**Project Title**

**ShopEZ: Effortless Online Shopping**

**Project Overview**

ShopEZ is a streamlined online shopping platform designed to offer users an intuitive and personalized shopping experience. Its user-friendly interface simplifies product discovery, personalized recommendations, and secure transactions. Sellers benefit from advanced order management tools and insightful analytics, empowering them to grow their businesses efficiently.

**Key Features**

**Seamless Checkout Process**

A fast, secure, and straightforward payment experience for users.

**Effortless Product Discovery**

Robust search and filter options to quickly find the perfect product.

**Personalized Shopping Experience**

AI-driven product recommendations tailored to user preferences.

**Efficient Order Management for Sellers**

Streamlined seller dashboard with order processing, inventory tracking, and sales insights.

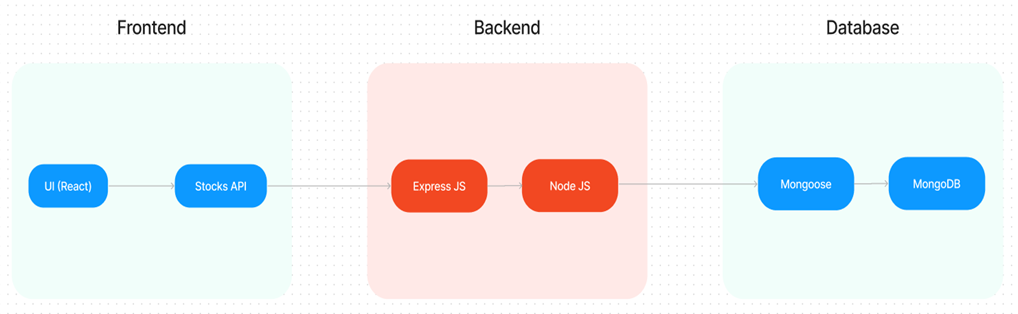
**Insightful Analytics for Business Growth**

Actionable data visualizations for sellers to optimize their strategies.

**Scenario**: Sarah's Birthday Gift

Sarah, a busy professional, is scrambling to find the perfect birthday gift for her best friend, Emily. She knows Emily loves fashion accessories, but with her hectic schedule, she hasn't had time to browse through multiple websites to find the ideal present. Feeling overwhelmed, Sarah turns to ShopEZ to simplify her search.

**Technical Architecture**

****

In this architecture diagram:

• The frontend is represented by the "Frontend" section, including user interface components such as User Authentication, Cart, Products, Profile, Admin  dashboard, etc.,

• The backend is represented by the "Backend" section, consisting of API endpoints for Users, Orders, Products, etc.,It also includes Admin Authentication and an Admin Dashboard.

•  The Database section represents the database that stores collections for Users, cart, Orders and Product.

**Client-Server Model**

**Client**

* **Frontend Framework**: React.js for building a dynamic and responsive interface.
* **Key Client Components**:
  + **Product List**: Displays available items.
  + **Cart**: Summarizes selected items for checkout.
  + **Checkout Page**: Captures user details for payment and delivery.

**Server**

* **Backend Framework**: Node.js with Express.js to handle API requests.
* **Database**: MongoDB for managing user accounts, products, and orders.
* **APIs**:
  + **Authentication**: Secure user and seller login/logout.
  + **Order Management**: Save, update, and retrieve orders.
  + **Payment Processing**: Integrate with payment gateways (e.g., Stripe, Razorpay).

**Data Flow**

1. Client sends a request to the server (e.g., product selection, payment initiation).
2. Server validates the request and updates the database.
3. Server responds with success or error messages, triggering frontend updates.

**Prerequisites**

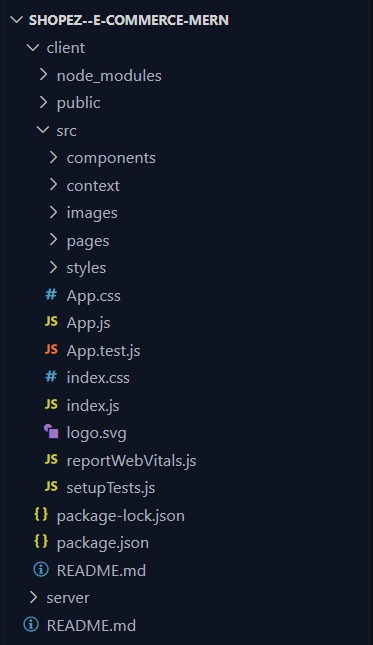
**For Buyers**

1. User Account:
   * Register or log in to access shopping cart and order history.
2. Valid Payment Method:
   * Credit/Debit card, PayPal, or UPI enabled for checkout.
3. Address Information:
   * Add or select saved delivery addresses.

