Full Stack Development with MERN

Project Documentation format

1. Introduction

- **Project Title:** ShopEZ - E-Commerce Platform
- **Team Members:**

1. HRITHIK KHANNA K B
2. NAVEEN R
3. MAGESH KUMAR S
4. THULASI RAM REDDY V T

2. Project Overview

- **Purpose:** The e-commerce application will serve both customers and vendors by providing a user-friendly platform where customers can browse, compare, and purchase a wide variety of products online. Vendors will have the ability to list and manage their products, track sales, and interact with customers, ensuring a seamless and efficient online shopping experience for all users.

- **Features:**

I. **Customer-Facing Features**

1. **User Registration and Login**: Secure authentication using JWT, social media login options, and password recovery.
2. **Product Browsing**: Extensive product catalog with search filters for categories, brands, price range, ratings, and more.
3. **Shopping Cart**: Easy-to-use cart functionality, including adding/removing items, viewing total cost, and applying discount codes.
4. **Order Placement**: Streamlined checkout process with multiple payment options, shipping address management, and order summary.
5. **Order Tracking**: Real-time tracking of order status from processing to delivery with notifications.
6. **Order History**: Ability to view past orders for easy reordering and to see order details.
7. **Product Reviews and Ratings**: Write and read reviews and ratings for products.
8. **Wishlist**: Save favorite items for future purchase.

II. **Vendor-Facing Features**

1. **Vendor Registration and Login**: Secure registration process with verification and profile setup.

2. **Product Management**: Add, edit, and remove products, including images, descriptions, prices, stock levels, and availability.
3. **Order Management**: Real-time dashboard for managing incoming orders, updating order status, and handling special instructions.
4. **Promotion Management**: Create and manage promotional offers and discounts to attract more customers.
5. **Inventory Management**: Track inventory levels and receive alerts for low-stock items.
6. **Sales Analytics**: View detailed analytics on sales performance, popular products, and customer behavior.

III. **Admin-Facing Features**

1. **Admin Dashboard**: Comprehensive dashboard providing an overview of application performance, user statistics, and system health.
2. **User Management**: View, manage, and moderate customer and vendor accounts, including the ability to suspend or delete accounts.
3. **Content Moderation**: Monitor and approve/reject product listings, vendor profiles, and customer reviews.
4. **Order Monitoring**: Oversee order activity across the platform, resolve disputes, and ensure compliance with platform policies.
5. **Analytics and Reporting**: Generate detailed reports on user activity, sales performance, and platform usage.
6. **Role-Based Access Control**: Define and manage different access levels for staff members, ensuring secure and appropriate access to features.
7. **Promotions and Advertisements**: Create and manage global promotions and advertisements to be displayed to users.
8. **Category and Brand Management**: Add, edit, and remove product categories and brands to keep the catalog organized and up-to-date.

3. Architecture

- **Frontend**

1. **Public Directory**
   - **Purpose**: Hosts static files and the main entry point for the web application.
   - **Key Files**:
     - **favicon.ico**: The icon displayed in the browser tab.
     - **index.html**: The starting file containing meta links, external sources, and script references.
     - **manifest.json**: Configures web app details like name, icons, and start URL.
     - **robots.txt**: Instructs web crawlers on which pages to index.
2. **Src Directory**
   - **Purpose**: Contains all source code and main logic of the React application.
   - **Key Files**:

        - **index.js**: The entry point of the React application.

3. **Components Directory**
    - **Purpose**: Houses reusable UI components, promoting modularity and code reuse.
    - **Key Components**:
        - **Footer.jsx**: The footer section.
        - **Login.jsx**: The login page.
        - **Navbar.jsx**: The navigation bar.
        - **PopularProducts.jsx**: Displays popular products.
        - **Register.jsx**: The registration page.
        - **ProductList.jsx**: Displays a list of products.
        - **Authentication.jsx**: Manages authentication (login/register).
        - **Home.jsx**: The main home page component.
4. **Context Directory**
    - **Purpose**: Manages global state efficiently using React Context API.
    - **Key Files**:
        - **GeneralContext.js**: Provides context providers and consumers for global state management.
5. **Pages Directory**
    - **Purpose**: Contains components that represent different routes in the application.
    - **Subdirectories**:
        - **Admin**: Components for administrative functions.
            - **Admin.jsx**: Main admin dashboard.
            - **AllOrders.jsx**: Displays all orders.
            - **AllProducts.jsx**: Displays all products.
            - **AllUsers.jsx**: Displays all users.
        - **Customer**: Components for customer-related functionalities.
            - **Cart.jsx**: Shopping cart.
            - **CategoryProducts.jsx**: Displays products by category.
            - **IndividualProduct.jsx**: Displays details of a single product.
            - **Profile.jsx**: Displays and edits user profiles.
        - **Vendor**: Components for vendor-specific operations.
            - **EditProduct.jsx**: Edits a product.
            - **NewProduct.jsx**: Adds a new product.
            - **VendorHome.jsx**: Vendor home page.
            - **VendorOrders.jsx**: Displays vendor-related orders.
6. **Styles Directory**
    - **Purpose**: Contains CSS stylesheets for individual components and global styles.
    - **Contents**: CSS files organized to maintain style consistency and separation of concerns.
7. **JavaScript Files**
    - **Purpose**: Contains essential JavaScript files for the app's functionality.
    - **Key Files**:
        - **App.js**: Main application component.
        - **App.test.js**: Tests for the app component.

- **index.js**: React application's entry point.
- **reportWebVitals.js**: Measures performance metrics.
- **setupTests.js**: Configuration for setting up tests.

8. **Root Files**
   - **Purpose**: Hosts configuration and metadata files.
   - **Key Files**:
     - **.gitignore**: Specifies files and directories to be ignored by Git.
     - **package.json**: Lists project dependencies and scripts for managing the application.

