

# Project Documentation

---

## Store Manager: Keep Track of Inventory

### [1]. Introduction

- **Project Title:** Store Manager: Keep Track of Inventory.  
**Team ID:** NM2025TMID36672.
- **Team Leader:** PRITHIKA.V ;  
**Mail I.D:** [prithivenkatesan2007@gmail.com](mailto:prithivenkatesan2007@gmail.com)
- **Team Member:**
  - PRIYADHARSHINI.S  
**Mail I.D:** [priyap8560@gmail.com](mailto:priyap8560@gmail.com)
  - ROSHINI.J  
**Mail I.D:** [roshinijohnjoseph@gmail.com](mailto:roshinijohnjoseph@gmail.com)
  - SHEEBA.G  
**Mail I.D:** [sheebawincy@gmail.com](mailto:sheebawincy@gmail.com)
  - THABITHA D  
**Mail I.D:** [thabitha7707@gmail.com](mailto:thabitha7707@gmail.com)

### [2].Project Overview

- **Purpose:** Inventory Management System (IMS) is a powerful web application designed to simplify and optimize inventory tracking in retail store.
- **Features:**
  - Real-time stock updates.
  - Cart and checkout system.
  - Sale record tracking.
  - Depleting stock alerts.

### [3]. Architecture

- **Frontend:** React.js, HTML, CSS, JavaScript
- **Backend :** Node.js.
- **Database:** Mongo DB (product catalog, sales and user data)

### [4]. Setup Instructions

- **Prerequisites:**
  - Node.js.
  - Visual Studio Code.
  - JavaScript.
  - Git Hub.
  - React.js
- **Installation Steps:**
  - # clone repository directory and install libraries.
  - # Initialize a new React application.
  - # Install dependencies

### [5]. Folder Structure

- public
  - vite.html
- src
  - app
  - assets
  - features ➤ pages
  - App.css
  - App.js
  - index.css
  - main.js
  - test.js
- .gitignore
- eslint.config.js

- index.html
- package.JSON
- package-lock.JSON
- README.md
- Vite.config.js

## [6]. Running the Application

- **Frontend:** npm install
- **Backend:** npm run dev
- **Access:** Visit <http://localhost:5174>

## [7].API Documentation

- **User:**
  - /api/auth/register → Register a new user.
  - /api/auth/login → Login with credentials.
- **Projects:**
  - /api/projects/create
  - /api/projects/:id • Applications: /api/apply
- **Charts:**
  - /api/cart/add → Add product to cart.
  - /api/sales → Fetch all sales records.

## [8]. Authentication

- Users must register and log in before accessing product management and checkout features.
- JWT tokens are used for secure authentication.

## **[9]. User Interface**

- Product Catalog.
- Cart.
- Inventory.
- Add New Product
  - Product name.
  - Product image URL.
  - Price.
  - Stock.
  - Tags.
- Sales Record.

## **[10]. Testing**

- Test individual components.
- Test context and reducers.
- Debug UI/UX issues.

## **[11].Known Issues**

- No barcode/QR code scanning support.
- Requires stable internet for real-time updates (if using MongoDB Atlas).

## **[12].Further Enhancements**

- Add barcode scanning for faster product entry.
- Generate automated sales reports (daily, weekly, monthly).

## **[13]. Screenshot**

