ADVANCE WEB DEVELOPMENT

TERM PROJECT

ONLINE SHOPPING STORE

SUBMITTED BY

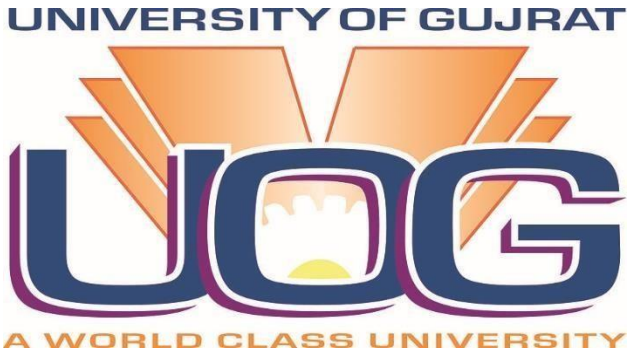
ROLL NO.	STUDENT NAME	SECTION
20021519-109	MUAHAMMAD FAHAD	A
20021519-012	MUHAMMAD TAHA	A

SUBMITTED TO

MR. MUHAMMAD ABO BAKAR ASLAM

SUBMISSION DATE

JUNE 25TH, 2024



GitHub Repository Link

https://github.com/offfahad/online-shopping-store-mern

1. Selection of Project

Our project does satisfy both conditions to some extent, it is based on RESTful API and Custom API on base of MongoDB.

1.1. Project Overview

Welcome to our **online shopping store application**! At our online store, we offer a wide range of high-quality products to meet your shopping needs. Whether you're looking for fashion, electronics, home appliances, or any other item, we have it all covered. This project is designed to provide a seamless online shopping experience by leveraging modern web technologies, RESTful APIs, and secure payment integration. The application allows users to register, log in, browse products, add items to their cart, and make payments using Stripe. Below is a detailed overview of the key features and functionalities of the project.

Key Features

User Authentication

Registration: Users can create an account by providing their name, email, and password. The system ensures that the email is unique to prevent duplicate accounts.

Login: Registered users can log in using their email and password. Secure authentication mechanisms ensure that user credentials are protected.

Product Browsing

Product Listings: The application fetches a wide range of products from the Fake Store API. Users can browse through various categories of products, view detailed descriptions, and check product images and prices.

Shopping Cart Management

Add to Cart: Users can add products to their cart directly from the product listings. Each cart item includes details such as the product ID, title, description, image, price, category, and quantity.

View Cart: Users can view the contents of their cart at any time. The cart displays all added items along with their quantities and total price.

Update Cart: Users can adjust the quantity of items in their cart or remove items altogether.

Secure Payment Processing

Stripe Integration: For a smooth and secure checkout process, the application integrates with Stripe. Users can proceed to payment from their cart, enter their payment details, and complete the purchase.

Payment Confirmation: Upon successful payment, users receive a confirmation, and the order is processed accordingly.

Technology Stack

Frontend:

- React: For building interactive user interfaces and managing the client-side of the application.
- Tailwind CSS: For styling the application with utility-first CSS classes, ensuring a responsive and modern design.

Backend:

- Node.js: For building the server-side application logic.
- Express.js: For creating RESTful APIs and managing server-side routing.

Database:

- MongoDB: For storing user information and cart details.

Authentication:

- JSON Web Tokens (JWT): For secure user sessions and authentication management.

External API:

- Fake Store API: For fetching product data that users can browse and add to their cart.

Payment Gateway:

- Stripe: For handling secure payments and processing transactions.

2. Creating ER Diagram

To create an ERD for our project, we need to define the entities and their relationships based on the functionality and data structures you've described. Here are the entities and their attributes:

1. User

- id (Primary Key)
- name

- email (Unique)
- password
- cart (Array of Cart Items)