- **Backend**

1. Server directory:

- Purpose: To host the server and to connect database and also to post and get API's

- Key files:

-index.js: it servers as the entry points for the backend servers for the application and it creates express application ,configures middleware connecting with mongoBD using mongoose, define the roots for various functions and starts the server.

-package.json: It lists project dependencies and scripts for managing the application.

-**Schema.js:** This file provides schema for various collections in a MongoDB database using Mongoose, an Object Data Modeling (ODM) library for MongoDB and Node.js.

- **Database**

1. Users

- Attributes: [list attributes like_id,username,password,email,usertype,approval,createdAt,updatedAt]

2. Admin

- Attributes: [list attributes like _id, Banner,categories]

3. Product

-Attributes:[list attributes like_id, Title, description ,mainImg,carousel,sizes,category,gender,price ,discount]

4. Orders

Attributes: [list attributes like _id,userId,name,email,mobile,address,pincode,title,description,mainImg,size,quantity,price,discount,paymentMethod,orderDate,orderStatus]

5. Cart

- Attributes: [list attributes like_id,userId,title,description,mainImg,size,quantity,price,discount]

-> It interacts with MongoDB using mongoose.connect with the database

4. Setup Instructions

- **Prerequisites**

Server Dependencies

1. bcrypt
2. body-parser
3. cors
4. express
5. mongoose

Client Dependencies

1 @testing-library/jest-dom
2 @testing-library/react
3 @testing-library/user-event
4 axios
5 bootstrap
6 react
7 react-dom
8 react-icons
9 react-router-dom
10 react-scripts
11 web-vitals

- **Installation:**

Cloning the project:

-git clone https://github.com/hrithikkb07/E-COMMERCE

Setting up the server:

-Navigate to server directory: cd server

-Install server dependencies: npm install

Setting up the client:

-navigate to server directory: cd server

-Install client dependencies: npm install

5. Folder Structure

- **Client**

  Public
  The public folder contains resources that are publicly accessible and served directly by the browser. It includes the favicon.ico, which is the icon displayed in the browser tab, and the index.html file, which is the main entry point for the application, containing essential meta tags and links to external resources. The manifest.json file provides configuration for the web app, detailing aspects such as the name, icons, and starting URL. Additionally, it includes logo192.png and logo512.png for different icon resolutions and the robots.txt file, which instructs web crawlers on which pages to index or ignore.

  Src
  The src directory is the heart of the React application, housing the core logic and components. The index.js file serves as the entry point, bootstrapping the React application. The App.js file contains the main application component, while App.test.js includes tests for this component. reportWebVitals.js is used for measuring performance metrics, and setupTests.js configures the testing environment. This directory encapsulates the structure and functionality of the application, ensuring organized and maintainable code.

  Components
  The components folder holds reusable UI components that are utilized throughout the application. These include Footer.jsx for the footer section, Login.jsx for the login page, Navbar.jsx for the navigation bar, PopularRestaurants.jsx for showcasing popular restaurants, Register.jsx for the registration page, and Restaurants.jsx for listing restaurants. The Authentication.jsx component handles authentication processes, while Home.jsx serves as the main home page component. This directory promotes reusability and consistency in the UI.

  Context
  The context directory is dedicated to managing global state within the application. It contains the GeneralContext.js file, which provides context providers and consumers, facilitating state management across different components. This setup enhances efficiency and organization by centralizing the state, making it easier to share data and functions between components without prop drilling.

Pages
The pages directory contains components that represent different routes or views within the application. It is subdivided into sections based on user roles.

Admin
The admin folder within pages includes components for the administrative section of the application. These components, such as Admin.jsx for the main dashboard, AllOrders.jsx for displaying all orders, AllProducts.jsx for managing products, and AllUsers.jsx for user management, are crucial for performing various administrative functions and maintaining control over the application's content and user base.

Customer
The customer folder within pages is focused on user-related components. It includes Cart.jsx for the shopping cart, CategoryProducts.jsx for displaying products by category, IndividualRestaurant.jsx for showing details of a specific restaurant, and Profile.jsx for user profile management. These components provide essential functionality for the customer experience, enabling users to browse, select, and purchase products, as well as manage their personal information.

Styles
The styles directory contains CSS files for styling the components and pages within the application. It includes individual stylesheets for specific components, ensuring a consistent look and feel across the application. Additionally, it has global styles that apply universally, maintaining visual coherence and enhancing user experience.

Root Files
The root of the project contains essential configuration and metadata files. The .gitignore file specifies which files and directories should be ignored by Git, preventing unnecessary files from being tracked. The package.json and package-lock.json files list the project's dependencies and their versions, ensuring a consistent development environment and facilitating dependency management. These files are fundamental for setting up, configuring, and maintaining the project's environment.

- **Server**

node_modules/: Similar to the client-side structure, this directory holds all the npm packages and dependencies required for the server-side application to run.

index.js: This file serves as the entry point for your Node.js application. It typically initializes the server, sets up routes, and connects to databases or other services.

package-lock.json: This file is automatically generated by npm. It locks down the versions of dependencies that were installed, ensuring consistency across different environments.

package.json: Contains metadata about the project and configuration details. It also lists all dependencies required by the project, scripts to run, and other metadata.

Schema.js: Presumably contains Mongoose schemas or other data structure definitions used for interacting with your MongoDB database.

6. Running the Application

- Provide commands to start the frontend and backend servers locally.
    - **Frontend:** `npm start` in the client directory.
    - **Backend:** `npm start` in the server directory.

7. API Documentation

User Authentication

- POST /api/user/register - Registers a new user.
- POST /api/user/login - Authenticates a user and returns a token.

User Management

- GET /api/user/- Retrieves user information by ID.
- PUT /api/user/- Updates user information by ID.

