Project Documentation

1. Introduction

- **Project Title: CookBook: Your Virtual Kitchen Assistant**
- **Team ID**
- **Team Leader:** ASVIDHA SHIRINE M I& 24bca03acas@gmail.com
- **Team Members:**
  - ABI B & 24bca01acas@gmail.com
  - FARZANA T & 24bca08acas@gmail.com
  - KUMUDHA B & 24bca25acas@gmail.com

2. Project Overview

- **Purpose:** SB Works connects clients and freelancers through project postings, bidding, and real-time communication.
- **Features:**
  - Project posting and bidding
  - Secure chat system
  - Feedback and review system
  - Admin control panel

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 user data, project information, applications, and chat messages

4. Setup Instructions

- **Prerequisites:**
  - Node.js

- 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
```

5. Folder Structure

```
SB-Works/
|-- client/                # React frontend
      |__components/
       L__ 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

- **Projects:**

- /api/projects/create
- /api/projects/:id • **Applications:** /api/apply

- **Chats:**
  - /api/chat/send – /api/chat/:userId

8. Authentication

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

9. User Interface

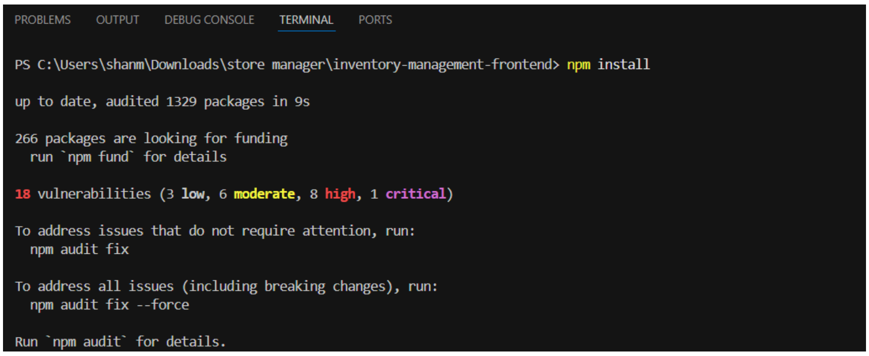
- Landing Page
- Freelancer Dashboard
- Admin Panel
- Project Details Page

10. Testing

- Manual testing during milestones
- Tools: Postman, Chrome Dev Tools

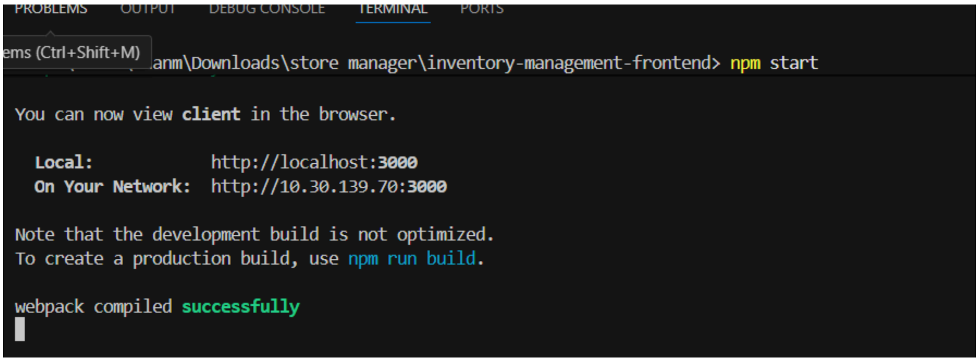
11. Screenshots or Demo

Step 1: Type In visual studio code-> Terminal-> new Terminal->Type npm install and press enter.



