E-Commerce Web-based Application

Miggy Antonio (015842723 louisemiguel.antonio@sjsu.edu) - Miqqy

Vincent Chen (015648568 vincent.g.chen@sjsu.edu) - dogboots

Richard Ngo (016306719 richard.t.ngo@sjsu.edu) - rng04

Leo Truong (015735174 leo.truong@sjsu.edu) - Leo-Truong

Mehek Singh (015578836 mehek.singh@sjsu.edu) - mehek2004

San José State University

CS 157A - Sec 01

Professor Ramin Moazeni

December 6, 2024

Table of Contents

Introduction

Our primary goal was to create an E-commerce web-based application that allows users to browse, purchase, and review products seamlessly. The project integrates a modern tech stack and emphasizes both functional and non-functional requirements to deliver an intuitive user experience.

The project includes essential E-commerce features such as user registration, a robust shopping cart system, categorization, and an admin interface for inventory management. It was developed using React.js and Next.js for the frontend and backend, with MySQL powering the database. Tailwind CSS was used for responsive and visually appealing UI/UX design. This project aimed to provide hands-on experience in web development and database design while fostering teamwork in a real-world scenario.

Functional Requirements

The E-commerce application implements the following key functionalities:

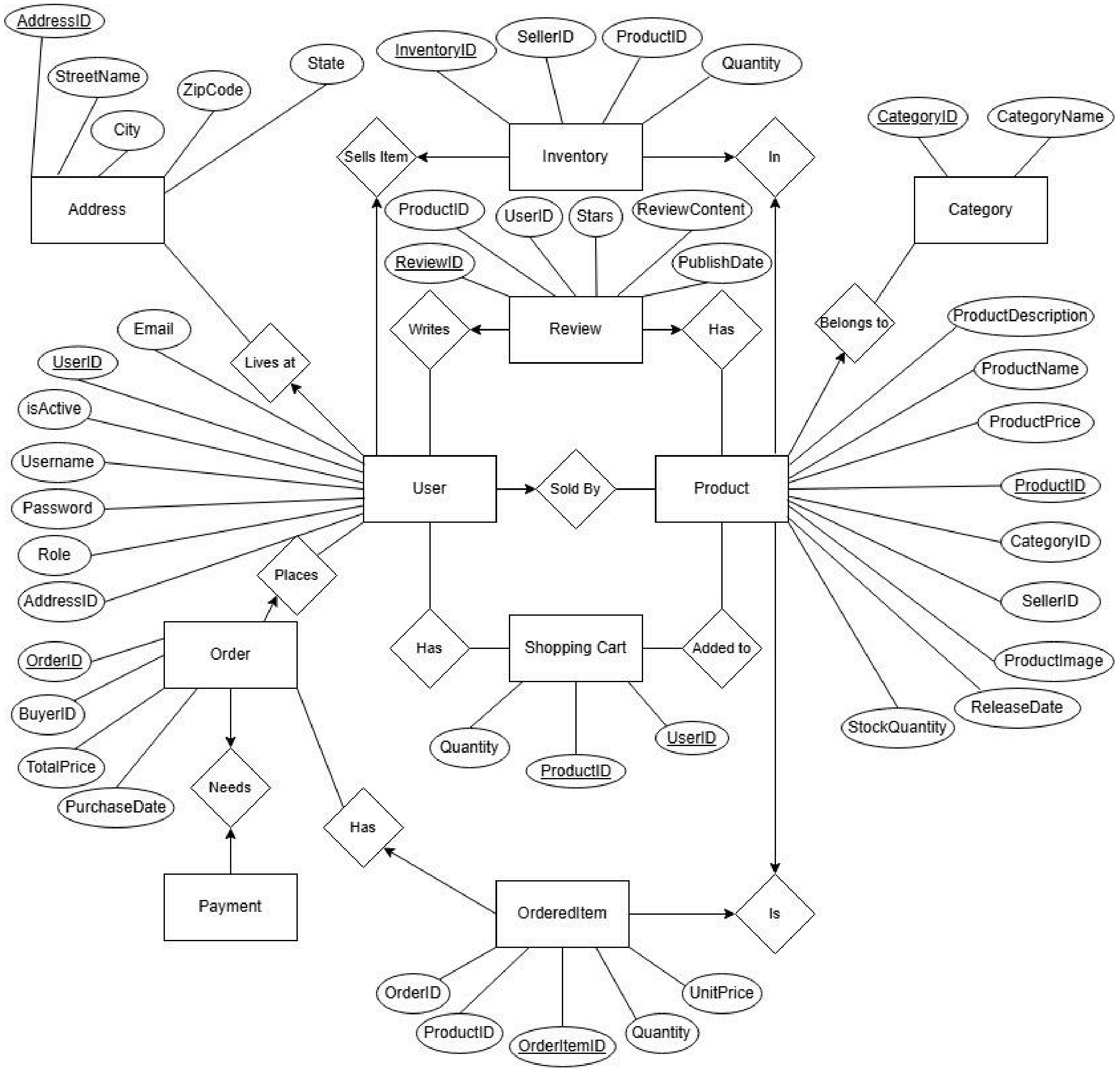
- User Account Management: User registration, login, and profile management.
- Product Browsing and Filtering: Users can view and filter products by categories.
- Shopping Cart: Users can add, update, or remove items from their cart in real time.
- Order Placement: Secure order placement with confirmation details.
- Review System: Users can leave reviews and ratings for purchased products.
- Admin Features: Administrators can add, update, or delete product listings.

