# MERN stack powered by MongoDB

# Naan Mudhalvan - Project Documentation

## Introduction

**Project Title:** E-Commerce Platform **(**ShopEZ)

**College:** Tagore Engineering College – 4127

**Department:** B.E CSE IV Year 7th Semester

**Team ID:** NM2024TMID02249

**Team Members:**

| S.No | Student Name | Register No. | Naan Mudhalvan ID | Role |
|---|---|---|---|---|
| **1.** | Prem R | 412721104036 | B99583D90CD57618BA84DA26921DAB85 | Backend |
| **2.** | Arivalan P | 412721104006 | 31B86F3578BF431940C4E458E7EB4DFA | Backend |
| **3.** | Naveen K | 412721104031 | 8E5980977AA96DB895D498D8DEDE72C9 | Frontend |
| **4.** | Sibiraj P | 412721104046 | DF5A04453EFE476BA6148DFC9DC16150 | Frontend |
| **5.** | Sivabalan K | 412721104047 | 0E83537C4DF00D58B7D4EAD13550E4F8 | Frontend |

**Project Overview**

**Purpose**: Our E-Commerce Platform is designed to provide a seamless and engaging shopping experience for users, along with a comprehensive and efficient management interface for administrators. It offers customers a wide range of products, while enabling administrators to manage inventory, track orders, and analyze sales data.

**Features**:

**Customer/User**:

- **Signup/Login**:
  - User registration with email.
  - Login with email and password, with a “Remember Me” option.
- **Product Browsing and Filtering**:
  - View all available products with filter, and sorting capabilities by category, price, and popularity.
  - View detailed product information, including images, descriptions, specifications, and reviews.
- **Wish list and Cart Management**:
  - Add products to the wish list for future reference.
  - Add products to the cart, view, and modify quantities.
  - Save items in the cart for later, with cart persistence across sessions.

- **Order Placement and Tracking**:
  - Secure checkout process with address input, order summary, and delivery options.
  - Order tracking with status updates and estimated delivery times.
- **User Profile**:
  - Editable profile information except for email (name, contact, address, etc.).
  - Order history view with downloadable receipts.

**Admin**:

- **User and Product Management**:
  - View, add, edit, and delete products and categories.
  - Manage users, with the ability to view user profiles and delete users if needed.
- **Order Management**:
  - View and update order statuses, including processing, shipping, and delivery.
  - Generate invoices and send order notifications.
- **Inventory and Sales Analytics**:
  - Track inventory levels, set low-stock alerts, and restock products as necessary.
  - View sales data, including sales volume, top products, revenue trends, and customer purchase behavior.
- **Promotions and Discounts**:
  - Create, update, and manage discount codes, sales events, and promotional offers.

**Architecture**

**Frontend**:

- **Developed using React** with responsive design and user-friendly interface, optimized for smooth navigation.
- **Bootstrap for styling** and layout, enhancing mobile and tablet responsiveness.

**Backend**:

- **Node.js and Express.js** to handle business logic and manage API endpoints for product listing, user actions, and order processing.
- **Middleware includes**:
  - **bcryptjs** for secure password hashing.
  - **jsonwebtoken** for user authentication and authorization.
  - **Multer** for managing image uploads and handling product image files.

**Database**:

- **MongoDB** serves as the database solution for storing all data, including user profiles, product listings, orders, and reviews.
- **Mongoose** ORM (Object-Relational Mapping) is used for defining schemas and handling data interactions efficiently.

This architecture and feature set ensure a robust, flexible, and scalable platform that caters to customer needs and streamlines administrative tasks, with a responsive, high-performance user experience.

## Setup Instructions

### Prerequisites:

- Node.js
- MongoDB Atlas Account.

### Installation:

1. **Clone the repository:**

   git clone cd online-learning-platform

2. **Install dependencies:**
   - **Frontend:**

     cd client

     npm install
   - **Backend:**

     cd server

     npm install
3. **Set up environment variables:**
   - Create **.env** file in the **server** folder with the following:

     **MONGODB_CONNECT_URL**='mongodb+srv://radhaprem:prem123@cluster0.yizr3.mongodb.net/?retryWrites=true&w=majority'

     **FRONTEND_URL**= http://localhost:3000
   - Create .**env.local** file in the **client** folder with the following:

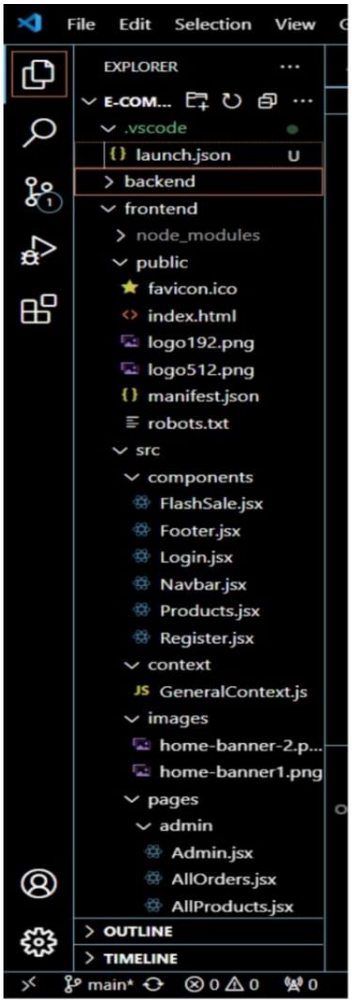
     **BACKEND_URL**= http://localhost:6001

## Folder Structure

### Client (Frontend):

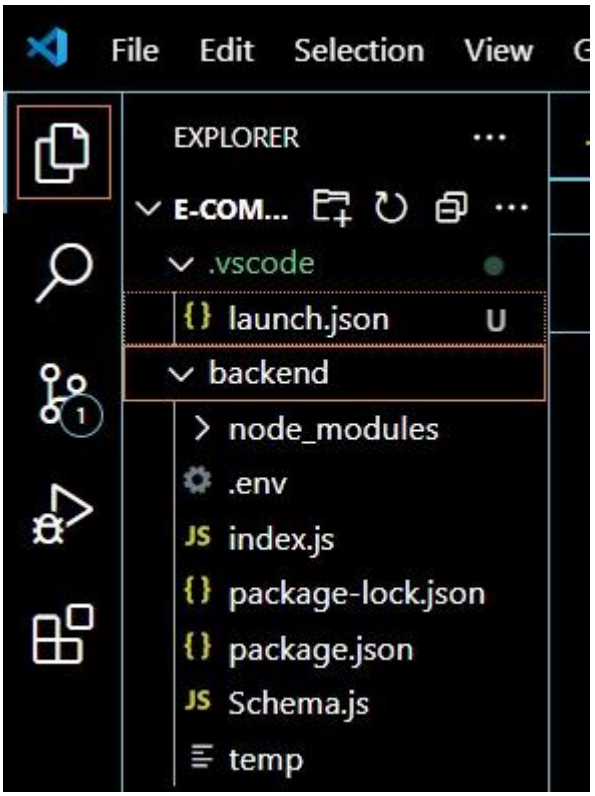
- **src/:** Contains all React components, pages, and assets.
  - **Components/:** Reusable UI components such as Footer, Header, Header Student and Sidebar

- **Pages/:** Folders such as Admin, User, Instructor and Navs containing Pages for different aspects of the application.



