# Meenakshi College Of Engineering

12, Vembuliamman Koil St, Annaji Nagar, KK Nagar West, Virugambakkam, Chennai, Tamil Nadu 600078 ( Affiliated to Anna University )



Naan mudhalvan - MERN stack powered by Mongo DB

Project Tittle

E-commerce application – using MERN

Submitted by

**Praveen kumar R 311421205065**

**Sreenivasan V S 311421205085**

**Karthikeyan M 311421205302**

**Thirumurugan M P 311421205096**

**Shop EZ: E-commerce**

**Application**

**-using mern**

**Table of Contents**

1. Introduction
2. Project Overview
3. Architecture
4. Setup Instructions
5. Folder Structure
6. Running the Application
7. API Documentation
8. Authentication
9. User Interface
10. Testing
11. Known Issues
12. Future Enhancements
13. Conclusion

1. **Introduction**

**Project Title**: *The Shop EZ – E-Commerce application*

**Project Description**:

*The Shop EZ is an e-commerce platform designed to provide a seamless and enjoyable shopping experience for users while empowering sellers to efficiently manage and grow their online businesses. Developed with the MERN stack (MongoDB, Express.js, React, and Node.js), The Shop EZ harnesses modern, full-stack web technologies to create a dynamic, secure, and scalable application that meets the needs of a fast-evolving e-commerce landscape. This platform caters to both small and large product catalogs, making it suitable for sellers of all sizes looking to reach a diverse audience of online shoppers.*

*The e-commerce industry has seen rapid growth, with a continually increasing demand for convenience, security, and variety in shopping experiences. Recognizing these needs, The Shop EZ was designed to address two primary audiences: consumers seeking a wide-ranging, user-friendly shopping environment and sellers requiring robust tools for product and sales management. Through an architecture built on modularity and scalability, The Shop EZ ensures high performance across user journeys, from browsing products to completing secure transactions. Its responsive design and user-friendly interface enhance accessibility across different devices, including desktops, tablets, and mobile phones, ensuring a consistent experience for every shopper.*

***Key Goals and Objectives***

*The primary goal of The Shop EZ is to deliver an end-to-end e-commerce solution that emphasizes usability, security, and flexibility. For users, this means a simple and intuitive interface for browsing, searching, and purchasing products. By creating a responsive, visually appealing layout and equipping it with features like a comprehensive catalog and easy navigation, The Shop EZ enables users to quickly find and purchase items with minimal friction.*

*Security and privacy are fundamental priorities, especially in an age where data security is paramount. The Shop EZ is built with secure user authentication managed through JSON Web Tokens (JWT), ensuring safe access to sensitive data and personalized features. The platform also integrates with secure payment gateways, allowing users to complete purchases confidently and safeguarding all personal information.*

*On the other hand, The Shop EZ provides sellers with tools to streamline their product and order management processes. Sellers can list products, update inventory, and view sales analytics—all through a dedicated seller dashboard. This tool serves as a central hub for managing product details, tracking low inventory items, monitoring customer engagement, and making data-driven decisions. Additionally, the platform’s architecture is scalable, meaning it can handle an increasing volume of traffic and data without impacting performance, making it ideal for sellers who are aiming to grow their customer base.*

***Platform Highlights***

*Some of the standout features that support these goals include:*

* ***Comprehensive Product Catalog****: Users can explore a wide range of products across categories with clear pricing, descriptions, and user reviews. This catalog is searchable and filterable, offering a tailored experience for users to find exactly what they need.*
* ***Streamlined Checkout Process****: With an easy-to-navigate checkout flow, The Shop EZ ensures a minimal number of steps from cart to payment, reducing cart abandonment and enhancing user satisfaction.*
* ***Order Tracking****: Users can track orders from purchase to delivery with real-time notifications, promoting transparency and building user trust.*
* ***Responsive and Accessible Design****: Built with React, the frontend is fully responsive, supporting accessibility across devices and ensuring a smooth experience for all users.*
* ***Enhanced Security and Privacy Measures****: The platform incorporates best practices in data protection, ensuring secure communication and storage of user information.*

*The Shop EZ aims to create a balanced ecosystem where both users and sellers are fully supported through every stage of their e-commerce journey. By emphasizing intuitive design, secure transactions, and efficient seller management, the platform strives to build a user-centric e-commerce solution adaptable to future growth and changing market demands.*

**2. Project Overview**

**Purpose**:

*The Shop EZ* is a comprehensive e-commerce application aimed at delivering a smooth and satisfying shopping experience for online shoppers, while also empowering sellers with the tools they need to effectively manage and scale their online stores. Built on the MERN stack (MongoDB, Express.js, React, and Node.js), *The Shop EZ* leverages modern web technologies to create a platform that is highly scalable, modular, and responsive. The primary objective of this project is to develop an end-to-end solution that addresses the needs of both consumers looking for a diverse product catalog and sellers seeking an efficient way to reach online customers.

E-commerce is rapidly growing and evolving, and *The Shop EZ* aims to keep pace by providing a solution that emphasizes ease of use, security, and flexibility. This platform has been developed with a focus on creating an accessible, user-friendly interface for both end-users and sellers, alongside an efficient backend infrastructure to support seamless transactions, data integrity, and system scalability. By focusing on the goals of a seamless user experience, secure payment processes, and an informative seller dashboard, *The Shop EZ* strives to offer a balanced e-commerce platform that caters to all stakeholders in the e-commerce ecosystem.

**Key Features**:

Below are the core features of *The Shop EZ*, which cater to the needs of both shoppers and sellers.

**1. Comprehensive Product Catalog**

*The Shop EZ* hosts an extensive product catalog that spans multiple categories, allowing users to explore a wide range of items from electronics to clothing, home goods, and more. Each product listing is meticulously detailed, featuring high-quality images, clear pricing information, detailed descriptions, and user reviews. The catalog structure allows users to easily browse, search, and filter products based on various criteria such as price, brand, and ratings, enhancing product discoverability and helping users find exactly what they’re looking for.

For sellers, the catalog is fully customizable through the seller dashboard. Sellers can manage product details, update images, and set promotional discounts, all of which reflect in real-time on the customer interface. This dual functionality ensures that both customers and sellers experience a dynamic and informative catalog.

**2. Shop Now Button and Quick Order Access**

Every product in the catalog has a convenient “Shop Now” button, enabling users to swiftly begin the purchasing process without needing to navigate to another page. When a user selects a product, the system takes them directly to an order details page, where they can specify shipping addresses, choose payment methods, and include any particular product specifications if needed. This approach optimizes the user journey by reducing clicks and offering an immediate path to purchase, contributing to a smoother shopping experience and higher conversion rates.

