INVENTRY MANAGEMENT SYSTEM

1. Introduction

• Project Title: Inventory Management System

• Team ID: NM2025TMID39594

• Team Leader: THOGESH.M & thogimarkson@gmail.com

• Team Members:

– RANJITH KUMAR.G & ranjithneymar321@gmail.com

– SARAN KUMAR.S & sarankumarsrini2007@gmail.com

– HARISH.R & harishharish96232@gmail.com

2. Project Overview

• Purpose: The Inventory Management System (IMS) is designed to help track and manage

inventory data efficiently, reducing manual errors and providing a centralized dashboard

for stock management.

• Features:

– User-friendly interface with React frontend

– Inventory item management (CRUD operations)

– Responsive styling with Tailwind CSS

– Scalable architecture for future backend and database integration

3. Architecture

• Frontend: React.js with Tailwind CSS

• Backend: Planned Node.js + Express.js (future integration)

Database: Planned MongoDB (future integration)

4. Setup Instructions

• Prerequisites: Node.js, npm, Git

Installation Steps

Clone the repository

git clone https://github.com/Shreyas-karthick/Inventory-Management-System.git

cd Inventory-Management-System

# Install dependencies

npm install

# Run application

npm start

5. Folder Structure

Inventory-Management-System/

|-- src/ # React frontend code

|\_\_ public/ # Static assets

|\_\_ package.json

|\_\_ tailwind.config.js

6. Running the Application

• Development:

cd Inventory-Management-System

npm start

• Access: Visit http://localhost:3000

7. API Documentation

Currently, the system is frontend-only. Suggested endpoints for backend integration:

• /api/items [GET, POST]

• /api/items/:id [GET, PUT, DELETE]

8. Authentication

Planned JWT-based authentication for secure login and middleware-protected routes once

backend is integrated.