2. Cart Item (Embedded in User)

- id
- title
- description
- image
- price
- category
- quantity

3. Product (Fetched from an external API)

- id (Primary Key)
- title
- description
- image
- price
- category

4. Payment

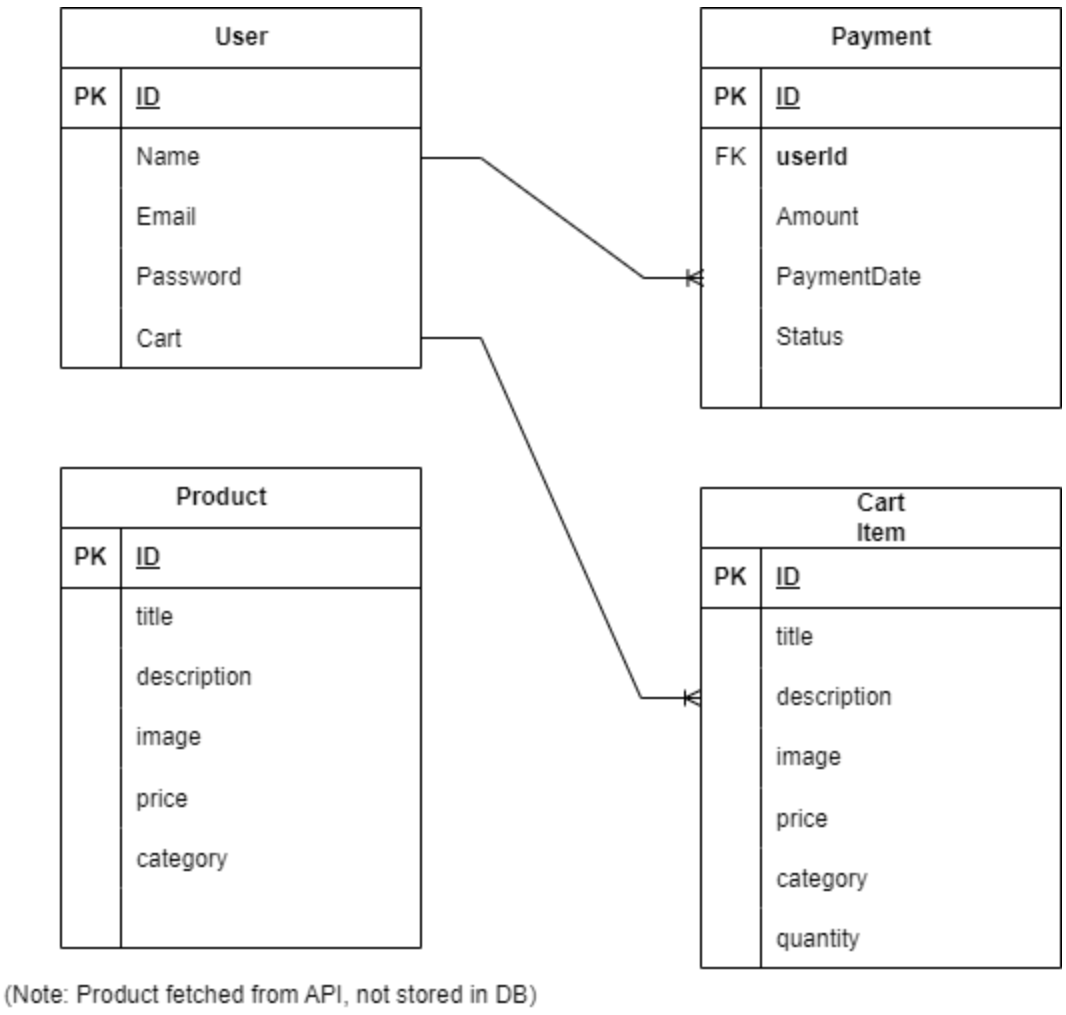
- id (Primary Key)
- userId (Foreign Key)
- amount
- paymentDate
- status

Relationships:

User to Cart Item: One-to-Many relationship (one user can have multiple cart items).

User to Payment: One-to-Many relationship (one user can have multiple payments).

Product: Represented but not stored in the database (fetched from an API).



(Note: Product fetched from API, not stored in DB)

3. Identify and List Down all Users

Identify all expected users that will interact with your project to perform specific operations/activities.

In our application, various users will interact with the system, each performing specific operations or activities. Below is a list of expected users along with their corresponding roles and activities:

1. Guest User

Operations

- Browse products
- View product details
- Register for a new account

- Log in to an existing account

2. Registered User

Operations

- Log in and log out
- Browse products
- View product details
- Add products to the cart
- View and manage the cart (update quantities, remove items)
- Proceed to checkout
- Make payments using Stripe

Admin User (Optional)

Operations

- Log in and log out
- Manage user accounts (create, read, update, delete)
- Manage products (create, read, update, delete)
- View and manage orders
- View payment history
- Generate reports

4. Identify all Use-Cases for all Users

Identifying all use-cases (operations/activities) for each type of user in our application helps in understanding the functional requirements and interactions within the system. Below are the use-cases categorized by user roles:

Use-Cases for Each User Role

1. Guest User

- View Products
- Browse through the list of available products.
- View product details including title, description, price, and images.
- Account Registration
- Create a new user account by providing name, email, and password.
- Authentication
- Log in to the application using registered email and password.

- Log out from the application.