**3. Order Details Page and Confirmation**

The order details page provides users with a comprehensive overview of their purchase, including shipping information, payment method, and any additional options selected during checkout. Once an order is placed, the user receives a confirmation notification that serves as an immediate acknowledgment of the transaction. This page also allows users to double-check all pertinent details before confirming, reducing errors and enhancing customer satisfaction.

The confirmation page ensures that users receive prompt feedback on their orders and can access a summary at any time. Additionally, users have the ability to view past order details in their account, which provides a complete purchase history and contributes to transparency and trust.

**4. Secure and Efficient Checkout Process**

Security and ease of use are top priorities for the checkout process in *The Shop EZ*. Personal and payment information are processed through secure, encrypted channels, ensuring that sensitive data is safeguarded. The platform integrates with popular payment gateways, providing users with multiple payment options for a flexible and convenient experience. The checkout process has been optimized to ensure a minimal number of steps, reducing cart abandonment rates and enhancing overall user satisfaction.

**5. Order Tracking and Notifications**

After completing an order, users can track their order status from the time of purchase until delivery. The system provides real-time updates on order processing, packaging, and shipping, giving users a clear view of where their order is at every stage. Notifications are sent via email or in-app messages, keeping users informed of their order’s progress. This feature enhances the customer experience by offering transparency and peace of mind, as users can stay informed without having to manually check the status.

**6. Seller Dashboard for Inventory and Sales Management**

One of the standout features of *The Shop EZ* is the comprehensive seller dashboard, which empowers sellers to manage their products, view order history, and gain insights into sales performance. Through the dashboard, sellers can list new products, update product details, and manage inventory in real-time. They can also access detailed analytics on customer engagement and order metrics, helping them make data-driven decisions to optimize their business.

The dashboard provides a holistic view of a seller’s operations, from tracking low inventory items to identifying high-performing products. Sellers can quickly view new orders, monitor pending shipments, and track the overall performance of their store. This tool is crucial for sellers looking to scale their operations while maintaining efficiency.

**7. User Account and Profile Management**

Each user on *The Shop EZ* has an account where they can manage their personal information, address book, order history, and preferences. The account area offers users the flexibility to update their profiles, add multiple shipping addresses, and securely store payment methods for faster checkout in future purchases. User profiles enhance the personalized shopping experience, making it easy for returning users to complete transactions and manage past orders.

**8. Enhanced Security and Data Privacy**

*The Shop EZ* is built with data security and privacy in mind. User authentication is managed using JWT (JSON Web Tokens), which allows for secure, session-based authentication. Passwords and sensitive user data are encrypted, and the platform follows best practices in data security to prevent unauthorized access. Payment information is processed through secure channels, ensuring compliance with data protection standards. These measures are designed to instill user trust and safeguard personal information at all stages of the shopping experience.

**9. Responsive and Accessible Design**

The platform’s frontend, built with React, ensures that the interface is fully responsive and accessible across all devices, including desktops, tablets, and smartphones. This responsive design enhances usability by ensuring the same high-quality experience regardless of screen size. Accessibility features, such as keyboard navigation and screen reader compatibility, further ensure that the platform can be used by individuals with disabilities.

**10. Scalable and Modular Architecture**

The modular architecture of *The Shop EZ*, supported by the MERN stack, allows for future scalability. Each component of the system—frontend, backend, and database—can be independently developed, updated, and scaled. This modularity is especially beneficial as the platform grows, making it easy to add new features, integrate third-party APIs, and handle larger volumes of data and users without compromising performance.

**11. Customer Support and Feedback System**

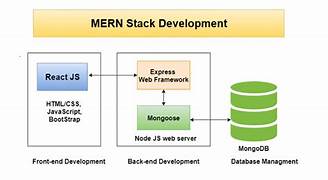
*The Shop EZ* includes a customer support system that allows users to contact support for order inquiries, product questions, and troubleshooting. Additionally, a feedback system enables users to leave reviews and ratings on purchased products, contributing to a community-driven catalog that assists future buyers in making informed decisions. This feedback loop also provides sellers with insights into customer satisfaction, helping them improve their offerings.

**3. Architecture**

The architecture of *The Shop EZ* follows a well-structured, modular design that leverages the MERN stack (MongoDB, Express.js, React, and Node.js) to create a scalable, efficient, and responsive e-commerce platform. Each component within this stack plays a unique role in delivering the functionalities required by the platform, enabling a full-stack, end-to-end solution. Here’s a detailed breakdown of the architectural components and how they interconnect to support *The Shop EZ*.

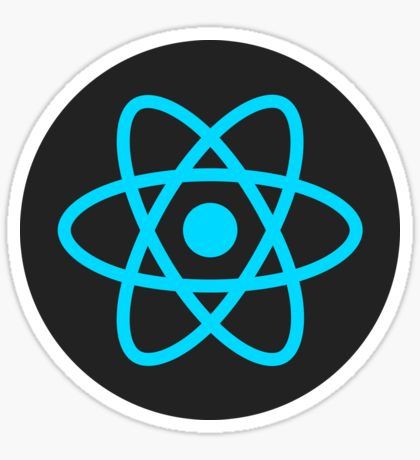
**Architecture Components**:

1. **Frontend Architecture (React)**
2. **Backend Architecture (Node.js and Express.js)**
3. **Database Architecture (MongoDB)**

****

**1. Frontend Architecture (React)**

The frontend of *The Shop EZ* is built using **React**, a popular JavaScript library for creating interactive, efficient, and responsive user interfaces. React’ s component-based architecture is especially well-suited for an e-commerce platform like *The Shop EZ*, as it enables reusable UI components and streamlined state management.



**Key Elements of the Frontend:**

* **Component-Based Structure**:
* **State Management (Context API / Redux)**:
* **Routing (React Router)**:
* **Responsive Design and Styling**:

**2. Backend Architecture (Node.js and Express.js)**

The backend of *The Shop EZ* is built on **Node.js** and **Express.js**, which provide a powerful and flexible environment for handling server-side logic, data processing, and business logic. This server-side architecture supports all core functionalities, including managing user data, processing orders, and handling payments.

**Key Elements of the Backend:**

* **Server and API Endpoints**:
* **Authentication and Authorization (JWT)**:
* **Business Logic and Validation**:
* **Order Processing and Notifications**:



**3. Database Architecture (MongoDB)**

The database architecture of *The Shop EZ* is centered around **MongoDB**, a NoSQL document database known for its flexibility, scalability, and ability to handle unstructured data. MongoDB is a great choice for e-commerce platforms, as it can easily adapt to changing data structures and provides fast access to large amounts of data.

