**SB Foods - Food Ordering Application**

**INTRODUCTION**

The **SB Foods Food Ordering Application** is a web-based solution designed to streamline online food ordering and delivery. Built using the **MERN stack** (MongoDB, Express.js, React.js, and Node.js), the application allows users to browse the menu, place orders, and track their delivery status efficiently. The platform provides a user-friendly interface for customers, restaurant staff, and administrators to manage the entire food ordering process, enhancing the overall customer experience.

**Key Features**

1. **User Registration & Login**: Users can register, log in, and manage their profiles securely.
2. **Menu Browsing**: A dynamic menu that allows users to explore available dishes, apply filters, and view details.
3. **Order Placement & Payment**: Secure order placement with integrated payment options.
4. **Order Tracking**: Real-time order status updates and notifications for customers.
5. **Admin Dashboard**: Admin functionalities for managing menu items, orders, and user accounts.

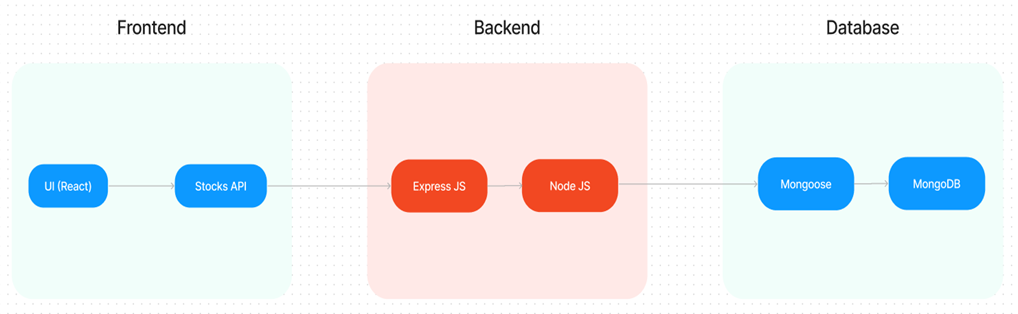
**DESCRIPTION**

The SB Foods application aims to provide a seamless online food ordering experience. Users can conveniently place orders from their favorite restaurants, track deliveries, and make payments securely. The app's robust backend ensures efficient data handling, while the intuitive frontend delivers a smooth user experience.

**Scenario**

**Example**: Sarah, a busy professional, wants to order lunch from SB Foods. She logs in, browses the menu, and places an order. She can track the delivery status in real-time and receives notifications when the food is on its way.

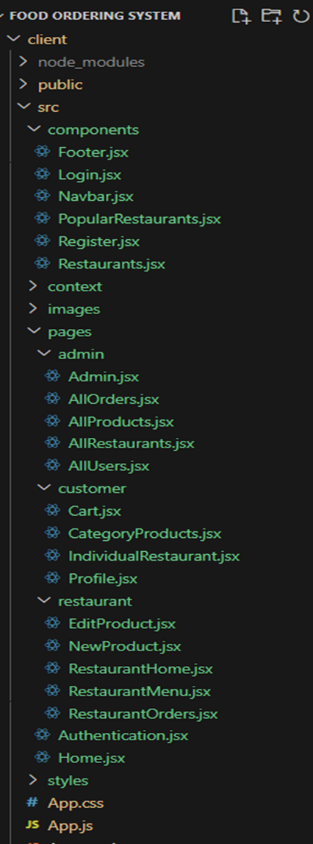
**TECHNICAL ARCHITECTURE**

****

The **SB Foods** application employs a **client-server model** using the MERN stack:

1. **Frontend**:
   * Built with **React.js** for a responsive and interactive user interface.
   * Uses **Material UI** and **Bootstrap** for styling and enhanced user experience.
   * Utilizes **Axios** for handling API requests to the backend.
2. **Backend**:
   * Developed using **Node.js** and **Express.js** to handle server-side logic.
   * Implements **RESTful APIs** for efficient communication between frontend and backend.
   * Uses **Mongoose** for database management and schema validation.
3. **Database**:
   * **MongoDB** stores user data, orders, menu items, and payment information.
   * Ensures secure data management and scalable performance.
4. **Security**:
   * Implements **JWT authentication** for secure user login and session management.
   * Ensures data protection with encryption and secure payment processing.