**Server (Backend):**

- **routes/:** Defines API routes for users, instructors, forums, and admin.
- **models/:** MongoDB schemas for Users, Courses, Instructor, Thread and OTP.
- **middleware/:** Authentication using jwt token.



## Running the Application

- **Frontend:**

  cd client

  npm start

- **Backend:**

  cd server

  node index.js

**User Routes:**

**POST** /register: Register a new user.
**POST** /login: Login an existing user.
**GET** /fetch-users: Fetch all registered users.

**Admin Routes:**

**GET** /fetch-banner: Fetch the banner content.
**POST** /update-banner: Update the banner content.
**GET** /fetch-categories: Fetch all categories.
**POST** /add-new-product: Add a new product.
**PUT** /update-product/:id: Update an existing product by ID.

**Product Routes:**

GET /fetch-products: Fetch all products.
GET /fetch-product-details/:id: Fetch details of a specific product by ID.

**Order Routes:**

GET /fetch-orders: Fetch all orders.
POST /buy-product: Place an order for a specific product.
PUT /cancel-order: Cancel an order.
PUT /update-order-status: Update the status of an order.
POST /place-cart-order: Place orders for all items in the cart.

**Cart Routes:**

GET /fetch-cart: Fetch all items in the cart.
POST /add-to-cart: Add a new item to the cart.
PUT /increase-cart-quantity: Increase the quantity of a cart item by ID.
PUT /decrease-cart-quantity: Decrease the quantity of a cart item by ID.
PUT /remove-item: Remove a specific item from the cart by ID.

## Authentication:

**JWT (JSON Web Tokens)**:

- **Token Generation**: Tokens are generated upon successful login, allowing secure access to user-specific features.
- **Middleware Validation**: Middleware checks token validity for protected routes, ensuring secure access control.

- **Email Verification**:
  - During signup, a verification link is sent to the user's email using **Node mailer** to verify the account.

## User Interface

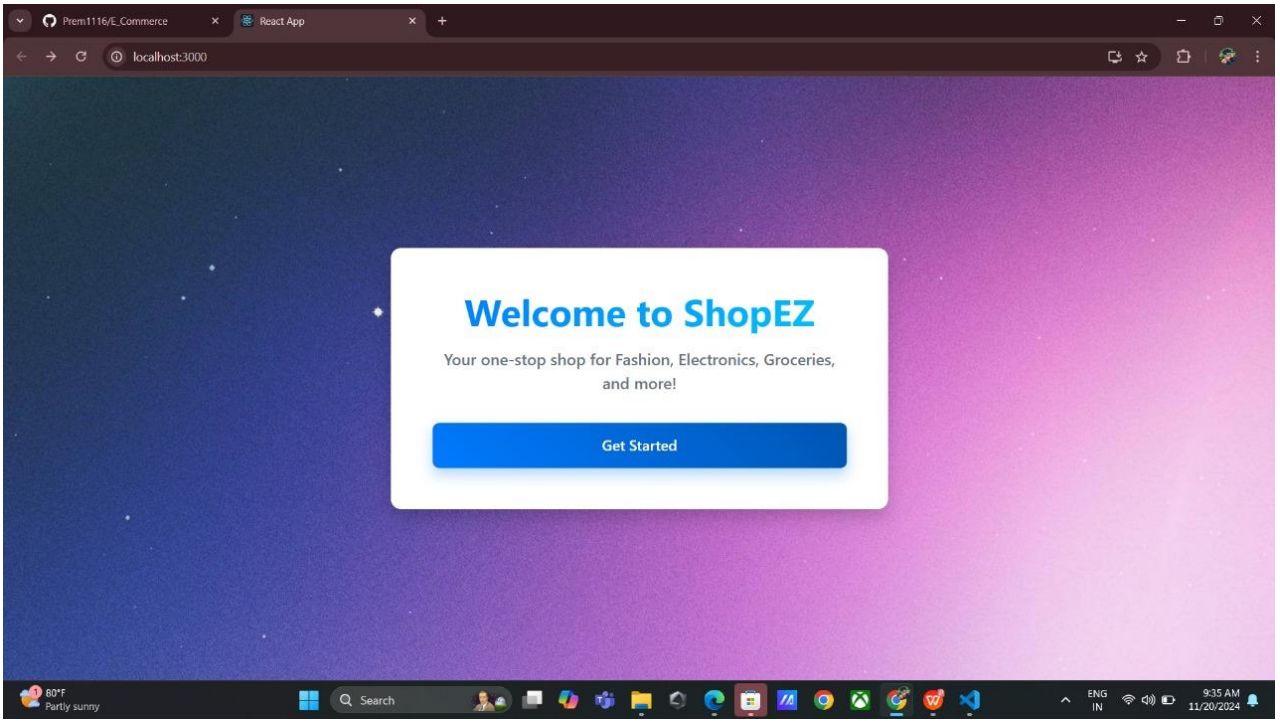
- **Customer Dashboard**:
  - A personalized dashboard where users can view their profile, order history, and wish list.
  - Accessible options to update profile details (excluding email) and manage orders.
  - View cart items, track orders, and add products to the wish list for future purchases.
- **Admin Dashboard**:
  - A streamlined interface that enables admin users to manage all products, orders, categories, and users.
  - Tools to add, edit, and delete products and categories, monitor stock levels, and view user profiles.
  - Admin-specific reports on sales, inventory, and user activity.

## Testing

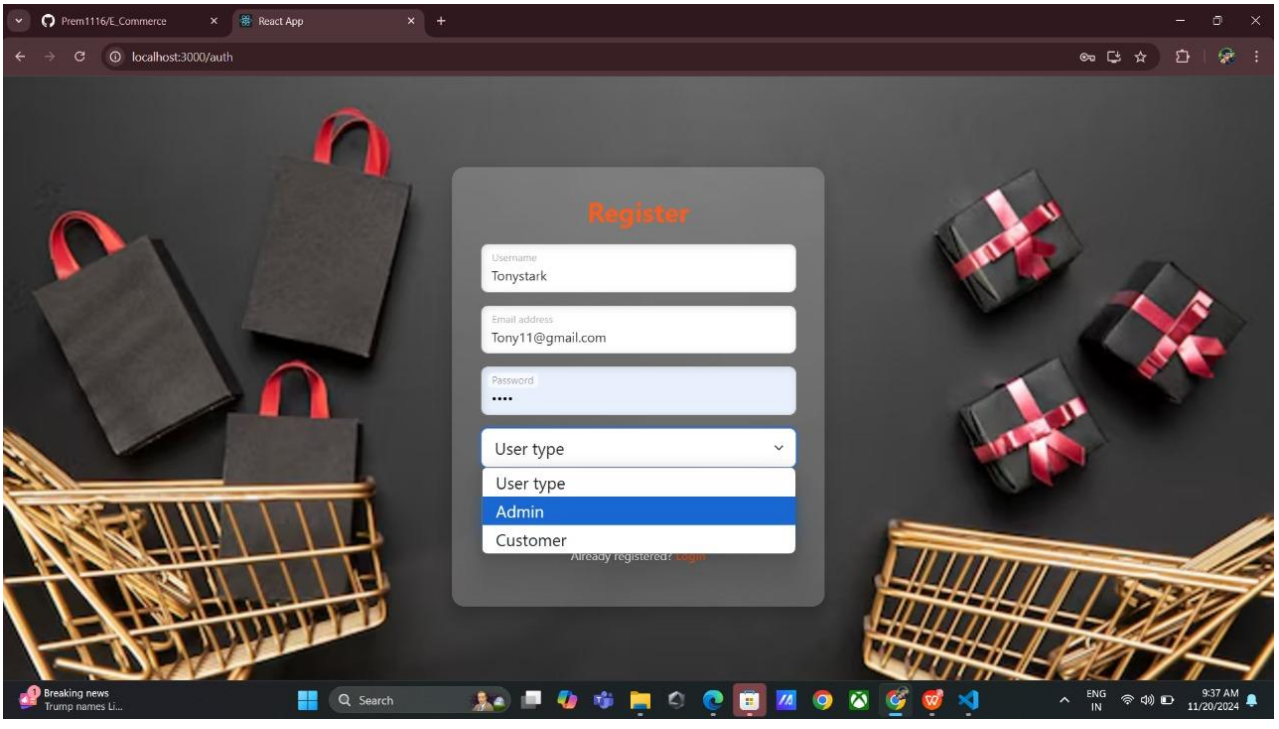
- **Manual Testing**:
  - Verified all major functionalities, including user signup/login, product search and filtering, cart management, order placement, and order tracking.
- **Postman Testing**:
  - Used for API endpoint testing, covering essential operations such as signup/login, product CRUD operations, cart updates, and order handling.