Product Management

- GET /api/products - Retrieves all products.
- POST /api/products - Creates a new product.
- GET /api/products/- Retrieves a product by ID.
- PUT /api/products/- Updates a product by ID.
- DELETE /api/products/- Deletes a product by ID.

Order Management

- GET /api/orders - Retrieves all orders.
- POST /api/orders - Creates a new order.
- GET /api/orders/- Retrieves an order by ID.
- PUT /api/orders/- Updates an order by ID.
- DELETE /api/orders/- Cancels an order by ID.

Cart Management

- GET /api/cart - Retrieves the current user's cart.
- POST /api/cart - Adds an item to the cart.
- PUT /api/cart/- Updates a cart item by ID.
- DELETE /api/cart/- Removes an item from the cart by ID.

8. Authentication

1. JWT Tokens (JSON Web Tokens):

• Client Side: When a user logs in with valid credentials, the server generates a JWT token and sends it back to the client.

• Server Side: The server signs the token using a secret key and includes user information (like user ID and possibly roles) in the payload.

• Storage: Typically stored in local storage or session storage in the browser to persist authentication state across sessions.

2. Token Expiry:

• JWT tokens have an expiration time (e.g., 1 hour). Clients can refresh tokens using a refresh token if supported.

• Refresh tokens are stored securely on the client side and are exchanged with the server to obtain a new JWT token without requiring the user to re-enter credentials.

3. Role-based Access Control (RBAC):

• Users are assigned roles (e.g., admin, customer, restaurant owner) that dictate their permissions.

• Access to certain routes or functionalities is restricted based on these roles.

4. Middleware:

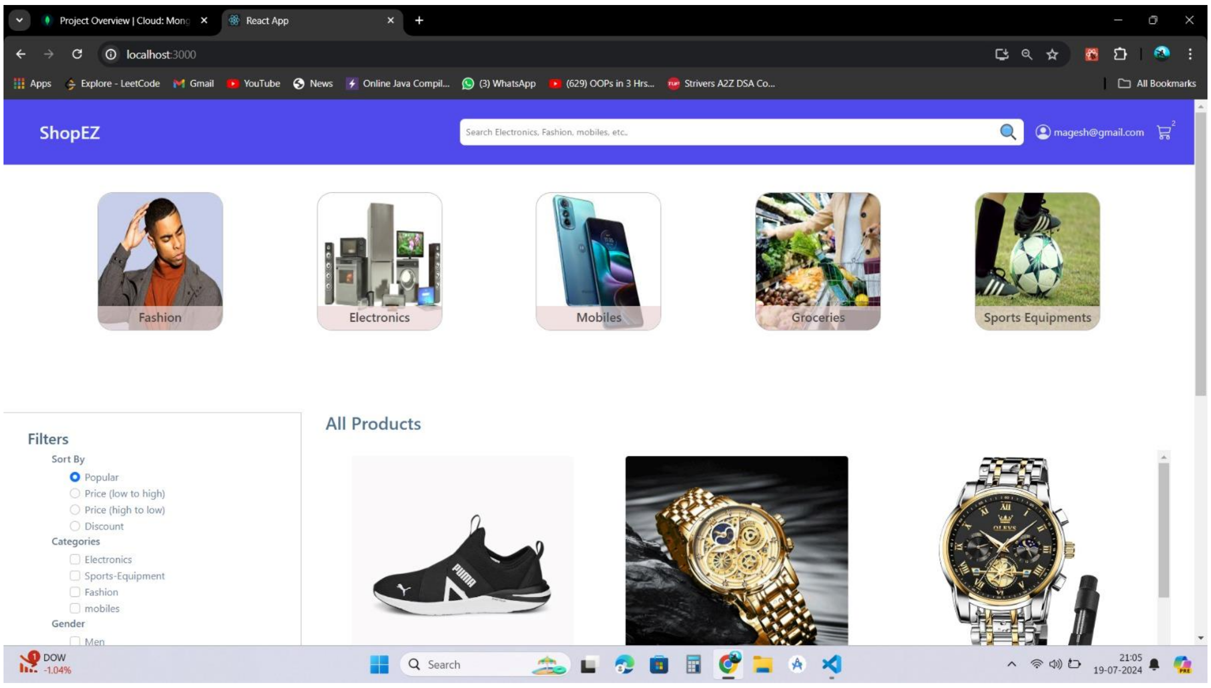
• Express middleware is often used to protect routes that require authentication.

• Before allowing access to protected resources, middleware verifies the validity of the JWT token.

