

E-Commerce Macelleria

Autori:

Leonardo Minaudo 297792

Michele Cairone 284972

E-Commerce Macelleria è una piattaforma web per la vendita online di alimenti quali carni e prodotti affini.

Navigando nell'e-commerce è possibile osservare nella home page tutti i prodotti acquistabili, applicare dei filtri di ricerca ad essi e aggiungerli al carrello impostando la quantità desiderata. Cliccando sulla card di un singolo prodotto si può aver modo di visualizzare ulteriori dettagli su di esso.

Un cliente che desidera effettuare un acquisto deve prima autenticarsi con le proprie credenziali (email e password). Qualora non disponesse già di un account può registrarsi cliccando sull'icona del profilo e successivamente sulla voce 'Registrati'. A tal punto l'utente sarà reindirizzato nella pagina di registrazione dove inserirà tutti i dati utili alla creazione del proprio account per poter effettuare l'accesso.

A login avvenuto il cliente può continuare a effettuare la propria spesa, accedere al proprio carrello, modificare le informazioni personali e visualizzare gli ordini già effettuati.

Il carrello, accessibile dall'icona specifica nella barra di navigazione, include tutti i prodotti precedentemente aggiunti con i relativi dettagli per ogni articolo e la possibilità di effettuare il check-out tramite pagamento con conto PayPal.

Gli ordini effettuati dai clienti sono fruibili dall'account del commerciante che accedendo può visualizzare nel dettaglio i singoli ordini e modificare il loro stato, in modo da tenere aggiornati i propri clienti sulla preparazione del collo. Il commerciante inoltre può aggiungere, rimuovere o modificare i prodotti in vendita nel proprio e-Commerce tramite un'apposita interfaccia.

L'E-Commerce è strutturato secondo un architettura **Model-View-Controller (MVC)** cioè composta da tre parti diverse: i dati (Model), la visualizzazione dei dati (View) e la gestione degli input (Controller). Questi tre componenti sono interconnessi: il Model viene mostrato tramite la View all'utente, il quale produce gli input con cui il Controller aggiorna il Model.

Si è scelto tale modello poiché ha grandi vantaggi nella gestione del codice, infatti

questo pattern favorisce lo sviluppo, il test e la manutenzione di ciascuna parte indipendentemente dall'altra.

La creazione delle diverse pagine e la gestione delle stesse è stata guidata tramite **Next.js**.

- **Next.js:**

Next.js è un framework JavaScript per applicazioni React. Permette di sviluppare applicazioni web, app mobile, desktop e web app progressive: è costruito secondo il principio di "Build once, run anywhere". Ha un router basato sul file system e sul concetto di pagina. Altre caratteristiche di Next.js sono suddivisione automatica del codice, routing automatico, hot code reloading (viene ricaricato solo il codice modificato) permette uno sviluppo rapido, efficiente e flessibile delle applicazioni web. Essendo basato su React utilizza **JSX** (JavaScript Syntax Extension) è un'estensione alla sintassi del linguaggio Javascript che fornisce un modo per strutturare il rendering dei componenti utilizzando una sintassi simile nell'aspetto all'HTML.



Le varie interfacce utente sono state progettate con l'ausilio di **HTML**, **CSS** e **Material-UI**.

- **HTML:**

Hypertext Markup Language, standard usato per strutturare i documenti del World Wide Web.

Rappresenta la struttura portante delle pagine web: su questa struttura si possono aggiungere modifiche grafiche, grazie ai fogli di stile CSS, ed elementi dinamici, grazie alla programmazione JavaScript.



- **CSS:**

Cascading Style Sheets, è un linguaggio usato per programmare la resa grafica di documenti scritti in un linguaggio di markup, come HTML e XML. Le classi CSS sono usate per specificare attributi grafici come font, dimensione, colore, spaziatura, bordo e posizione degli elementi all'interno di una rappresentazione web.



- **Material-UI:**

Material-UI è una libreria che consente di importare e utilizzare diversi componenti per creare un'interfaccia utente nelle applicazioni React. Ciò consente di risparmiare una notevole quantità di tempo poiché non si ha bisogno di scrivere tutto da zero.



L'interattività delle pagine è stata curata con **JavaScript**.

- **JavaScript:**

JavaScript è un linguaggio di programmazione orientato agli eventi, comunemente utilizzato nella programmazione Web lato client (esteso poi anche al lato server) per la creazione, in siti web e applicazioni web, di effetti dinamici interattivi tramite funzioni di script invocate da *eventi* innescati a loro volta in vari modi dall'utente sulla pagina web in uso.



L'E-Commerce è stato reso in grado, con l'utilizzo di **Redux**, di memorizzare vari stati quali un avvenuto login e l'aggiornamento/mantenimento del carrello e dei suoi prodotti al suo interno.

- **Redux:**

Redux è un contenitore di stato prevedibile per app JavaScript. Redux fornisce un modo semplice per centralizzare lo stato e la logica di un'applicazione web: per questo è ideale nella programmazione di applicazioni a pagina singola (SPA, Single Page Application).



L'interoperabilità tra i dati e la loro visualizzazione/modifica è stata gestita tramite la libreria **Axios**.

- **Axios:**

Axios è una libreria JavaScript che permette di connettersi con le API di backend e gestire le richieste effettuate tramite il protocollo HTTP.

Il vantaggio di Axios risiede nel suo essere *promise-based*, permettendo quindi l'implementazione di codice asincrono. Il codice asincrono permetterà, in una pagina, di caricare più elementi contemporaneamente invece che in maniera sequenziale, snellendo sensibilmente i tempi di caricamento.



Le richieste che inoltra il client sono gestite lato server tramite il linguaggio **PHP**.

- **PHP:**

PHP è un linguaggio di scripting open source generico, usato per lo più nello sviluppo web. Progettato per creare pagine web dinamiche, integrate efficacemente con i database.



Il server si occupa di eseguire le opportune Query SQL, in base alla richiesta ricevuta, ad un database opportunamente creato per la permanenza dei dati.

Per maggiore sicurezza il sistema non memorizza le password in chiaro degli account ma ne salva il loro hash che verrà successivamente confrontato con l'hash generato in fase di autenticazione.

Per la fase di pagamento è stato implementato l'API REST PayPal che permette di poter simulare un acquisto tramite apposite credenziali.

L'intero sistema è stato sviluppato, dopo la fase di progettazione, tramite l'utilizzo dell'IDE Visual Studio Code e implementato localmente con un server Apache contenente al suo interno le API server per poter comunicare con il DBMS (DataBase Management System).