**TITLE: Inventory Management System**

**1.Introduction**

**Team ID**: NM2025TMID37999

**Team Leader** : Jayalakshmi.M (📧 mohanm53371@gamail.com)

**Team Members:**

Jesmitha.R(📧 jasmitharavi7@gmail.com)

Jayapradha.A (📧 jayapradha222006@gmail.com)

Mohana.M (📧 mohanam11092007@gmail.com)

**2. Project Overview**

📝 Purpose:

The Inventory Management System helps businesses track, manage, and update tile stock levels with real-time accuracy.

✨ Key Features:

📦 Tile product registration (SKU, color, size, category)

📉 Low stock alerts

🏢 Supplier & purchase order tracking

💵 Sales history & reporting

🧾 Barcode/QR code scanning

👤 Role-based access (Admin, Manager, Staff)

**3. System Architecture**

Frontend: React.js + Tailwind CSS

Backend: Node.js + Express.js

Database: MongoDB

Authentication: JWT

💻 Example API Flow:

graph TD

A[User] --> B[React Frontend]

B --> C[Express API]

C --> D[MongoDB Database]

**4. Setup Instructions**

Prerequisites: Node.js, MongoDB, Git, VS Code

🔧 Installation Steps:

# Clone the repository

git clone https://github.com/example/inventory-management.git

# Install frontend dependencies

cd client

npm install

# Install backend dependencies

cd ../server

npm install

**5. Folder Structure**

Inventory-Management/

│-- 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

🌐 Visit → http://localhost:3000

**7. API Documentation**

📌 User APIs

POST /api/user/register

POST /api/user/login

📌 Inventory APIs

POST /api/inventory/add

GET /api/inventory/:id

PUT /api/inventory/update/:id

DELETE /api/inventory/delete/:id

📌 Example Code – Add Inventory Item

// server/controllers/inventoryController.js

exports.addItem = async (req, res) => {

try {

const item = new Inventory(req.body);

await item.save();

res.status(201).json({ message: "Item added successfully", item });

} catch (err) {

res.status(400).json({ error: err.message });

}

};

**8. Authentication**

🔑 JWT-based authentication

🔐 Role-based access (Admin, Staff, Manager)

**9. User Interface**

Dashboard showing stock summary

Product listing with filters

Supplier panel

Sales report generation

**10. Testing**

🛠 Tools: Postman, Jest, Chrome DevTools

📌 Example Jest Test:

test("should create a new inventory item", async () => {

const res = await request(app)

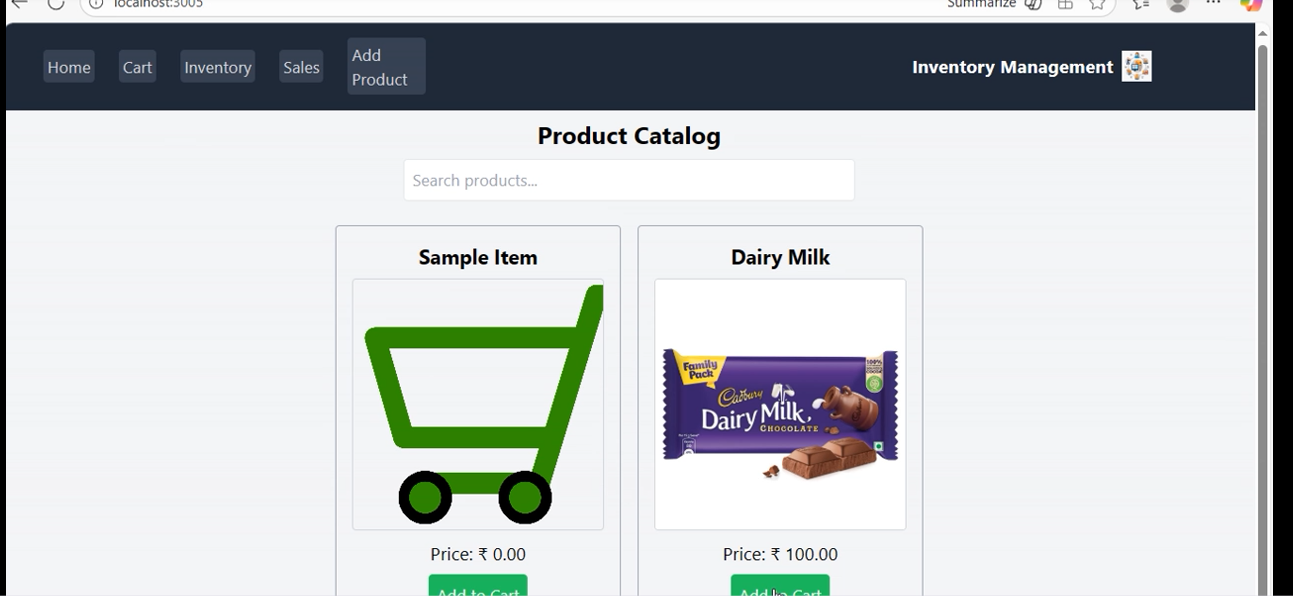
.post("/api/inventory/add")

.send({ name: "Tile A", quantity: 100 });

expect(res.statusCode).toBe(201);

});

**11. Screenshots / Demo**

****

**12. Known Issues**

Offline mode not fully supported

Bulk import feature pending

**13. Future Enhancements**

📱 Mobile App

📊 AI-powered demand forecasting

🌍 Multi-language support

☁ Cloud backup