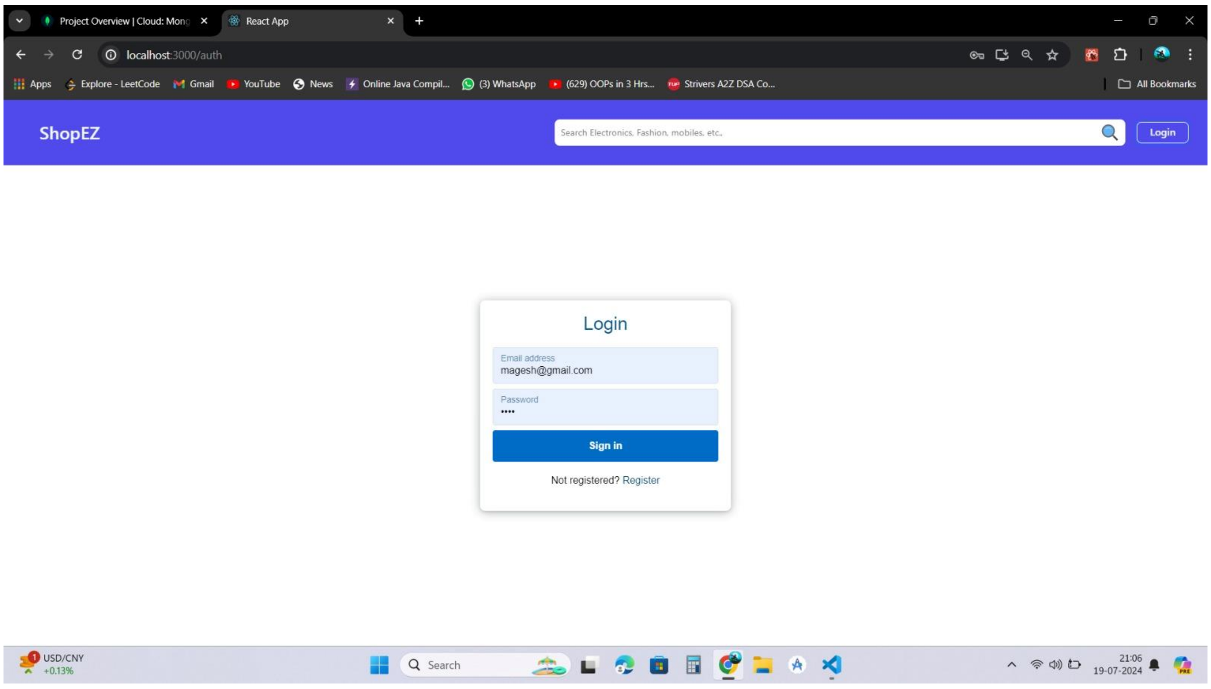
• If the token is valid, the server extracts user information from the token's payload and grants access based on the user's role and permissions.

9. User Interface

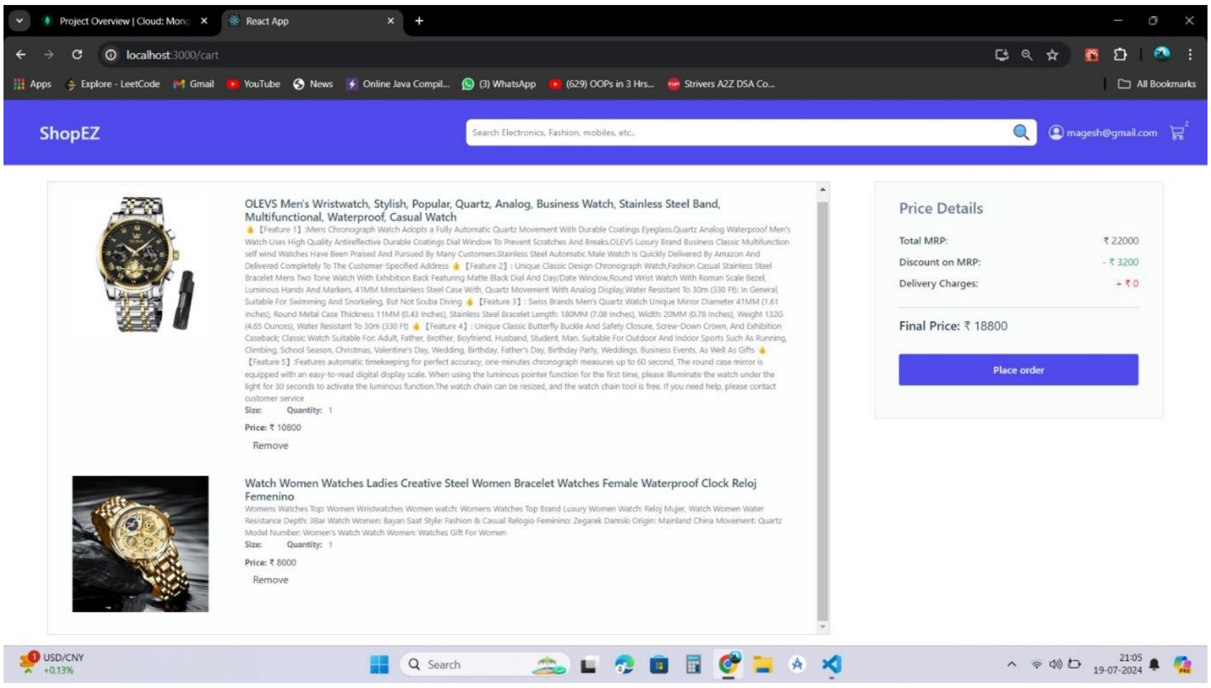
- **Home Page**



- Login Page



- Cart


localhost:3000/cart

ShopEZ

OLEVS Men's Wristwatch, Stylish, Popular, Quartz, Analog, Business Watch, Stainless Steel Band, Multifunctional, Waterproof, Casual Watch

👍【Feature 1】:Mens Chronograph Watch Adopts a Fully Automatic Quartz Movement With Durable Coatings Eyeglass.Quartz Analog Waterproof Men's Watch Uses High Quality Antireflective Durable Coatings Dial Window To Prevent Scratches And Breaks.OLEVS Luxury Brand Business Classic Multifunction self wind Watches Have Been Praised And Pursued By Many Customers.Stainless Steel Automatic Male Watch Is Quickly Delivered By Amazon And Delivered Completely To The Customer-Specified Address 👍【Feature 2】: Unique Classic Design Chronograph Watch,Fashion Casual Stainless Steel Bracelet Mens Two Tone Watch With Exhibition Back Featuring Matte Black Dial And Day/Date Window,Round Wrist Watch With Roman Scale Bezel, Luminous Hands And Markers, 41MM Mmstainless Steel Case With, Quartz Movement With Analog Display,Water Resistant To 30m (330 Ft): In General, Suitable For Swimming And Snorkeling, But Not Scuba Diving 👍【Feature 3】: Swiss Brands Men's Quartz Watch Unique Mirror Diameter 41MM (1.61 Inches), Round Metal Case Thickness 11MM (0.43 Inches), Stainless Steel Bracelet Length: 180MM (7.08 Inches), Width: 20MM (0.78 Inches), Weight 132G (4.65 Ounces), Water Resistant To 30m (330 Ft) 👍【Feature 4】: Unique Classic Butterfly Buckle And Safety Closure, Screw-Down Crown, And Exhibition Caseback; Classic Watch Suitable For: Adult, Father, Brother, Boyfriend, Husband, Student, Man. Suitable For Outdoor And Indoor Sports Such As Running, Climbing, School Season, Christmas, Valentine's Day, Wedding, Birthday, Father's Day, Birthday Party, Weddings, Business Events, As Well As Gifts 👍【Feature 5】:Features automatic timekeeping for perfect accuracy, one-minutes chronograph measures up to 60 second, The round case mirror is equipped with an easy-to-read digital display scale. When using the luminous pointer function for the first time, please illuminate the watch under the light for 30 seconds to activate the luminous function.The watch chain can be resized, and the watch chain tool is free. If you need help, please contact customer service

