Store Manager: Keep Track of Inventory

# Introduction:

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

* + **Team ID: NM2025TMID35238**
  + **Team Leader: D. DHARANEESWARAN & k.d.dharaneeswaran2007@gmail.com**

## Team Members:

* + - C. MANIKANDAN & [manikandan85872790@gmil.com](mailto:manikandan85872790@gmil.com)
    - S. KARTHICK & Karthicksuresh042@gmail.com
    - J. RAJESHWARAN & rajeshwaran2876@gmail.com
    - R. PRAVEEN & preaveen29@gmail. com

# Project Overview

* + - **Inventory Management**: Inventory Management helps maintain healthy stock levels in a store and acquire them in time.
    - **Stock Updates:** Stock will automatically update on sale of products, and it can be updated on adding new stock.
    - **Cart:** Products can be added to cart for a particular sale and quantity can be added to each product.
    - **Checkout at Cart:** Upon checkout, cart is cleared, inventory is updated, and a sale record is made.
    - **Adding New Products to Inventory:** New products can be added to the inventory by providing product name, image URL, price, stock, tags.
    - **Alert View for Depleting Stock:** Depleting stocks are shown in red background, and alert count can be updated as per requirement.
    - **Search Functionality for Products:** Products in inventory and product catalog can be searched.
    - **Sale Records:** All sale records are stored with sale value, products and datetime.
  + **Purpose:**

The purpose of inventory management is to balance stock levels to meet customer demand and operational needs while minimizing costs and optimizing cash flow. Key objectives include preventing overstocking and stockouts, reducing carrying costs, improving order fulfillment, enhancing customer satisfaction, and providing accurate data for better decision-making and business profitability.

* **Key Purposes of Inventory Management**

**Balancing Stock Levels:**

The primary goal is to have the right amount of inventory at the right time to satisfy customers and support operations without tying up excessive capital in stock.

**Cost Reduction:**

By minimizing excess inventory, businesses reduce carrying costs (storage, insurance, potential obsolescence) and associated expenses.

**Customer Satisfaction:**

Ensuring products are available when customers want them prevents stockouts and canceled orders, which improves customer service and loyalty.

**Optimizing Cash Flow:**

Reducing excess inventory frees up cash that would otherwise be tied up in unsold goods, improving the company's financial liquidity.

**Preventing Stockouts:**

Effective management ensures sufficient stock to meet demand, preventing lost sales and frustrated customers.

**Improving Operational Efficiency:**

Proper inventory control streamlines the supply chain by ensuring the availability of raw materials, components, and finished goods, which supports smooth production and fulfillment.

**Enabling Data-Driven Decisions:**

Inventory management systems provide valuable data on inventory levels, sales trends, and demand patterns, enabling businesses to make informed decisions about purchasing, stocking, and promotions.

**Protecting Against Obsolescence:**

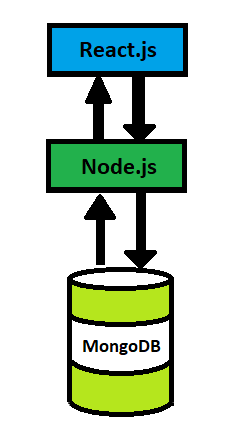
By managing inventory closely, companies can avoid large quantities of products becoming obsolete, which reduces waste and financial loss. how does inventory management improve business profitability.

## Features:

* + - Project posting and bidding
    - Secure chat system
    - Feedback and review system
    - Admin control panel

# Architecture

* + **Frontend:** 
    - Frontend, also known as client-side, is the part of a website or web application that users see and interact with directly, responsible for the user interface (UI).
    - In this project we used React.js with Bootstrap and Material UI as User Interface.
  + **Backend:** 
    - A backend, also known as the server-side, is the part of an application that users don't see or directly interact with but is essential for its operation. It consists of the servers, databases, and the business logic that process data, manage information, and handle processes like authentication and payments, ensuring the frontend functions correctly.
    - In this project we use Node.js and Express.js managing server logic and API endpoints as backend.
  + **Database:** 
    - A database is an organized collection of structured information or data, stored electronically in a computer system to enable efficient storage, retrieval, and management of large amounts of information.
    - In this project we used MongoDB stores user data, project information, applications, and chat messages.



