**Frontend Development with React.js**

**Store Manager – Frontend Documentation**

**1.Introduction**

**Project Title: Store Manager**

**Team Members:**

**[K Tamil kala ] – Frontend Developer**

**[K Sangavi] – UI/UX Designer**

**[S Savvanthi] – State Management & Routing**

**[K Kokila] – Documentation & Testing**

**2. Overview**

**Purpose:**

**The Store Manager project is a web-based application designed to simplify inventory, sales, and product management for retail stores. It allows store managers to keep track of stock levels, manage product categories, process sales, and generate reports through a user-friendly interface.**

**Features:**

**Dashboard for quick store insight Product management (add, edit, delete products)Inventory tracking with stock alerts Sales management and history logs. User authentication and role-based access (admin/manager). Responsive design for desktop and mobile**

**3.Architecture**

**Component structure :**

**App.js – Root component handling routing.**

**Navbar – Navigation across dashboard, products, sales, etc. Dashboard – Summary of sales, inventory, and performance**

**Pro – Displays all products with CRUD operations. Sales – Manages sales transactions.**

**Auth – Login/Register components for authentication.**

**State Management:**

1. **Context API for global state (authentication, user roles).**
2. **Local state (useState, useReducer) for component-specific interactions (form handling, modals).**
3. **Routing:**
4. **Implemented using react-router-dom.**
5. **Routes include /login, /dashboard, /products, /sales, /reports.**
6. **PrivateRoute wrapper for protected routes.**

**4.Setup Instructions**

**Prerequisites:**

**Node.js (>= 16.x)**

**Npm or yarn package manager**

**Git**

**Installation:**

1. **Clone the repository :**

**Git clone** [**https://github.com/your-repo/store-manager.git**](https://github.com/your-repo/store-manager.git)

**Cd store-manager/client**

1. **Install dependencies:**

**Npm install**

1. **Configure environment variables in .env:**

**REACT\_APP\_API\_URL=http://localhost:5000/api**

**5.Folder Structure**

**Store-manager/**

**client/**

**public/**

**src/**

**assets/ # Images, icons, static files**

**components/ # Reusable UI components (Navbar, Button, Modal)**

**pages/ # Page-level components (Dashboard, Products, Sales)**

**context/ # Context API for global state. / # Custom React hooks**

**utils/ # Helper functions**

**styles/ # Global CSS / styled-components**

**App.js # Main application entry**

**Utilities:**

**fetchWrapper.js – Handles API calls.**

**useAuth.js – Custom hook for authentication state.**

**Validators.js – Input validation helpers.**

**6.Running the Application**

**Start the development server:**

**Cd client**

**Npm start**

**Application runs on** [**http://localhost:3000**](http://localhost:3000)**.**

**7.Component Documentation**

**Key Components:**

* + **Navbar – Navigation links, user profile dropdown.**
  + **DashboardCard – Reusable info card for sales/inventory.**
  + **ProductTable – Displays products with search, sort, and actions.**
* **Reusable Components:**
  + **Button – Custom styled button supporting variants (primary, secondary).**
* **Modal – For confirmations (delete product, update stock).**
  + **FormInput – Reusable input with validation support.**

**8.State Management**

**Global State:**

* **AuthContext → Stores user authentication, roles, and JWT tokens.**
* **StoreContext → Manages product list, sales history, and stock data.**

**Local State:**

* **Handled via useState in forms, search filters, and modals.**

**9.User Interface**

* **(Screenshots to be inserted after UI implementation) Dashboard with metrics cards. Product management page with search and CRUD options. Sales entry page with invoice generator.**
* **Login page with authentication.**

**10.Styling**

**CSS Frameworks/Libraries:**

* + **Tailwind CSS for utility-first styling.**
  + **React Icons for iconography.**
  + **Theming:**
  + **Custom color palette defined in tailwind.config.js. Dark mode toggle under consideration.**

**11.Testing**

**Testing Strategy:**

* **Unit tests with Jest & React Testing Library for components.**
* **Integration tests for API calls and context.**
* **E2E tests planned using Cypress.**

**Code Coverage:**

* **Configured via Jest with coverage reports generated on build.**

**12.Screenshots or Demo**

**[ ] Insert screenshots of Dashboard, Products, and Sales pages.**

**[ ] Provide hosted demo link (Netlify/Vercel).**

**13.Known Issues**

* **Product search filter needs optimization for large datasets.**
* **Sales reporting page still under development.**
* **Mobile responsiveness for tables requires improvement.**

**14.Future Enhancements**

* + **Add barcode scanner support for product entry.**
  + **Implement dark mode theme.**
  + **Advanced analytics and chart visualizations.**
  + **Offline mode with PWA support.**