Size: **Quantity:** 1

Price: ₹ 10800

Remove

Watch Women Watches Ladies Creative Steel Women Bracelet Watches Female Waterproof Clock Reloj Femenino

Womens Watches Top: Women Wristwatches Women watch: Womens Watches Top Brand Luxury Women Watch: Reloj Mujer, Watch Women Water Resistance Depth: 3Bar Watch Women: Bayan Saat Style: Fashion & Casual Relogio Feminino: Zegarek Damski Origin: Mainland China Movement: Quartz Model Number: Women's Watch Watch Women: Watches Gift For Women

Size: **Quantity:** 1

Price: ₹ 8000

Remove

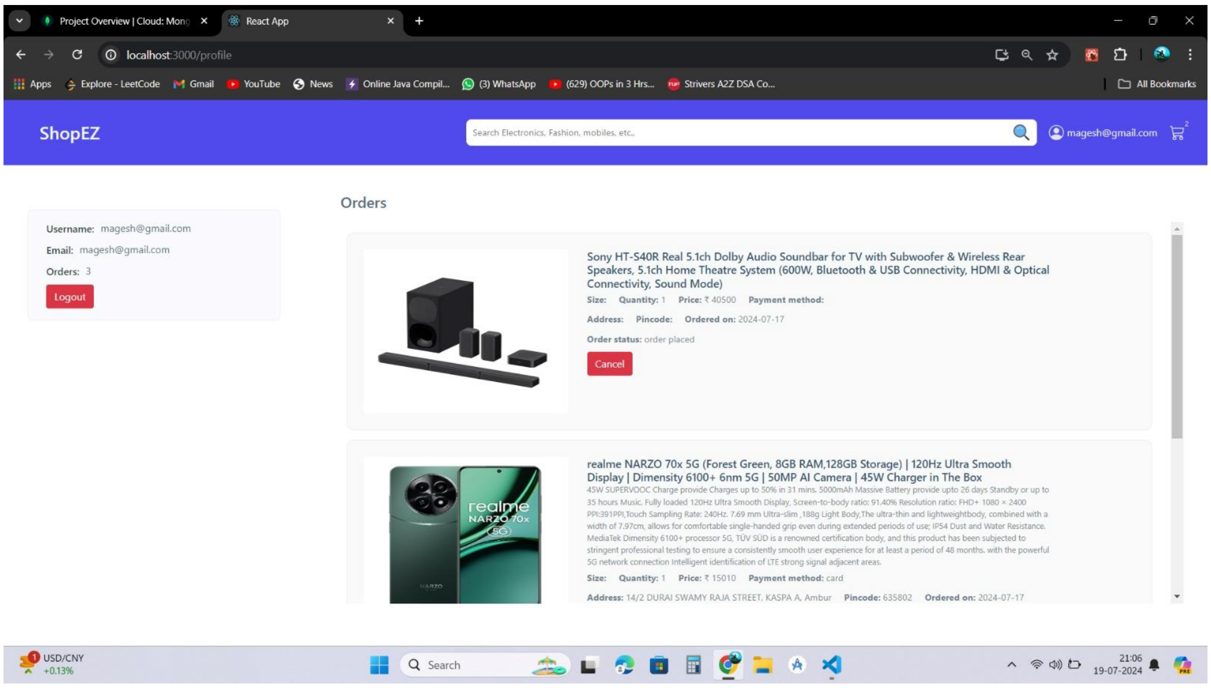
Price Details

Total MRP:	₹ 22000
Discount on MRP:	- ₹ 3200
Delivery Charges:	+ ₹ 0

Final Price: ₹ 18800

Place order

- User Profile


localhost:3000/profile

ShopEZ

Username: magesh@gmail.com

Email: magesh@gmail.com

Orders: 3

Logout

Orders

Sony HT-S40R Real 5.1ch Dolby Audio Soundbar for TV with Subwoofer & Wireless Rear Speakers, 5.1ch Home Theatre System (600W, Bluetooth & USB Connectivity, HDMI & Optical Connectivity, Sound Mode)

