mantenere separate le responsabilità tra le due parti dell'applicazione.

4. Tecnologie utilizzate

Le tecnologie principali utilizzate nel progetto sono:

Frontend:

- React
- Vite
- React Router
- Axios

Backend:

- Node.js
- Express
- dotenv
- bcrypt
- jsonwebtoken

Persistenza:

- PostgreSQL come database
- Prisma come ORM

5. Database e modello dei dati

Il database utilizzato è PostgreSQL ed è gestito tramite Prisma.

Le principali entità presenti sono:

- User, che contiene email, password (salvata come hash) e ruolo;
- Product, che contiene le informazioni relative ai prodotti inseriti;
- Review, che è prevista per eventuali funzionalità future.

Il modello dei dati è stato pensato in modo semplice ma estendibile, anche se alcune parti potrebbero essere migliorate.

6. Autenticazione e autorizzazione

Per l'autenticazione viene utilizzato JWT.

Dopo il login, il backend genera un token che contiene le informazioni principali dell'utente, come l'identificativo e il ruolo.

Il token viene poi utilizzato dal frontend per accedere alle rotte protette.

Il backend verifica il token ad ogni richiesta, senza mantenere sessioni aperte. Questo rende l'applicazione stateless.

7. Sicurezza

Per quanto riguarda la sicurezza, le password vengono salvate come hash utilizzando bcrypt.

Le rotte sensibili sono protette tramite autenticazione e controllo del ruolo dell'utente.

Sono state adottate alcune misure base di sicurezza, anche se l'applicazione non è pensata per un contesto produttivo reale.

8. Avvio dell'applicazione

Per avviare l'applicazione in locale è necessario configurare le variabili d'ambiente del backend, avviare il server Express e successivamente il frontend React.