

# RELAZIONE PROGETTO

Davide Faroldi Lo Presti – 320013

## REQUISITI FUNZIONALI

1. Gestione del catalogo
  - a. Gli Employee possono aggiungere, modificare e rimuovere prodotti
  - b. I prodotti sono suddivisi in categorie e sottocategorie e ogni prodotto presenta immagini, caratteristiche e recensioni
  - c. È possibile monitorare le scorte
2. Carrello degli acquisti
  - a. Gli utenti possono inserire e rimuovere articoli dal carrello
  - b. Il costo totale degli articoli è sempre visibile
3. Ricerca prodotti
  - a. È disponibile una barra di ricerca per trovare prodotti
4. Gestione ordini
  - a. Gli ordini vengono confermati automaticamente
  - b. Gli Employee possono monitorare gli ordini e gestire i rimborsi
5. Recensioni e valutazioni
  - a. Gli utenti possono lasciare recensioni e voti ai prodotti
6. Gestione del personale
  - a. Gli Employee hanno accesso a una sezione riservata per gestire inventario, ordini e altri Employee
  - b. Si possono generare report mensili e gestire le spedizioni

## REQUISITI NON FUNZIONALI

1. Prestazioni: il sistema garantisce tempi di risposta rapidi
2. Sicurezza: è assicurata la conformità agli standard PCI-DSS per la protezione dei dati
3. Usabilità: l'interfaccia è intuitiva e semplice da utilizzare
4. Affidabilità: l'e-commerce deve funzionare senza interruzioni e minimizzando i guasti
5. Compatibilità: il sistema è accessibile da diversi browser
6. Integrazione con servizi esterni: vengono utilizzati servizi di terze parti come Stripe

## ARCHITETTURA DEL SISTEMA

1. **Frontend**

- a. L'interfaccia visibile agli utenti, è sviluppato con React.js, garantendo un'esperienza interattiva grazie alla creazione di una SPA (Single Page Application)

## **2. Backend**

- a. L'elaborazione delle richieste degli utenti e della gestione dei dati è stato realizzato con Node.js e Express, consentendo la comunicazione tra frontend e il database

## **3. Database**

- a. Database relazionale che archivia le informazioni riguardanti utenti, prodotti ordini e spedizioni.
- b. Il back end utilizza il modulo MySQL2 per gestire le operazioni di lettura e scrittura

## **4. Comunicazione tra i livelli**

- a. Frontend interagisce con il back end inviando richieste http
- b. Le API RESTful definiscono le operazioni disponibili per il frontend
- c. Le risposte del back end vengono restituite in formato JSON per un'elaborazione efficiente

# **TECNOLOGIE UTILIZZATE**

## **1. Frontend**

- a. Framework: React.js
- b. Linguaggi: JavaScript, JSX, CSS
- c. Gestione stato: useState
- d. Routing: react-router-dom

## **2. Backend**

- a. Framework: Node.js con Express.js
- b. Database: MySQL
- c. Autenticazione: JWT (JSON Web Token)

## **3. Strumenti aggiuntivi**

- a. Ambiente di sviluppo: Visual Studio Code
- b. Database: PhpMyAdmin
- c. Elaborazione pagamenti: Stripe
- d. Testing API: Postman
- e. Test di carico: Apache JMeter
- f. Hosting: XAMPP

# **STRUTTURA DEL DATABASE**

- Accessories: memorizza gli accessori in vendita
- AccessoriesCart: memorizza gli accessori salvati nei carrelli

- Cart: identifica i carrelli degli utenti
- Categories: identifica le categorie di prodotti
- Customers: memorizza i dati degli utenti registrati
- Detailsales: relazione tra i prodotti con gli acquisti
- Employees: memorizza i dati del personale
- Instruments: memorizza i prodotti “strumenti”
- InstrumentsCart: memorizza gli strumenti salvati nei carrelli
- ProductsCart: memorizza i prodotti nel carrello
- Reviews: memorizza le reviews dei vare prodotti
- Sales: memorizza le informazioni dei pagamenti
- Services: memorizza i servizi in vendita
- ServicesCart: memorizza i servizi salvati nei carrelli
- Shipments: memorizza le spedizioni di vendite

## **MANUALE UTENTE**

### **1. Registrazione e accesso**

- a. Gli utenti possono creare un account nella sezione di login
- b. Dopo l’accesso, è possibile modificare i dati dal menu Account

### **2. Ricerca e acquisto**

- a. I prodotti possono essere cercati nell’apposita barra di ricerca
- b. Il carrello permette di aggiungere e rimuovere articoli
- c. Gli acquisti possono essere effettuati sia dalla pagina del prodotto che dalla sezione carrello

### **3. Gestione ordini e recensioni**

- a. Gli utenti possono recensire i prodotti
- b. Gli employee possono vedere gli ordini fatti

### **4. Servizio clienti**

- a. Il servizio di assistenza è accessibile attraverso un modulo dedicato o via social media

## **FASE DI TESTING**

### **1. Obiettivi**

- a. Il processo di testing ha verificato il corretto funzionamento del sistema, valutando la conformità ai requisiti e le prestazioni

### **2. Strumenti di testing**

- a. Postman per il controllo delle API

### **3. Tipologie di Test**

- a. Test unitari: verificano singole unità del codice,

- b. Test di integrazione: valutano il corretto funzionamento tra moduli interconnessi
- c. Test funzionali: simulano scenari reali

## **CONCLUSIONE**

Questa relazione ha descritto il processo di sviluppo dell'e-commerce di strumenti musicali, analizzando i requisiti, l'architettura, il database e gli strumenti utilizzati. Le fasi di testing hanno garantito il corretto funzionamento del sistema, migliorandone l'affidabilità e la sicurezza.