Architecture

The project follows a modular architecture:

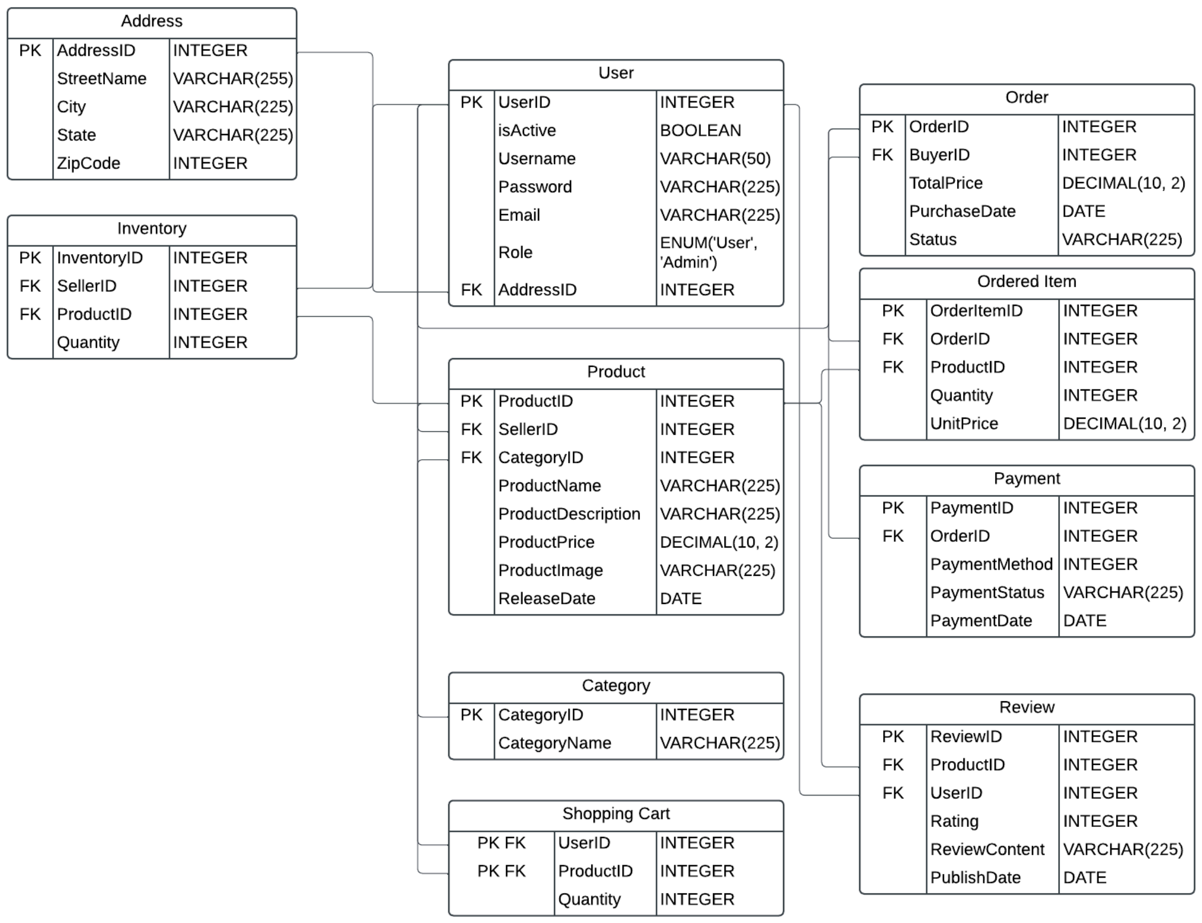
- Frontend: React.js for dynamic components and state management, ensuring seamless user interactions.
- Backend: Next.js for creating API routes that fetch and manipulate data from the database.
- Database: MySQL handles data storage with normalized tables for users, products, orders, and reviews.
- Styling: Tailwind CSS offers quick and customizable styles for consistent UI/UX.

ER Data Model



This is the Entity Relationship(ER) diagram for the e-commerce website and in it are showcased the 10 entities, their attributes, and how they are related to each other. An important detail of this ER diagram is that every single attribute was single-valued and atomic. This ER diagram was used as a blueprint to implement the database structure.

