

E-commerce per strumenti musicali e materiali audio

Tecnologie utilizzate:

- Frontend: il frontend è stato sviluppato con React, utilizzando librerie di supporto tra le quali react-hook-form e react router per facilitare la costruzione dell'interfaccia utente e la gestione delle richieste verso il server. Sono state implementate pratiche di style responsive attraverso l'uso di CSS Modules e media queries per adattare il layout a schermi di varie dimensioni.
- Backend: il server è stato sviluppato utilizzando Node.js con il framework Express che consente la creazione di endpoint API per la gestione di operazioni CRUD sui dati. La gestione dei file è facilitata da Multer per consentire all'amministratore del servizio web di caricare le immagini associate ai prodotti.
- Database: ogni endpoint è progettato per gestire specifiche richieste HTTP e interagire con il database PostgreSQL tramite la libreria postgres.

Obiettivi e funzionalità

Il progetto ha come obiettivo la creazione di una piattaforma di e-commerce dedicata alla vendita di strumenti musicali e materiale audio.

Gli utenti dopo aver effettuato l'accesso vengono indirizzati alla homepage, che li accoglie con una selezione di articoli.

Da qui gli utenti possono navigare attraverso le varie sezioni "About" e "Services", per conoscere la piattaforma e i servizi disponibili e ricercare i prodotti desiderati tramite la barra di ricerca. Ogni utente può aggiungere i prodotti al proprio carrello che è sempre accessibile per visualizzare i prodotti selezionati e completare facilmente l'acquisto tramite il checkout.

Per facilitare la gestione delle operazioni, l'amministratore ha accesso ad una dashboard che funge da centro di controllo per monitorare l'attività del sito.

Tramite la dashboard l'amministratore può visualizzare il numero totale degli ordini ricevuti e l'elenco degli utenti registrati sulla piattaforma.

L'amministratore ha la possibilità di aggiungere, modificare ed eliminare i prodotti del catalogo in modo semplice e veloce.

Attraverso la dashboard, la piattaforma permette di caricare le immagini che verranno gestite dalla libreria Multer.

Funzionamento di Multer

Multer è un middleware per Node.js che semplifica la gestione dei file caricati, riceve il file immagine dal frontend attraverso un form HTML consentendo di specificare dettagli come il formato (es: .jpg, .png), la dimensione massima e il percorso di salvataggio. Una volta validato il file viene salvato nella cartella di destinazione specificata nel backend (nel nostro caso './assets')

API RESTful

Le API RESTful svolgono un ruolo fondamentale nel nostro servizio, ci permettono infatti di facilitare la comunicazione tra il frontend e il server, essenziale per la gestione degli utenti, dei prodotti e degli ordini.

Ogni endpoint API è stato progettato per eseguire operazioni specifiche utilizzando metodi HTTP standard per supportare operazioni CRUD. Abbiamo utilizzato JSON come formato di trasferimento per le richieste HTTP poiché è leggero e facilmente leggibile sia dagli sviluppatori che dalle macchine stesse.

Le API RESTful implementano una gestione degli errori robusta e una validazione delle richieste per garantire l'integrità dei dati quali name, price per i prodotti e validità del formato (es: numeri positivi per i prezzi)

In caso di errori le API restituiscono messaggi di errore per spiegare l'errore specifico.

Esempio endpoint API

L'endpoint POST ci consente di aggiungere nuovi prodotti al catalogo.

L'amministratore può infatti inviare una richiesta contenente i dettagli del prodotto in formato JSON. Il server RESTful si occupa dello "spacchettamento" delle comunicazioni JSON sia in un senso che nell'altro. Nel caso del Database, ogni chiave di un pacchetto JSON corrisponde ad una colonna e al contrario saranno le funzioni di chiamata stesse a farlo per poi presentare i dati sul frontend.

Di seguito è riportato un esempio del payload che può essere inviato:

```
{
  "name": "Chitarra Acustica",
  "price": 199.99,
  "image": "chitarra.jpg"
}
```

Questa richiesta viene inviata all'URL /api/products. Una volta ricevuta, il sistema crea un nuovo articolo nel database, rendendolo così disponibile per gli utenti che navigano nel catalogo.