**APPLICATION FLOW**

****

**Customer Flow**

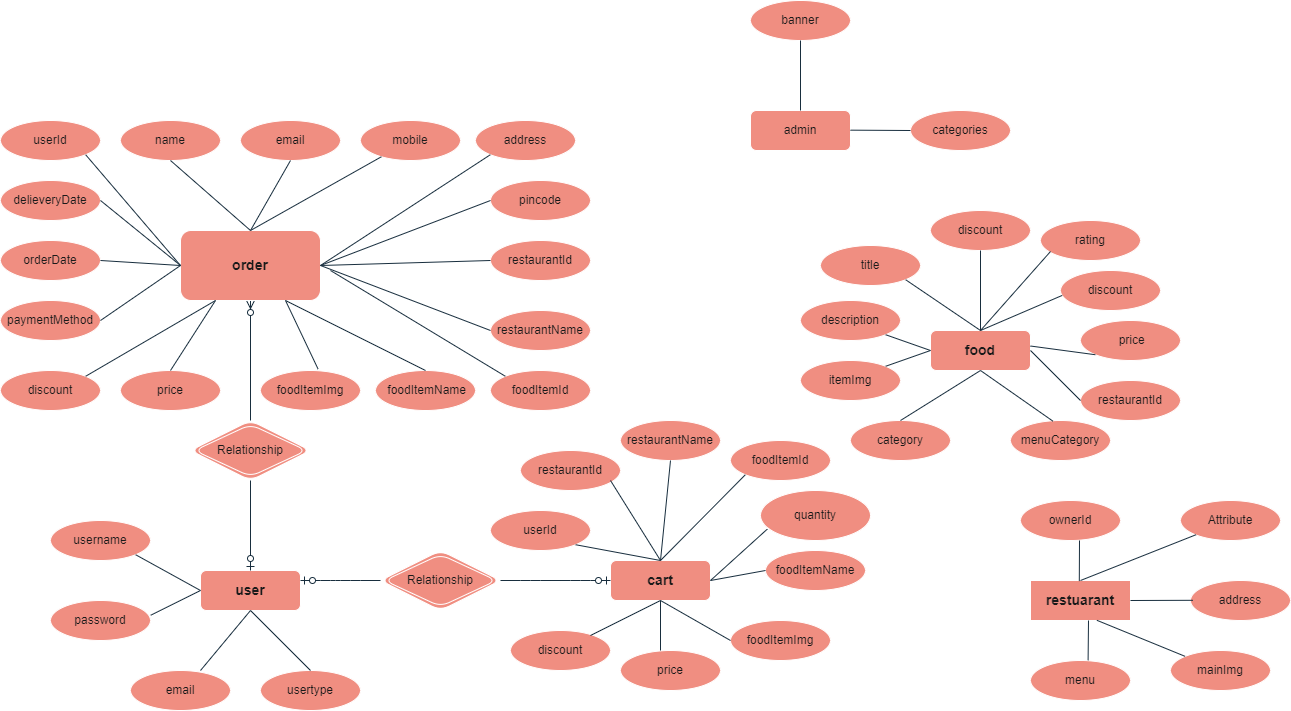
1. **User Registration and Login**:
   * Users can sign up with their details or log in using their email and password.
   * Passwords are securely hashed before storage.
2. **Menu Browsing & Order Placement**:
   * After logging in, users can browse the available dishes, add items to the cart, and place an order.
   * Users can view order summaries, apply discount codes, and proceed with payment.
3. **Order Tracking**:
   * Users receive notifications upon successful order placement.
   * Real-time updates on the order status are available on the dashboard.

**Admin Flow**

1. **Order Management**:
   * The admin dashboard allows monitoring of incoming orders and updating their statuses.
   * Administrators can manage menu items, add new dishes, and handle customer queries.
2. **User Management**:
   * Admins have control over user accounts, allowing for blocking or enabling accounts as needed.

**ER DIAGRAM**

The ER diagram illustrates the relationships between key entities in the SB Foods database, including users, orders, and menu 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**

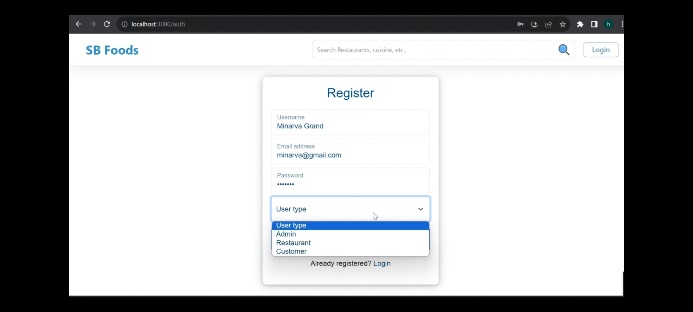
* Define schemas for User, Order, and MenuItem.
* Implement secure payment integration.