2. Registered User

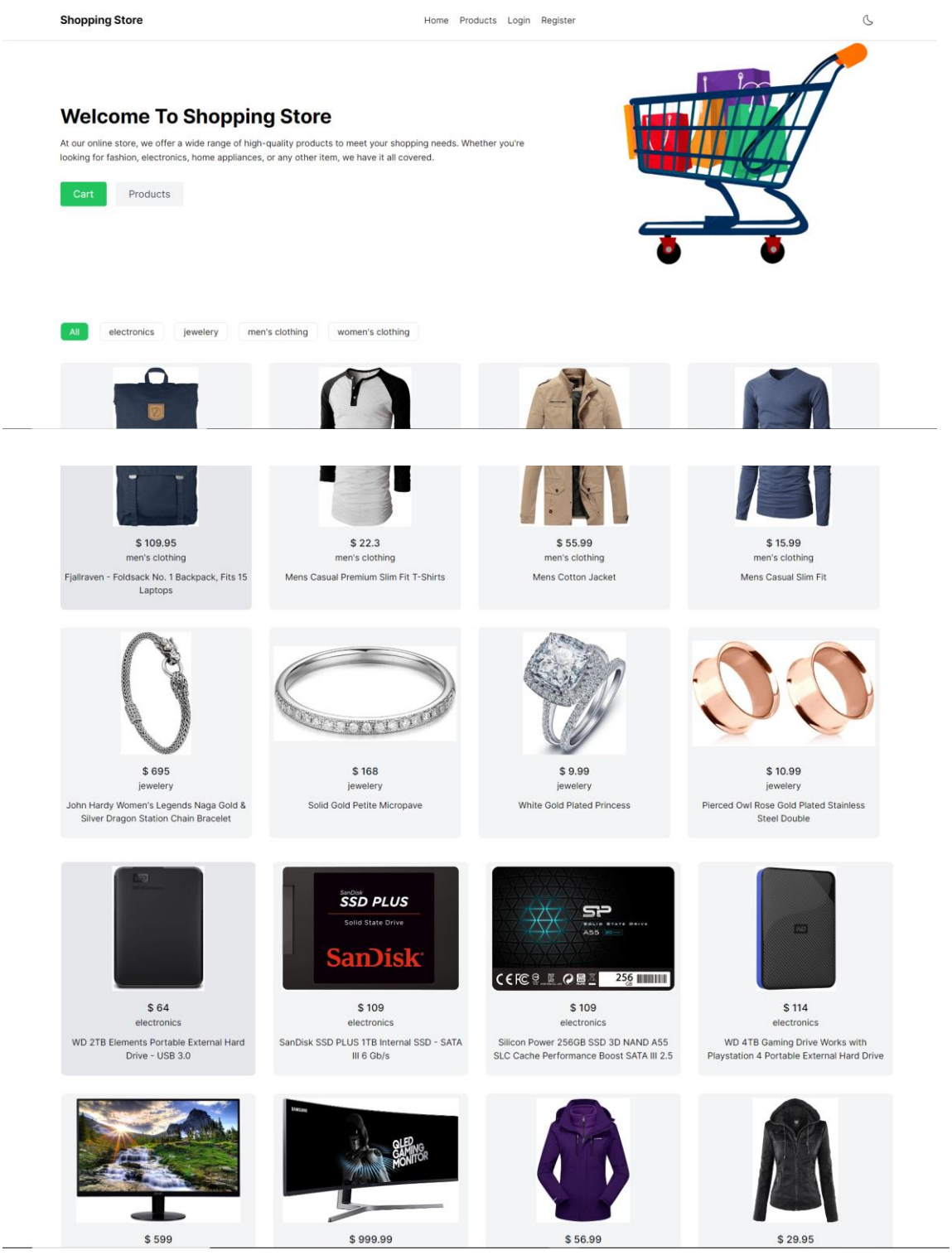
- General User Activities
- View Products: Browse through the list of available products.
- View Product Details: Access detailed information about each product.
- Log Out: End the current session and log out from the application.
- Shopping Activities
- Add to Cart: Add products to the shopping cart.
- View Cart: See all items currently in the cart.
- Update Cart: Modify quantities or remove items from the cart.
- Clear Cart: Remove all items from the cart in one action.
- Checkout and Payment
- Proceed to Checkout: Initiate the checkout process to purchase items in the cart.
- Enter Shipping Details: Provide shipping address and preferred shipping method.
- Payment: Make payment securely using Stripe integration.
- Order Confirmation: Receive confirmation of the order after successful payment.
- Account Management
- View Profile: Access and view personal information such as name and email.
- Update Profile: Modify user information including name, email, and password.
- View Order History: Review past orders and their details.