## Screenshots:

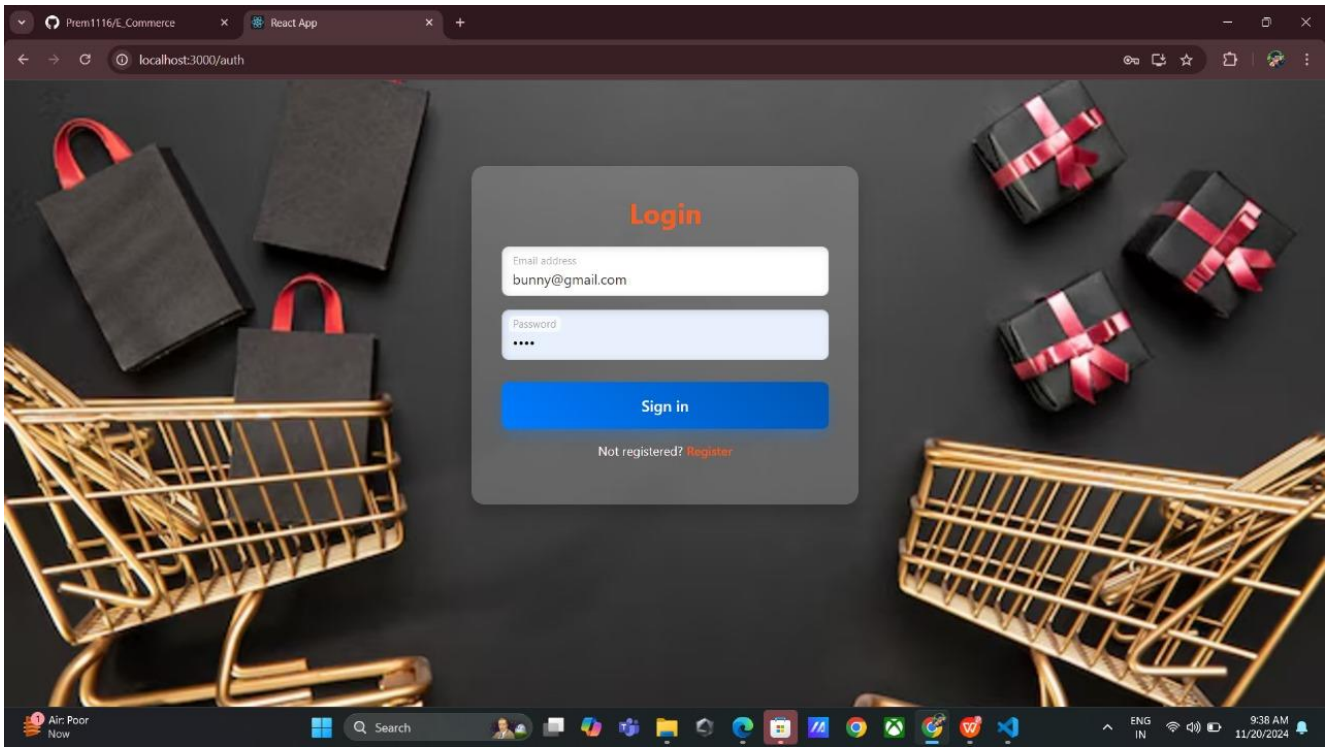
### 1. Landing Page



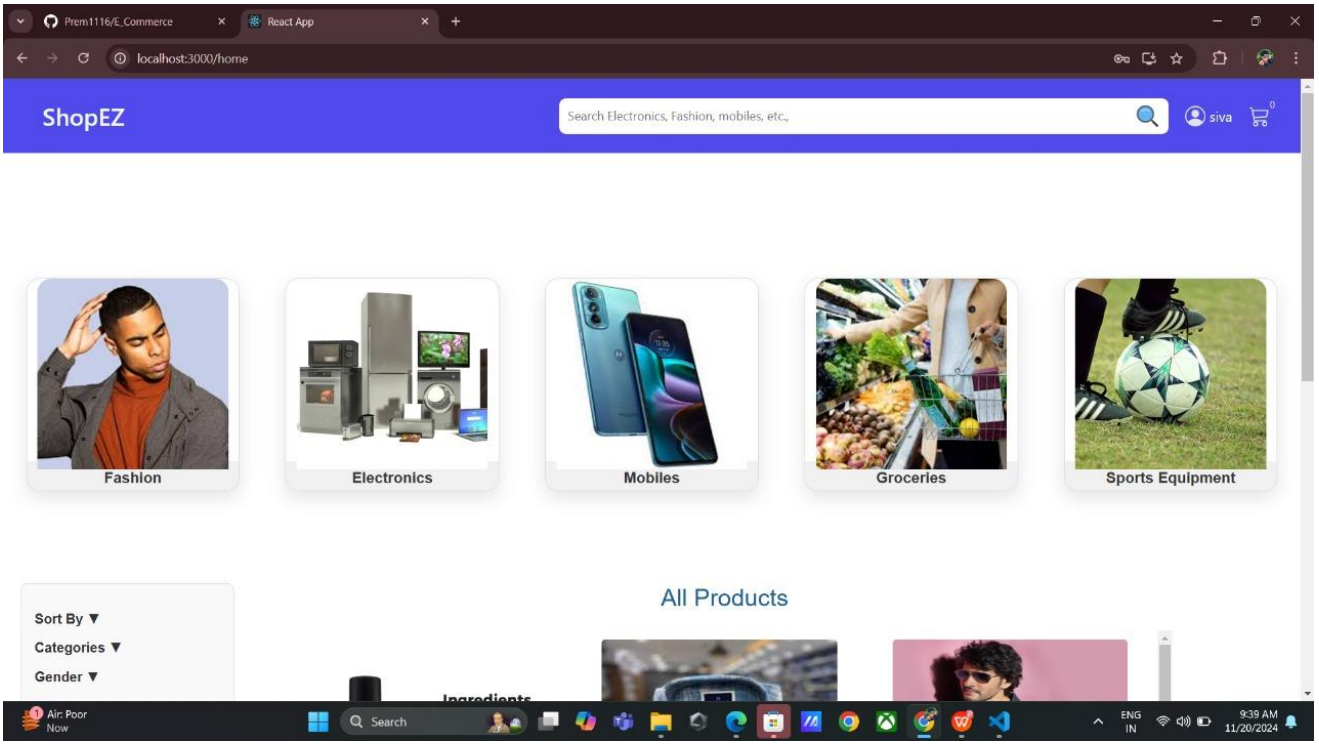

