Restaurant P.O.S - Documentation

Project Overview

This React-based web application provides a comprehensive restaurant management platform. It includes features for order management, analytics, inventory control, kitchen display, and customer interaction. The project uses modular components, context APIs, Firebase for authentication, and provides dashboards for various stakeholders (admin, chef, waiter).

Project Structure

The source code is organized into the following directories and files:

- assets/: Contains images and other static assets.

- components/: Contains all reusable UI components grouped by functionality.

- constants/: Contains global constants and Firebase config.

- contexts/: React context logic (authentication).

- pages/: Page-level components for routing and major views.

- utils/: Utility functions, e.g., for analytics data.

- App.jsx: Main application component with routing.

- main.jsx: Entry point of the application.

- index.css: Global styles.

Module Details

Each module in the application handles specific functionality as outlined below:

Pages

These components represent distinct pages in the app, such as AdminDashboard, Analytics, ChefOrders, etc.

Analytics Components

Provides metrics and insights into different aspects of the restaurant like customers, employees, finances, inventory, and sales.

Home Components

Used on the user-facing home screen. Displays popular items, recent orders, etc.

Table Order Components

Used for order handling per table including invoices, item quantity control, and order details.

Shared Components

Components like Header and BottomNav used across the app for consistent layout.

Utils and Config

Includes Firebase configuration and helper functions to mock or process data.

Authentication Context

The `AuthContext.jsx` file sets up and provides authentication state throughout the app using Firebase Auth.

How to Launch the Code

Follow these steps to set up and run the project locally:

1. Install Node.js (preferably v16 or higher) and npm.
2. Open a terminal and navigate to the project directory.
3. Run `npm install` to install dependencies.
4. Ensure Firebase is correctly configured in `src/constants/firebase.js`.
5. Run `npm run dev` to start the development server.
6. The app should be accessible at `http://localhost:5173`.

Future Improvements

- Process Payments (Card, QR, Split)
- Mobile App Development
- Advanced AI Forecasting
- Self-service kiosks or QR-based table ordering.
- Real-time notifications and offline mode support.

Features

- Real-time order management and table assignment
- Inventory tracking with analytics integration
- Role-based access control (Admin, Waiter, Chef, Customer)
- Financial, sales, and employee analytics dashboards
- Customer order history and popular items suggestions
- Printable invoices and digital billing
- Firebase-authenticated user management system

User Roles

- Admin – Full access to all system functionalities including analytics and user management.
- Manager – Full access to analytics.

- Waiter – Manages customer orders, assigns tables, views inventory.
- Chef – Views and updates kitchen orders and their status.

Core Functionalities

- Order Management – Place, update, and fulfill table orders.
- Inventory Management – Track stock levels, usage, and restocking.
- Analytics – View sales, financial, employee, and inventory analytics.
- Authentication – Secure Firebase-based login for different roles.
- Invoice Generation – Generate and print invoices for customers.
- Role-based Dashboards – Tailored views and actions depending on user role.

Database Structure

The system uses Firebase Firestore as the backend database. Below are the primary collections used:

- Categories – Stores categories of menu items.
- Invoice – Contains generated invoices for orders.
- OrderDetails – Item-level breakdown for each order.
- employees – Information about restaurant staff.
- expenses – Financial records and expense tracking.
- inventory – Tracks current stock and consumption.
- items – List of available dishes and their metadata.
- orders – All active and historical orders.
- settings – Application settings and configuration.
- tables – Table metadata and occupancy state.
- users – User login and role assignment data.

GitHub Code Link

https://github.com/makram24/React-Pos