

# Frontend Development with React.js — Project Documentation

## Introduction

Project Title: Store Manager — Keep Track of Inventory

Team ID: NM2025TMID48238

Team Size: 4

Team Members:

- Team Leader: NANDHUPRIYA S — madhanselvan72@gmail.com
- Team Member: LOGESWARI G — glogeswari0306@gmail.com
- Team Member: MEGALA A — megala.as@gmail.com
- Team Member: HEMALATHA R — ravim256643@gmail.com

## Project Overview

Purpose: To provide a simple and efficient way for store owners to manage their inventory.

Features:

- Product management (add, update, delete products)
- Stock level tracking with real-time updates
- Low-stock alerts and highlights
- Dashboard with product statistics
- Basic sales and stock reports
- User login page

## Architecture

Component Structure: App, Auth, Dashboard, Products, ProductForm, Reports, Shared components.

State Management: React Context + Reducer (global), useState (local).

Routing: Implemented with react-router-dom. Routes include /login, /dashboard, /products, /products/:id, /reports.

## Setup Instructions

Prerequisites: Node.js v16+, npm/yarn, Git.

Installation:

1. git clone
2. cd client
3. npm install
4. Configure .env (e.g., REACT\_APP\_API\_URL=http://localhost:5000)
5. npm start

## Folder Structure

/client /public /src /api /assets /components /pages /context /hooks /styles /utils App.js index.js

## Running the Application

- Development: npm start @ http://localhost:3000
- Build for production: npm run build
- Testing: npm test

## Component Documentation

- ProductList: Displays products with edit/delete options.
- ProductForm: Form for adding or editing products.
- DashboardCard: Displays single statistics.

## State Management

- Global: Context (Products, Auth, Toasts).
- Local: Forms, modals, small UI states.

## User Interface

- Dashboard with stats - Table for product management - Modals for adding/editing products - Reports with charts

## Styling

Tailwind CSS used. Light theme by default.

# Testing

Unit Tests: Reducers, utility functions (Jest).

Component Tests: ProductForm, Dashboard (React Testing Library).

Optional: Cypress for end-to-end testing.

## Known Issues

- No advanced role-based control
- Reports limited to basic stock/sales
- Needs backend or mock server for functionality

## Future Enhancements

- Role-based access control
- Barcode scanner integration
- CSV import/export
- Advanced analytics and charts
- Offline support with caching