Database Design



Address

PK	AddressID	INTEGER
	StreetName	VARCHAR(255)
	City	VARCHAR(225)
	State	VARCHAR(225)
	ZipCode	INTEGER

Inventory

PK	InventoryID	INTEGER
FK	SellerID	INTEGER
FK	ProductID	INTEGER
	Quantity	INTEGER

User

PK	UserID	INTEGER
	isActive	BOOLEAN
	Username	VARCHAR(50)
	Password	VARCHAR(225)
	Email	VARCHAR(225)
	Role	ENUM('User', 'Admin')
FK	AddressID	INTEGER

Product

PK	ProductID	INTEGER
FK	SellerID	INTEGER
FK	CategoryID	INTEGER
	ProductName	VARCHAR(225)
	ProductDescription	VARCHAR(225)
	ProductPrice	DECIMAL(10, 2)
	ProductImage	VARCHAR(225)
	ReleaseDate	DATE

Category

PK	CategoryID	INTEGER
	CategoryName	VARCHAR(225)

Shopping Cart

PK FK	UserID	INTEGER
PK FK	ProductID	INTEGER
	Quantity	INTEGER

Order

PK	OrderID	INTEGER
FK	BuyerID	INTEGER
	TotalPrice	DECIMAL(10, 2)
	PurchaseDate	DATE
	Status	VARCHAR(225)

Ordered Item

PK	OrderItemID	INTEGER
FK	OrderID	INTEGER
FK	ProductID	INTEGER
	Quantity	INTEGER
	UnitPrice	DECIMAL(10, 2)

Payment

PK	PaymentID	INTEGER
FK	OrderID	INTEGER
	PaymentMethod	INTEGER
	PaymentStatus	VARCHAR(225)
	PaymentDate	DATE

Review

PK	ReviewID	INTEGER
FK	ProductID	INTEGER
FK	UserID	INTEGER
	Rating	INTEGER
	ReviewContent	VARCHAR(225)
	PublishDate	DATE

This is the database design for our e-commerce web-based application. The e-commerce database stores critical information such as users, products, orders, and reviews related to an online store. This allows our users to access and manipulate certain data through the front end of our application, allowing for efficient retrieval of data needed for an e-commerce website to function properly. It also allows our admins to efficiently retrieve and analyze data required to run the online store.

Major Design Decisions

We considered many factors when deciding on the technology stack for our project. We evaluated React.js, Vue.js, and Angular for the frontend user interface. We chose React.js because it has a large community, many helpful resources, and many third-party tools that make development easier. React also allowed us to create reusable components, which saved time and kept our code organized. For the backend, we decided to use Next.js. This framework was ideal because it provides an easy way to create API endpoints that handle data transfer between the frontend and the database. Its built-in features like routing and server-side rendering also made it a great choice for a smooth and efficient application. The database was built using MySQL since it was the primary database technology taught in class and is the required database for this project. MySQL allowed us to design relational tables for users, products, orders, and reviews, which matched our project requirements. To style the website, we used Tailwind CSS. This framework helped us create a clean and responsive design quickly. Tailwind’s utility-first approach lets us write fewer lines of custom CSS while still giving us a lot of control over the look and feel of the site.

These decisions ensured that our project was manageable for a student team while still allowing us to meet the technical and design goals of the application.

Implementation Details

To ensure a fully functional website, we had to address many issues. Firstly, we needed a way to recognize that a user was logged in. Using the sessionStorage property, we could display the user's ID upon login. Another issue was figuring out how to display data once a user clicks on a category. We created an API folder that stored all the backend javascript files that queried MySQL data. To establish a database connection, we stored a file called db.js in the lib folder and imported that file to every backend file to ensure data could be retrieved. Once the HTTP requests were fully working, we created div boxes that would display the attributes of products and allow users to write reviews and change quantity. The image paths of products were stored in the database and the public/images folder would have the same path name to ensure the frontend could display the proper image. Every page in the website was stored in the pages directory (excluding the starting page) and website components such as the navigation bar were stored in the components folder. Pages such as the review and checkout page mainly utilized the POST request to api endpoints to insert data into the database while GET requests were used to fetch product information. Overall, many of the tools from Next.js and React such as links, router, and hooks were used to implement our website successfully.

Demonstration

App demo link

Conclusion

The E-commerce web-based application met its main goals, providing a platform with many features that was easy to use and navigate. Our final updates focused on improving the design to make the website more visually appealing and user-friendly. We also conducted user testing to find and fix small issues that could make the site harder to use.

While the project will not continue after the semester, it has the potential to grow in the future. For example, advanced payment systems could be added to make the checkout process more secure and convenient. The application could also be expanded to include more product categories and features, like personalized recommendations or advanced search tools.

If developed further, this project could add many new functions and integrations. These updates would show its scalability and how it could be applied in real-world scenarios. Overall, the project highlights our ability to create a practical and functional system while leaving room for exciting possibilities in the future.