# Setup Instructions

## Prerequisites:

* + - **Node.js**

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the local environment. It provides a scalable and efficient platform for building network applications.

**Install Node.js and npm** on your development machine, as they are required to run JavaScript on the server-side.

* Download: <https://nodejs.org/en/download/>
* Installation instructions: <https://nodejs.org/en/download/package-manager/>

* **React.js**:

React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.

Install React.js, a JavaScript library for building user interfaces.

* Create a new React app:

npx create-react-app my-react-app

Replace my-react-app with your preferred project name.

* Navigate to the project directory:

cd my-react-app

* Running the React App:

With the React app created, you can now start the development server and see your React application in action.

* Start the development server:

npm start

This command launches the development server, and you can access your React app at [http://localhost:3000](about:blank) in your web browser.

* **HTML, CSS, and JavaScript**: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.

* **Version Control**: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

• **GitHub:** Download and installation instructions can be found at:

https://git- scm.com/downloads 

* **Development Environment**: Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.
* Visual Studio Code: Download from <https://code.visualstudio.com/download>
* Sublime Text: Download from <https://www.sublimetext.com/download>
* WebStorm: Download from [https://www.jetbrains.com/webstorm/download](https://www.jetbrains.com/webstorm/download%20)

To get the Application project from drive:

Follow below steps:

**Install Dependencies:**

• Navigate into the cloned repository directory and install libraries:

cd store

npm install

* **Start the Development Server**:

• To start the development server, execute the following command:

npm start

**Access the App:**

• Open your web browser and navigate to [http://localhost:3000](http://localhost:3000/).

• You should see the application's homepage, indicating that the installation and setup were successful.

 You have successfully installed and set up the application on your local machine. You can now proceed with further customization, development, and testing as needed.

* + - MongoDB
    - Git
    - 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
  + **Chats:**

