Project Documentation

**Project Title:Store Manager:Keep Track of Inventory.**

# Introduction

## Project Title: Store Manager:Keep Track of Inventory.

* + **Team ID :NM2025tmid33837**
  + **Team Leader:** Siyam S siyam11718@gmail.com

## Team Members:

* + - BALA DINESH B [baladinesh2225@Gmail.Com](mailto:baladinesh2225@Gmail.Com)
    - JANITH KUMARA S [janithkd902@Gmail.Com](file:///C:\Users\admin\Downloads\janithkd902@Gmail.Com)
    - BALA KUMARAN A [ryderbala1@Gmail.Com](file:///C:\Users\admin\Downloads\ryderbala1@Gmail.Com)

# Project Overview

* + **Purpose:** A "Store Manager: Keep Track of Inventory" project is a system that helps manage and track stock levels, product information, and inventory data to ensure efficient inventory management and minimize stockouts or overstockia.

## Features:

* + - Inventory Tracking
    - Low Stock Alerts
    - Data Analysis
    - Barcode Scanning
    - Supplier Management

# Architecture

* + **Frontend:** React.js with Bootstrap and Material UI
  + **Backend:** Node.js and Express.js managing server logic and API endpoints
  + **Database:** MongoDB stores user data, project information, applications, and chat messages

# Setup Instructions

## Prerequisites:

* + - Node.js
    - MongoDB
    - github
    - React.js
    - Express.js **–** Mongoose **–** Visual Studio Code

## Installation Steps:

# Clone the repository git clone

# Install client dependencies cd client npm install

# Install server dependencies cd

../server npm install

# Folder Structure

SB-Works/

|-- client/ # React frontend

| components/ L pages/

| server/ # Node.js backend

| routes/

| models/

| controllers/

# Running the Application

## Frontend:

cd client npm start

**Backend:**

cd server npm start

* + **Access:** Visit [http://localhost:3000](http://localhost:3000/)

