**Project Documentation**

**1. Introduction**

**Project Title:** Store Manager Pro: Intelligent Retail Solution

**Team ID:** NM2025TMID40003

**Team Leader**: KAUSHIKA [kaushikaa83@gmail.com]

**Team Members:**

JENITA G[sac2427csc5463@ssacollegechennai.com]

JOTHIKA L[Jothika sac2427csc5412@ssacollegechennai.com]

KAVINISHA H[sac2427scs5501@ssacollegechennai.com]

YAMINI S[sac2427csc5301@ssacollegechennai.com]

**2. Project Overview**

Store Manager project aims to enhance overall store operations by streamlining inventory management, improving customer service protocols, and optimizing staff workflows. Led by the store manager, the initiative focuses on increasing sales performance, reducing stock discrepancies, and elevating the in-store experience through targeted training and layout redesign. By integrating data-driven decision-making and team accountability, the project seeks to create a more efficient, engaging, and profitable retail environment.

**Features:**

– Centralized inventory with stock alerts

– Automated billing with multiple payment options

– Customer loyalty and rewards management

– Employee attendance and payroll tracking

– Sales forecasting with visual reports

– Secure role-based access with admin controls

**3. Architecture**

Frontend: Angular + Bootstrap for responsive and dynamic UI

Backend: Django REST Framework for secure API development

Database: PostgreSQL (optimized for large-scale retail data)

**4. Setup Instructions**

Prerequisites:

– Python 3.12

– PostgreSQL

– Node.js (for frontend)

– Git

– VS Code / PyCharm

Installation Steps:

# Clone repository

git clone

# Setup frontend

cd frontend

npm install

# Setup backend

cd ../backend

pip install -r requirements.txt

**5. Folder Structure**

store-manager-pro/

├── frontend/ # Angular UI

│ ├── components/

│ ├── services/

│ └── assets/

├── backend/ # Django REST API

│ ├── apps/

│ ├── models/

│ ├── views/

│ └── utils/

└── README.md

**6. Running the Application**

Frontend:

cd frontend

ng serve

Backend:

cd backend

python manage.py runserver

Access: Visit http://localhost:4200

**7. API Documentation**

Authentication:

– POST /api/auth/register – Register user

– POST /api/auth/login – User login

Products:

– GET /api/products – Fetch all products

– POST /api/products – Add product

– PUT /api/products/:id – Update product

Sales:

– POST /api/sales – Record new sale

– GET /api/sales/report – Sales summary

**8. Authentication**

– JWT (JSON Web Token) for session handling

– Multi-role support: Admin, Cashier, Supervisor

**9. User Interface**

– Dashboard with revenue & sales trends

– Inventory management with search & filters

– Billing page with receipt generation

– Customer loyalty & rewards tracking

– Employee attendance and payroll view

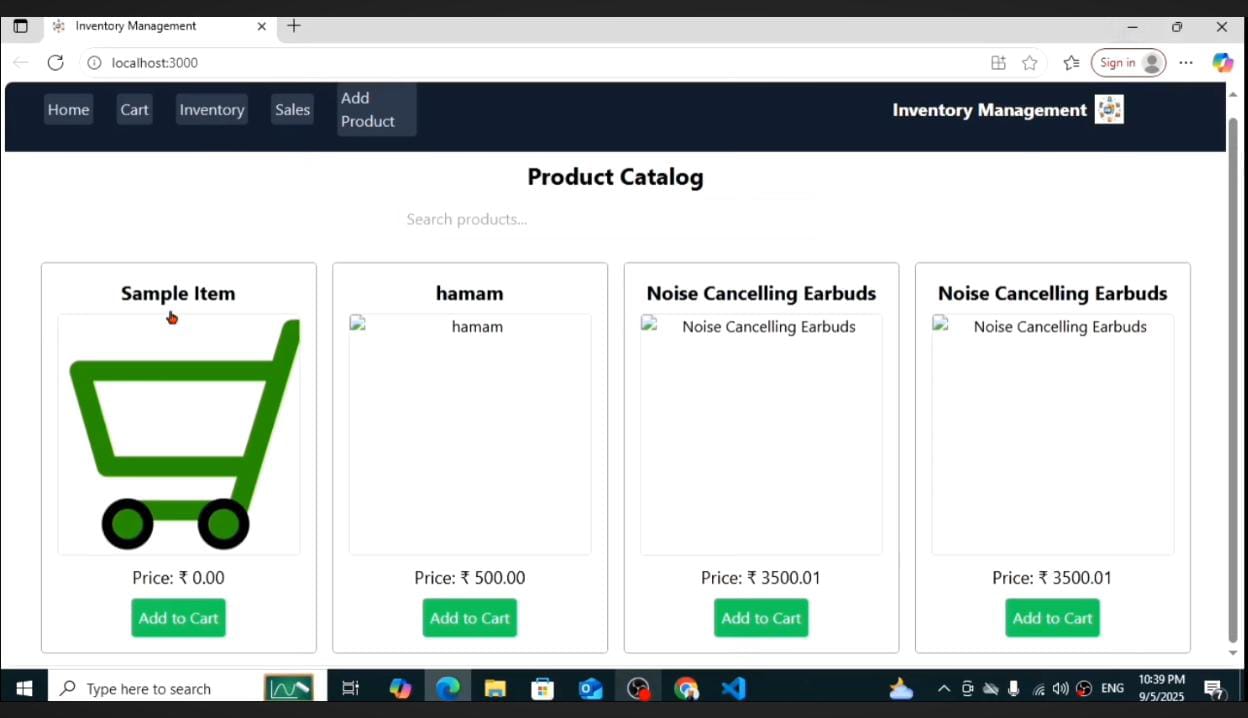
**10. Testing**

– API Testing with Postman

– Unit tests using PyTest & Jasmine (for Angular)

– Browser testing on Chrome, Firefox, Edge

1. **Screenshots or Demo**



**12. Known Issues**

– Some graphs load slowly on low-end devices

– No SMS notifications yet

– Limited multi-language support

**13. Future Enhancement**

– Mobile app for on-the-go store management

– QR code & barcode integration

– AI-driven personalized customer recommendations

– Integration with payment gateways (UPI, PayPal, Stripe)