

Taizo Shop: E-commerce per prodotti animali

Autore: Eleonora Martella 307680

Taizo Shop è un e-commerce per la vendita di prodotti animali nel quale vengono trattati prodotti per cani, gatti, uccelli, roditori e pesci.

Nella HomePage possiamo visualizzare i prodotti in offerta e attraverso la barra di navigazione si possono ricercare i prodotti singolarmente e accedere alle diverse categorie e sotto-categorie.

La card di ogni singolo prodotto contiene una descrizione principale, una descrizione più dettagliata, la lista degli ingredienti nel caso di generi alimentari e i diversi prezzi in base al formato.

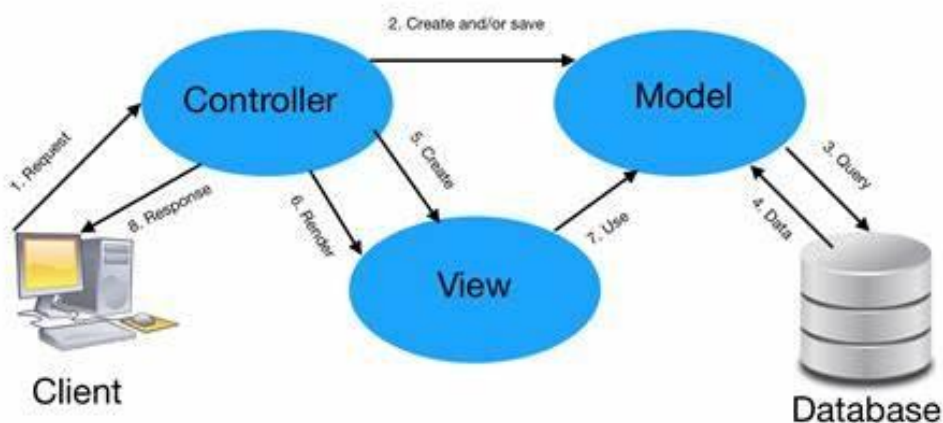
Un utente loggato può acquistare, visionare i suoi ordini, aggiungere prodotti alla wishlist e infine modificare le sue informazioni.

Il pagamento può essere effettuato con un account paypal o semplicemente inserendo una carta di credito valida.

Ogni singolo ordine effettuato può essere visionato dall'admin, il quale può cambiare lo stato dell'ordine in modo da tenere aggiornati gli utenti sulla preparazione e consegna del collo.

L'Admin può aggiungere, modificare e rimuovere prodotti dalla sezione di GESTIONE.

L'E-commerce è strutturato secondo un architettura **MODEL-VIEW-CONTROLLER** un pattern architetturale molto diffuso nello sviluppo di sistemi software, in particolare nell'ambito della programmazione orientata agli oggetti e in applicazioni web, in grado di separare la logica di presentazione dei dati dalla logica di elaborazione che rende operativa un'applicazione



E' composta da DATI(Model), VISUALIZZAZIONE DEI DATI e CONTROLLER:

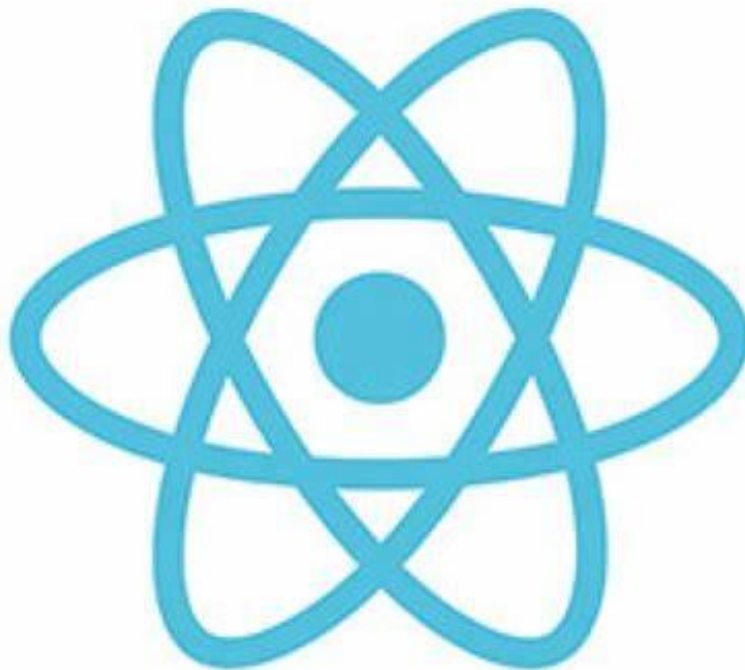
- il *model* fornisce i metodi per accedere ai dati utili all'applicazione;
- la *view* visualizza i dati contenuti nel model e si occupa dell'interazione con utenti e agenti;

- il *controller* riceve i comandi dell'utente (in genere attraverso la view) e li attua modificando lo stato degli altri due componenti.

Questo modello permette una gestione del codice molto più semplice in quanto ogni parte è indipendente dall'altra.