Size: **Quantity:** 1 **Price:** ₹ 40500 **Payment method:**

Address: **Pincode:** **Ordered on:** 2024-07-17

Order status: order placed

Cancel

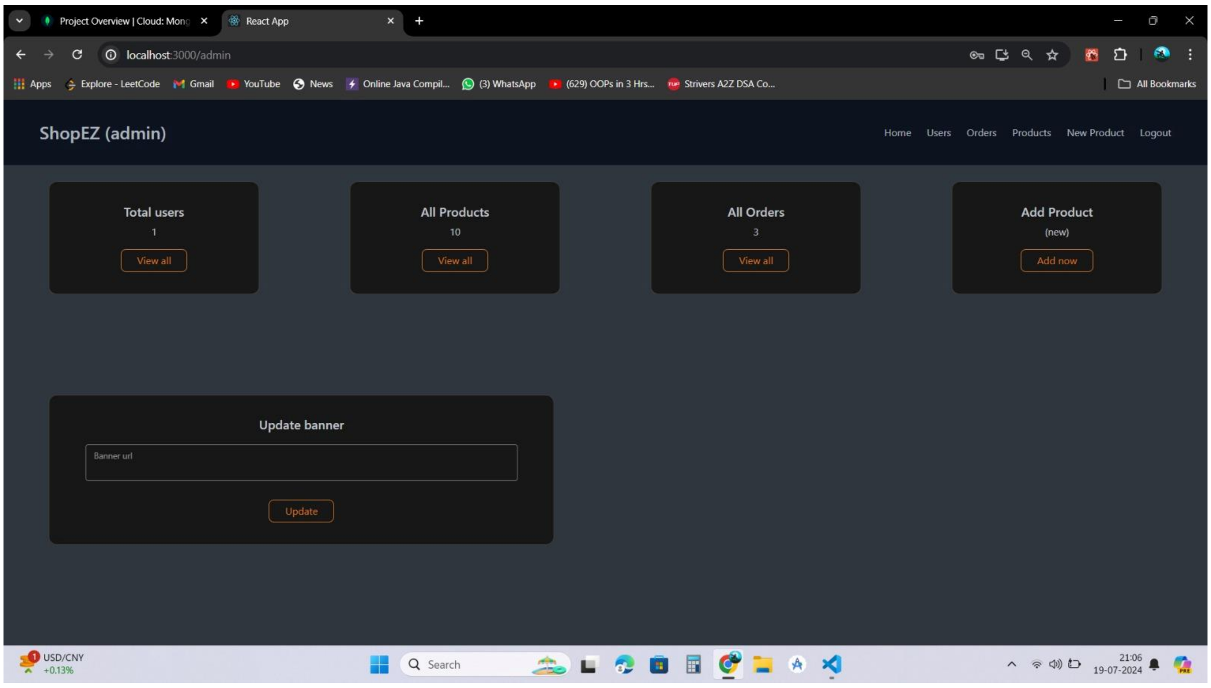
realme NARZO 70x 5G (Forest Green, 8GB RAM,128GB Storage) | 120Hz Ultra Smooth Display | Dimensity 6100+ 6nm 5G | 50MP AI Camera | 45W Charger in The Box

45W SUPERVOOC Charge provide Charges up to 50% in 31 mins. 5000mAh Massive Battery provide upto 26 days Standby or up to 35 hours Music. Fully loaded 120Hz Ultra Smooth Display, Screen-to-body ratio: 91.40% Resolution ratio: FHD+ 1080 × 2400 PPI:391PPI,Touch Sampling Rate: 240Hz. 7.69 mm Ultra-slim ,188g Light Body,The ultra-thin and lightweightbody, combined with a width of 7.97cm, allows for comfortable single-handed grip even during extended periods of use; IP54 Dust and Water Resistance. MediaTek Dimensity 6100+ processor 5G, TÜV SÜD is a renowned certification body, and this product has been subjected to stringent professional testing to ensure a consistently smooth user experience for at least a period of 48 months. with the powerful 5G network connection Intelligent identification of LTE strong signal adjacent areas.

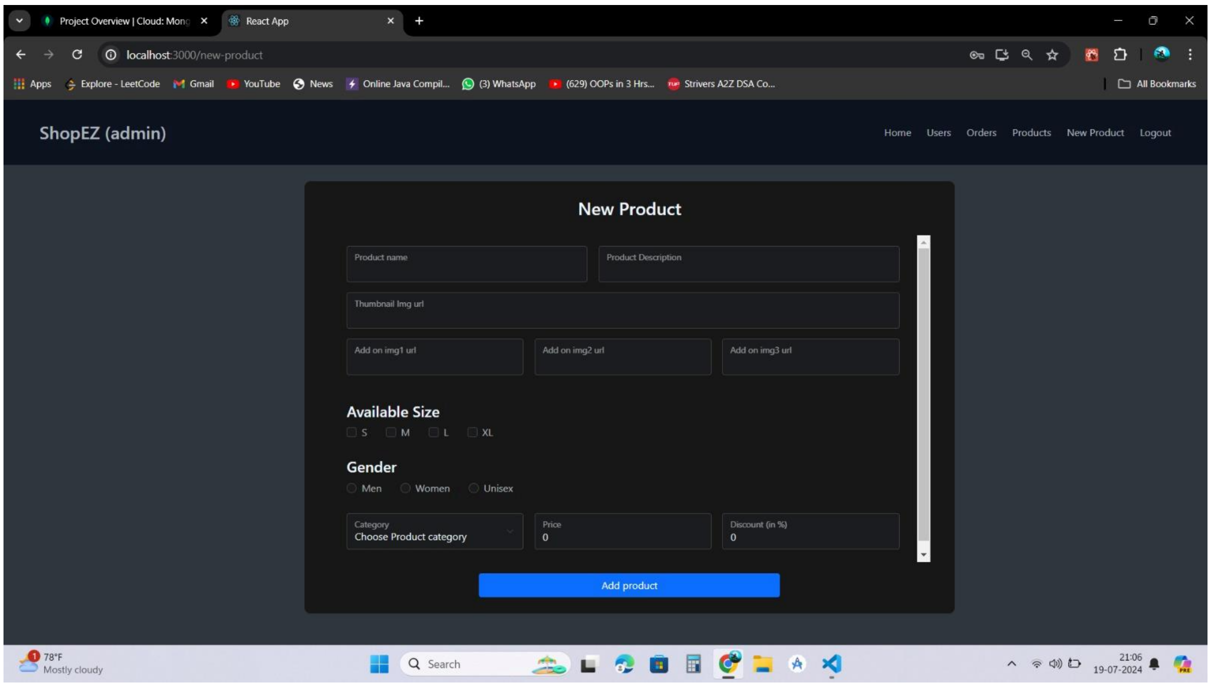
Size: **Quantity:** 1 **Price:** ₹ 15010 **Payment method:** card