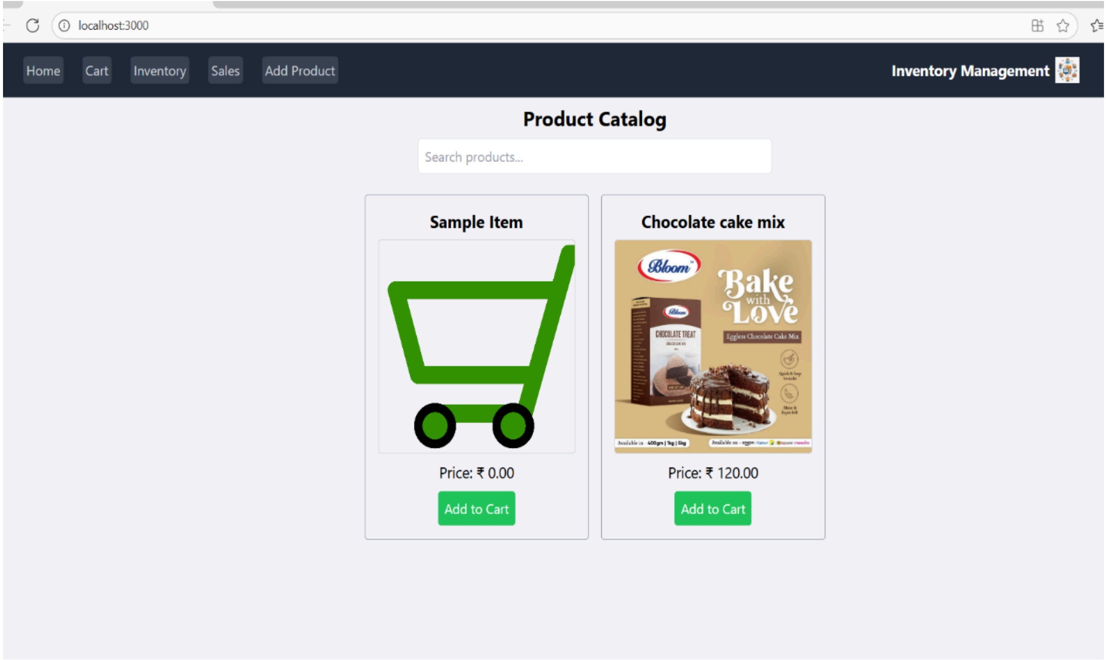
```
PS C:\Users\shanm\Downloads\store manager\inventory-management-frontend> npm install

up to date, audited 1329 packages in 9s

266 packages are looking for funding
  run `npm fund` for details

18 vulnerabilities (3 low, 6 moderate, 8 high, 1 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

Step 2: Type npm start and press enter

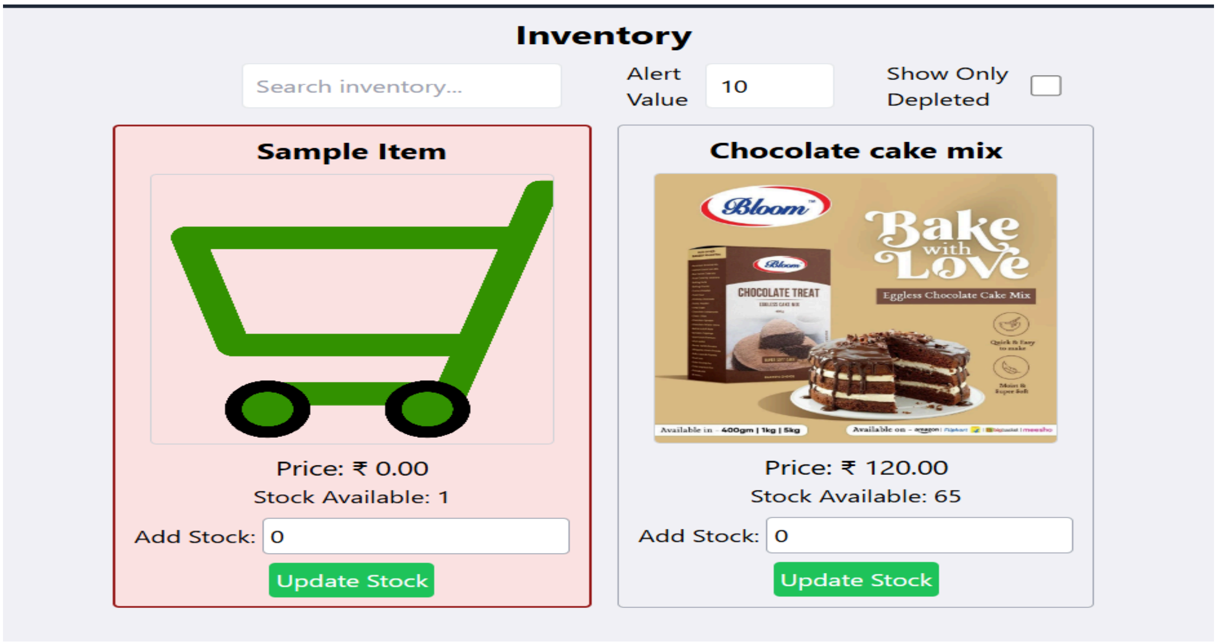