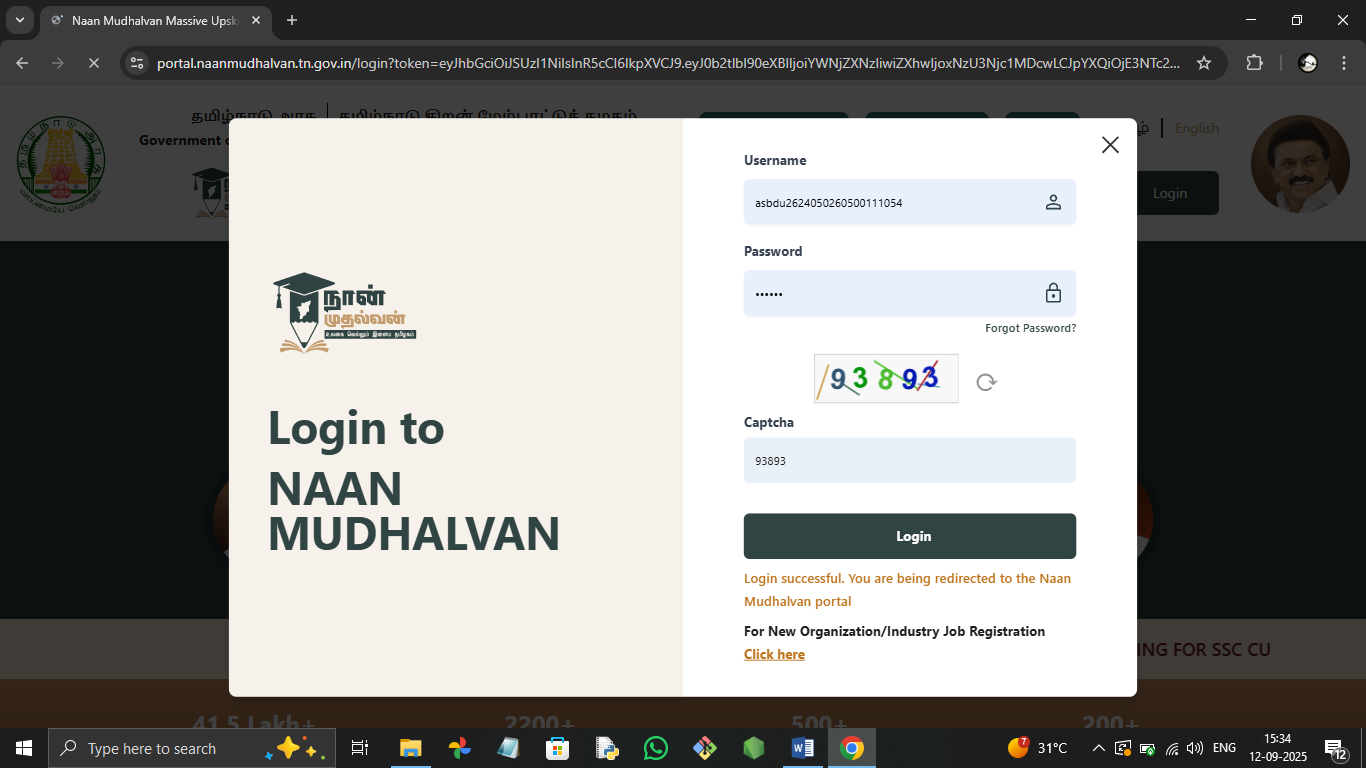
## Project

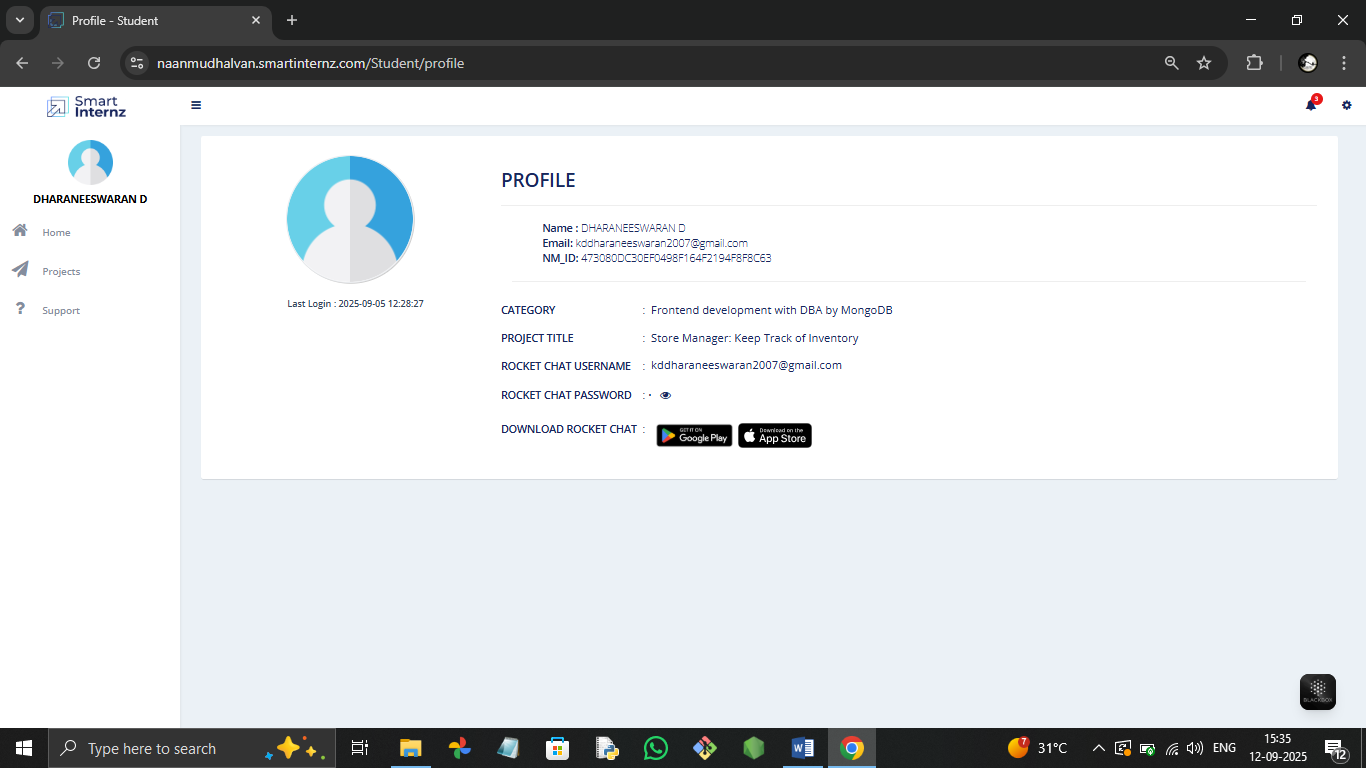
* /api/projects/create
* /api/projects/:id • **Applications:** /api/apply
* /api/chat/send
* /api/chat/:userId