### 2. Register Page (customer and admin)



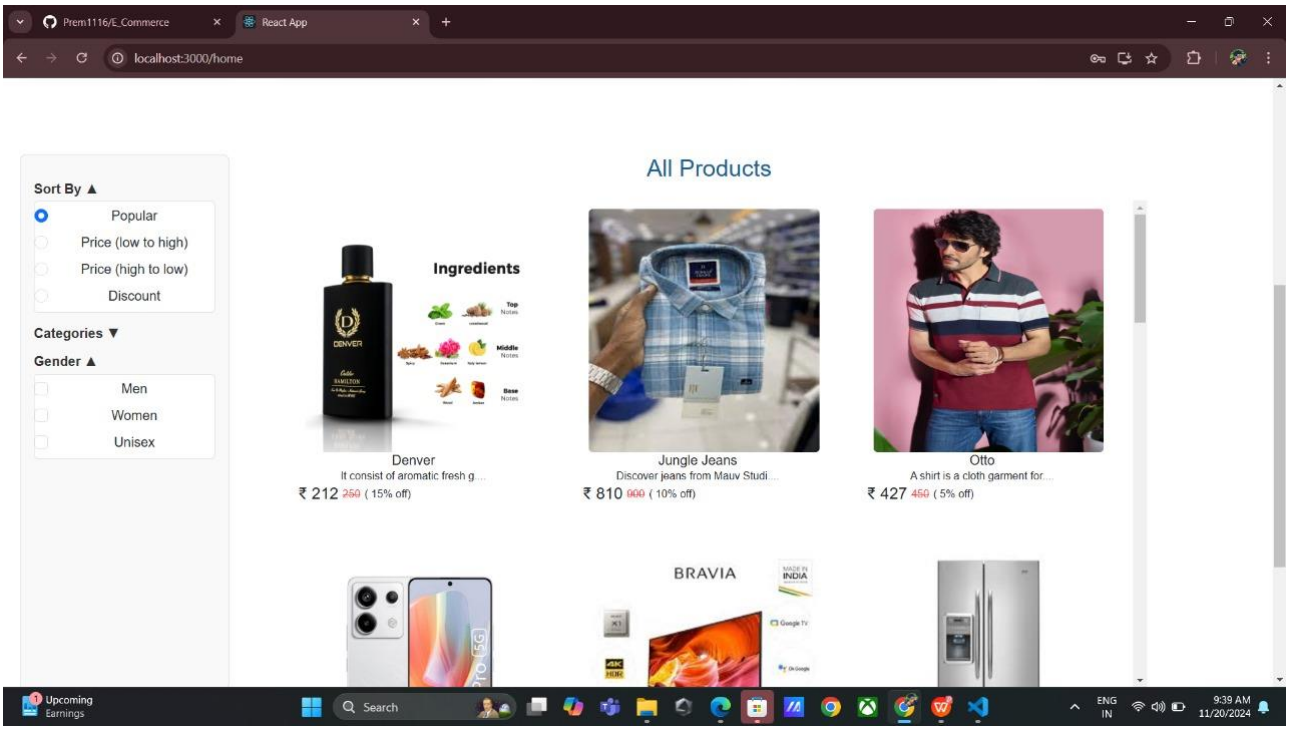
### 3. User Login



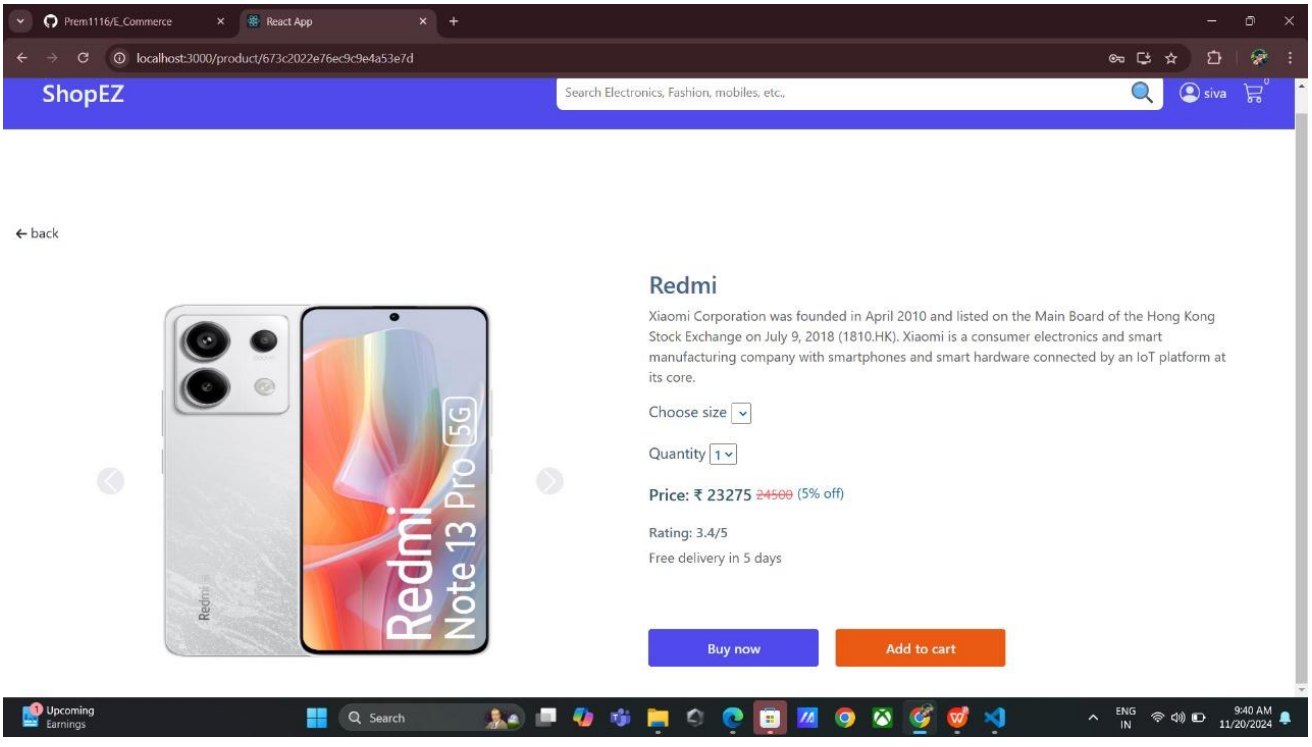