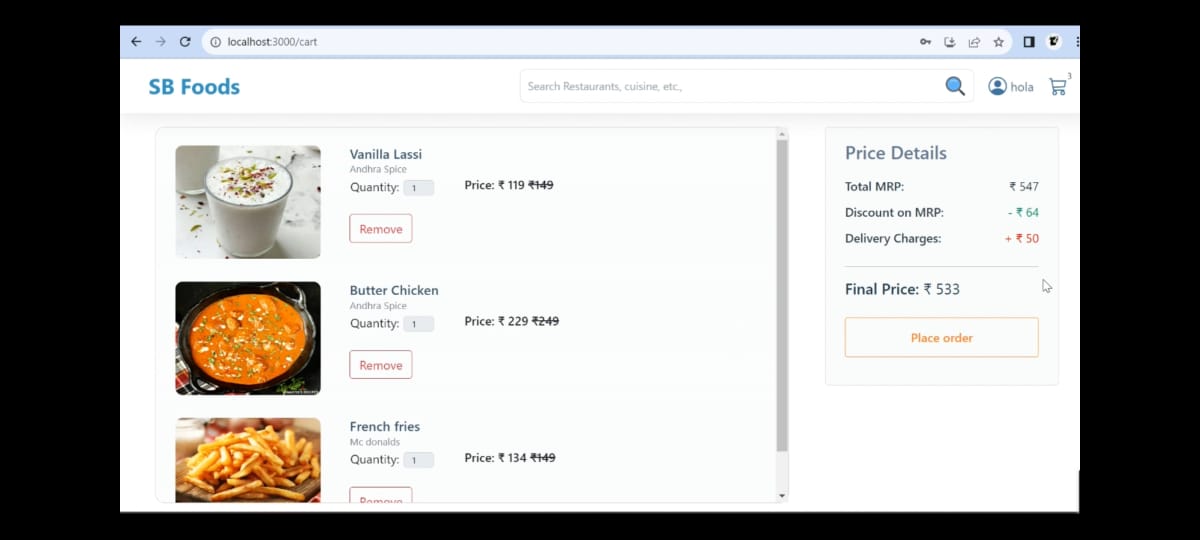
**Milestone 4: Frontend Development**

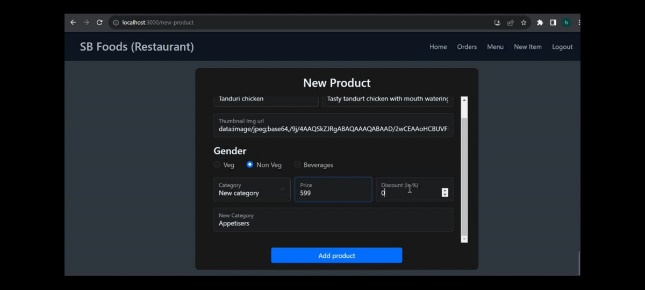
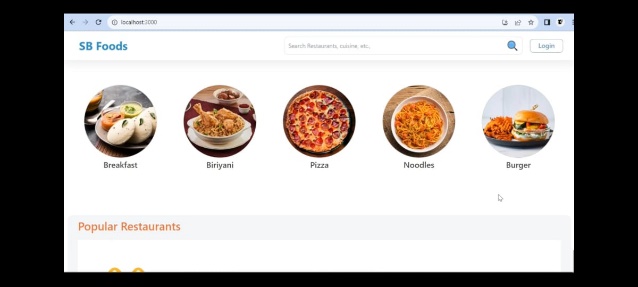
* Build responsive UI components for customer interaction.
* Integration of APIs with frontend using Axios.

**Milestone 5: Testing & Deployment**

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

**OUTPUT**





**FUTURE ENHANCEMENTS**

1. Integration of a **mobile app** version.
2. Advanced **analytics dashboard** for restaurant performance insights.
3. AI-based **personalized recommendations** for users.