1. Introduction

Project Title: ShopEZ: One-Stop Shop For Online Purchases

Team Members and Roles:

- Yadla Anudeep - Team Leader & Backend Developer
- Jagannadham Likithanand - Frontend Developer
- Khandavalli Madhu Sai Santosh Kumar - Database & Integration
- Muthyam Venkata Ramana - UI/UX Designer

2. Project Overview

Purpose:

ShopEZ is designed to offer users a seamless and efficient online shopping experience. It acts as a one-stop destination for purchasing various categories of products, providing a user-friendly interface for customers and an administrative portal for sellers.

Features:

- User registration and login with secure authentication
- Browsing and searching products by category
- Wishlist and cart management
- Order placement and history tracking
- Admin panel for managing inventory and users

3. Architecture

Frontend:

The frontend is built using React.js with a component-based architecture. It manages routing, product display, form validation, and state management using React Hooks.

Backend:

Backend is built with Node.js and Express.js. It exposes RESTful APIs for authentication, product management, cart operations, and orders.

Database:

MongoDB is used for data storage. Mongoose is used for schema design and managing database queries. Collections include Users, Products, Cart, and Orders.

4. Setup Instructions

Prerequisites:

- Node.js (v16+ recommended)
- MongoDB (Local or Atlas)

Installation:

```
# Clone the repository
$ git clone <your-repo-url>

# Install server dependencies
$ cd server
$ npm install

# Set up environment variables in a .env file
MONGO_URI=<your-mongo-uri>
JWT_SECRET=<your-secret>

# Install client dependencies
$ cd ../client
$ npm install
```

5. Folder Structure

Client:

```
- /client
  - /src
    - /components
    - /pages
    - /context
    - /utils
    - App.js, index.js
```

Server:

```
- /server
  - /controllers
  - /routes
  - /models
  - /middleware
  - server.js, .env
```

6. Running the Application

Frontend:

```
cd client
npm start
```

Backend:

```
cd server
npm start
```

7. API Documentation

Auth APIs:

- POST /api/auth/register: Register a new user

- POST /api/auth/login: Login and receive JWT

Product APIs:

- GET /api/products: Get all products

- GET /api/products/:id: Get product by ID

Cart APIs:

- POST /api/cart: Add to cart

- GET /api/cart: Get user cart

Order APIs:

- POST /api/orders: Place order

- GET /api/orders: View user orders

8. Authentication

Authentication is handled using JWT tokens. On login, users receive a token stored in local storage and passed via headers for protected routes. Middleware on the backend validates the token for authorization.

9. User Interface

The UI is responsive and includes:

- Navigation bar with login/logout, cart, wishlist

- Category-wise product browsing

- Admin login panel

- Product cards with add-to-cart and wishlist options

10. Testing

Testing is done manually using Postman for APIs and through end-user testing on the UI. Integration testing is applied to verify frontend and backend flow.

11. Screenshots or Demo

Screenshot included below.

12. Known Issues

- Mobile responsiveness can be further improved

- Admin login does not have role-based routing yet

- Error messages need enhancement for failed operations

13. Future Enhancements

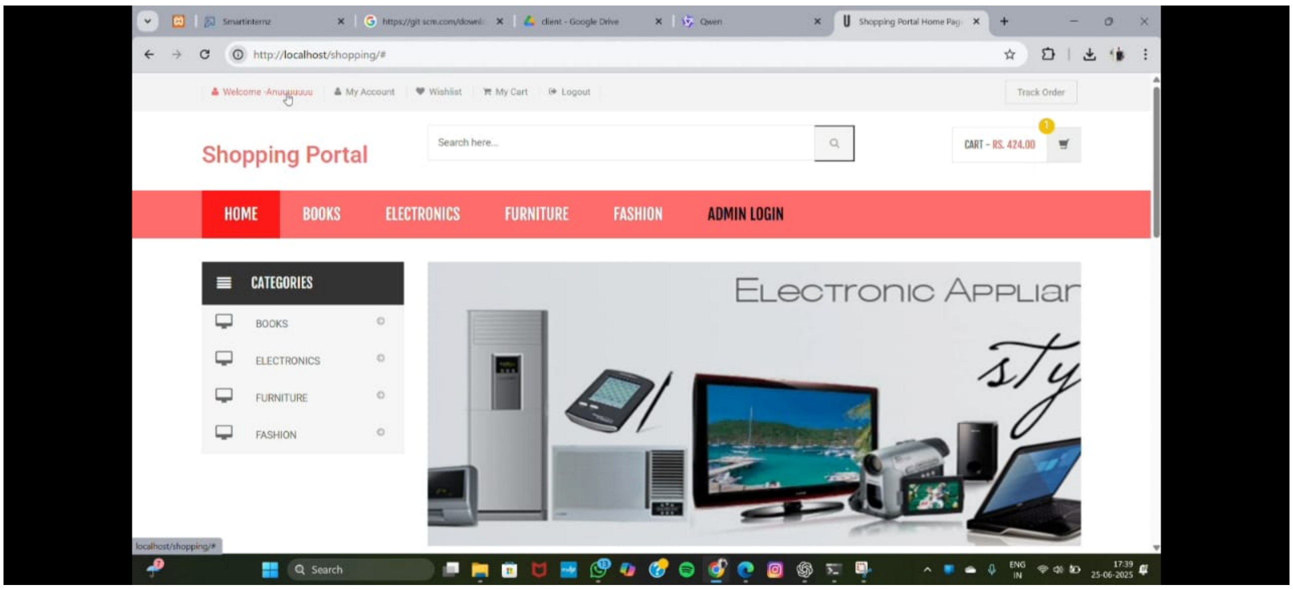
- Integrate payment gateway like Razorpay or Stripe

- Add product ratings and reviews

- Enable multi-language support

- Add stock notifications and shipping status tracking

UI Screenshot


http://localhost/shopping/#
Welcome -Anuuuuuuu
My Account
Wishlist
My Cart
Logout
Track Order
Shopping Portal
Search here...
CART - RS. 424.00
HOME
BOOKS
ELECTRONICS
FURNITURE
FASHION
ADMIN LOGIN
CATEGORIES
BOOKS
ELECTRONICS
FURNITURE
FASHION
Electronic Applian