

Project Documentation

Project Title

1.Introduction

- **Project Title:** INVENTORY STORY MANAGEMENT
- **Team ID:** Nm2025tmid37818
- **Team Leader:** P.Pooja pjayanthi521992@gmail.com

- **Team Members:**

G.Nithya Dharshini

nithyakathika23@gmail.com

T.Pradiksha

pradiksha.thangaraju@gmail.com

S.Monika

Mon633977@gmail.com

2.Project Overview

Purpose: Manage stock, sales, and purchases in a retail store.

Features:

- Add, edit, and delete products
- Track stock levels
- Generate sales reports
- User authentication (Admin, Staff)
- Dashboard with charts

3.Architecture

Frontend: React.js with Material UI

Backend: Node.js + Express.js

Database: MongoDB (Products, Users, Transactions)

4.Setup Instructions

Prerequisites:

- Node.js
- MongoDB
- Git

Installation Steps:

Clone the repository git clone

Install client dependencies cd client && npm
install

Install server dependencies cd ../server &&
npm install

5. Folder Structure

Store-Inventory-Management/

|

|— client/ # React frontend

| |— components/ # Reusable UI components

| |— pages/ # React pages

| |— context/ # State management

| |— styles/ # CSS / Tailwind styles

| |— utils/ # Helper functions

|

|— server/ # Node.js + Express backend

| |— routes/ # API routes

| |— models/ # MongoDB schemas

| |— controllers/ # Business logic

| |— middlewares/ # Authentication & validation

| |— config/ # Database & environment configs

|

└─ README.md

6. Running the Application

Frontend: cd client && npm start

Backend: cd server && npm start

Access at: <http://localhost:3000>

7. API Documentation

Products: POST /api/products/create, GET /api/products/:id, PUT /api/products/:id, DELETE /api/products/:id

Users: POST /api/users/register, POST /api/users/login

Reports: GET /api/reports/sales, GET /api/reports/inventory

8. Authentication

JWT-based login system

Middleware to protect private routes

9. User Interface

- Dashboard for stock overview,
- Products & Categories for easy management,
- Orders & Reports for tracking sales,
- User Management with secure login.

10. Testing

- Unit Testing – Verified each module (Products, Orders, Reports).
- Integration Testing – Checked smooth interaction between frontend and backend.
- User Testing – Ensured the interface is simple and error-free.
- Result – Application runs successfully with correct data flow.

11. ScreenShot or Demo

[Demo Video](#)

12. Known Issues

- Limited to single-store management (no multi-branch support).
- No barcode/QR code integration.
- Basic reporting features only.
- Mobile application not yet available.

13. Future Enhancements

- Barcode/QR Code support for quick product updates.

- Mobile app for easy inventory access.
- Cloud storage for real-time multi-store management.
- AI-based stock prediction to avoid shortages.
- Automated supplier orders when stock is low.
- Advanced reports & analytics for better decision-making.