### 4. User Dashboard

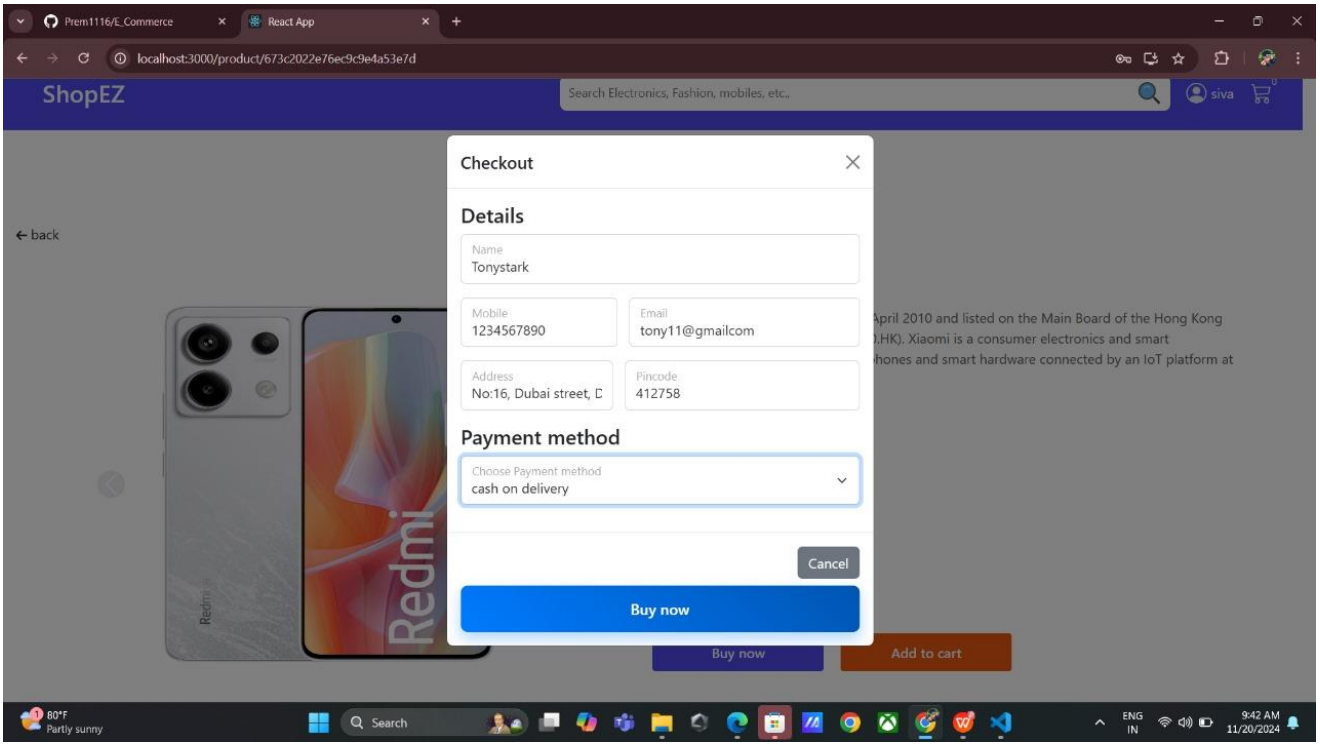


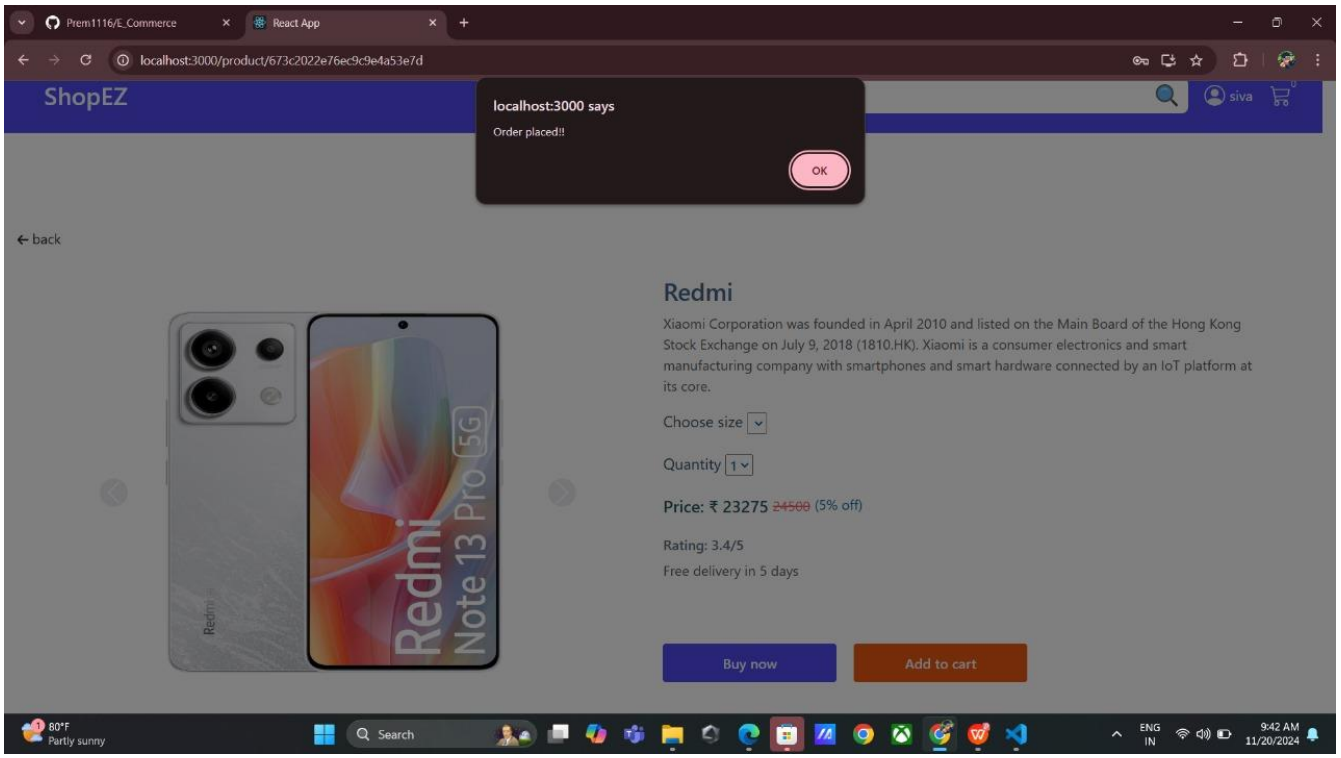
- **All Products - section with Categories**



