Restaurant Web Application Documentation

This documentation provides an overview of the features and functionality of a restaurant website built using the MERN stack (MongoDB, Express, React, and Node.js). The website allows users to view food items, place orders, create an account, and sign in. It also includes an admin panel for managing food items and reviewing and accepting or rejecting orders.

Tech Stack

Frontend: React, Redux
Backend: Node.js, Express
Database: MongoDB
Authentication: JWT (JSON Web Tokens)
State Management: Redux
UI Library: Material-UI

Features

1. User Authentication
2. Users can create an account with their email address and password.
3. Users can log in with their registered email and password.
4. User passwords are securely hashed and stored in the database using bcrypt.
5. Authentication is implemented using JWT (JSON Web Tokens) for secure and stateless authentication.
6. Food Items
7. Users can view a list of food items available in the restaurant.
8. Food items are categorized and displayed with relevant details such as name, description, price, and image.
9. Users can view food item details and add items to their cart.
10. Cart functionality allows users to view and modify their selected food items before placing an order.
11. Orders
12. Users can place orders for the selected food items in their cart.
13. Users can view their order history and order status.
14. Admin users can view all orders and change their status (accepted, rejected).
15. Admin Panel
16. Admin users can access a secured admin panel for managing food items and orders.
17. Admins can add, edit, and delete food items.
18. Admins can view all orders, review order details, and update their status.

User Interface (UI)

The user interface (UI) of the restaurant website is designed to be user-friendly and visually appealing, providing a seamless experience for users. The website follows a responsive design approach, making it accessible on various devices, including desktops, tablets, and mobile phones. The UI is implemented using Material-UI, a popular UI library for React applications, which provides a rich set of pre-designed UI components that are customizable and reusable.

API Endpoints

The backend API for the restaurant website is built using Node.js and Express, providing RESTful endpoints for handling various operations. The API is secured using JWT authentication to ensure secure communication between the client and server. Below are some of the API endpoints:

- **/api/auth**: Handles user authentication, including user registration and login.
- **/api/food**: Handles CRUD (Create, Read, Update, Delete) operations for food items, accessible only by admin users.
- **/api/orders**: Handles CRUD operations for orders, including placing orders, retrieving order history, and updating order status.
- **/api/users**: Handles user-related operations, such as retrieving user details and updating user information.

Database Schema

The restaurant website uses MongoDB, a popular NoSQL database, for storing food items, orders, and user information. The database schema includes the following collections:

- foods: Stores food item details, such as name, description, price, and image.
- orders: Stores order details, including order items, total amount, and status.
- users: Stores user information, including email, password, and role (admin or regular user).

Security

The restaurant website follows industry-standard security practices to ensure data privacy and protection. Some of the security measures implemented in the website include:

- Passwords are securely hashed using bcrypt.
- JWT (JSON Web Tokens) are used for authentication, ensuring secure and stateless authentication.
- Admin functionalities are protected by role-based access control (RBAC), ensuring that only authenticated