

# Project Documentations

CookBook: Your Virtual Kitchen Assistant

## 1. Introduction

- CookBook: Your Virtual Kitchen Assistant
- Team ID : NM2025TMID46903
- Team Leader: M. Oviya & [oviya9385@gmail.com](mailto:oviya9385@gmail.com)
- Team Members:

S.No	Name	Email ID
1.	V. Mageshwari	<a href="mailto:magimageshwari345@gmail.com">magimageshwari345@gmail.com</a>
2.	S. Malini	<a href="mailto:malinisaravanan2007@gmail.com">malinisaravanan2007@gmail.com</a>
3.	S. Monisha	<a href="mailto:monishakumar4377@gmail.com">monishakumar4377@gmail.com</a>
4.	V. Monisha	<a href="mailto:monishaverrrasamy2006@gmail.com">monishaverrrasamy2006@gmail.com</a>

## 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/
    |-- 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

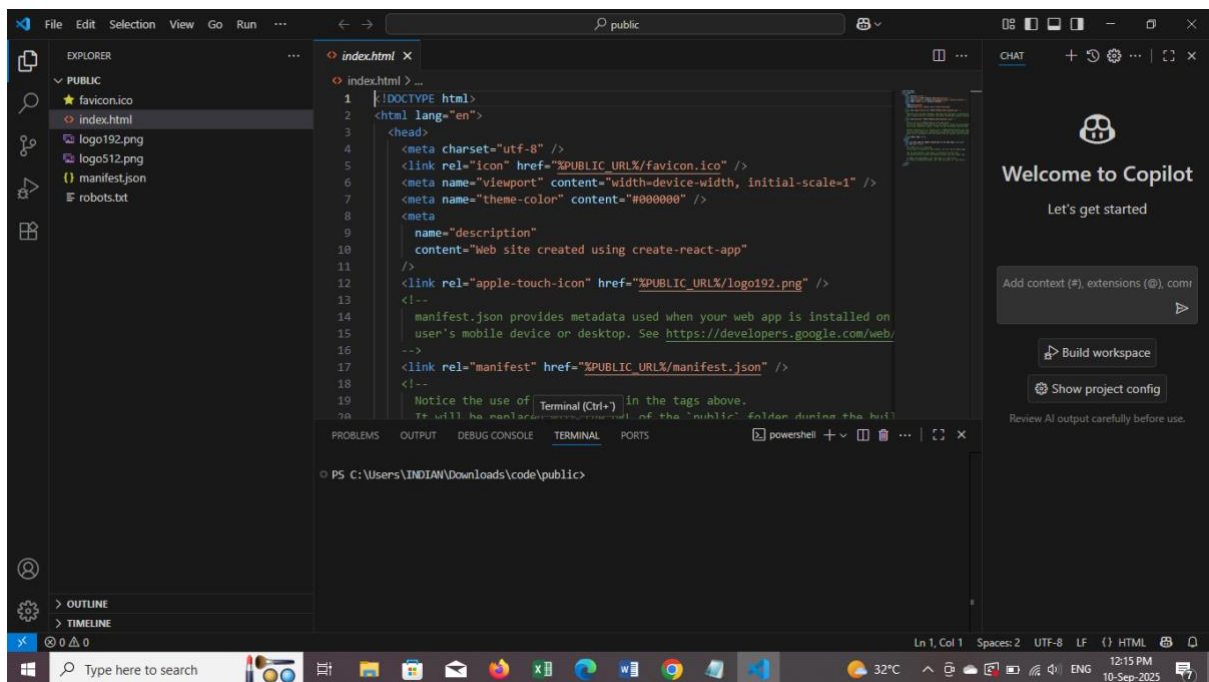
- Landing Page
- Freelancer Dashboard
- Admin Panel

