

# Project Documentation

## 1. Introduction

### **Project Title: Inventory Management System**

Team ID: NM2025TMID38010

Team Leader: **YUVASRI M** (sumathi4843@gmail.com)

Team Members:

**VARSHINI** (varshinikvm6@gmail.com)

**GEETHALAKSHMI V** (geetha13047@gmail.com)

**HEMAVATHI A** (hemavathia475@gmail.com)

## 2. Project Overview

The Inventory Management System is designed to streamline stock tracking and sales management. Many businesses rely on manual processes, which cause errors and inefficiencies. This project provides an automated, simple, and user-friendly interface where products, stock, and sales are managed efficiently. It ensures accurate updates and supports real-time decision-making.

## 3. Features

- User-friendly dashboard
- Add, update, and delete products
- Track sales records
- Search and filter products
- Real-time stock updates
- Report generation

## 4. Architecture

- Frontend: React.js with Tailwind CSS
- Backend: JSON (local state management, simulated backend)
- Database: MySQL (planned for integration)

## 5. Setup Instructions

- Prerequisites: Node.js, MySQL, Git, Visual Studio Code
- Clone the repository: `git clone`
- Navigate to project folder: `cd inventory-management-frontend`
- Install dependencies: `npm install`
- Start the application: `npm start`
- Access via: `http://localhost:3000`

## 6. Folder Structure

```
inventory-management-frontend/ |-  
- public/  
|-- src/  
|-- components/  
|-- Cart/  
|-- Inventory/  
|-- Product/  
|-- Sales/  
|-- context/  
|-- hooks/  
|-- pages/
```

## 7. Running the Application

```
Frontend: cd inventory-management-frontend npm  
start Access the application at  
http://localhost:3000
```

## 8. API Documentation

Currently, the project runs on a frontend-only architecture using context and hooks for state management. Backend and API endpoints can be integrated with Node.js and Express in the future.

## 9. Authentication

At present, authentication is not implemented. Future versions can integrate JWT-based authentication for secure login.

## 10. User Interface

- Login Page
- Dashboard with inventory overview
- Add/Update Product Page
- Sales Management Page
- Reports Page
- Error/Not Found Page

## 11. Testing

Manual testing was carried out during development using browser dev tools. Future testing can include Jest and Postman for API validation.

## 12. Screenshots or Demo

Screenshots of the application can be added here.

### **13. Known Issues**

- Authentication not implemented
- Backend API not connected
- Data persistence limited to local state

### **14. Future Enhancements**

- Integrate backend with Node.js/Express.js
- Implement MySQL database support
- Add JWT authentication and role-based access
- Generate detailed sales and inventory reports
- Improve UI with more analytics dashboards