**Key Elements of the Database:**

* **Collections**:  
  MongoDB stores data in collections, each representing a different entity within *The Shop EZ*. Key collections include:
  + **Users**: Stores user data such as names, email addresses, hashed passwords, and user roles (buyer or seller).
  + **Products**: Contains details about each product, including title, description, price, category, images, and stock availability.
  + **Carts**: Temporarily stores products added to a user’s cart, with each entry differentiated by user ID.
  + **Orders**: Stores completed orders, with references to user details, cart items, payment information, and order status.
  + **Admin Data**: Stores admin-specific data such as banner images, promotional categories, and site announcements.



1. **API Layer and Middleware**

The API layer in *The Shop EZ* serves as the bridge between the frontend and backend. It manages requests from the client, applies necessary business logic, interacts with the database, and sends responses back to the frontend.

**Key Elements of the API Layer:**

* **API Endpoints**:  
  The backend exposes RESTful API endpoints to facilitate communication between the client and server. Each endpoint corresponds to a specific resource, allowing CRUD operations on products, users, orders, and cart items. Examples of key endpoints include:
  + **/api/products**: To get product lists, retrieve single products, or manage products (for admins).
  + **/api/users**: To register new users, log in, and manage user profiles.

**4. Setup Instructions**

**Prerequisites**

Before starting, ensure that your system has the following software dependencies installed:

1. **Node.js and npm**: Node.js is required to run the JavaScript code for both the backend server and frontend React application. npm (Node Package Manager) is needed to install packages.
   * **Installation**: [Download Node.js and npm](https://nodejs.org/) and follow the instructions based on your operating system.
   * **Verify Installation**: Open a terminal or command prompt and type node -v and npm -v to confirm that both are installed.
2. **MongoDB**: MongoDB is used as the database for *The Shop EZ*. You can either use a local MongoDB instance or set up a MongoDB Atlas account for a hosted database.
   * **Installation** (local): [Download MongoDB](https://www.mongodb.com/try/download/community) and follow the setup instructions based on your operating system.
   * **Optional**: For a hosted database, create a MongoDB Atlas account at [MongoDB Atlas](https://www.mongodb.com/cloud/atlas) and follow the instructions to set up a cluster.

**Step-by-Step Setup Instructions**

**Step 1: Install Dependencies**

The project is divided into two main folders: client for the React frontend and server for the Node.js backend. Each folder has its own set of dependencies that need to be installed.

1. **Backend Dependencies**:
   * Navigate to the server folder:

**cd server**

* + Install the backend dependencies listed in package.json by running

**npm install**

**2.Frontend Dependencies**:

* + Open a new terminal window and navigate to the client folder:

**cd client**

* + Install the frontend dependencies by running:

**npm install**

**Step 2: Start the Application**

After configuring the environment variables and installing dependencies, you can start the backend and frontend servers.

1. **Starting the Backend Server**:
   * In the terminal, navigate to the server folder if you’re not already there and run:

**npm start**

* + This command will start the backend server on the port specified in the .env file (typically port 5000). You should see output in the terminal confirming that the server is running and connected to the MongoDB database.

1. **Starting the Frontend Server**:
   * In another terminal window, navigate to the client folder and run:

**npm start**

* + This will start the React development server, typically running on port 3000.

**Step 3: Verify Database Connection**

To ensure the database is connected correctly, navigate to the MongoDB collections and check if any initial data is available. Additionally, you can confirm database connectivity by checking for log messages in the backend terminal indicating a successful connection.

1. **Folder Structure**

* **Client (Frontend)**:
  + **src**
    - **components**: Reusable UI components like ProductCard, CartItem, and OrderSummary.
    - **pages**: Full-page views such as ProductList, ProductDetails, Cart, Checkout, and Dashboard.
    - **services**: Handles API calls to the backend for CRUD operations.
    - **styles**: Global stylesheets and CSS modules for specific components.
    - **utils**: Utility functions for data manipulation and formatting.
* **Server (Backend)**:
  + **/routes**: Contains endpoint routes for users, products, cart, orders, and authentication.
  + **/controllers**: Business logic functions for handling requests, such as order processing or cart updates.
  + **/models**: Mongoose schemas representing database collections (e.g., User, Product, Order).
  + **/middlewares**: Middleware for authentication, error handling, and validation.
  + **/config**: Configuration files for database connection and environment variables.

**6. Running the Application**

**Commands to Start Servers**:

**Frontend**: **cd client**

**npm start**

**Backend: cd server**

**npm start**

**7. API Documentation**

* **User Routes**:
  + POST /api/users/signup - Registers a new user.
  + POST /api/users/login - Logs in a user, returning a JWT for session management.
* **Product Routes**:
  + GET /api/products - Retrieves all products with pagination.
  + GET /api/products/:id - Retrieves detailed information for a specific product.
* **Cart Routes**:
  + POST /api/cart - Adds a product to the user’s cart.
  + GET /api/cart/:userId - Retrieves the user’s current cart contents.
  + DELETE /api/cart/:productId - Removes a product from the cart.
* **Order Routes**:
  + POST /api/orders - Places an order, transferring items from cart to orders.
  + GET /api/orders/:userId - Retrieves all orders for a specific user.

Each endpoint provides detailed documentation, including method types, required parameters, and example responses.

8. **Authentication**

Authentication in *The Shop EZ* is implemented using JSON Web Tokens (JWT) for secure user identity verification and session management. When a user registers or logs in, the server generates a unique JWT containing the user's ID and other necessary data, which is signed with a secret key stored on the server. This token is then sent to the client (usually as a cookie or in local storage), and it accompanies each subsequent request to verify the user's identity.

For each request that requires authentication, the backend checks for the presence of a valid token. If the token is valid and not expired, the server grants access to protected routes and resources, such as viewing the cart, placing an order, or accessing the seller dashboard. If the token is invalid or absent, the server denies access.

The authentication system ensures both security and user convenience by eliminating the need for repeated logins. This JWT-based approach also simplifies handling stateless sessions and allows for secure authorization, enabling role-based access control for users, admins, and sellers in the platform.

**9. User Interface**

**Screenshots/GIFs**:

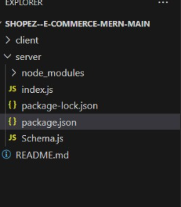
* + Product catalog view showing products, filters, and search bar.
  + Product details view with "Shop Now" option.
  + Cart interface with item details, quantity updates, and checkout option.
  + Order confirmation and tracking page.
  + Seller dashboard with order and inventory management.