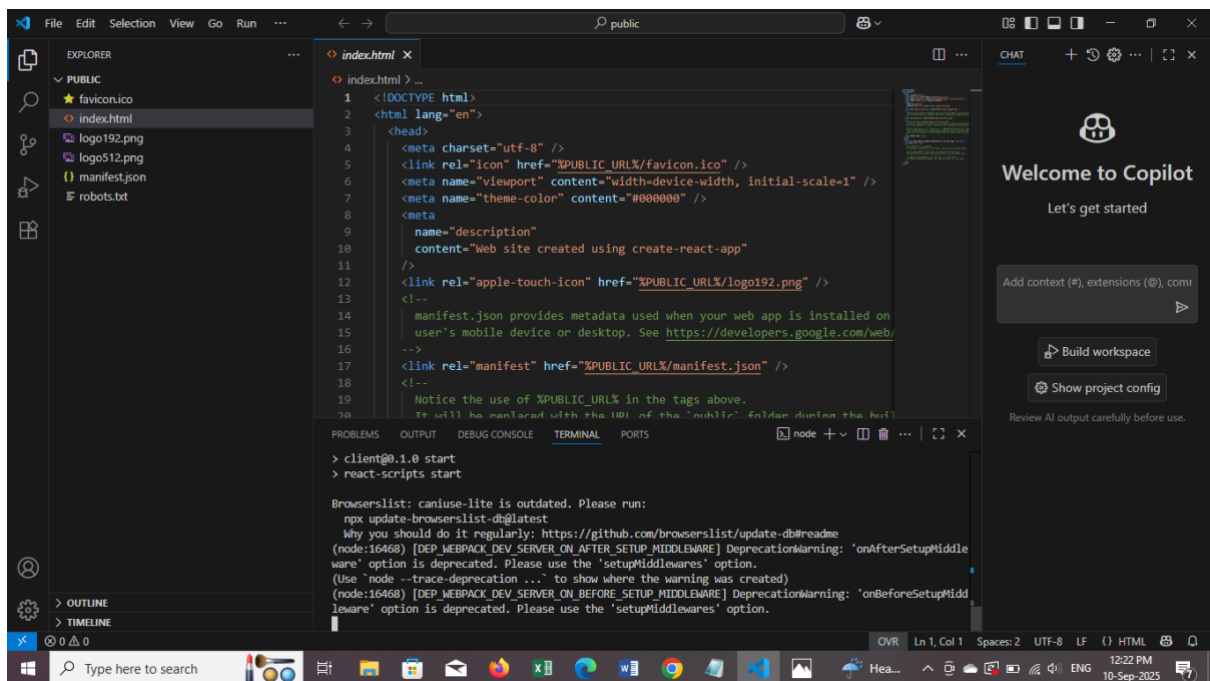
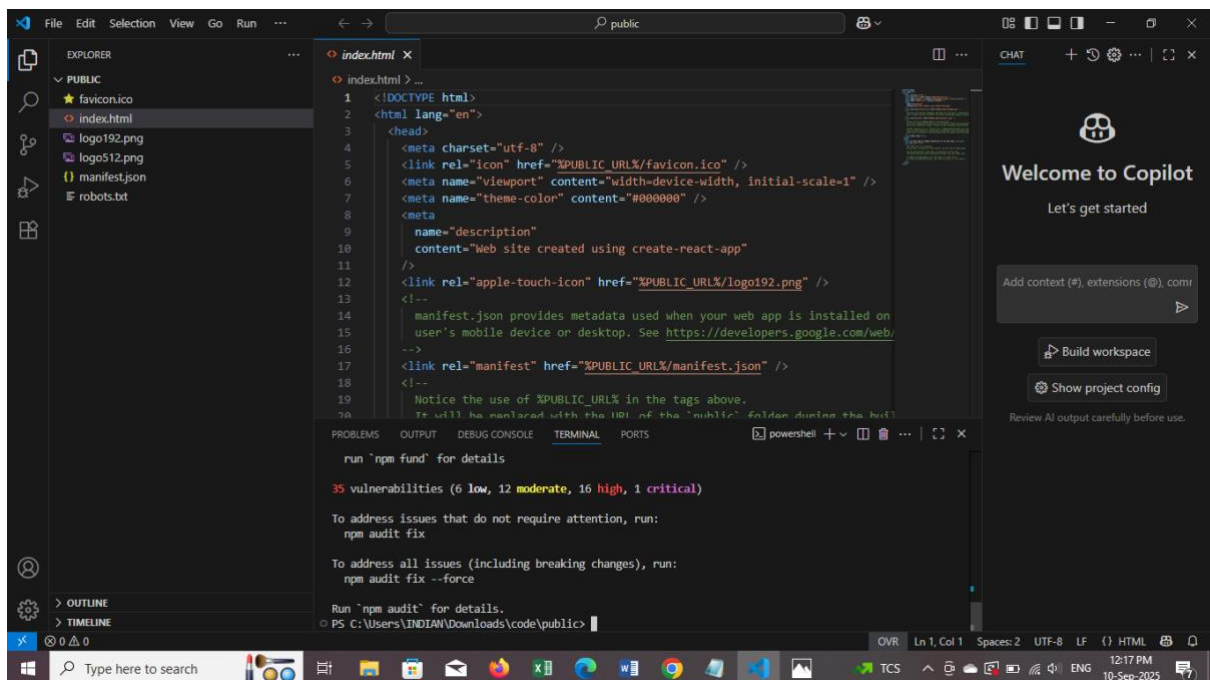
- Project Details Page