Address: 14/2 DURAI SWAMY RAJA STREET, KASPA A, Ambur **Pincode:** 635802 **Ordered on:** 2024-07-17

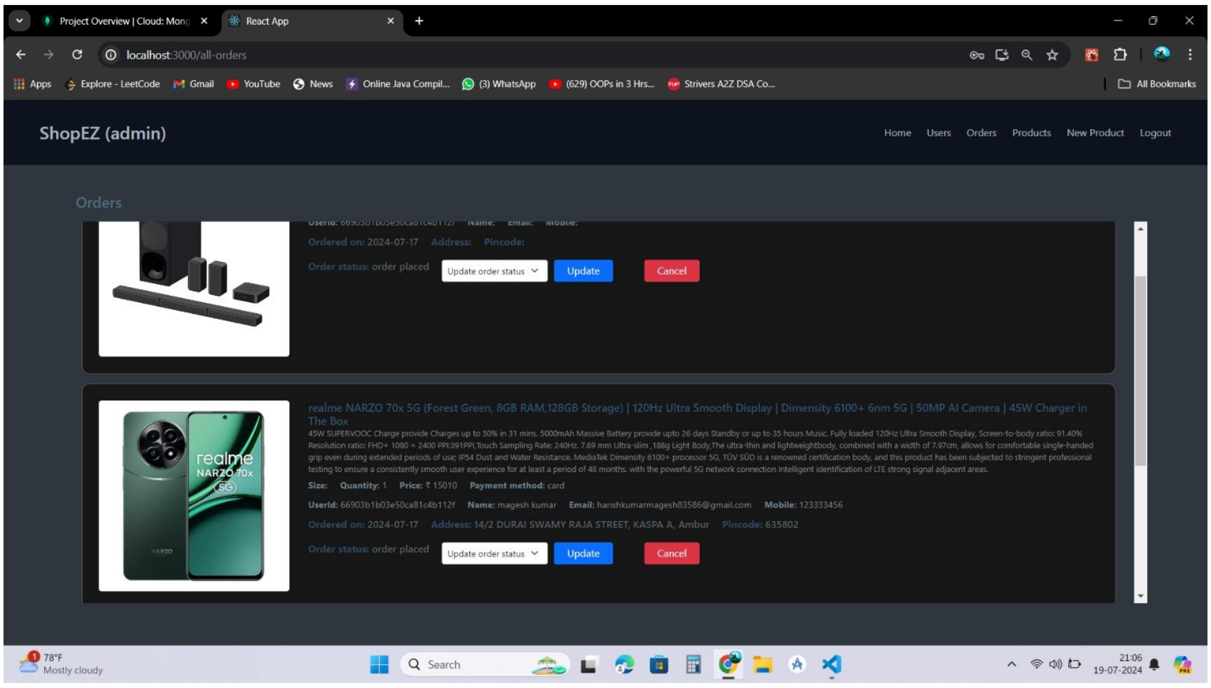
- Admin UI



- New Product adding



- Update order status



ShopEZ (admin)

Home Users Orders Products New Product Logout

Orders

Ordered on: 2024-07-17 **Address:** **Pincode:**

Order status: order placed

realme NARZO 70x 5G (Forest Green, 8GB RAM,128GB Storage) | 120Hz Ultra Smooth Display | Dimensity 6100+ 6nm 5G | 50MP AI Camera | 45W Charger in The Box

45W SUPERVOOC Charge provide Charges up to 50% in 31 mins. 5000mAh Massive Battery provide upto 26 days Standby or up to 35 hours Music. Fully loaded 120Hz Ultra Smooth Display, Screen-to-body ratio: 91.40% Resolution ratio: FHD+ 1080 × 2400 PPI:391PPI,Touch Sampling Rate: 240Hz. 7.69 mm Ultra-slim ,188g Light Body,The ultra-thin and lightweightbody, combined with a width of 7.97cm, allows for comfortable single-handed grip even during extended periods of use; IP54 Dust and Water Resistance. MediaTek Dimensity 6100+ processor 5G, TÜV SÜD is a renowned certification body, and this product has been subjected to stringent professional testing to ensure a consistently smooth user experience for at least a period of 48 months. with the powerful 5G network connection Intelligent identification of LTE strong signal adjacent areas.

