ERP System Documentation

Welcome to the documentation for our simplified ERP (Enterprise Resource Planning) system! This document will guide you through the functionalities, technologies used, and how to set up and interact with the system.

Table of Contents

Project Overview

Our ERP system is designed to streamline basic business operations. It consists of several key components:

- **Dashboard:** Provides an overview of the system's features and key metrics.
- **Items Management**: Allows for the management of product listings, including adding, editing, and deleting items.
- **Orders Management:** Facilitates viewing and handling orders, including updating order status and deleting orders.
- **Calendar View (Optional):** Displays orders on their expected delivery dates in a calendar interface.

Functionality

1. Dashboard

The dashboard serves as the central hub of our ERP system, providing users with an overview of key metrics and functionalities. Here's a breakdown of the features implemented:

- **Responsive Design**: The dashboard is designed to adapt seamlessly to various screen sizes, ensuring a consistent user experience across devices.
- **Navigation Bar**: A navigation bar is incorporated to facilitate easy navigation between different components of the system, including Items, Orders, and Calendar.
- **Quick Navigation Buttons**: Two buttons are placed strategically on the dashboard, allowing users to navigate directly to the Items and Orders management pages, enhancing user accessibility.
- **API Integration:** APIs are utilized to fetch and display real-time data on the dashboard, including the total number of products and orders. This ensures that users have up-to-date information at their fingertips.

2. Items Management

The Items Management component enables users to efficiently manage product listings. Here are the key functionalities implemented:

- **Mock Data Usage**: Static mock data from a db.json server is utilized for seamless data fetching and manipulation, ensuring a smooth user experience.
- **CRUD Operations**: Users can perform CRUD operations on product listings, including adding, editing, and deleting items. Proper validation is implemented to maintain data integrity and accuracy.
- **User-Friendly Interface**: An intuitive user interface is designed to enhance user interaction, making it easy for users to add, edit, and view product details.

3. Orders Management

The Orders Management module facilitates the handling of orders efficiently. Here's what this component offers:

- **Real-time Data Fetching**: Orders are fetched dynamically from the JSON server, ensuring that users have access to the latest order information.

- **Order Card Component**: Each order is presented in a visually appealing card format, containing options to update order status, delete orders, and view detailed order information.
- **Enhanced User Interaction**: Users can seamlessly update order status, delete orders, and view detailed order information with just a few clicks, enhancing the overall user experience.

4. Calendar Component

The Calendar component provides users with a visual representation of order delivery dates. Here's how it enhances the user experience:

- **Full Calendar Integration**: The FullCalendar module is utilized to design and implement the calendar view, meeting all specified requirements.
- **Order Visualization**: Orders are displayed on their respective delivery dates on the calendar, providing users with a clear overview of their delivery schedules.
- **Interactive Popup**: Clicking on a delivery date opens a popup modal, displaying detailed order information, enhancing user engagement and interaction.

Additional Details

During the project implementation, the following techniques and practices were employed:

- **Responsive Design**: All components are designed to be responsive, ensuring optimal user experience across devices.
- **Proper Code Structure**: The codebase is structured in a modular manner, with separate components for dashboard, items, orders, calendar, and navigation bar, enhancing code readability and maintainability.
- **Maintainability**: Emphasis is placed on writing clean, maintainable code, ensuring that future modifications and enhancements can be easily implemented.
- **Component Division**: The project is divided into separate components, such as dashboard, items, orders, calendar, and navigation bar, improving code organization and maintainability.

Technology Stack

Our front-end application is built using React, a popular JavaScript library for building user interfaces. We chose React for its modularity, component-based architecture, and efficient state management.

Code Structure

The codebase is organized into separate components for better readability and maintainability:

- **Dashboard:** Contains the dashboard component.
- **Items**: Manages product listings.
- **Orders**: Handles orders management.
- **Calendar**: Implements the calendar view (optional).
- **Navbar**: Provides navigation functionality.

Documentation

The README document in our project repository contains comprehensive instructions for setting up and interacting with the front-end application. It includes clear and concise guidelines for running the application, navigating through different sections, and utilizing its features. Additionally, screenshots are provided to demonstrate the functionality of the website.

Setup Instructions

To set up the ERP system locally, follow these steps:

1. Clone the project repository from GitHub.
2. Navigate to the project directory.
3. Install dependencies using npm install.
4. Start the development server with [npm start].
5. Start JSON Server db.json in new terminal using [json-server –watch db.json –port 8000]
6. Access the application in your web browser at http://localhost:3000.

Interacting with the System

Once the application is running, you can interact with the ERP system by navigating through the different sections using the navigation bar. Perform CRUD operations on products and orders as needed. Use the calendar view to visualize orders based on their expected delivery dates.