**10. Testing**

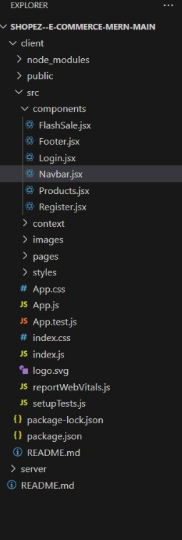
* **Strategy and Tools**:
  + **Unit Testing**: Testing individual components and functions using Jest.
  + **Integration Testing**: Testing interactions between backend services and database using tools like Mocha and Chai.
  + **End-to-End Testing**: Verifying the entire user flow from browsing to checkout with Cypress.

**11. Screenshots :**

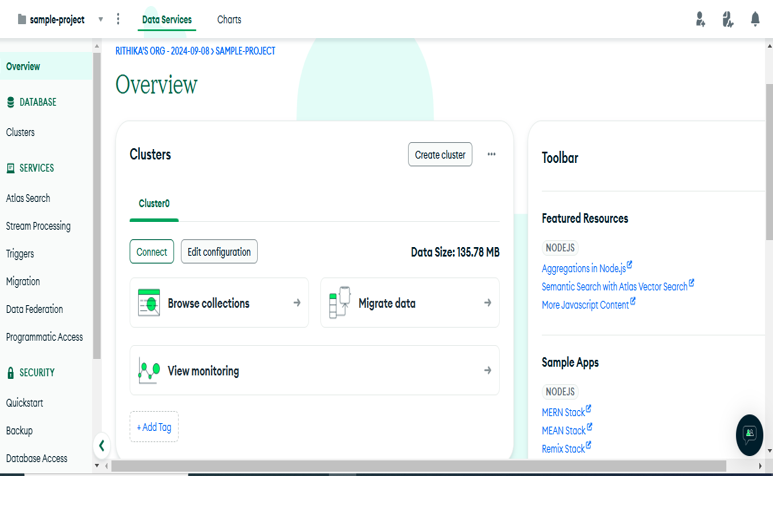
* **Back-end functionality**



* **Front-end functionality**

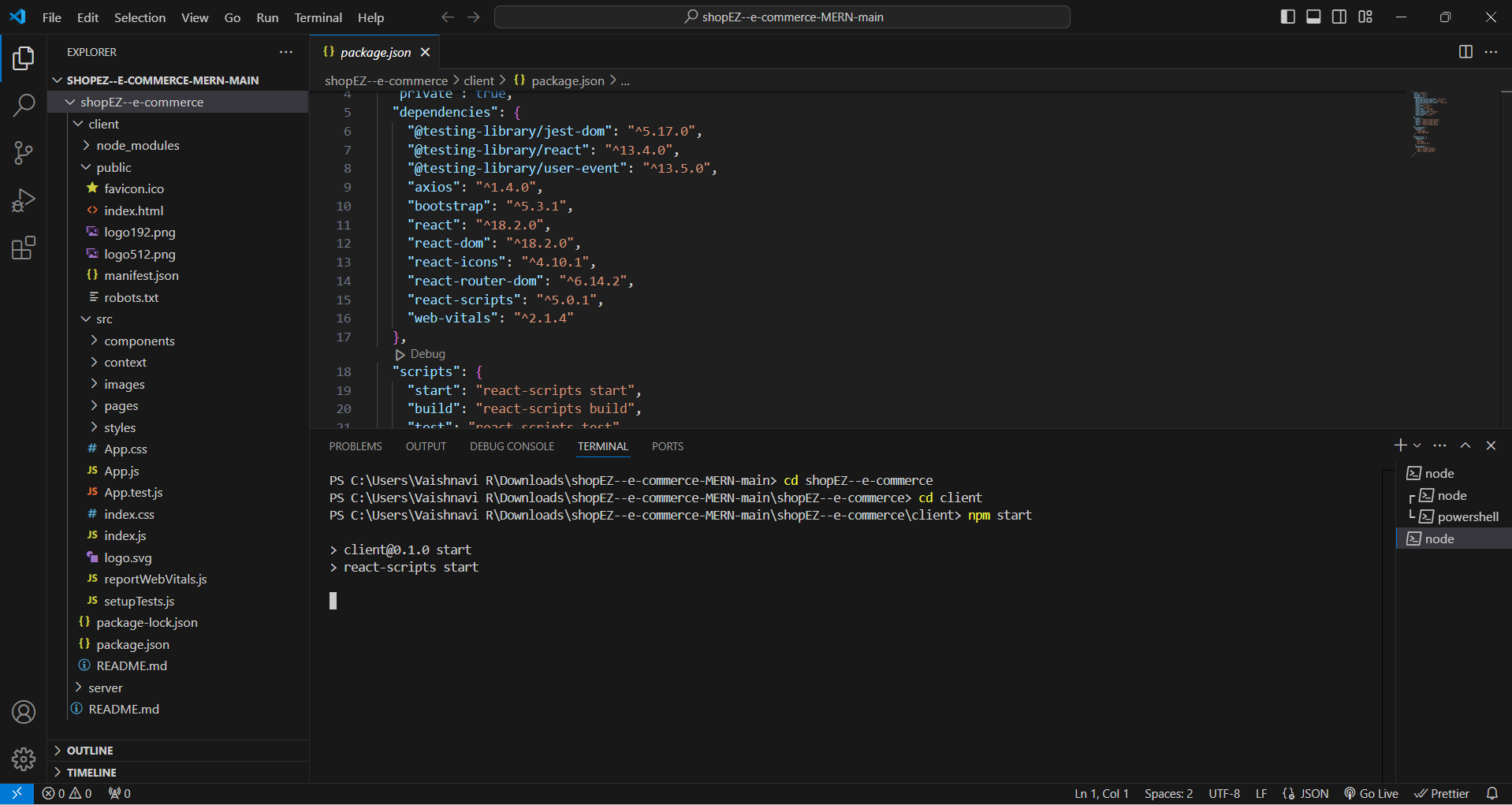


Database:

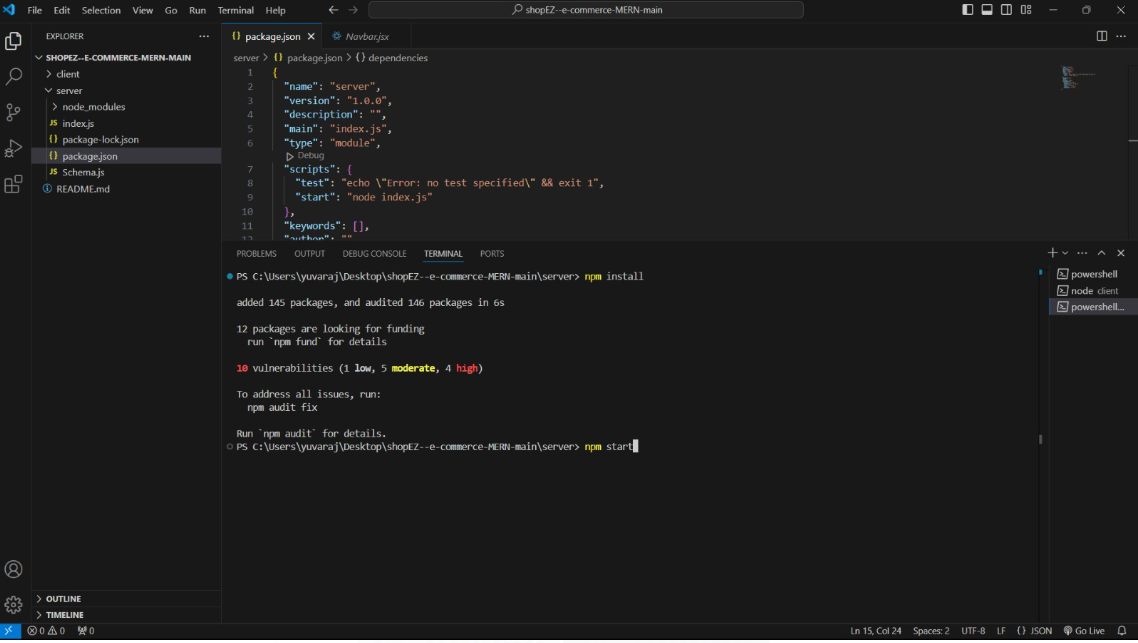


**Project execution**

Front-end:

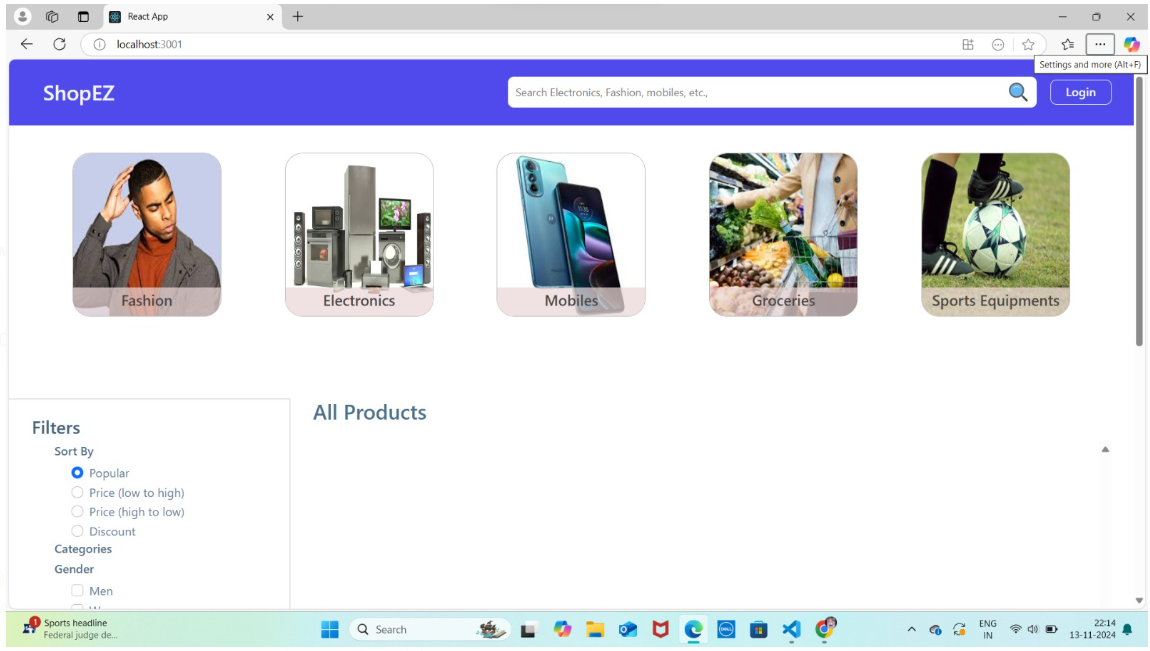


Back-end:

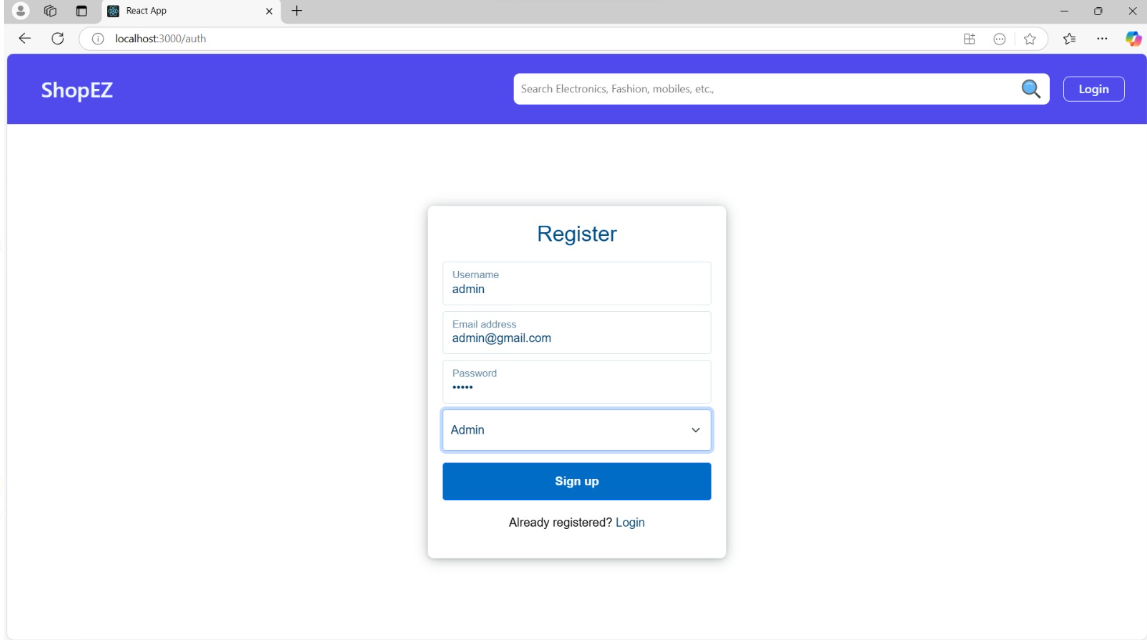


**OUTPUT :**

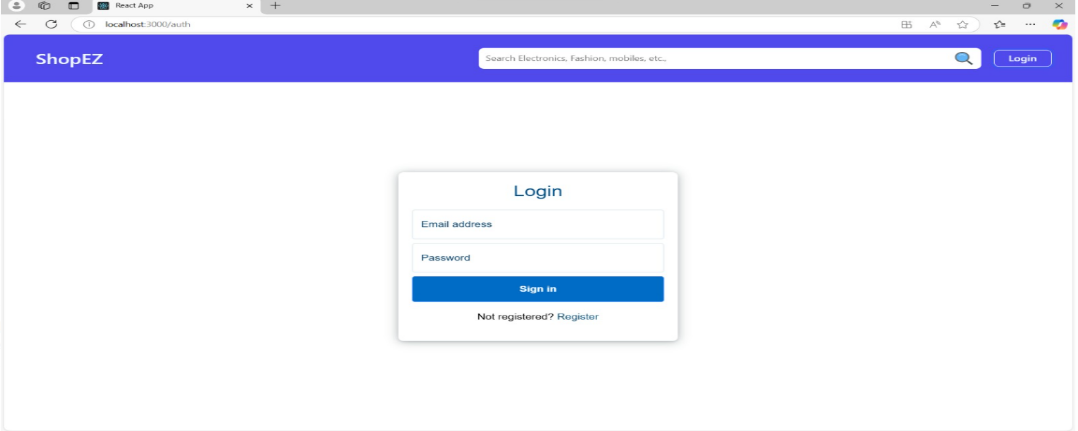
**The home page**

****

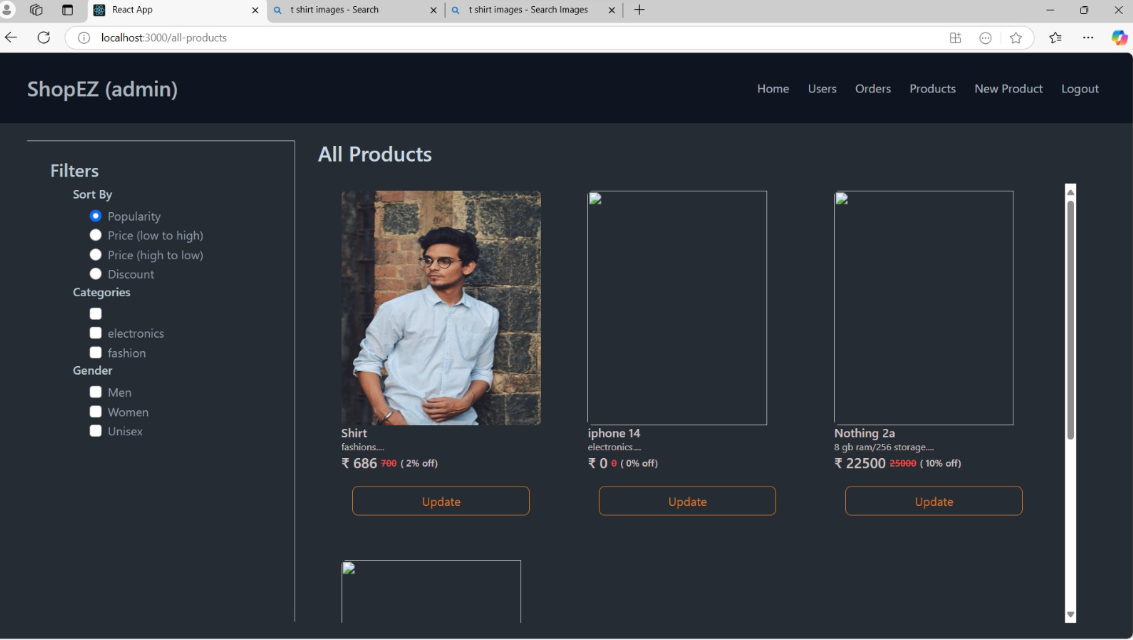
**Sign-up page**

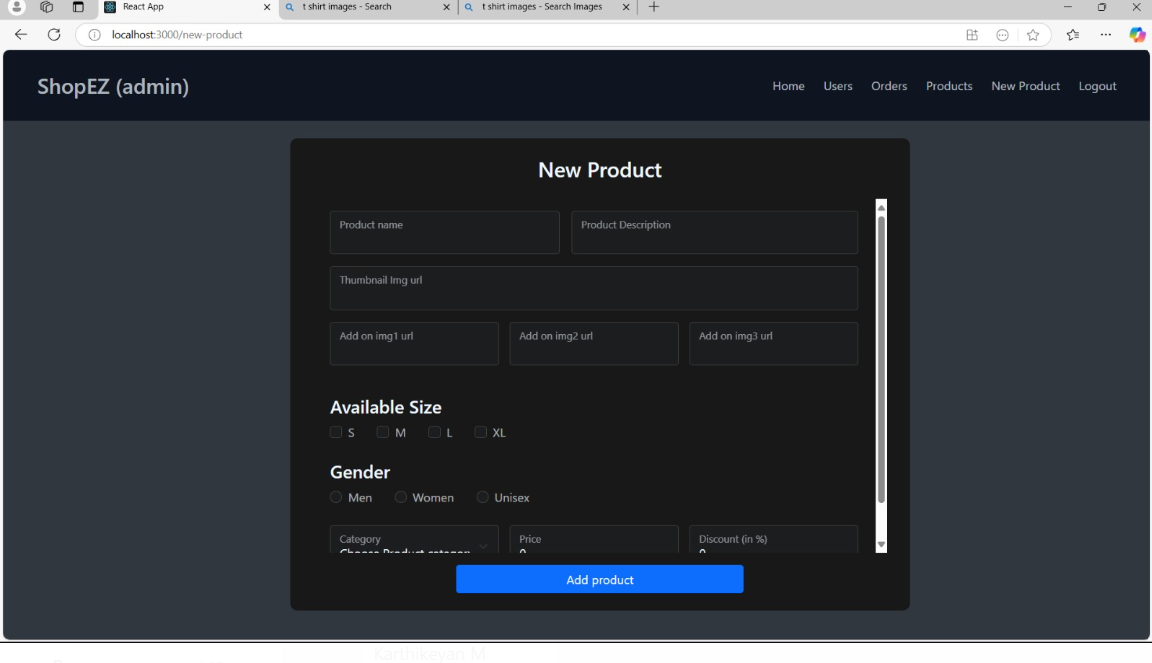
****

**Login page**

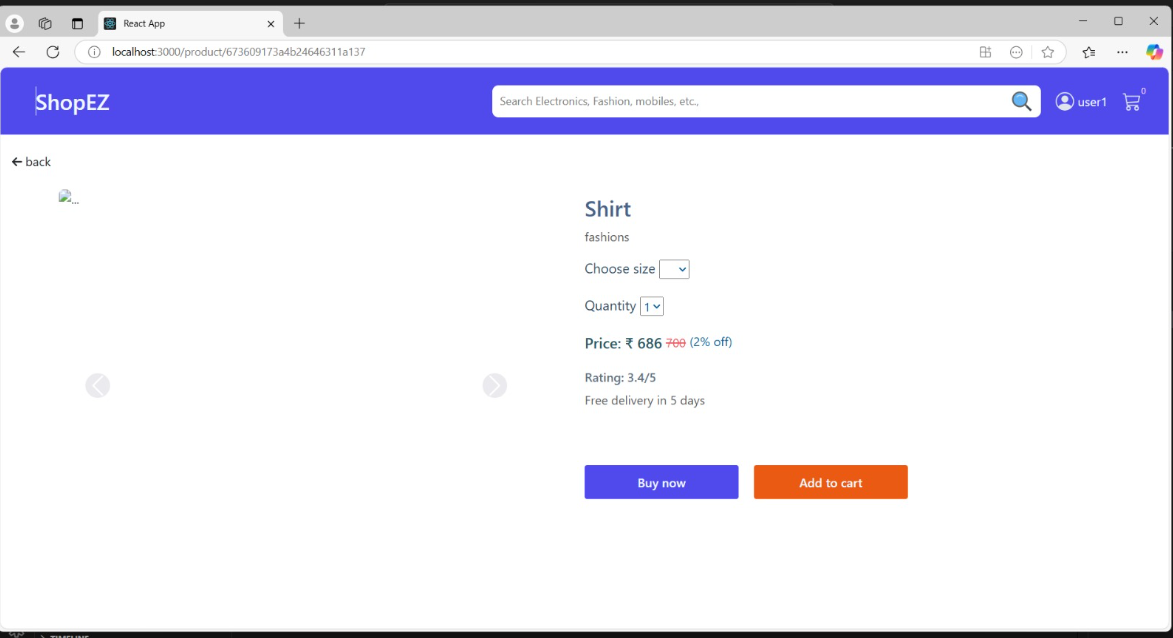
****

**All Products**

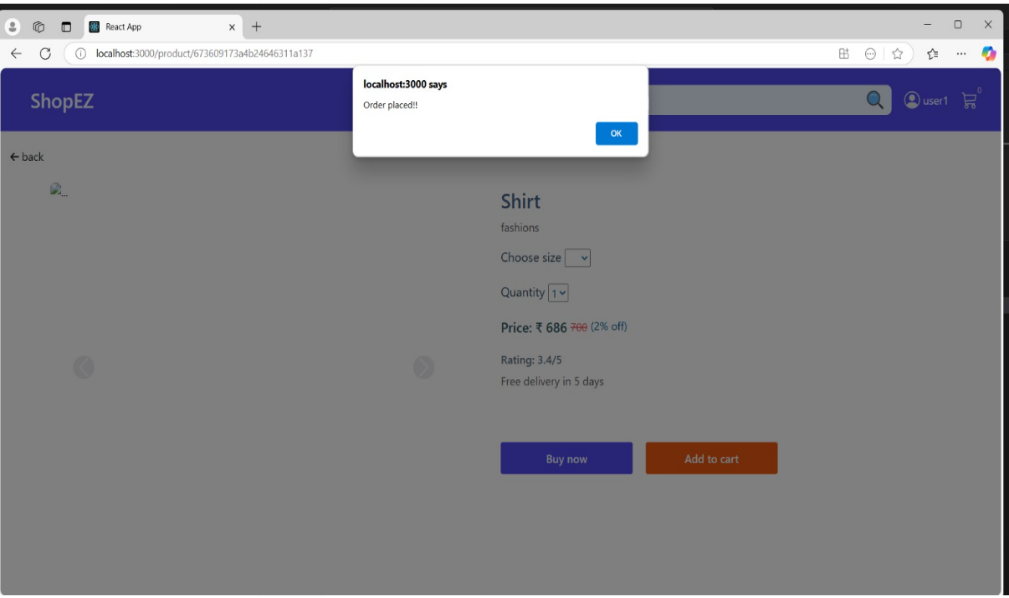
****

**NEW PRODUCTS**

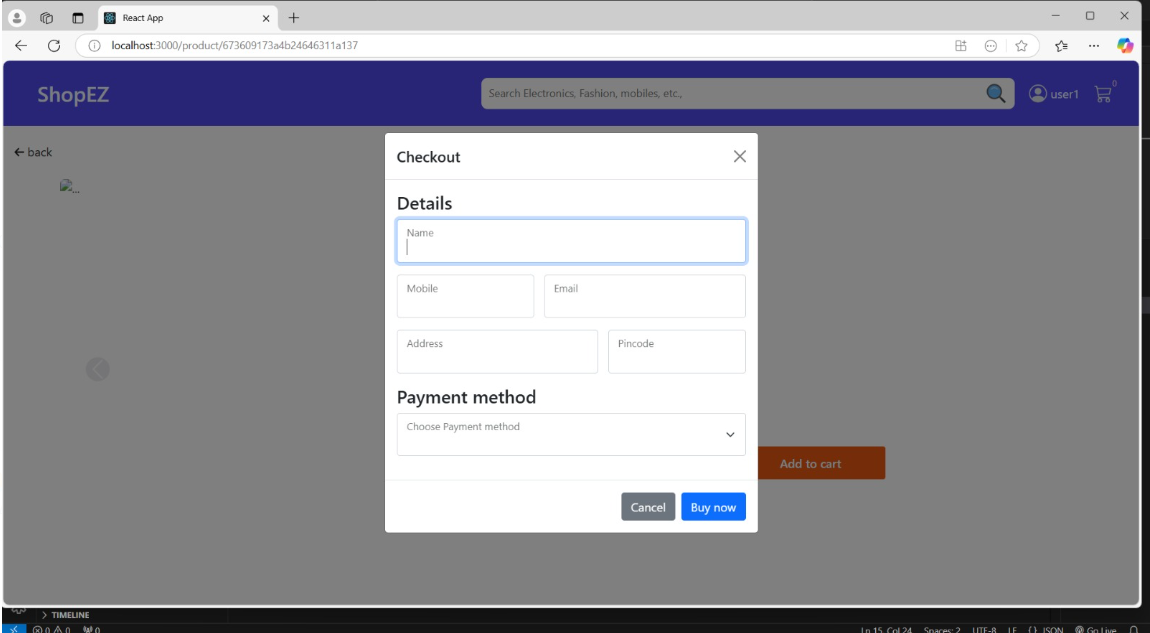
**Add to cart**

****

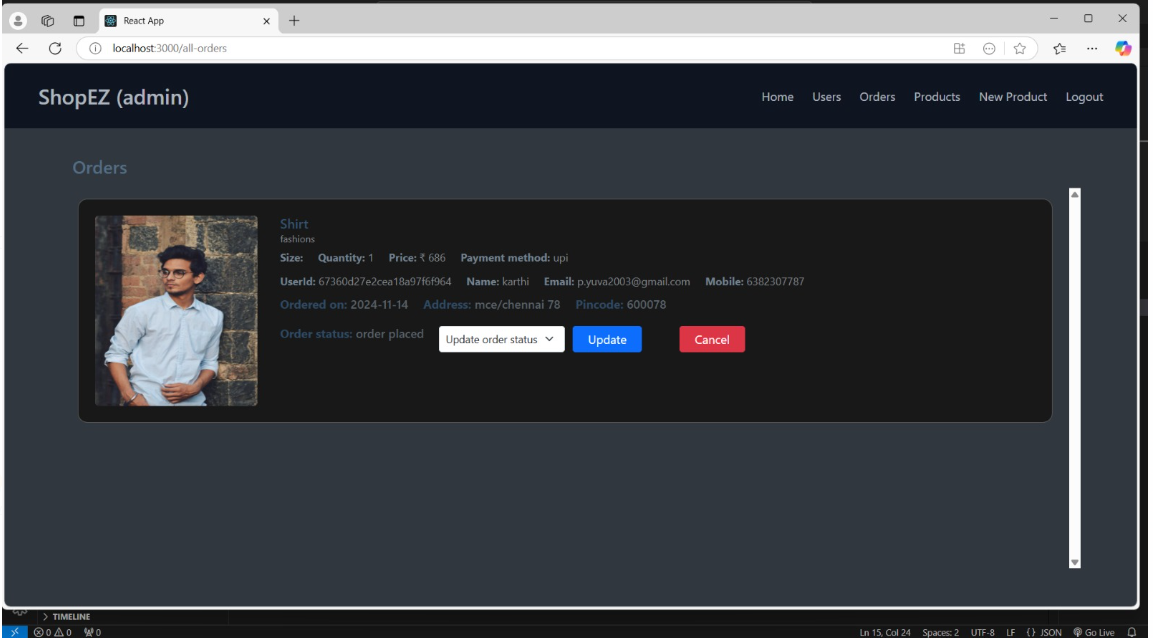
**Order placed**

****

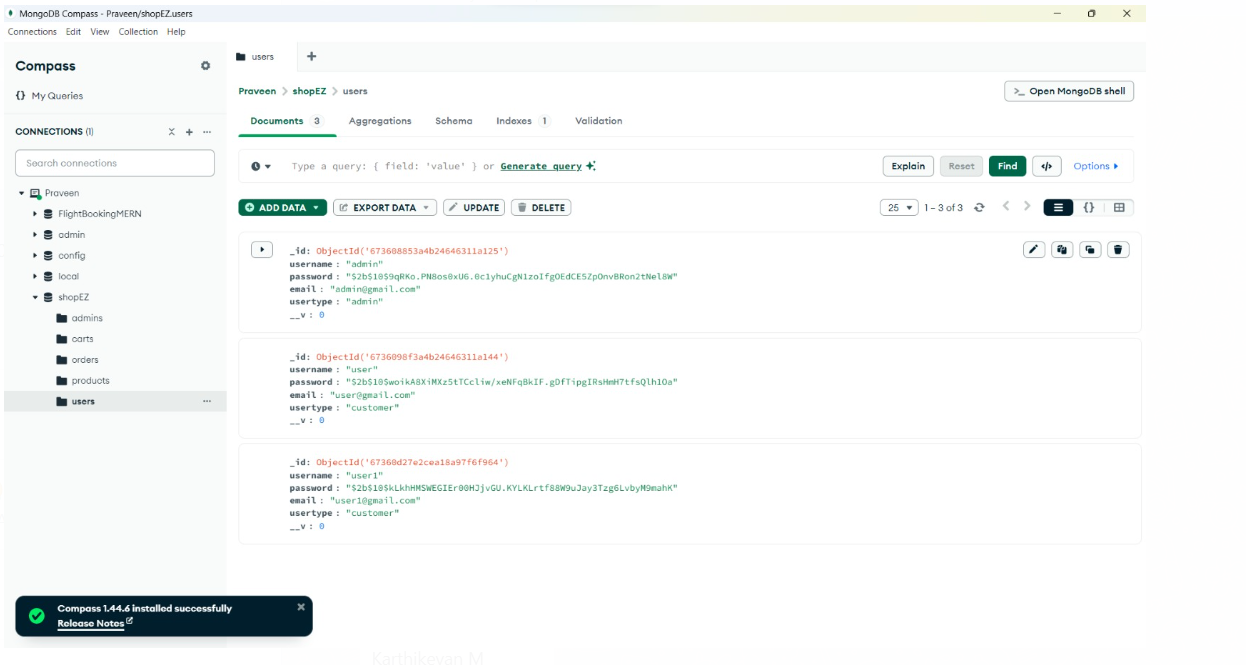
**Payment**

****

**Order details**

****

**Database**

****

**12. Known Issues**

* **Bugs/Issues**:
  + Some delays in updating the cart upon item removal.
  + Potential UI glitches on certain browsers.
  + Limited filtering options for product categories.

**13. Future Enhancements**

* **Planned Improvements**:
  1. **Advanced Filtering**: Adding multi-criteria filtering options.
  2. **Recommendation System**: Implementing product recommendations based on user behavior.
  3. **Enhanced Analytics for Sellers**: Offering more detailed insights into product performance and customer trends.
  4. **Expanded Payment Options**: Adding multiple payment gateways for user convenience.

This document serves as an extensive guide for developing, using, and maintaining *The Shop EZ* e-commerce platform.

**14. Conclusion**

*The Shop EZ* offers a modern, user-friendly e-commerce platform that simplifies the shopping experience for users while providing powerful tools for sellers to manage their inventories and sales. By leveraging the MERN stack (MongoDB, Express.js, React, and Node.js), the platform ensures robust performance, scalability, and security. The system is designed to meet the needs of both shoppers and sellers, offering a seamless user interface, secure checkout process, and a comprehensive backend that manages products, orders, and user interactions efficiently.

The architecture supports future scalability with the potential for adding new features such as personalized recommendations, advanced filtering options, and multi-payment gateway support. Additionally, the seller dashboard enhances seller management capabilities, making it easier to track orders and adjust inventory. With a focus on security and user experience, *The Shop EZ* is positioned to be a reliable and dynamic e-commerce solution, making online shopping accessible, enjoyable, and efficient.

Through continuous testing, feedback, and future enhancements, *The Shop EZ* will remain adaptable to evolving market trends and user needs, providing long-term value for both customers and sellers.

Top of Form

Bottom of Form