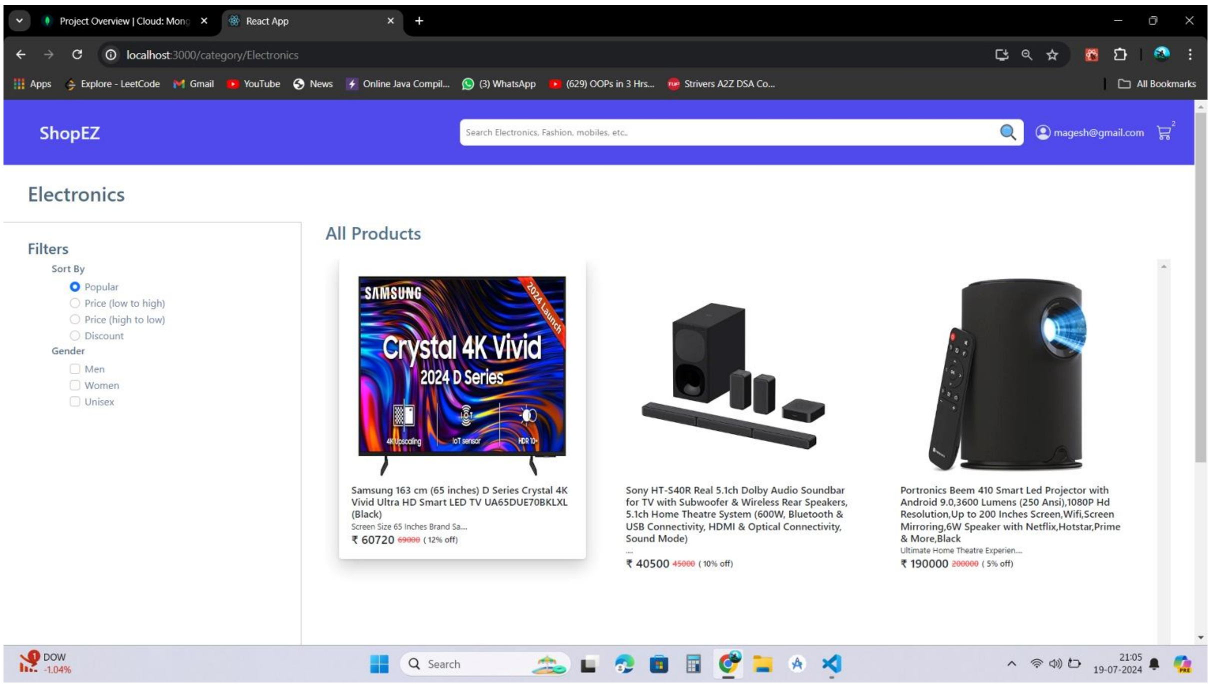
Size: **Quantity:** 1 **Price:** ₹ 15010 **Payment method:** card

UserId: 66903b1b03e50ca81c4b112f **Name:** magesh kumar **Email:** harishkumarmagesh83586@gmail.com **Mobile:** 123333456

Ordered on: 2024-07-17 **Address:** 14/2 DURAI SWAMY RAJA STREET, KASPA A, Ambur **Pincode:** 635802

Order status: order placed

- Electronics Category



ShopEZ

Electronics

Filters

Sort By

[✓ Popular] Price (low to high) Price (high to low) Discount

Gender

Men Women Unisex

All Products

Samsung 163 cm (65 inches) D Series Crystal 4K Vivid Ultra HD Smart LED TV UA65DUE70BKLXL (Black)
Screen Size 65 Inches Brand Sa....
₹ 60720 ~~69000~~ (12% off)

Sony HT-S40R Real 5.1ch Dolby Audio Soundbar for TV with Subwoofer & Wireless Rear Speakers, 5.1ch Home Theatre System (600W, Bluetooth & USB Connectivity, HDMI & Optical Connectivity, Sound Mode)
....
₹ 40500 ~~45000~~ (10% off)

Portronics Beem 410 Smart Led Projector with Android 9.0,3600 Lumens (250 Ansi),1080P Hd Resolution,Up to 200 Inches Screen,Wifi,Screen Mirroring,6W Speaker with Netflix,Hotstar,Prime & More,Black
Ultimate Home Theatre Experien....
₹ 190000 ~~200000~~ (5% off)

10. Testing

Unit Testing:

Backend:

Jest: A testing framework for JavaScript applications, ideal for unit testing functions and modules.

Frontend:

Jest: Included by default with Create React App for unit testing React components.

React Testing Library: For testing React components with a focus on testing user interactions and component behavior.

Integration Testing:

Backend:

Supertest: Used in combination with Jest to test API endpoints and ensure they function correctly when integrated with each other.

Frontend: React Testing Library: Also used for integration tests, verifying the interaction between different components.

API Testing:

Backend:

Postman: For manual and automated API testing. Postman allows you to send requests to your API endpoints and validate responses. You can also create test scripts in Postman to automate and validate API responses.

11. Screenshots or Demo

https://drive.google.com/drive/folders/1JnnI5RVgfO-VsIvSREV5rRFNhyj5mNhu

12. Known Issues

BG-001: Product Images Not Loading

- **Description**: Some product images fail to load, affecting the browsing experience for users.

BG-002: Payment Gateway Not Processing Payments

- **Description**: Users are unable to complete purchases due to payment processing failures.

BG-003: Push Notifications Not Received

- **Description**: Users are not receiving push notifications for order updates or promotions.

BG-004: Checkout Page Crashes

- **Description**: The checkout page occasionally crashes, preventing users from completing their purchases.

13. Future Enhancements

1. Personalized Product Recommendations

- **Description**: Implement a recommendation engine that uses machine learning to suggest products based on users' browsing and purchase history. This will enhance the user experience by providing personalized shopping suggestions, potentially increasing sales.

2. Enhanced Search Functionality with AI

- **Description**: Upgrade the search feature with AI-powered capabilities such as natural language processing and voice search. This will make it easier for users to find products using conversational language or voice commands, improving the overall usability of the platform.

3. Augmented Reality (AR) for Product Visualization

- **Description**: Introduce AR features that allow users to visualize products in their real-world environment. This is particularly useful for items like furniture, home decor, and fashion, enabling users to see how products would look and fit before making a purchase.

4. Loyalty Program Integration

- **Description**: Develop a comprehensive loyalty program that rewards customers with points for purchases, reviews, and referrals. This can include tiered membership levels with exclusive benefits and discounts, fostering customer retention and encouraging repeat purchases.

5. Advanced Analytics Dashboard for Vendors

- **Description**: Provide vendors with an advanced analytics dashboard that offers detailed insights into their sales performance, customer behavior, inventory levels, and marketing campaign effectiveness. This will empower vendors to make data-driven decisions and optimize their operations on the platform.