

SINOSSI CAPSTONE PROJECT

THE GENOESE HOUSEKEEPER

Sintesi

La mia applicazione web ha come obiettivo fornire agli utenti la possibilità di selezionare una lista di prodotti, presenti generalmente in tutti i supermercati, e restituire la selezione dei supermercati più vicini e con le migliori offerte associate ai prodotti da noi selezionati.

Per quanto riguarda le tecnologie che utilizzerò per lo sviluppo Front-End: JavaScript, HTML/CSS, SASS, Bootstrap, React Redux. Per quanto riguarda il Back-End, invece, utilizzerai Java, Spring Boot, PostgreSQL e JPA. Per l'autenticazione degli utenti utilizzerò il sistema basato su JSON Web Token (JWT).

EXTRA: Offrire un sistema di tracciamento per guidare l'utente nella scelta del percorso migliore per fare la spesa sfruttando le migliori offerte e sconti offerti nei vari negozi.

Strumenti e Tecnologie:

Front-End

HTML: è il linguaggio di markup alla base della struttura delle pagine web, tra i principali vantaggi vi è la facilità e molti strumenti/librerie di supporto. Mi permetterà di immettere i dati e dare una struttura ad albero dei vari elementi(tag).

CSS: permette la formattazione delle pagine web creando stili complessi e personalizzati per i diversi elementi delle pagine anche grazie a selettori avanzati. Separando il codice dal foglio HTML posso distinguere la struttura dallo stile rendendo il tutto più leggibile e mantenibile.

SASS: è un preprocessore di CSS che permette di aggiungere a CSS nativo funzionalità avanzate come le variabili, regole innestate(nested rules) e la possibilità di organizzare il codice in partials che vengono automaticamente importati in una versione finale auto compilata correttamente CSS.

JavaScript: è un linguaggio di programmazione versatile basato sulla programmazione funzionale che supporta anche la programmazione orientata agli oggetti. Grazie alla sua capacità di manipolare gli elementi HTML e CSS mi consentirà di creare interfacce utente dinamiche e interattive e di supportare framework moderni come Angular, React e Vue.

React Redux: React mi permette di compartimentalizzare le singole classi/funzioni permettendo di rendere il codice leggibile e mantenibile. Redux aiuta alla gestione delle relazioni tra componenti in applicazioni complesse permettendomi di utilizzare uno store condiviso facilitandomi la modifica di stato e la propagazione tra i componenti delle props.

Back-End

Java: i vantaggi di java sono prima di tutto l'affidabilità e sicurezza (non puoi accedere ai puntatori di memoria), la portabilità e la vasta comunità di supporto. Inoltre supporta pienamente librerie come Hibernate e JPA per gestire agilmente la persistenza dei database.

JPA: è una specifica di ORM (Object-Relational Mapping) per java che consente di mappare oggetti java su tabelle di database relazionali e viceversa. Questo mi semplifica lo sviluppo dell'applicazione permettendomi di concentrarmi sullo sviluppo delle funzionalità.

Spring Boot: è un framework di sviluppo che riduce la quantità di codice ripetuto. Mi offre una configurazione automatica dell'applicazione basata sulle iniezione delle dipendenze. Usando una programmazione dichiarativa e l'uso di classi POJO(Beans) aiuta a concentrarsi sulla descrizione di cosa deve essere fatto, piuttosto che sul come deve essere fatto.

PostgreSQL: è un sistema di gestione di database open source che permette mediante interfaccia grafica di gestire con elevata affidabilità, scalabilità e flessibilità la gestione di un DB.

Conclusioni:

La mia applicazione web si propone di semplificare l'esperienza di acquisto degli utenti, fornendo un elenco aggiornato dei negozi vicini e/o con i prezzi migliori per i prodotti desiderati. Con l'aggiunta di filtri personalizzati gli utenti possono ottimizzare l'output della propria ricerca e risparmiare tempo e denaro nei loro acquisti.

V 1.0.0

Ho scelto questo progetto perché abbraccia tutte le principali funzionalità trattate durante il corso gestendo sia la profilazione dell'utente e le classiche interazione Front-End (form, modali, caroselli, etc...), che la gestione di un DB relazionale dove andare a fare query customizzate.

V 1.5.0

Non avendo mai unito le due parti dello sviluppo mi aspetto di imparare a connettere i due aspetti del web development. Vorrei prendere dimestichezza nella costruzione delle relazioni tra tabelle dovendo strutturare le informazioni in base alle funzionalità che voglio sviluppare. Non è parte del core del progetto ma vorrei scrivere la mia prima applicazione in TypeScript per prendere dimestichezza con questo linguaggio sempre più utilizzato.

V 2.0.0

Infine, l'implementazione del tracciamento del percorso migliore, permetterebbe di offrire un interessante feature agli utenti permettendogli di cacciare le migliori offerte con un percorso efficiente.