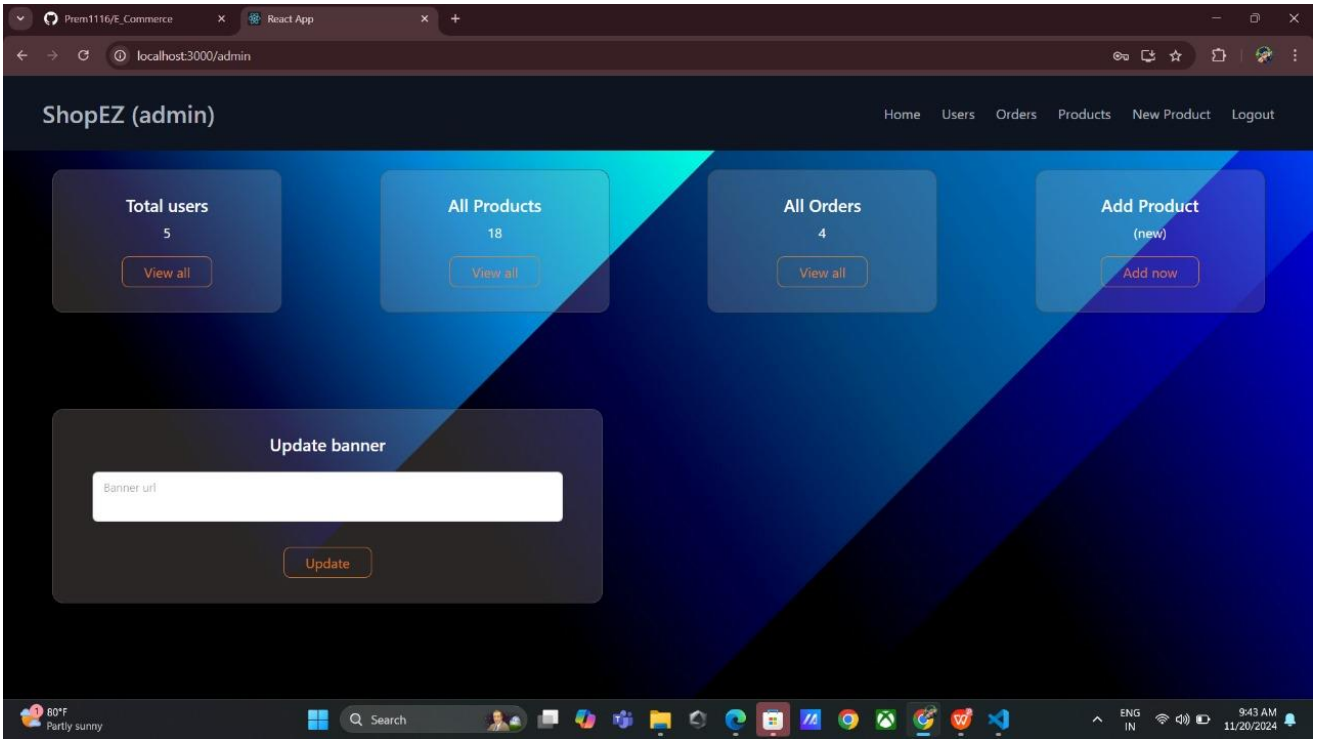

## 5. Placing Order



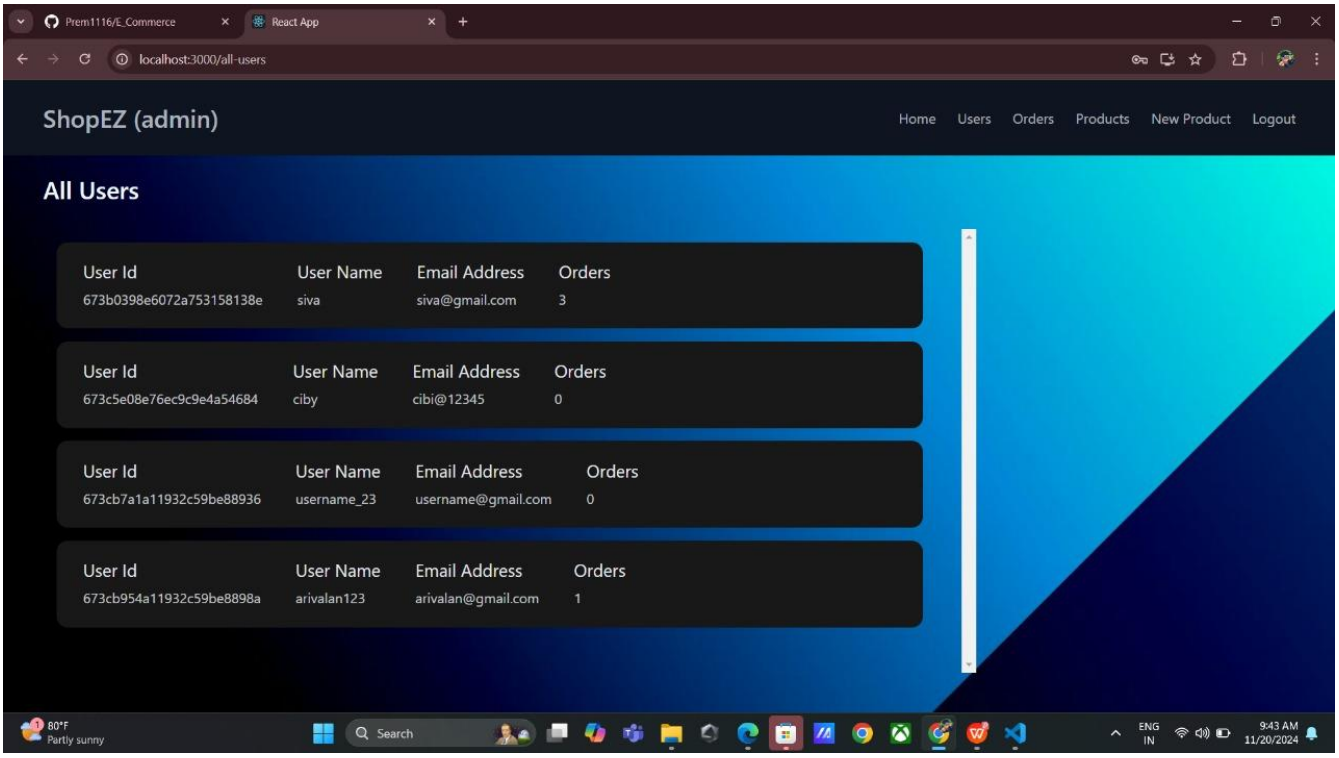

Prem1116/E_Commerce
React App
localhost:3000/product/673c2022e76ec9e4a53e7d
ShopEZ
Search Electronics, Fashion, mobiles, etc.
siva
Checkout
Details
Name
Tonystark
Mobile
1234567890
Email
tony11@gmailcom
Address
No:16, Dubai street, D
Pincode
412758
Payment method
Choose Payment method
cash on delivery
Cancel
Buy now
← back
Redmi
Buy now
Add to cart
80°F
Partly sunny
Search
ENG
IN
9:42 AM
11/20/2024


Prem1116/E_Commerce
React App
localhost:3000/product/673c2022e76ec9e4a53e7d
ShopEZ
localhost:3000 says
Order placed!!
OK
siva
← back
Redmi
Note 13 Pro 5G
Redmi
Xiaomi Corporation was founded in April 2010 and listed on the Main Board of the Hong Kong Stock Exchange on July 9, 2018 (1810.HK). Xiaomi is a consumer electronics and smart manufacturing company with smartphones and smart hardware connected by an IoT platform at its core.
Choose size
Quantity 1
Price: ₹ 23275 24500 (5% off)
Rating: 3.4/5
Free delivery in 5 days
Buy now
Add to cart
80°F
Partly sunny
Search
ENG
IN
9:42 AM
11/20/2024

