

# ERP System Documentation

## Table of Contents

1. [Project Overview](#)
  2. [Features](#)
  3. [Technology Stack](#)
  4. [Setup Instructions](#)
    - [Prerequisites](#)
    - [Installation](#)
  5. [Project Structure](#)
  6. [Detailed Component Overview](#)
    - [AuthContext](#)
    - [Sidebar](#)
    - [Dashboard](#)
    - [HR Module](#)
    - [Finance Module](#)
    - [Inventory Module](#)
    - [Projects Module](#)
    - [Role Management](#)
  7. [Routing](#)
  8. [Role-Based Access Control](#)
  9. [Real-Time Data Handling](#)
  10. [UI/UX Enhancements](#)
  11. [Error Handling and Validation](#)
  12. [Deployment](#)
  13. [Contributing](#)
  14. [License](#)
  15. [Contact Information](#)
-

## Project Overview

The ERP System is a comprehensive application designed to manage various aspects of a business, including Human Resources (HR), Finance, Inventory, and Projects. The system is built with a modular approach, allowing different departments to manage their data and processes efficiently. The application is equipped with role-based access control to ensure that sensitive data and functionalities are restricted to authorized users.

## Features

- **User Authentication and Authorization:** Secure login and signup with role-based access control.
- **Dashboard:** Overview of all modules with quick access links.
- **HR Module:** Manage employee data including adding, updating, and deleting employee records.
- **Finance Module:** Track financial transactions with options to add, update, and delete records.
- **Inventory Module:** Manage inventory items, including stock levels, pricing, and supplier information.
- **Projects Module:** Track and manage various projects within the organization.
- **Role Management:** Admin functionality to manage user roles and permissions.
- **Real-Time Updates:** Automatic updates of data across the application using Supabase's real-time features.
- **Responsive Design:** A user-friendly interface designed with Material UI, optimized for various devices.

## Technology Stack

- **Frontend:**
  - React.js
  - Material UI
- **Backend:**
  - Supabase (Database and Authentication)
- **Other Libraries:**
  - `react-router-dom` for routing
  - `@mui/icons-material` for icons

# Setup Instructions

## Prerequisites

Before setting up the project, ensure that you have the following installed:

- Node.js (v14 or higher)
- npm (v6 or higher)
- A Supabase account with a project created
- Git

## Installation

### 1. Clone the repository:

```
bash
Copy code
git clone https://github.com/yourusername/erp-system.git
cd erp-system
```

### 2. Install dependencies:

```
bash
Copy code
npm install
```

### 3. Create a `.env` file in the root directory and add your Supabase credentials:

```
plaintext
Copy code
REACT_APP_SUPABASE_URL=your-supabase-url
REACT_APP_SUPABASE_ANON_KEY=your-supabase-anon-key
```

### 4. Start the development server:

```
bash
Copy code
npm start
```

### 5. Access the application: Open your browser and navigate to <http://localhost:3000>.

## Project Structure

```
src/
├── components/
│   ├── AdminSignup.js
│   ├── Dashboard.js
│   ├── Finance.js
│   ├── HR.js
│   ├── Inventory.js
│   ├── Login.js
│   ├── Projects.js
│   ├── RoleSelection.js
│   ├── Sidebar.js
│   └── Signup.js
├── context/
│   └── AuthContext.js
├── styles/
│   └── index.css
├── App.js
├── index.js
├── supabaseClient.js
└── .env
```

## Detailed Component Overview

### AuthContext

- **Purpose:** Handles user authentication, session management, and role-based access control.
- **Key Methods:**
  - `login(email, password)`: Logs in a user and retrieves their roles.
  - `signup(email, password)`: Registers a new user.
  - `logout()`: Logs out the current user.

### Sidebar

- **Purpose:** Provides navigation links to different modules of the ERP system.
- **Features:**
  - Conditionally renders links based on user roles.
  - Styled using Material UI with icons for a modern look.

### Dashboard

- **Purpose:** Serves as the landing page after login, providing quick access to all modules.

- **Features:**
  - Displays a welcome message and navigation shortcuts.

## HR Module

- **Purpose:** Manages employee data.
- **Features:**
  - Add, edit, delete employee records.
  - Displays employee data in a table with real-time updates.

## Finance Module

- **Purpose:** Manages financial transactions.
- **Features:**
  - Add, edit, delete financial transactions.
  - Displays transaction data in a table with real-time updates.

## Inventory Module

- **Purpose:** Manages inventory items.
- **Features:**
  - Add, edit, delete inventory items.
  - Displays inventory data in a table with real-time updates.

## Projects Module

- **Purpose:** Manages organizational projects.
- **Features:**
  - Add, edit, delete project records.
  - Displays project data in a table with real-time updates.

## Role Management

- **Role Selection:**
  - Allows admins to select a role for a new user before creating the account.
- **AdminSignup:**
  - Displays the selected role and allows admins to create new user accounts with the specified role.

## Routing

The application uses `react-router-dom` for navigation. Here's a summary of the routes:

- `/login`: Login page
- `/signup`: Signup page
- `/dashboard`: Main dashboard (protected)
- `/hr`: HR Module (protected)
- `/finance`: Finance Module (protected)
- `/inventory`: Inventory Module (protected)
- `/projects`: Projects Module (protected)
- `/admin/roles`: Manage User Roles (admin only)
- `/admin/signup`: Admin Signup page (admin only)

## Role-Based Access Control

- **Roles:** Admin, Manager, Employee
- **Permissions:**
  - **Admin:** Full access to all modules, including role management.
  - **Manager:** Access to HR, Finance, Inventory, and Projects modules.
  - **Employee:** Access to view-only data in HR, Finance, Inventory, and Projects modules.

## Real-Time Data Handling

- **Supabase** is used for real-time data updates across the application.
- Data changes are automatically reflected in the UI without requiring a manual refresh.

## UI/UX Enhancements

- **Material UI:** The entire application UI is built using Material UI components for a consistent and responsive design.
- **Icons:** Icons from `@mui/icons-material` are used in the sidebar and action buttons.
- **Notifications:** Toast notifications are used to provide feedback to the user.

## Error Handling and Validation

- **Form Validation:** All forms include basic validation to ensure required fields are filled.
- **Error Messages:** Clear error messages are displayed to users in case of any issues (e.g., failed login, invalid data).

# Deployment

## Netlify (Frontend)

1. **Create a Netlify Account.**
2. **Connect the GitHub Repository.**
3. **Set Environment Variables:**
  - Add `REACT_APP_SUPABASE_URL` and `REACT_APP_SUPABASE_ANON_KEY` in Netlify's environment settings.
4. **Deploy the Site:**
  - The site will be automatically built and deployed after pushing changes to the main branch.

## Supabase (Backend)

- **Supabase** handles the backend, including the database and authentication.
- Ensure your Supabase project is correctly configured with the necessary tables and roles.

# Contributing

1. **Fork the Repository:** Create a fork of this repository on GitHub.
2. **Create a Branch:** Create a new branch for your feature or bug fix.
3. **Submit a Pull Request:** Once your changes are ready, submit a pull request for review.

# Contact Information

- **Author:** Soham Patil
- **Email:** sohammpatil0711@gmail.com
- **GitHub:** <https://github.com/NotSoham07>