3. Admin User (Optional)

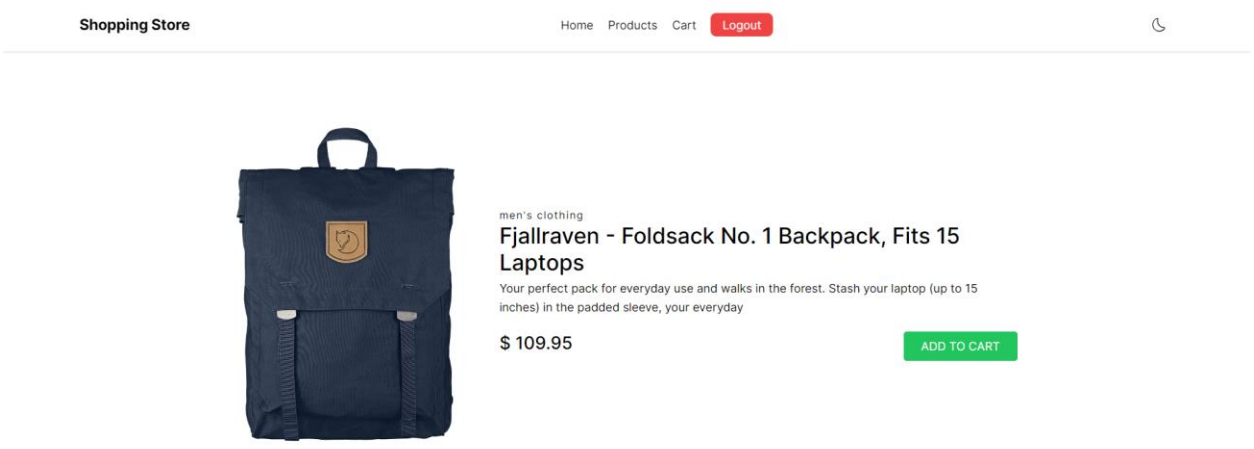
- User Management
- Manage Users: Create, read, update, and delete user accounts.
- View User Details: Access detailed information about individual users.
- Product Management
- Manage Products: Create, read, update, and delete products available in the store.
- View Product Details: Access detailed information about each product.

