**Project Documentations**

**Store Manager:Keep Track of Inventory**

**1. Introduction**

**• Store Manager:Keep Track of Inventory**

**• Team ID :** NM2025TMID46911

**• Team Leader**:  Madhumitha K &madhum8329@gmail.com

**• Team Members:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Name** | **Email ID** |
| **1.** | PAVITHRA K | Sridevipavi603@gmail.com |
| **2.** | PAVITHRA M | Pavithra809842@gmail.com |
| **3.** | POOJASRI R | sripooja3960@gmail.com |
| **4.** | PREMAVATHI M | Kavithapremavathi@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

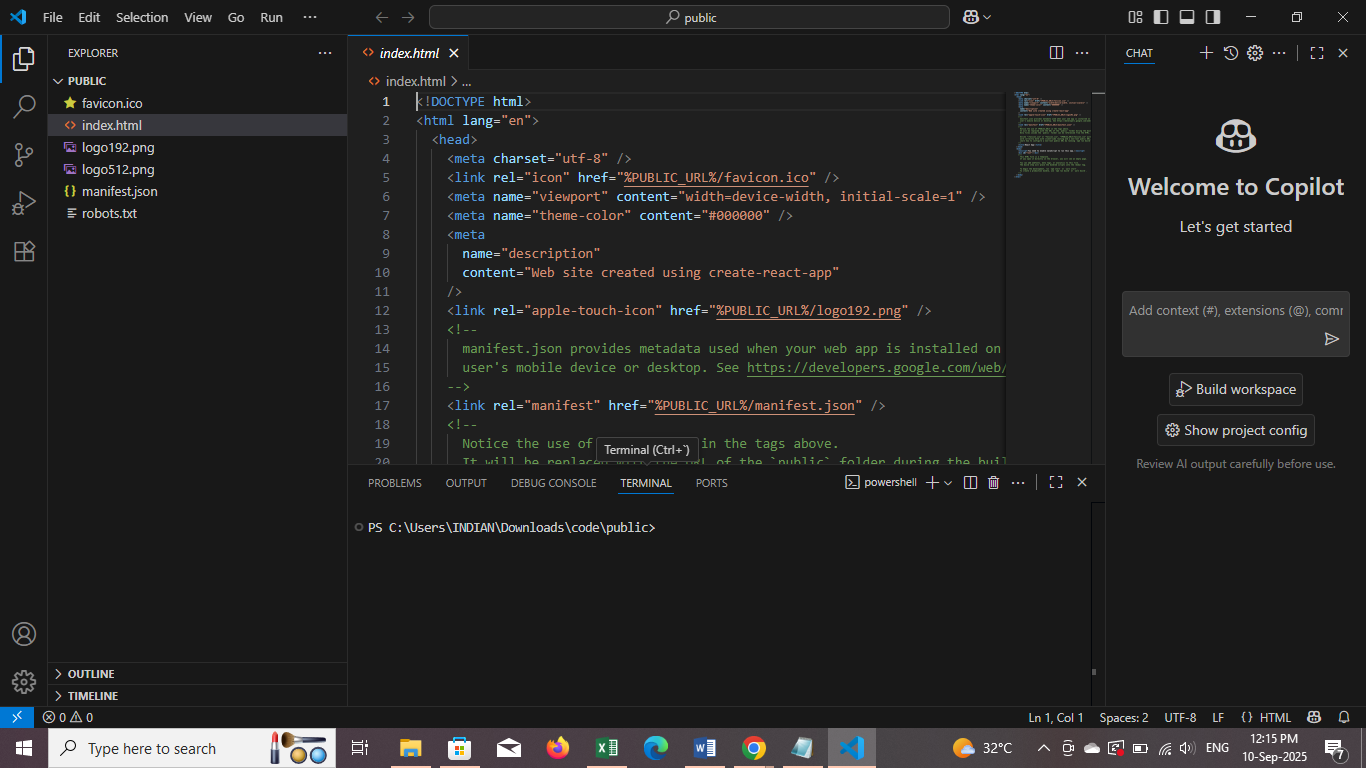
• Project Details Page

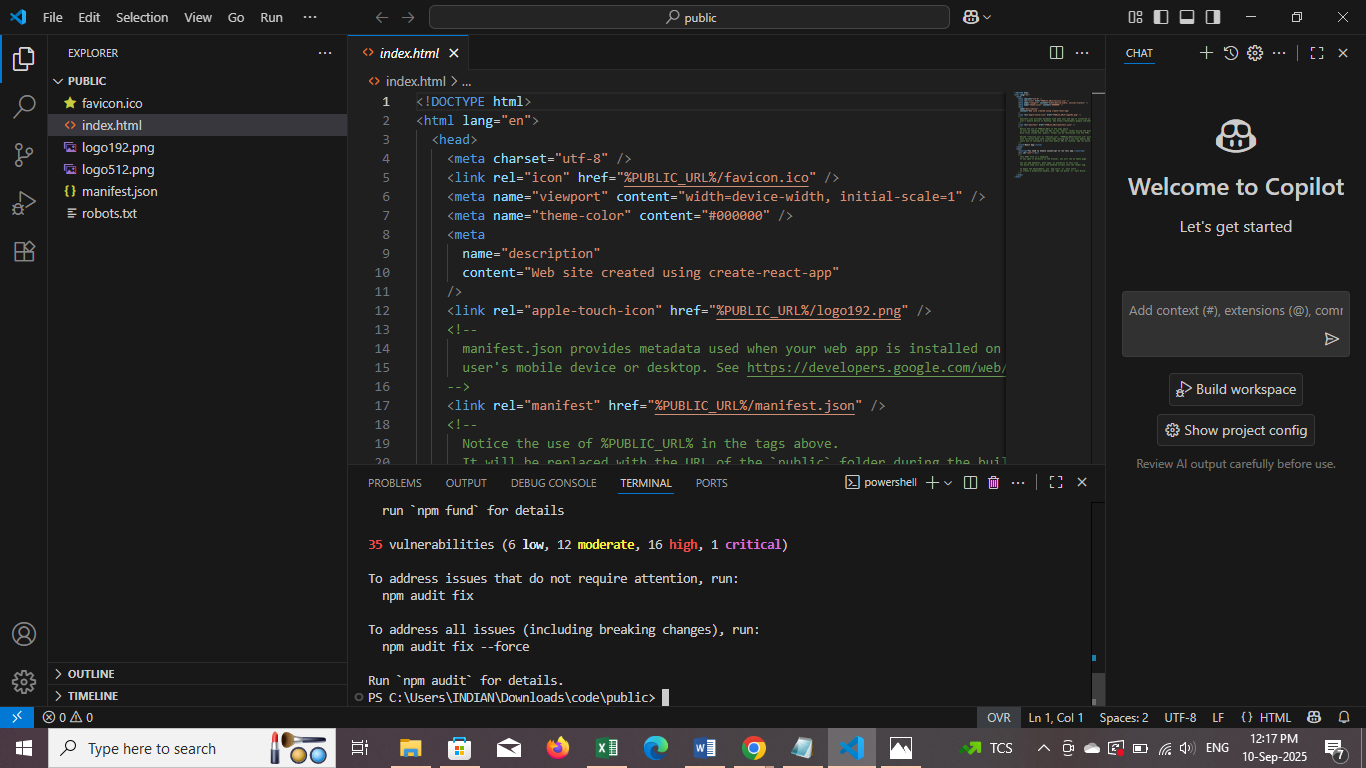
**10. Testing**

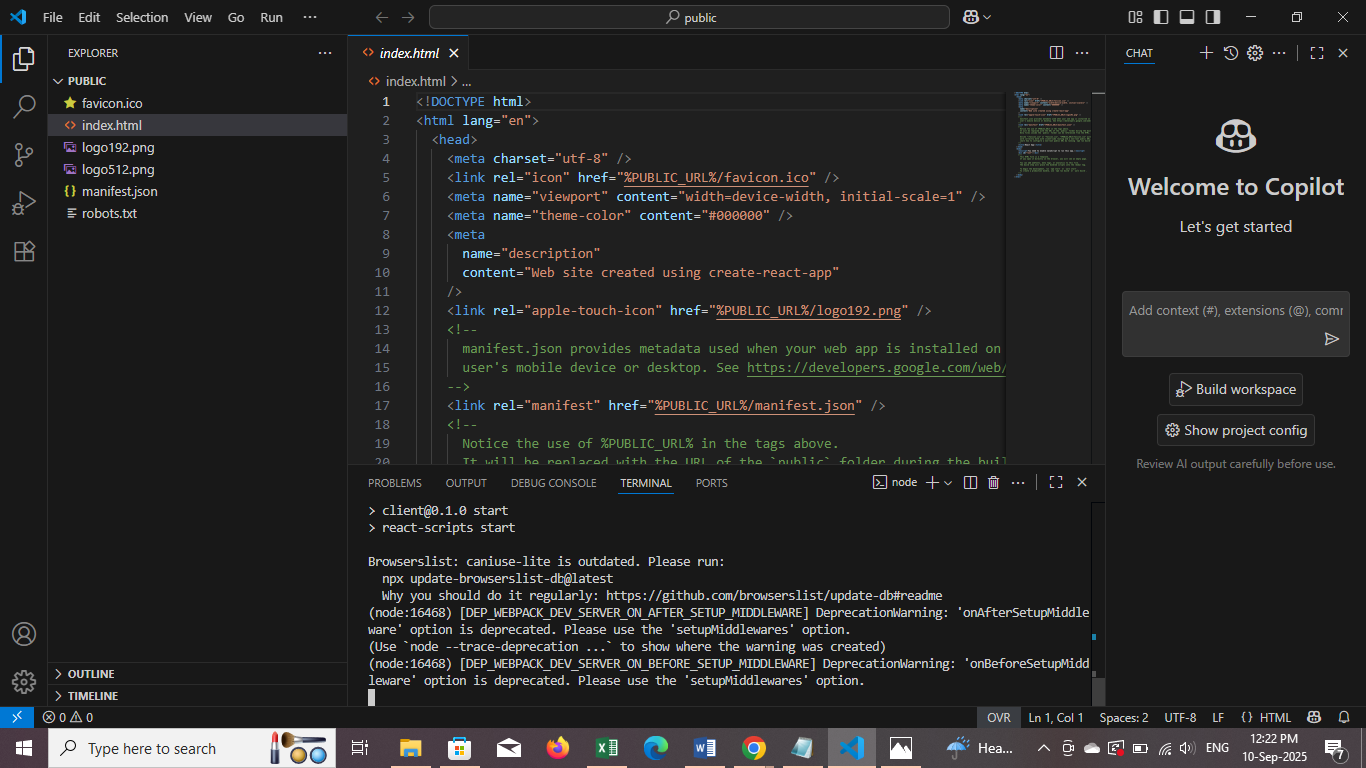
• Manual testing during milestones

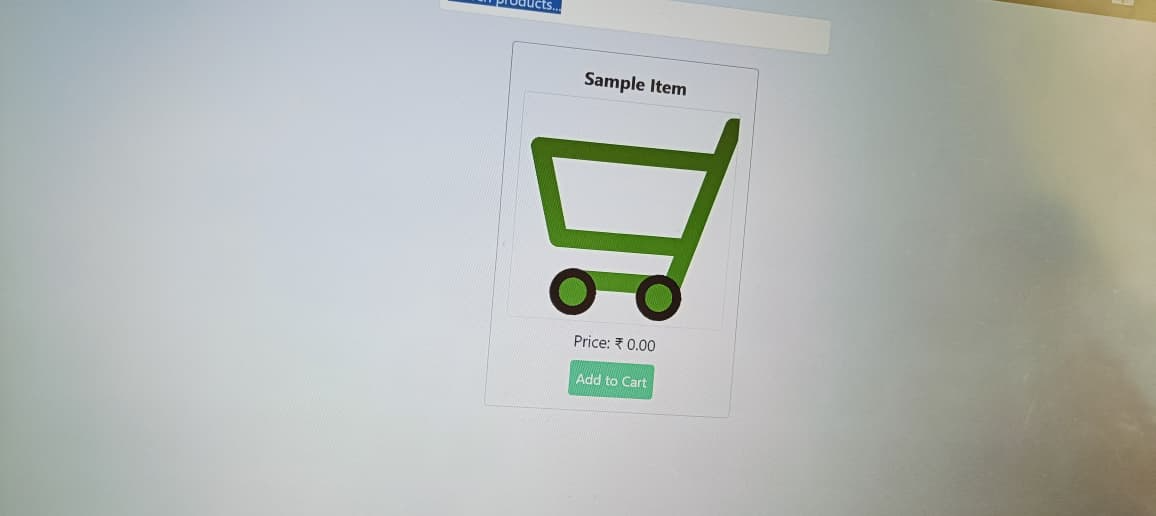
• **Tools**: Postman, Chrome Dev

