**RELAZIONE PROGETTO**

Davide Faroldi Lo Presti – 320013

**REQUISITI FUNZIONALI**

1. Gestione del catalogo
   1. Gli Employee possono aggiungere, modificare e rimuovere prodotti
   2. I prodotti sono suddivisi in categorie e sottocategorie e ogni prodotto presenta immagini, caratteristiche e recensioni
   3. È possibile monitorare le scorte
2. Carrello degli acquisti
   1. Gli utenti possono inserire e rimuovere articoli dal carrello
   2. Il costo totale degli articoli è sempre visibile
3. Ricerca prodotti
   1. È disponibile una barra di ricerca per trovare prodotti
4. Gestione ordini
   1. Gli ordini vengono confermati automaticamente
   2. Gli Employee possono monitorare gli ordini e gestire i rimborsi
5. Recensioni e valutazioni
   1. Gli utenti possono lasciare recensioni e voti ai prodotti
6. Gestione del personale
   1. Gli Employee hanno accesso a una sezione riservata per gestire inventario, ordini e altri Employee
   2. 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**
   1. 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**
   1. 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**
   1. Database relazionale che archivia le informazioni riguardanti utenti, prodotti ordini e spedizioni.
   2. Il back end utilizza il modulo MySQL2 per gestire le operazioni di lettura e scrittura
4. **Comunicazione tra i livelli**
   1. Frontend interagisce con il back end inviando richieste http
   2. Le API RESTful definiscono le operazioni disponibili per il frontend
   3. Le risposte del back end vengono restituite in formato JSON per un’elaborazione efficiente

**TECNOLOGIE UTILIZZATE**

1. **Frontend**
   1. Framework: React.js
   2. Linguaggi: JavaScript, JSX, CSS
   3. Gestione stato: useState
   4. Routing: react-router-dom
2. **Backend**
   1. Framework: Node.js con Express.js
   2. Database: MySQL
   3. Autenticazione: JWT (JSON Web Token)
3. **Strumenti aggiuntivi**
   1. Ambiente di sviluppo: Visual Studio Code
   2. Database: PhpMyAdmin
   3. Elaborazione pagamenti: Stripe
   4. Testing API: Postman
   5. Test di carico: Apache JMeter
   6. Hosting: XAMPP

**STRUTTURA DEL DATABASE**

* Accessories: memorizza gli accessori in vendita
* AccessoriesCart: memorizza gli accessiori 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**
   1. Gli utenti possono creare un account nella sezione di login
   2. Dopo l’accesso, è possibile modificare i dati dal menu Account
2. **Ricerca e acquisto**
   1. I prodotti possono essere cercati nell’apposita barra di ricerca
   2. Il carrello permette di aggiungere e rimuovere articoli
   3. Gli acquisti possono essere effettuati sia dalla pagina del prodotto che dalla sezione carrello
3. **Gestione ordini e recensioni**
   1. Gli utenti possono recensire i prodotti
   2. Gli employee possono vedere gli ordini fatti
4. **Servizio clienti**
   1. Il servizio di assistenza è accessibile attraverso un modulo dedicato o via social media

**FASE DI TESTING**

1. **Obiettivi**
   1. Il processo di testing ha verificato il corretto funzionamento del sistema, valutando la conformità ai requisiti e le prestazioni
2. **Strumenti di testing**
   1. Postman per il controllo delle API
3. **Tipologie di Test**
   1. Test unitari: verificano singole unità del codice,
   2. Test di integrazione: valutano il corretto funzionamento tra moduli interconnessi
   3. Test funzionali: simulano scenari reali

**CONCLUSIONE**

Questa relazione ha descritto il processo di sviluppo dell’ecommerce 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.