**For Sellers**

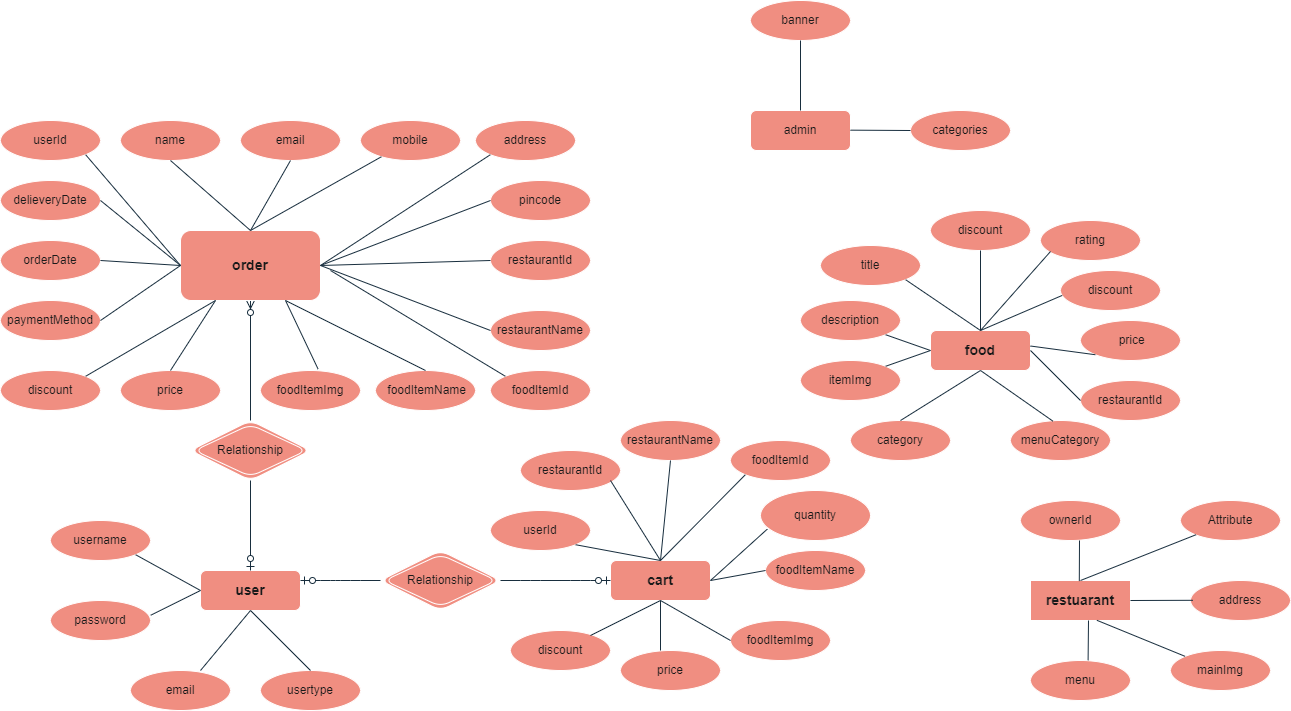
1. Seller Account:
   * Register and list products on the platform.
2. Product Inventory:
   * Maintain updated stock information to avoid order cancellation.
3. Dashboard Access:
   * Utilize tools for order management and analytics.

**Application Flow**

****

**ER DIAGRAM**

The ER diagram illustrates the relationships between key entities in the ShopEZ database, including users, orders, and items.



**PRE-REQUISITES**

To develop and run the SB Foods application, the following tools and technologies are required:

* **Node.js & npm**: For backend development.
* **MongoDB**: For database management.
* **React.js**: For frontend development.
* **Express.js**: For server-side logic.
* **Axios**: For making HTTP requests.
* **Material UI & Bootstrap**: For styling the frontend.

**Installation Guide**:

1. Install Node.js and npm from [Node.js Downloads](https://nodejs.org/).
2. Set up MongoDB from [MongoDB Downloads](https://www.mongodb.com/try/download/community).
3. Initialize the React project:

bash

Copy code

npx create-react-app sb-foods

1. Install backend dependencies:

bash

Copy code

npm install express mongoose bcryptjs jsonwebtoken

1. Start the development server:

bash

Copy code

npm start

**PROJECT STRUCTURE**

**Frontend**

* src/
  + components/
  + pages/
  + services/
  + App.js

**Backend**

* models/
* routes/
* controllers/
* middleware/

**MILESTONES**

**Milestone 1: Project Setup**

* Initial setup of the frontend and backend directories.
* Installation of required dependencies.

**Milestone 2: Backend Development**

* API development for user authentication, menu management, and order handling.
* Implementation of CRUD operations for users, orders, and menu items.