# Authentication

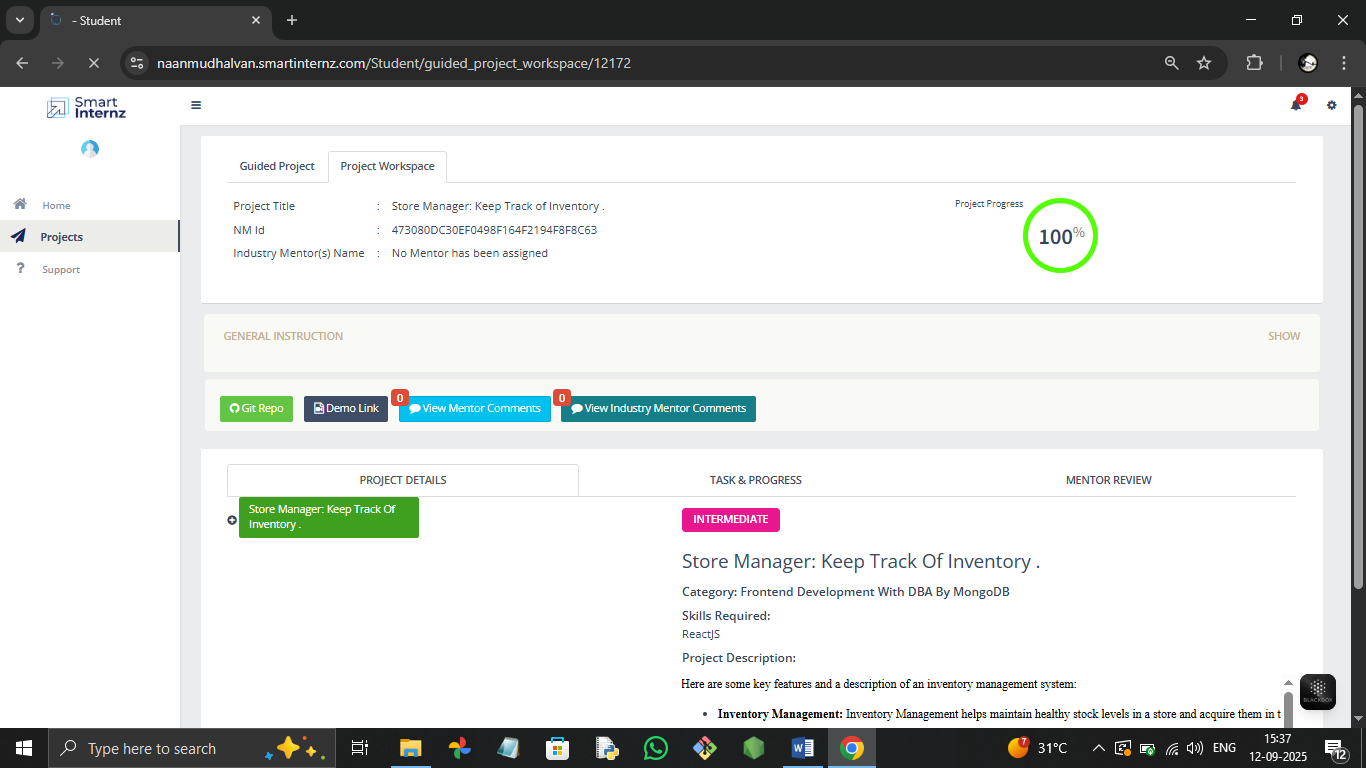
* + JWT-based authentication for secure login
  + Middleware protects private routes

