Project Documentation

# Project Title

Storage Manager for Inventory Tracking

# Team Details

Team ID : NM2025TMID40058

Team Size : 4

Team Leader : KANMANI S

Email id:

sac2427bca5231@ssacollegechennai.com

Team member : JESLYN VERONICA K

Email id:

[sac2427bca5783@ssacollegechennai.com](mailto:sac2427bca5@ssacollegechennai.com)

Team member : JOANNA MARIA D

Email id:

[sac2427bca5249@ssacollegechennai.com](mailto:sac2427bca5249@ssacollegechennai.com)

Team member : KEERTHI J

Email id:

[sac2427bca5352@ssacollegechennai.com](mailto:sac2427bca5352@ssacollegechennai.com)

# 1. Introduction

An Inventory Manager (or Storage Manager) is a system designed to keep track of products, their quantities, and their movement across sales and purchases. In any business—whether retail, wholesale, or e-commerce—managing stock efficiently is crucial to avoid overstocking, understocking, or financial losses.  
  
The storage manager acts as the central hub for monitoring:  
- What items are available in stock  
- How many units are sold  
- When new items are added  
- Real-time updates on product prices and availability  
  
By using an inventory management system, businesses can:  
- Track stock levels accurately  
- Update records automatically when a sale or new product entry happens  
- Prevent errors caused by manual record-keeping  
- Improve decision-making, like knowing when to restock or which products sell the most

# 2. Project Overview

Purpose: The Storage Manager helps businesses manage inventory by tracking product stock, sales, purchases, and availability in real time.

Features:

* – Product catalog management
* – Add and update product details
* – Sales and purchase record tracking
* – Real-time stock monitoring
* – User-friendly interface

# 3. Architecture

• Frontend: React.js with Bootstrap and Material UI

• Backend: Node.js and Express.js managing server logic and API endpoints

• Database: MongoDB stores product data, sales records, and user information

# 4. Setup Instructions

Prerequisites:

* – Node.js
* – MongoDB
* – Git
* – React.js
* – Express.js – Mongoose – Visual Studio Code

Installation Steps:

* # Clone the repository: git clone <repository\_url>
* # Install client dependencies: cd client && npm install
* # Install server dependencies: cd ../server && npm install

# 5. Folder Structure

Storage-Manager/

|-- client/ # React frontend

| |-- components/

| |-- pages/

|-- server/ # Node.js backend

| |-- routes/

| |-- models/

| |-- controllers/

# 6. Running the Application

Frontend:

cd client && npm start

Backend:

cd server && npm start

Access: Visit http://localhost:3000

# 7. API Documentation

• User: /api/user/register, /api/user/login

• Products: /api/products/create, /api/products/:id

• Sales: /api/sales/create, /api/sales/:id

# 8. Demo Link

<https://drive.google.com/file/d/1-ooaT0G9_--CaTP6t3fKYsxK3LmcmDFg/view?usp=drivesdk>

# 9. Authentication

JWT-based authentication for secure login. Middleware protects private routes.

# 10. User Interface

• Product Catalog Page

• Add Product Page

• Sales Record Page

• Admin Dashboard

# 11. Testing

• Manual testing during milestones

• Tools: Postman, Chrome Dev Tools

# 12. Known Issues

Some features may need optimization for large-scale inventory data.

# 13. Future Enhancements

• Add barcode/QR code scanning for products

• Mobile app integration

• Advanced analytics and reporting