**11. Screenshots or Demo**

****





ss

**12. Known Issues**

1. Inaccurate Inventory Counts

2.Stockouts and Overstocking

3.Inventory Shrinkage

4.Poor Inventory Visibility

**13. Future Enhancements**

**Features**

1. 1. Real-time Inventory Tracking: Monitor stock levels in real-time to prevent stockouts and overstocking.

2. Automated Inventory Updates: Use barcode scanning or RFID technology to automate inventory updates and reduce manual errors.

3. Inventory Forecasting: Analyze sales trends and historical data to predict inventory needs and optimize stock levels.

4. Low-Stock Alerts: Set up notifications for low stock levels to ensure timely replenishment.

5. Inventory Reporting: Generate detailed reports on inventory levels, stock movements, and product performance.

**Technical Enhancements**

1. Inventory Management Software: Implement dedicated systems that offer insights and reporting functions to manage stock levels effectively, providing real-time updates and automation features.

2. Barcode Scanning and RFID Technology: Utilize barcode scanning to automate updates and reduce manual entry errors. RFID technology allows for bulk scanning, real-time tracking, and improved inventory accuracy.

3. Automated Replenishment Software: Analyze sales trends and inventory levels to set ideal reorder points, ensuring timely product availability and minimizing stockouts.

4.Demand Forecasting Tools: Use historical data to predict inventory needs, reducing excess inventory costs and improving order fulfillment rates.

5.Real-Time Inventory Tracking Systems: Provide up-to-the-minute stock insights, preventing overselling and stockouts.