E-COMMERCE ABBIGLIAMENTO

**Specifica requisiti:**

L’e-commerce sviluppato permette di completare gli acquisti tramite pagamenti online attraverso carte, permette di aggiungere prodotti al carrello, di rimuovere prodotti nel carrello o di modificarne la quantità; per procedere con l’acquisto l’utente deve potersi registrare o fare il login in modo tale da permettere anche la visualizzazione degli acquisti passati; possibilità di effettuare ricerche mediante uso di filtri di ricerca o in maniera testuale e visualizzazione dei prodotti più nel dettaglio

**Progettazione e Implementazione:**

Per progettare il sistema abbiamo deciso di avvalerci dello stack MERN, dopo accurate ricerche ci è sembrata la soluzione più interessante, adatta, popolare e open-source quindi anche utile a nostro avviso da apprendere per la nostra crescita come professionisti.

Per la parte di **Frontend** come da stack MERN, attraverso React.js abbiamo creato le interfacce utente grazie alla sua semplicità, velocità e riusabilità dei componenti e ovviamente con l’uso di HTML e CSS; per quanto riguarda lo scambio di dati tra backend e frontend abbiamo fatto uso della libreria Axios per mandare richieste http in maniera asincrona e ricevere i dati.

Per rendere più piacevole il frontend abbiamo importato da Material-UI degli elementi grafici e aggiunto effetti alle immagini quando passiamo con il cursore sulle immagini per rendere l’esperienza dell’utente un po' più piacevole e per i pagamenti abbiamo deciso di affidarci a Stripe a nostro avviso molto interessante, ben documentata e una realtà in forte crescita.

Per la parte di **Backend,** Node.js grazie alla sua efficienza, velocità e il modo di gestione delle operazioni in modo asincrono è ottimo per creare applicazioni web real-time e npm permette di installare moduli in maniera facile e veloce, per rendere lo sviluppo più comodo abbiamo installato nodemon che a ogni cambiamento restarta il backend molto comodo ci ha fatto risparmiare tempo,

per creare le REST API utili per il nostro sistema abbiamo fatto uso di Express e utilizzato il middleware CORS per abilitare la Cross-Origin Resource Sharing, per quanto riguarda la gestione delle password tramite bcrypt garantiamo la funzione di hashing delle password, abbiamo utilizzato JWT per garantire la sicurezza nella verifica delle credenziali; per questo abbiamo utilizzato dotenv per leggere le variabili di configurazioni contenenti le nostre chiavi private JWT, Stripe e la stringa di collegamento al nostro database MongoDB.

Abbiamo optato per un database NoSQL dato che di esperienza con database SQL ne abbiamo già avuta, abbiamo potuto apprezzare la facilità di uso e la possibilità di caricare dati in formato JSON cosa che abbiamo fatto per popolare le nostre collection di dati del database e grazie a Mongoose abbiamo potuto creare e validare gli schemi delle nostre collezioni.

Per salvare e rendere più piacevole l’esperienza d’uso quando si fa un acquisto è richiesto l’indirizzo di spedizione e i dati dell’utente, questi se precedentemente inseriti vengono recuperati dai Cookie che abbiamo settato.

Dato che non abbiamo un profilo Stripe business per effettuare i pagamenti, il numero di carta da test tra le tante disponibili all’atto di prova secondo documentazione Stripe che abbiamo usato è 4242424242424242 cvc 3 cifre qualsiasi, data di scadenza non anteriore alla data attuale e cap numero casuale.