# User Interface

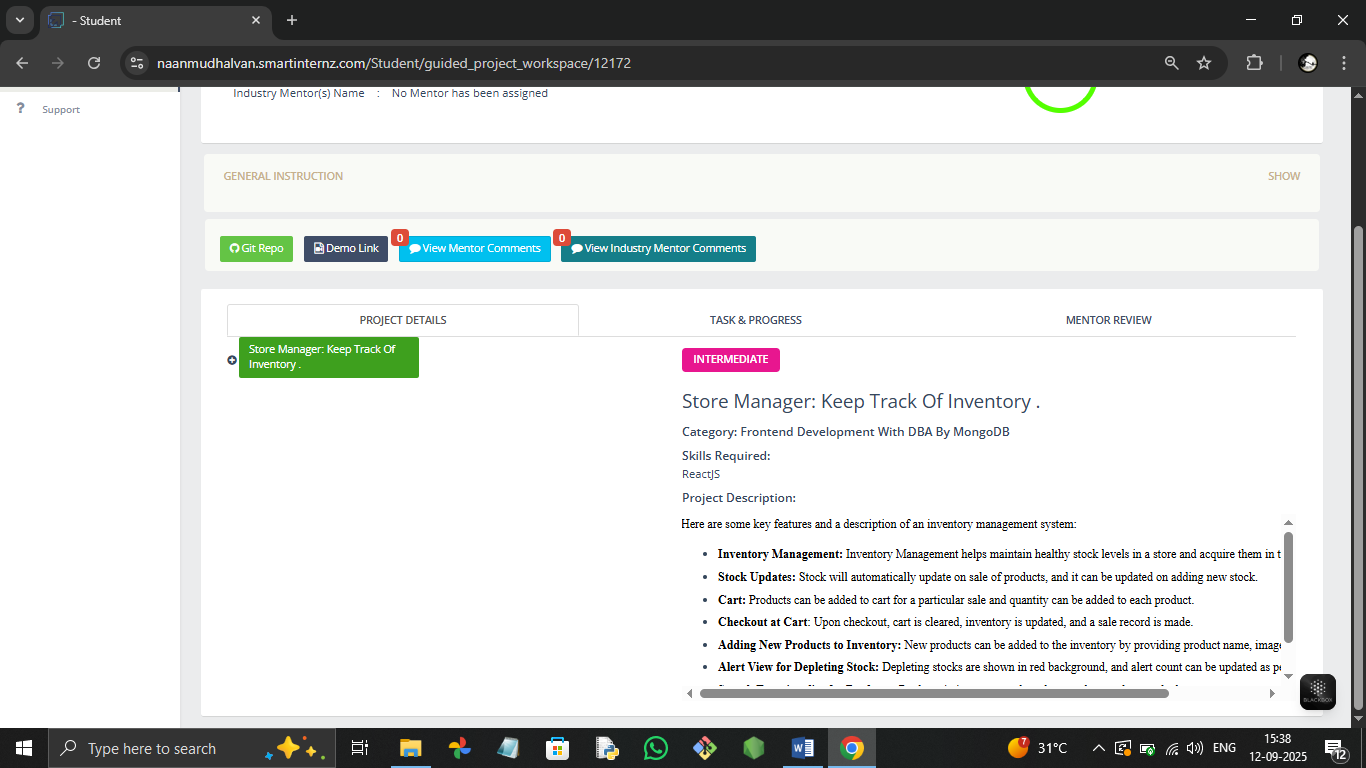
* + Landing Page
  + 
  + Freelancer Dashboard



* + Admin Panel



* Project Details Page



# Testing

* + Manual testing during milestones

Manual testing is the process of a human tester checking a software application's quality by executing test cases without using automated tools or scripts. The tester interacts with the software as an end-user to identify defects, bugs, and other issues that could negatively impact the user experience. This primitive testing technique is crucial for finding critical bugs and ensuring an application's design, functionality, and performance meet requirements.