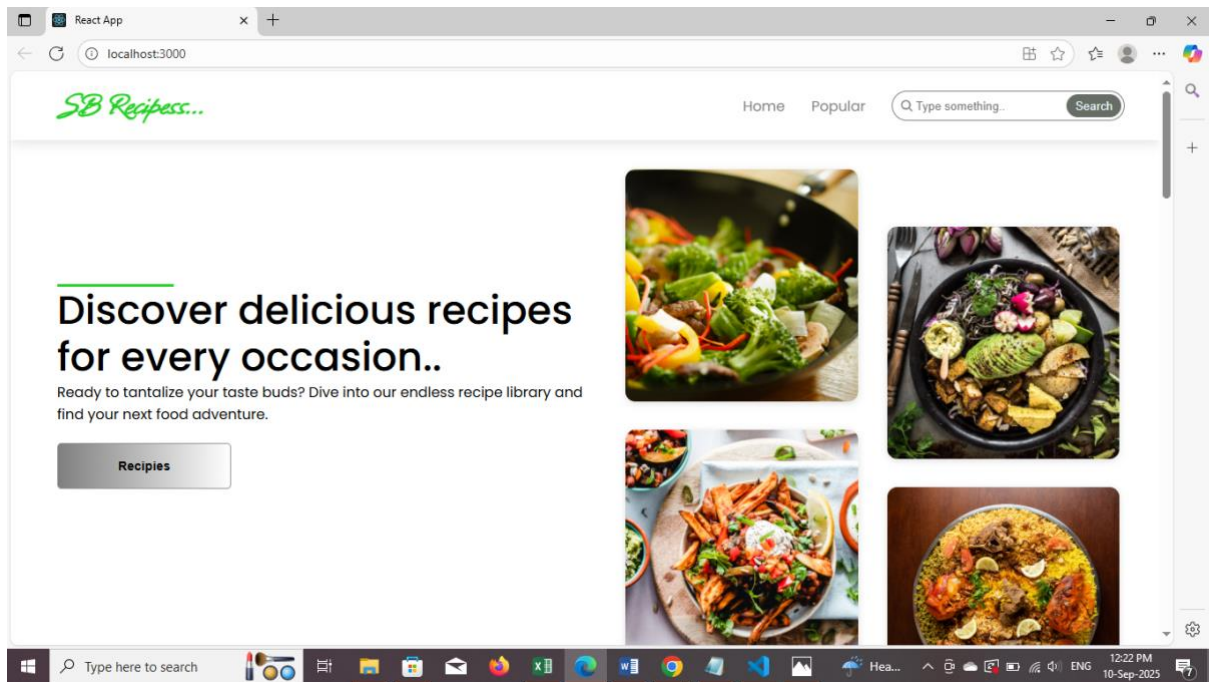
## 10. Testing

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

## 11. Screenshots or Demo







## 12. Known Issues

1. Meal planning and organization
2. Finding recipes based on dietary restrictions or preferences
3. Generating grocery lists
4. Step-by-step cooking instructions
5. Reducing food waste
6. Discovering new recipes

## 13. Future Enhancements

### Features

1. Personalized recipe suggestions: Based on dietary preferences, cooking history, and nutritional needs.
2. Smart meal planning: Automatically generates meal plans based on seasonal ingredients, sales, and dietary needs.
3. Voice assistant integration: Allows users to interact with the app hands-free while cooking.
4. Recipe customization: Enables users to modify recipes based on ingredient availability, dietary needs, and personal preferences.

5. Social sharing: Allows users to share favorite recipes and cooking experiences with friends and family.

### **Technical Enhancements**

1. Image recognition: Enables the app to recognize ingredients, dishes, and cooking techniques.
2. Machine learning: Improves recipe suggestions and meal planning based on user behaviour and preferences.
3. Integration with smart kitchen devices: Enables seamless control of kitchen appliances and gadgets.