# API Documentation

## User:

* + - /api/user/register
    - /api/user/login

**Projects:**

/api/projects/create

/api/projects:id

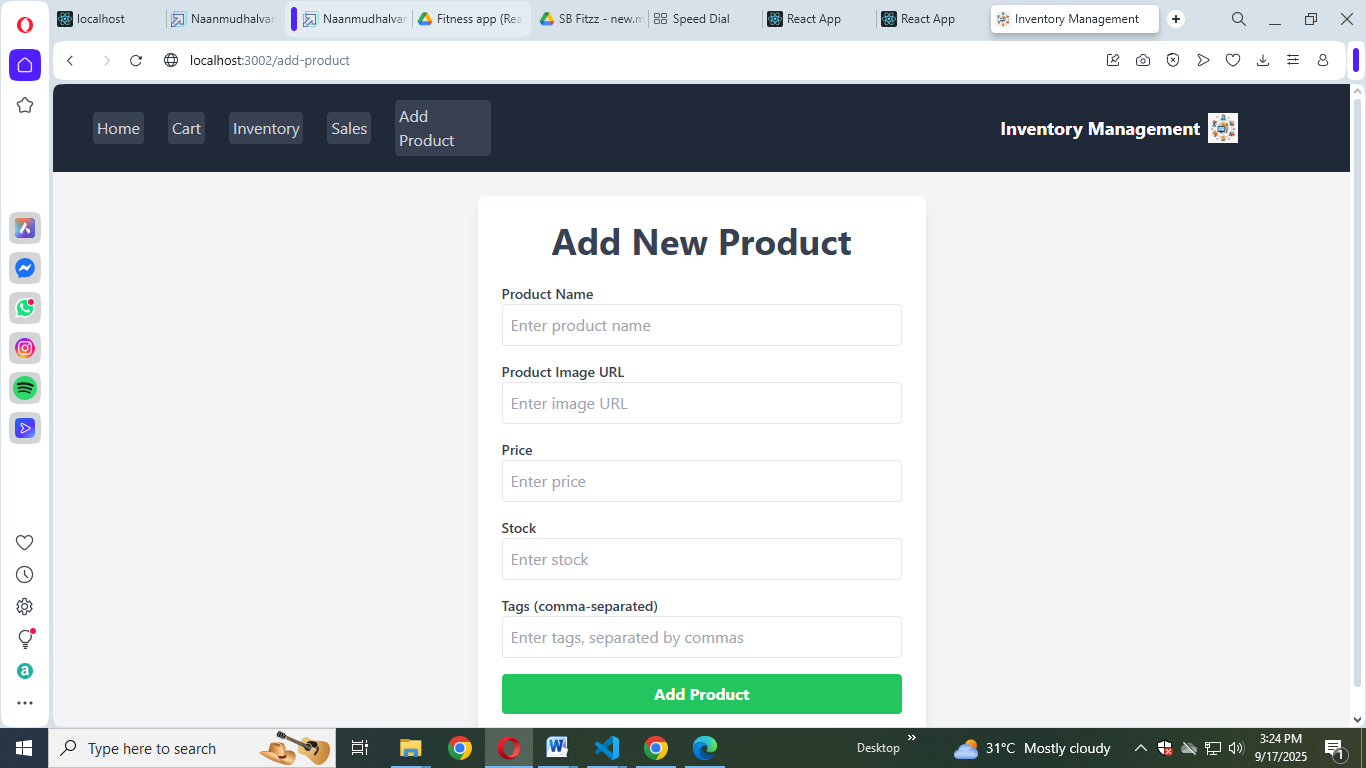
