**Inventory Management Frontend**

**1. Introduction**

**Project Title: Inventory Management Frontend**

**Team Members:**

**Poornitha K [Team Leader ] – Frontend Developer**

**Moulithilak M S – Designer**

**Guru Prasad P V – Quality Assurance Tester**

**Dharshini V – Project Manager**

**2. Project Overview**

**Purpose:**

- The Inventory Management Frontend is designed to provide a user-friendly interface for managing inventory items, tracking stock levels, and facilitating inventory operations for businesses.

**Features:**

- Dashboard for inventory overview

- Add, edit, and delete inventory items

- Search and filter functionality

- Responsive design for desktop and mobile

- Integration with backend API

**3. Architecture**

Component Structure:

- Major components include Dashboard, ItemList, ItemForm, SearchBar, and Navigation.

- Components interact via props and shared state, with reusable UI elements for forms and lists.

**State Management:**

- Utilizes React's Context API for global state (e.g., inventory data, user authentication).

- Local state managed with useState and useReducer hooks within components.

**Routing:**

- Uses `react-router-dom` for client-side routing between pages such as Dashboard, Items, and Settings.

**4. Setup Instructions**

**Prerequisites:**

- Node.js (v14 or higher)

- npm (v6 or higher)

**Installation:**

**1. Clone the repository:**

```powershell

git clone <repository-url>

```

**2. Navigate to the client directory:**

```powershell

cd inventory-management-frontend

```

**3. Install dependencies:**

```powershell

npm install

```

4. Configure environment variables in a `.env` file as needed.

**5. Folder Structure**

**Client:**

- `src/components/` - Reusable React components

- `src/pages/` - Page-level components (Dashboard, Items, etc.)

- `src/assets/` - Images, icons, and static files

- `src/styles/` - CSS or styling files

**Utilities:**

- `src/utils/` - Helper functions for API calls, formatting, and custom hooks

**6. Running the Application**

- To start the frontend server locally:

```powershell

npm start

```

Run this command inside the `inventory-management-frontend` directory.

**7. Component Documentation**

**Key Components:**

- `Dashboard`: Displays inventory summary and statistics

- `ItemList`: Shows a list of inventory items

- `ItemForm`: Form for adding/editing items

- `SearchBar`: Allows searching/filtering items

**Reusable Components:**

- `Button`, `Modal`, `InputField` - Used across multiple pages for consistent UI

**8. State Management**

**Global State:**

- Managed via Context API for inventory data and user session

- State flows from context provider to child components

**Local State:**

- Managed with useState/useReducer for form inputs, modal visibility, and filters

**9. User Interface**

- Screenshots or GIFs showcasing UI features should be added here:

- Dashboard view

- Item management form

- Search and filter interactions

**10. Styling**

**CSS Frameworks/Libraries:**

- Uses CSS Modules and/or Styled-Components for scoped styling

- May use a framework like Bootstrap or Material-UI (specify if used)

Theming:

- Custom themes can be implemented via CSS variables or styled-components theme provider

**11. Testing**

**Testing Strategy:**

- Unit tests for components using Jest and React Testing Library

- Integration tests for form submissions and API calls

- End-to-end tests with Cypress (if implemented)

**Code Coverage:**

- Jest coverage reports to ensure adequate test coverage

**12. Screenshots or Demo**

- Add screenshots or a link to a live demo here:

- ![Dashboard Screenshot](src/assets/dashboard.png)

- [Live Demo](<demo-url>)

**13. Known Issues**

- List any known bugs or issues here:

- [Example] Search functionality may not filter by all fields

- [Example] Responsive layout issues on some mobile devices

**14. Future Enhancements**

**Potential future features:**

- Add user authentication and roles

- Implement advanced analytics and reporting

- Add animations and enhanced UI transitions

- Improve accessibility and mobile responsiveness