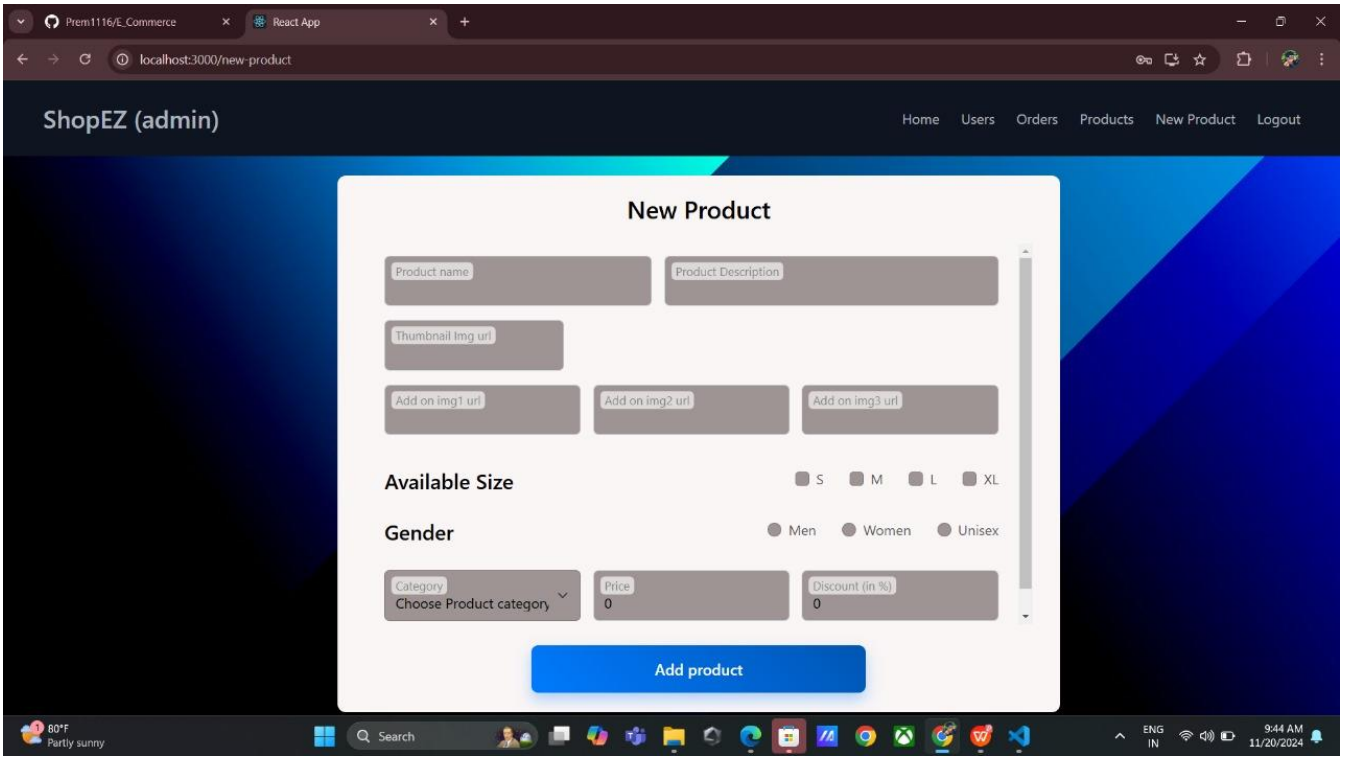
## 6. Admin Dashboard

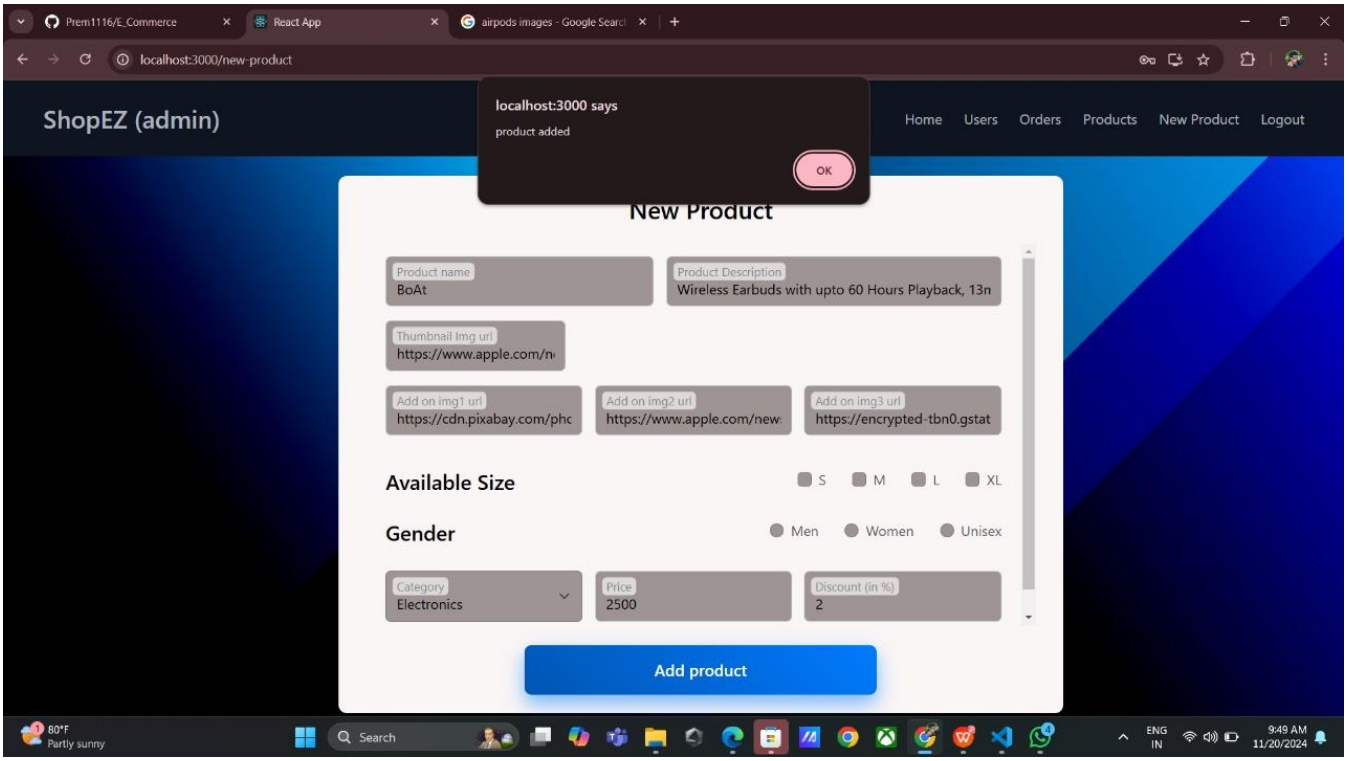



- **View users' login details**

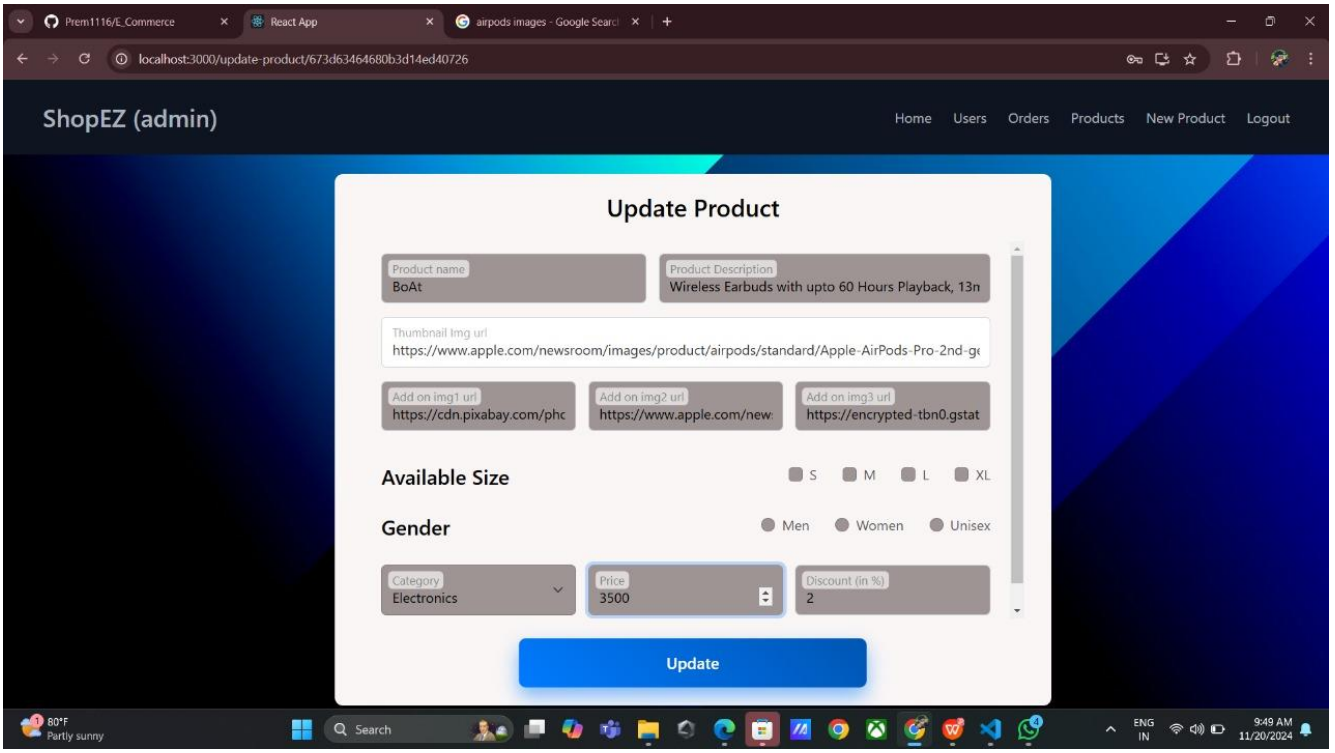


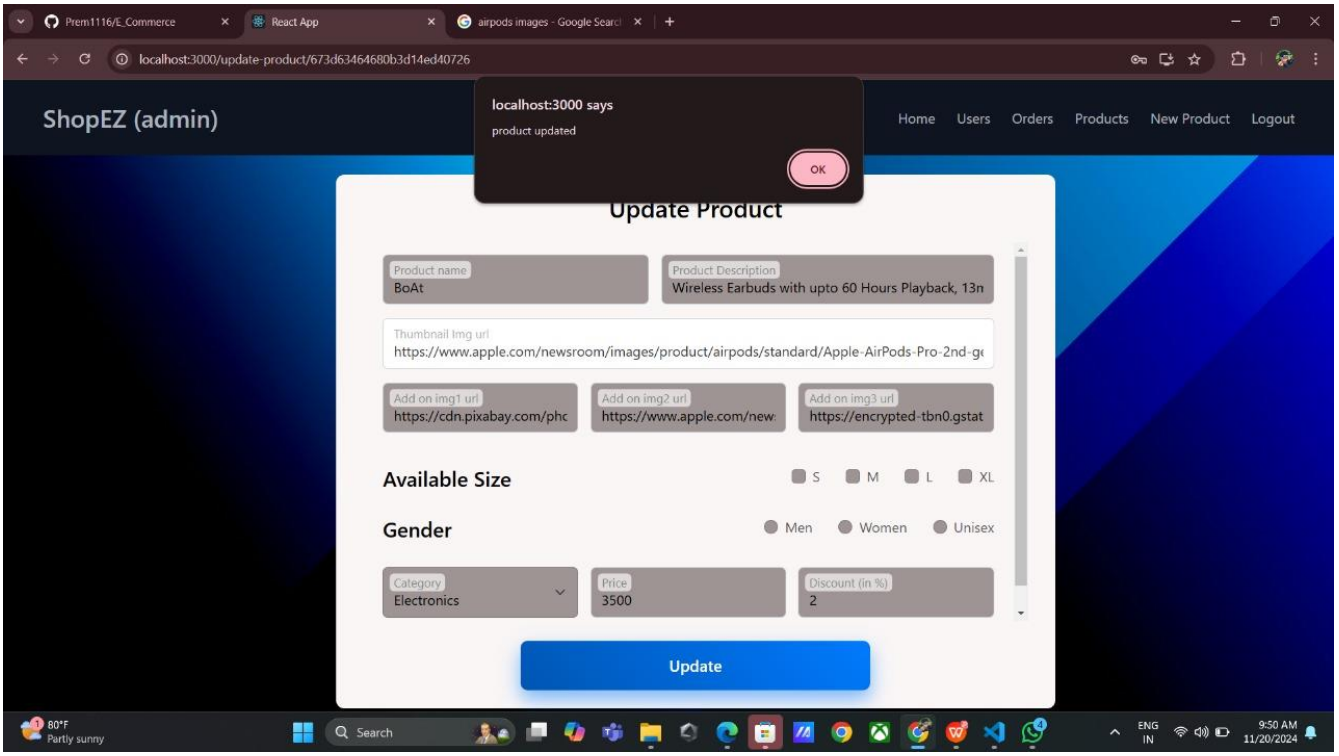
- **Adding a new product from Admin side**



## 7. Update Product



## Known Issues

- **Admin Login**: Currently, the admin panel lacks a dedicated login system, accessed only through a protected route.
- **Error Handling for File Uploads**: There is limited error handling for invalid product image or document uploads.
- **No Payment Gateway**: Due to project requirements, the platform currently lacks a real payment gateway, and checkout is replaced with a mock payment process.
- **Profile and Product Editing Limitations**: When editing product details or user profiles, the platform doesn't automatically fetch the previously uploaded images or documents, requiring users to re-upload them.

## Future Enhancements

- **Payment Gateway Integration**: Implement a secure payment gateway (e.g., Stripe, PayPal) to facilitate actual transactions during checkout.
- **Advanced Analytics for Admin**: Add in-depth analytics and reporting for the admin, including sales metrics, product performance, and customer insights.
- **Recommendation Engine**: Incorporate machine learning algorithms to recommend products based on user preferences, browsing history, and purchase patterns.
- **Live Customer Support via WebRTC**: Integrate WebRTC to enable live chat or video support for customers seeking assistance in real time.

## Video Demo Link:

https://drive.google.com/file/d/1chJoFQiLUsDbOXK1UxDU24AeN4x_AwHk/view?usp=sharing