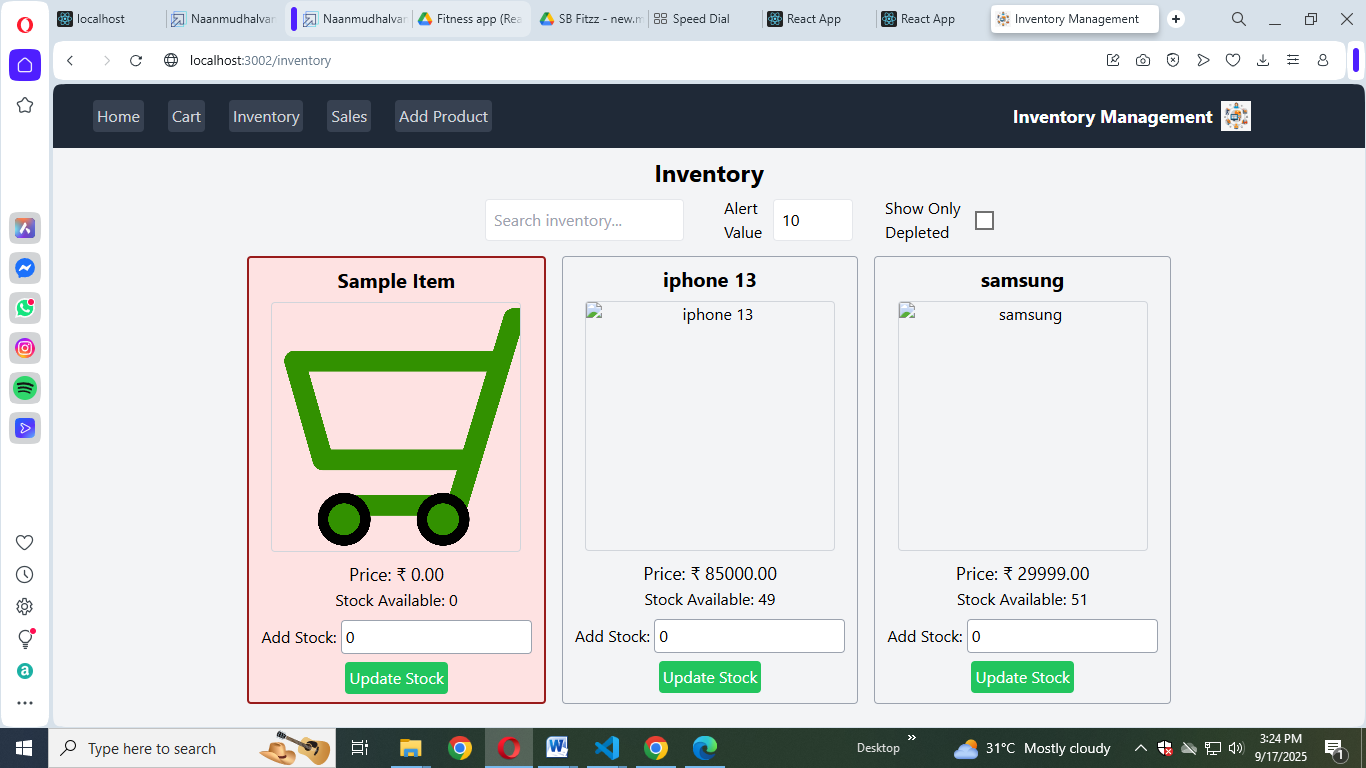
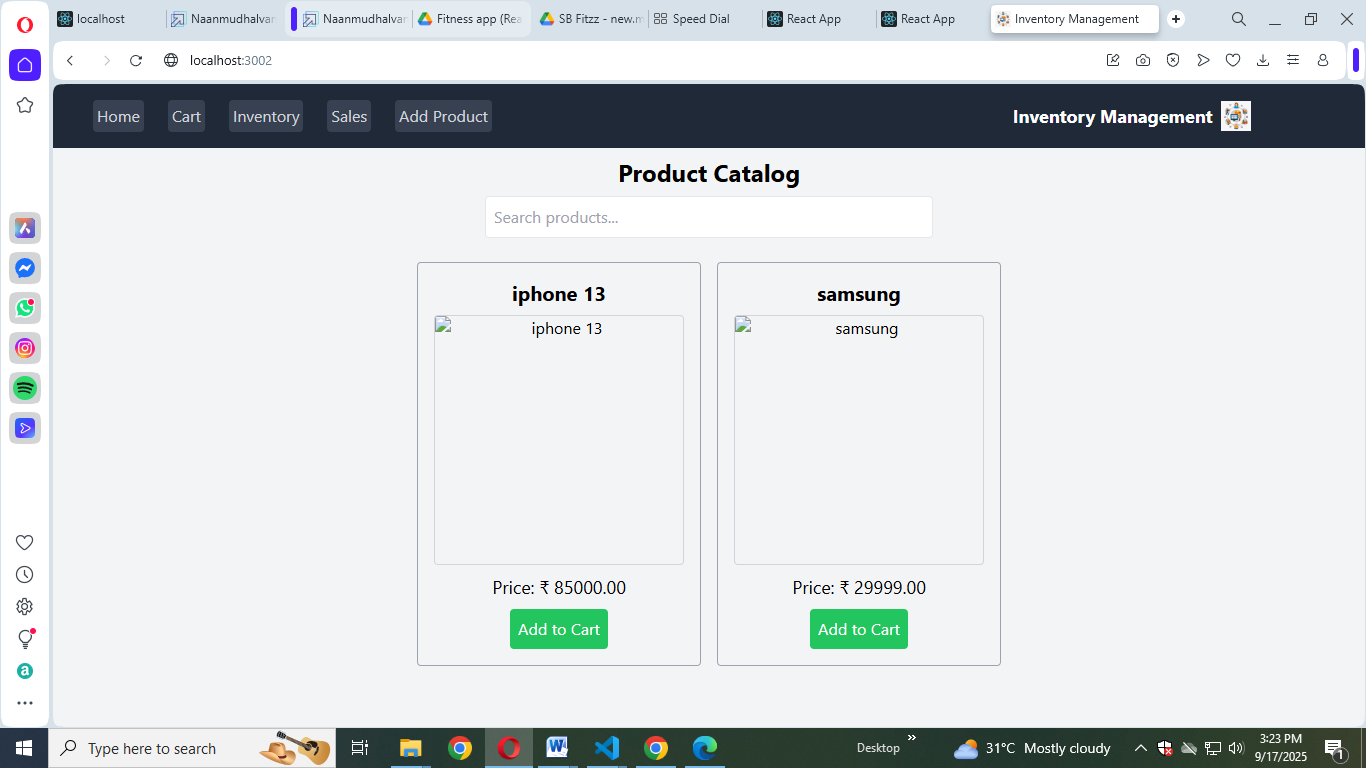
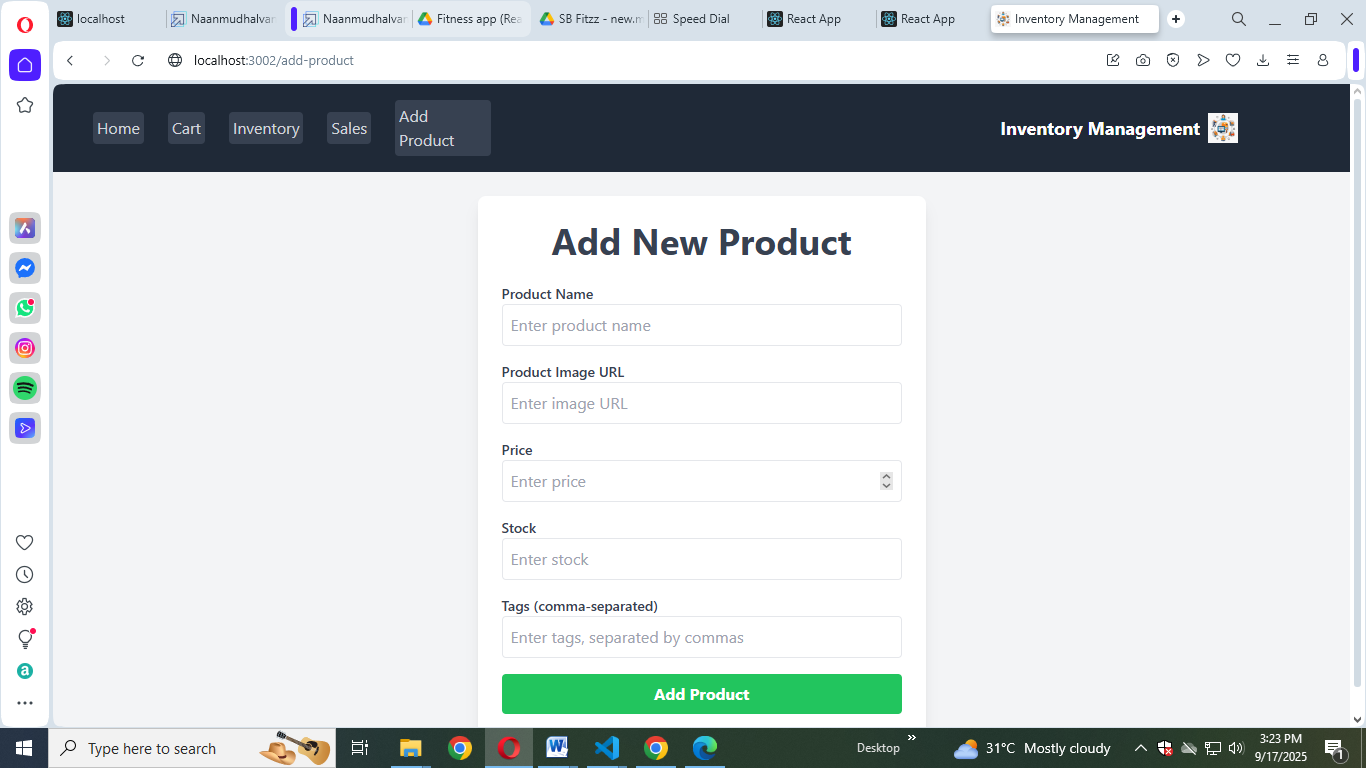
Applications:/api/apply