```
anm\Downloads\store manager\inventory-management-frontend> npm start

You can now view client in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://10.30.139.70:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Step 3: Project will run in localhost:3000



Step 4: Go to inventory for adding the product details and stock



Inventory

Search inventory... Alert Value 10 Show Only Depleted □

Sample Item

Price: ₹ 0.00

Stock Available: 1

Add Stock: 0

Update Stock

Chocolate cake mix

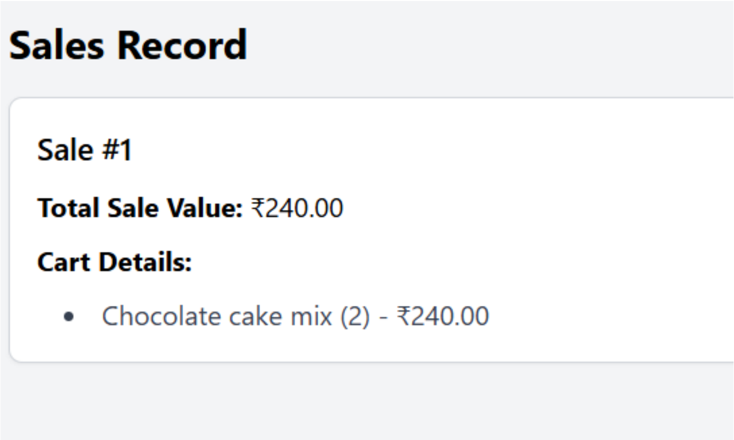
Price: ₹ 120.00

Stock Available: 65

Add Stock: 0

Update Stock

Step 5 : Go to sales to check the sales Record and go to



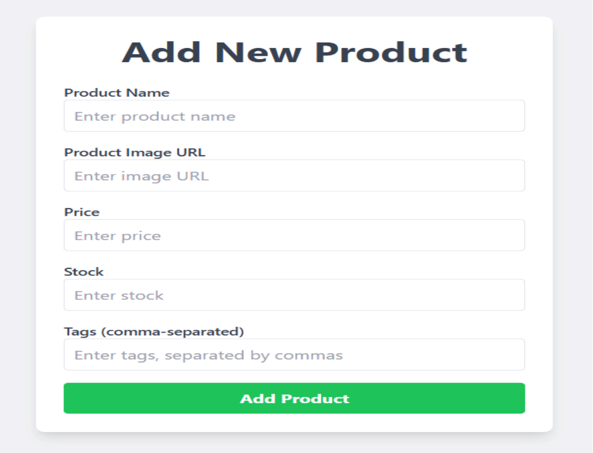
Sales Record

Sale #1

Total Sale Value: ₹240.00

Cart Details:

- Chocolate cake mix (2) - ₹240.00



Add New Product

Product Name: Enter product name

Product Image URL: Enter image URL

Price: Enter price

Stock: Enter stock

Tags (comma-separated): Enter tags, separated by commas

Add Product

12. Known Issues : Nill

13. Future Enhancements

- Barcode Scanner Integration
- Automatic Low Stock Alerts
- Supplier Management
- Multi user access
- Mobile App version