**Milestone 3: Database Development**

* Define schemas for User, Order, and MenuItem.
* Implement secure payment integration.

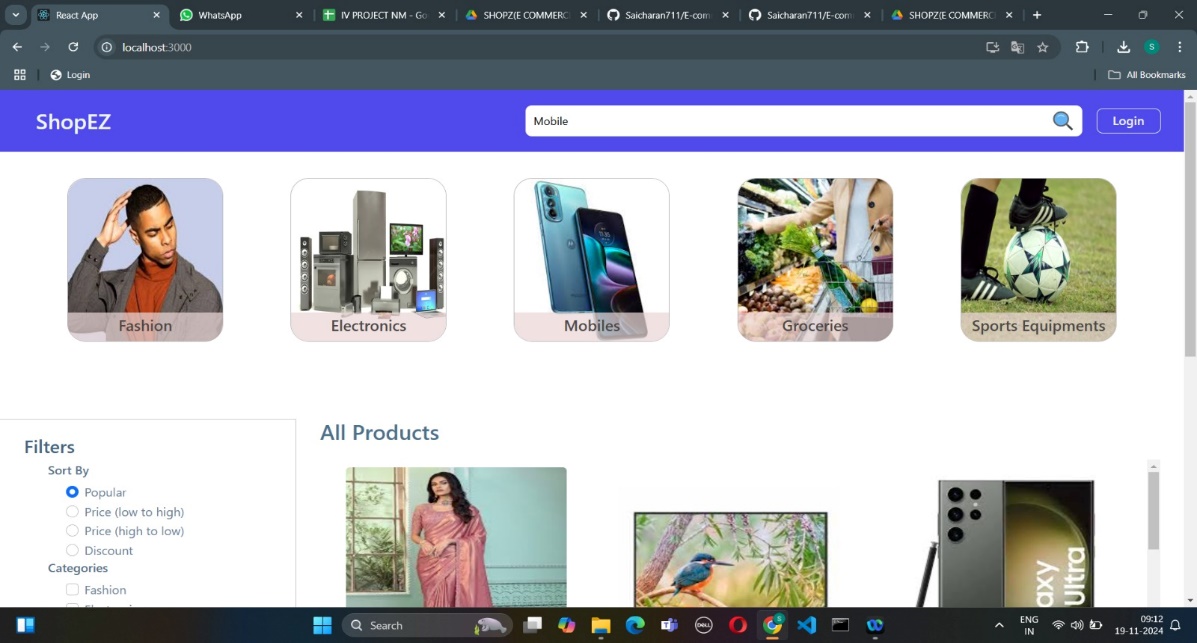
**Milestone 4: Frontend Development**

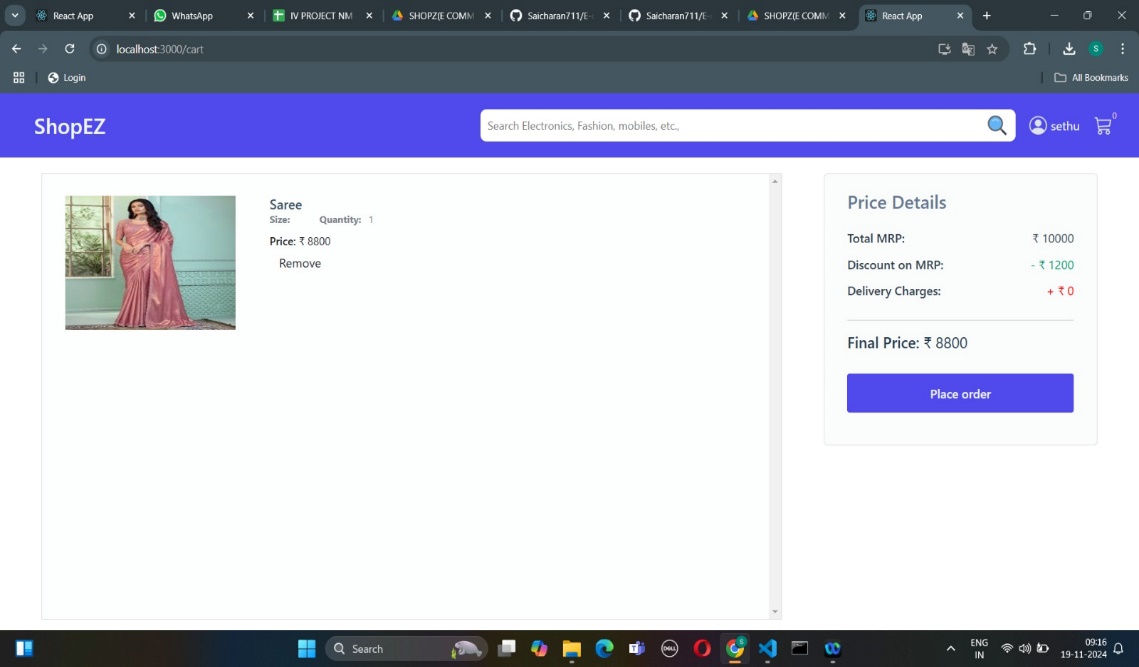
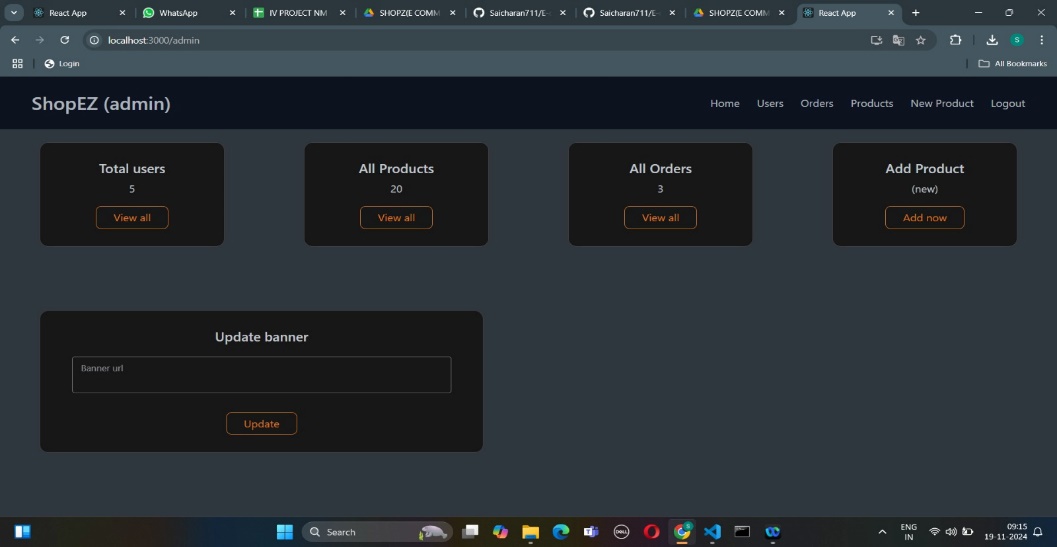
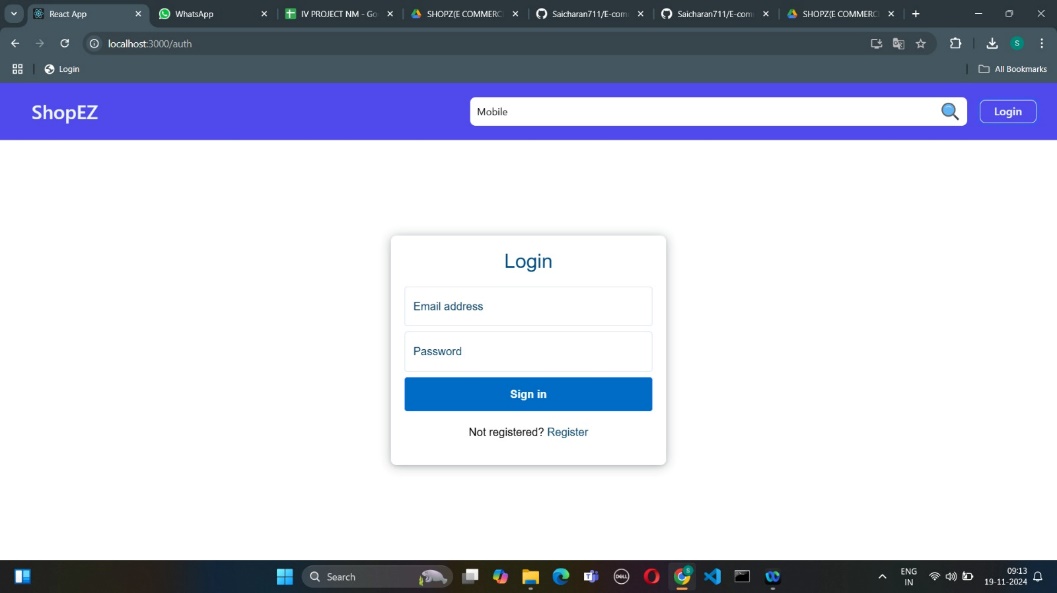
* Build responsive UI components for customer interaction.
* Integration of APIs with frontend using Axios.

**Milestone 5: Testing & Deployment**

* Conduct end-to-end testing to ensure functionality.
* Deploy the application on platforms like **Heroku** or **Vercel**.

**OUTPUT**

****

****