User Interface

• Landing Page

• Inventory Dashboard

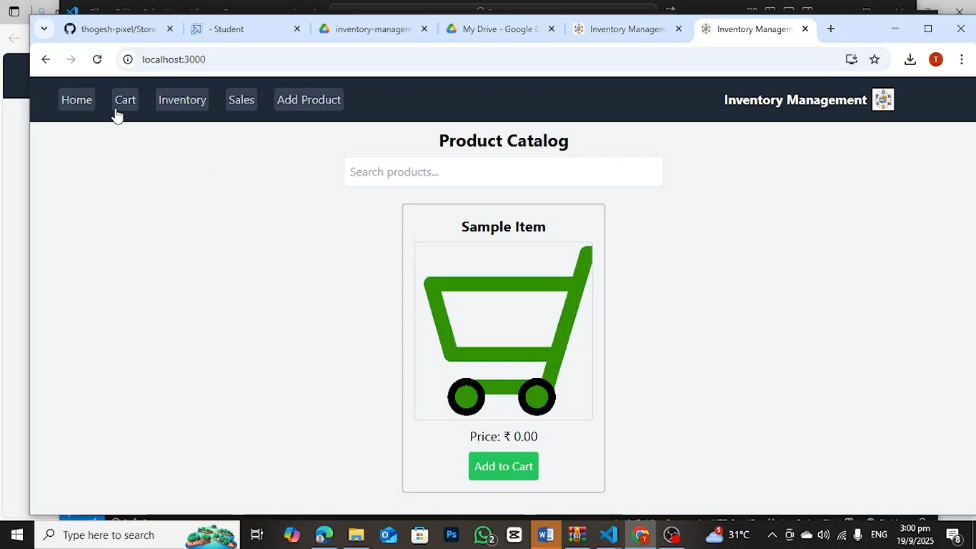
• Item Management Pages

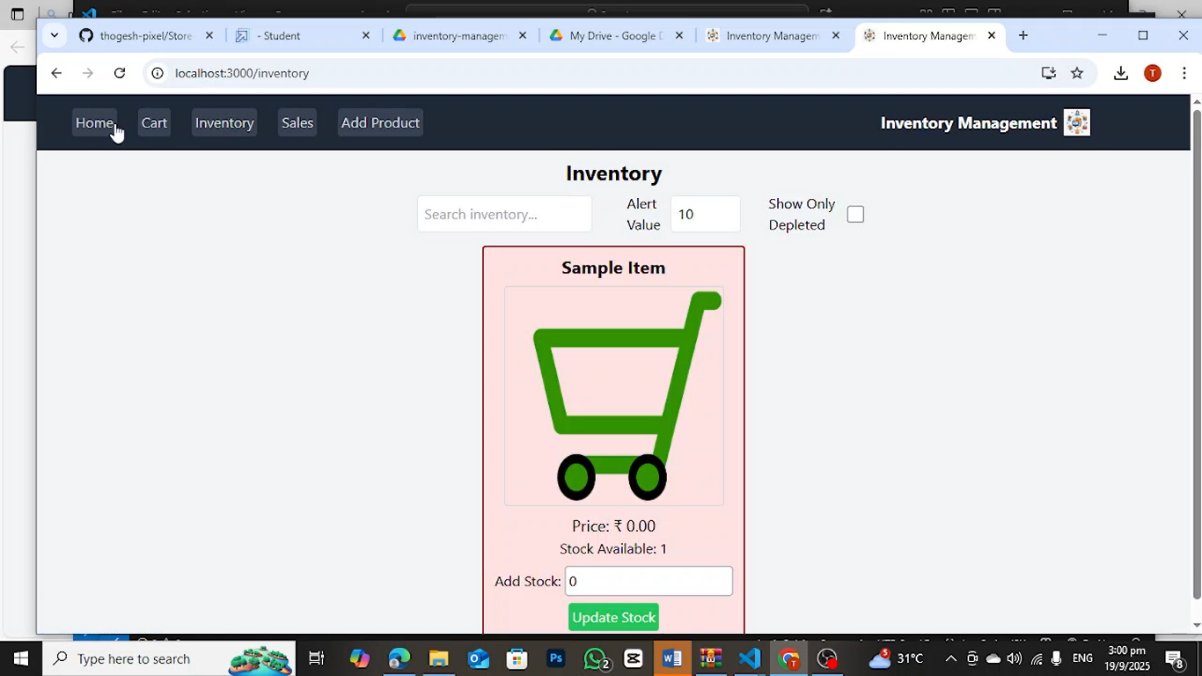
10. Testing

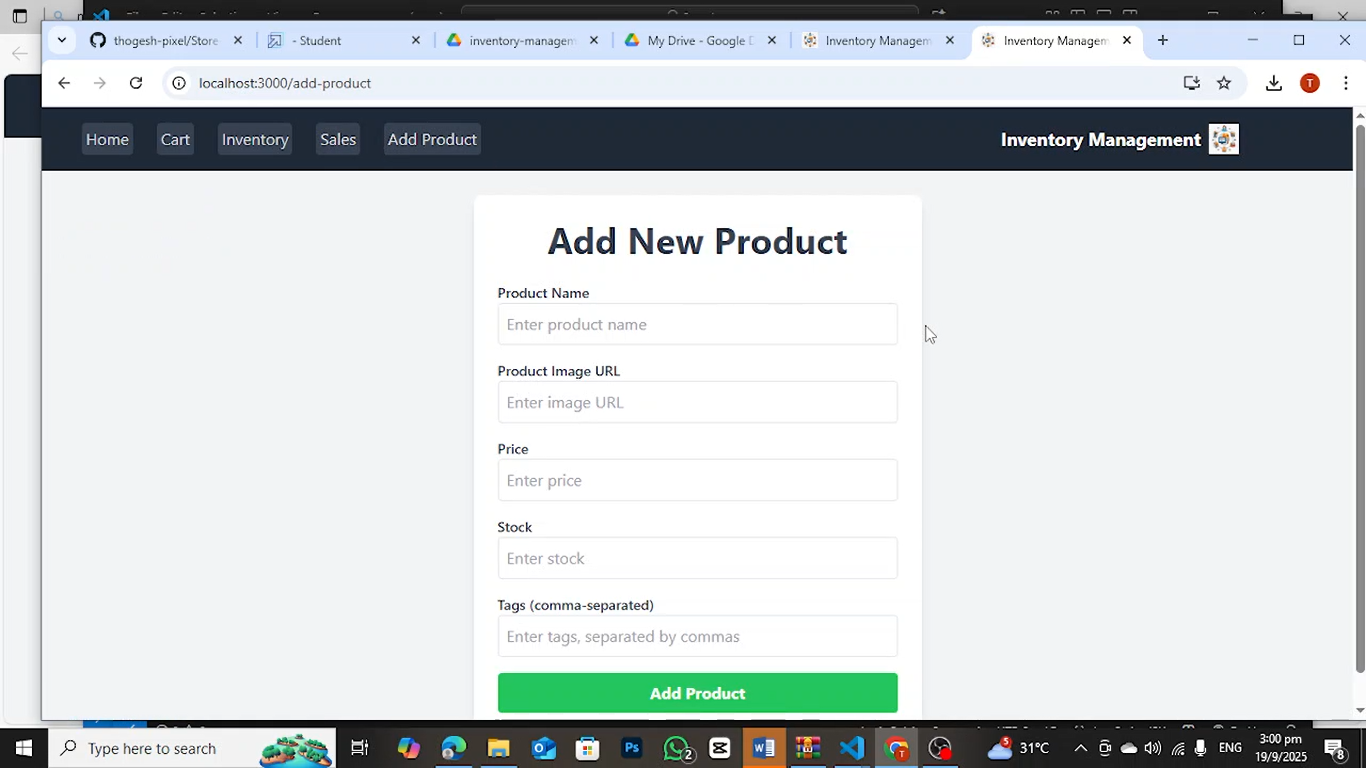
• Manual testing during development

• Tools: Chrome Dev Tools, Jest (planned), React Testing Library

11. Screenshots







Demo link : https://drive.google.com/file/d/1K2V87OLbM0-bq-TKSwNe1I-KyUCwtTch/view?usp=sharing

Known Issues

• Currently frontend only, no backend or database integration

• No authentication implemented yet

13. Future Enhancements

• Backend API integration (Node.js + Express.js)

• Database support (MongoDB)

• Authentication & Authorization (JWT)

• Reporting dashboards & stock alerts

• Search, filter, and sorting features