La creazione e gestione di ogni pagina è stata possibile grazie all'utilizzo di diverse tecnologie.

REACT



React JS

E' stato utilizzato principalmente **React** che è una libreria open-source, front-end, Javascript per la creazione di interfacce utente. Utilizzato come base nello sviluppo di applicazioni a pagina singola. React è basato su JSX è un'estensione alla sintassi del linguaggio JS che fornisce un modo per strutturare il rendering dei componenti utilizzando una sintassi molto simile a quella di HTML

NEXT



Next è un framework JavaScript per applicazioni React permette la suddivisione del codice, routing automatico, e la ricarica del solo codice modificato permettendo così uno sviluppo rapido, flessibile e efficiente delle applicazioni web

Per la grafica delle interfacce utente sono stati utilizzati **HTML, CSS** e **MATERIAL-UI**

HTML



E' un linguaggio di markup, nato per la formattazione e impaginazione di documenti ipertestuali disponibili nel web 1.0, oggi è utilizzato principalmente per il disaccoppiamento della struttura logica di una pagina web (definita appunto dal markup) e la sua rappresentazione grafica, gestita tramite gli stili CSS per adattarsi alle nuove esigenze di comunicazione e pubblicazione all'interno di internet, gli elementi dinamici vengono gestiti da Javascript

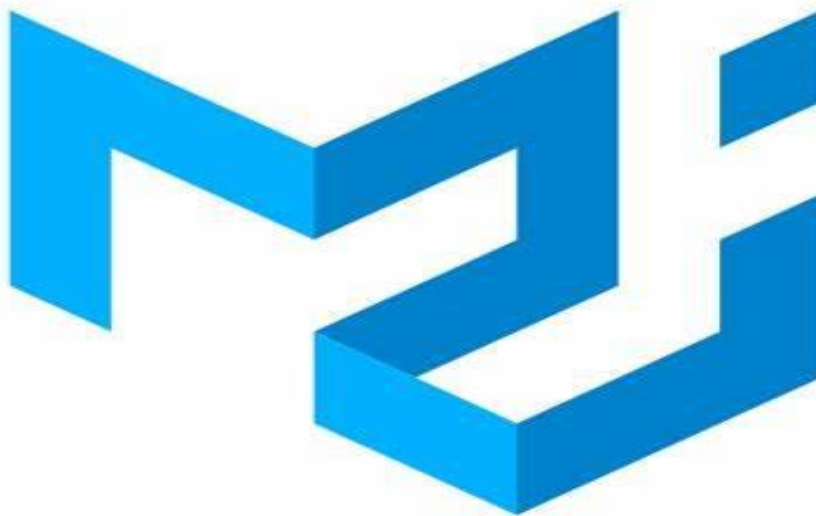
CSS



E' un linguaggio usato per definire la formattazione di documenti HTML,XML

L'introduzione del CSS si è resa necessaria per separare i contenuti delle pagine HTML dalla loro formattazione o layout e permettere una programmazione più chiara e facile da utilizzare, sia per gli autori delle pagine stesse sia per gli utenti, garantendo contemporaneamente anche il riutilizzo del codice ed una sua più facile manutenzione.

MATERIAL-UI



Si tratta di una libreria che consente di importare e utilizzare diversi componenti per creare le interfacce utente per applicazioni React.

Le pagine sono state rese interattive grazie a **Javascript**

JAVASCRIPT



JavaScript è un linguaggio di programmazione orientato agli eventi, utilizzato nella programmazione lato client per la creazione di effetti dinamici tramite funzioni di script invocate da eventi che vengono attivati quando l'utente interagisce con la pagina web

Con **Redux** invece è stato possibile memorizzare alcuni stati come ad esempio il login da parte dell'utente e un eventuale aggiornamento/mantenimento del carrello con relativi prodotti

REDUX



Redux

Redux è una libreria JavaScript open source per la gestione e la centralizzazione dello stato dell'applicazione. È comunemente usato con librerie come React o Angular per la costruzione di interfacce utente. Redux è una piccola libreria con un'API semplice e limitata progettata per essere un contenitore prevedibile per lo stato dell'applicazione.

L'interoperabilità tra i dati e la loro visualizzazione/modifica è stata gestita tramite **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 suo vantaggio risiede nell'essere promise based, in quanto permette l'implementazione di un codice di tipo ASINCRONO permettendo così di caricare più elementi contemporaneamente ottimizzando i tempi di caricamento

La gestione dei dati invece è stata affidata a **MongoDB**



MongoDB è un DBMS non relazionale, orientato ai documenti.

Classificato come un database di tipo NoSQL, MongoDB si allontana dalla struttura tradizionale basata su tabelle dei database relazionale in favore di documenti in stile JSON con schema dinamico , rendendo l'integrazione di dati di alcuni tipi di applicazioni più facile e veloce.

Mongoose è una libreria ODM (Object Data Modeling) per MongoDB.

Mongoose aiuta gli sviluppatori con la modellazione dei dati, l'applicazione dello schema, la convalida del modello e la manipolazione generale dei dati.