4. UI Screens

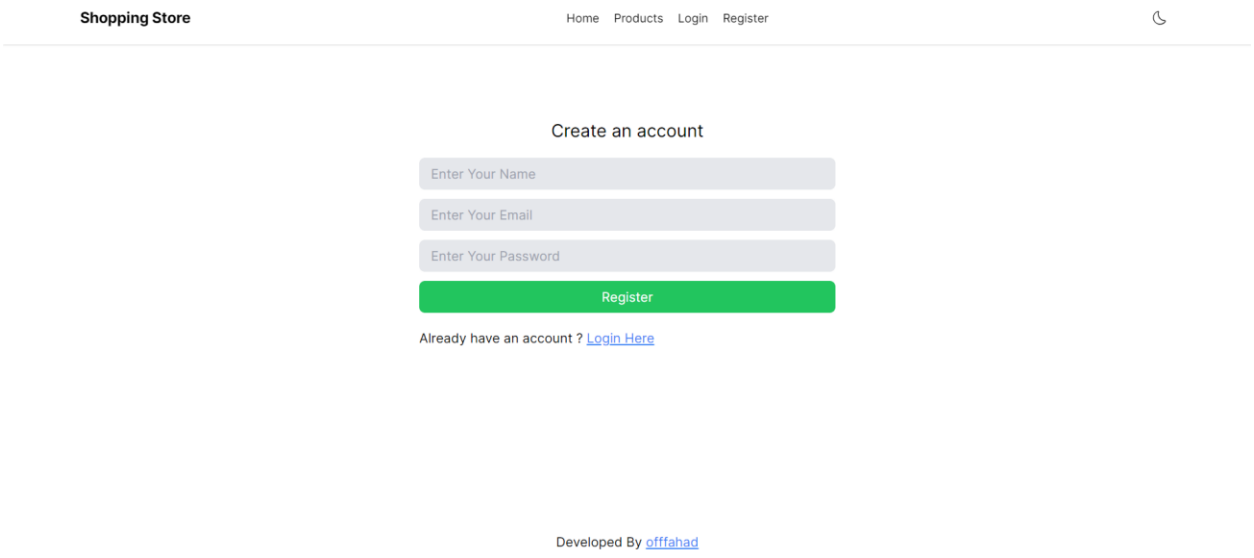
1) Home Screen and Products Screens



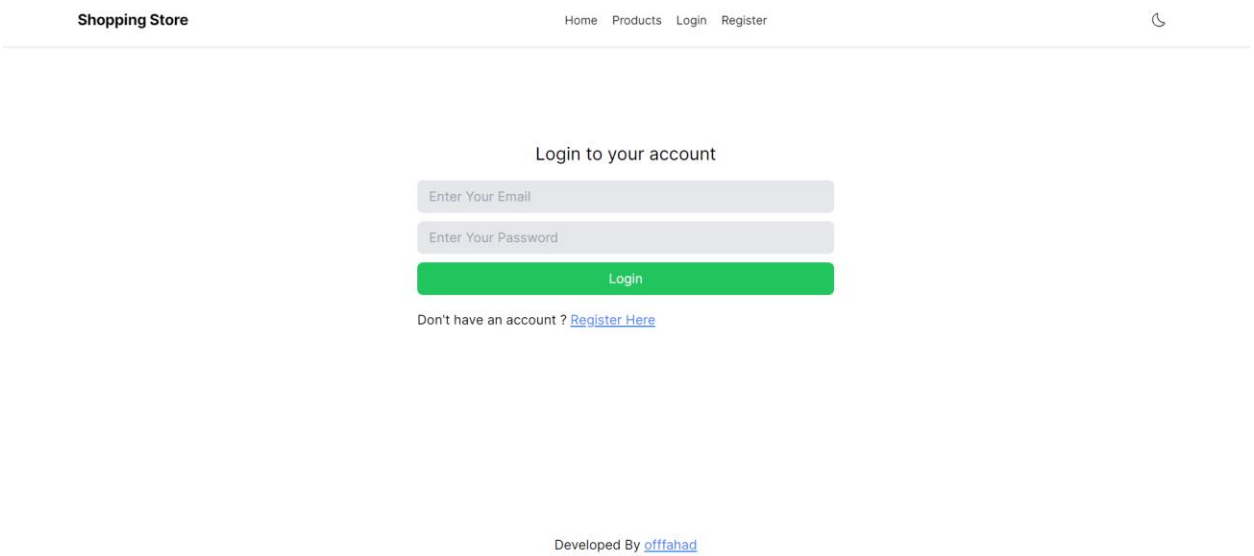
2) Product Detail Screen



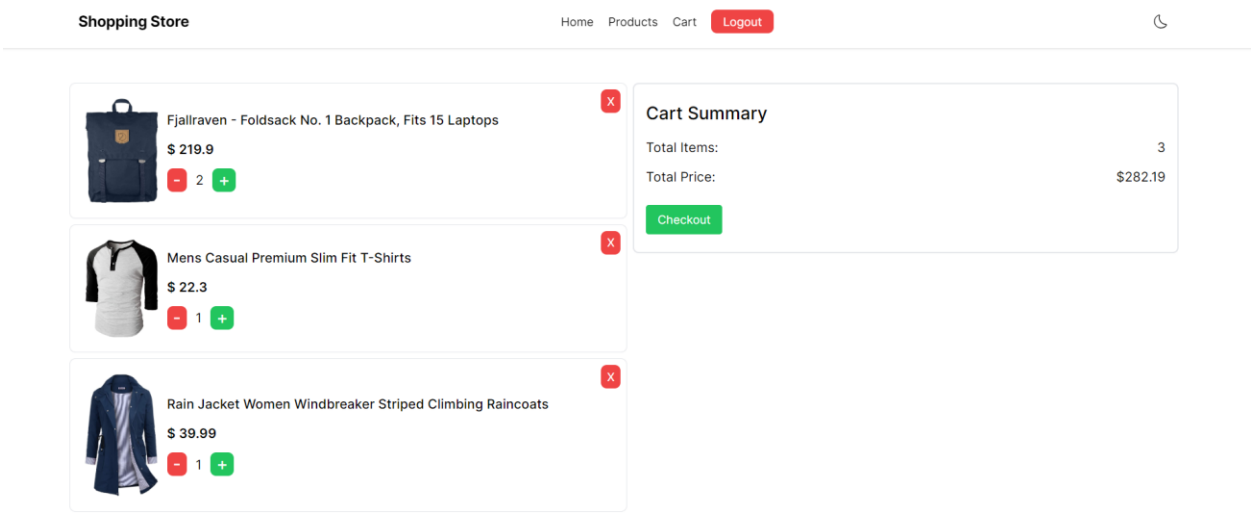
3) Register Screen



4) Sign In Screen



5) After Signing Cart Screen



6) Payment Screen