# Authentication

* + Username/Password Authentication
  + Email/Password Authentication

# User Interface

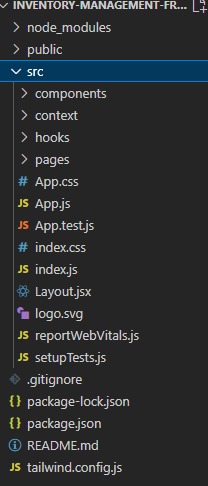
* + Landing Page
  + Inventory Dashboard:
  + Admin Dashboard UI
  + Procurement Dashboard



# Testing

* + Manual testing during milestones
  + Automated Testing Tools
  + Tools: Postman, Chrome Dev Tools

1. **Screenshots or Demo**

****

1. **Known Issues**

1. Data Inconsistency: Inaccurate or inconsistent data due to manual entry errors or system glitches.

2. Inventory Discrepancies: Differences between actual stock levels and recorded inventory levels.

3. Implement Data Validation: Validate user input to prevent data inconsistencies.

4. Conduct Regular Inventory Audits: Regularly verify inventory levels to prevent discrepancies.

1. **Future Enhancements**

1. AI-Powered Forecasting: Use machine learning to predict inventory demand.

2. Automated Reordering: Automatically reorder stock when inventory levels are low.

3. Mobile App: Access and manage inventory on-the-go with a mobile app.

4. Real-Time Notifications: Receive notifications for low stock, stockouts, or other inventory events.

5. Advanced Analytics: Gain deeper insights into inventory trends and performance.

6. Customizable Reports: Create custom reports with relevant metrics and KPIs.

7. Integration with E-commerce: Streamline online sales and inventory management.