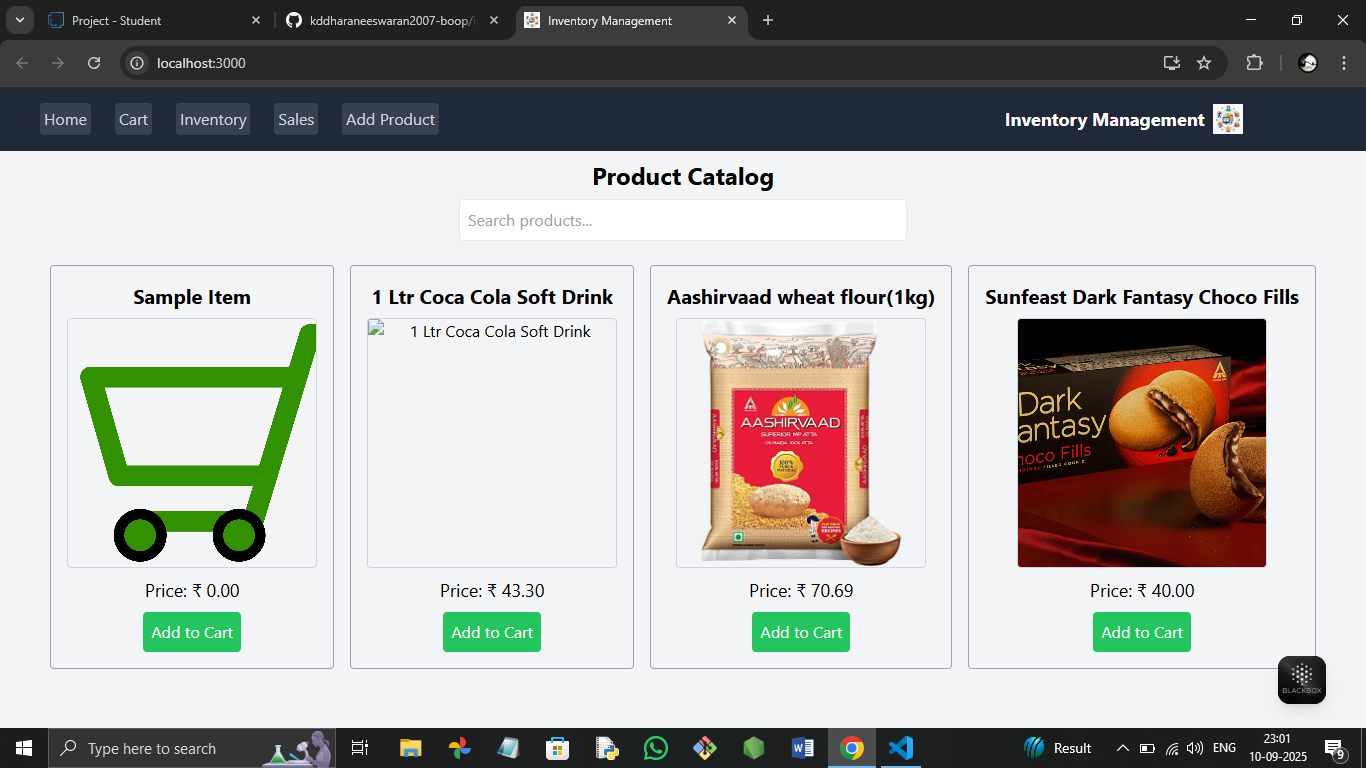
Key aspects of manual testing

* Human interaction:

A human tester physically performs the steps outlined in test cases, simulating real user behavior.

* No automation tools:
  + - Test execution relies on the tester's actions, rather than pre-programmed scripts or automated tools.
  + Focus on user experience:
    - Manual testers check the software's look, feel, and usability, ensuring a smooth and intuitive user journey.
  + Defect identification:
    - The primary goal is to discover bugs, errors, and issues that can cause friction for end-users.
    - Tools: Postman, Chrome Dev Tools

1. **Screenshots or Demo**



1. **Known Issues:**

Npm installalition process is little bit hard

1. **Future Enhancements**

In future we maintain the stock details for hourly based for fast customer access