

Project Documentation

Cook book Your Virtual Kitchen Assistant

1. Introduction

- ***Project Title: Cook Book***
- ***Team ID: NM2025TMID30426***
- ***Team Leader: R. Megha shree -Mail ID: meghashree742007@gmail.com***
- ***Team Members & Mail ID:***
 - _C.Manoranjitham & manoranjitham.mkd@gmail.com***
 - P.Nisha & nishasri974@gmail.com***
 - M.Manjula & manju26102610@gmail.com***
 - P.Sivabharathy & sivabharathi706@gmail.com***

2. Project Overview

- ***Purpose: SB Works connects clients and freelancers through project postings, bidding, and real-time communication***
- ***Features:***
 - Project posting and binding***
 - 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.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 fronted

|_ 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***

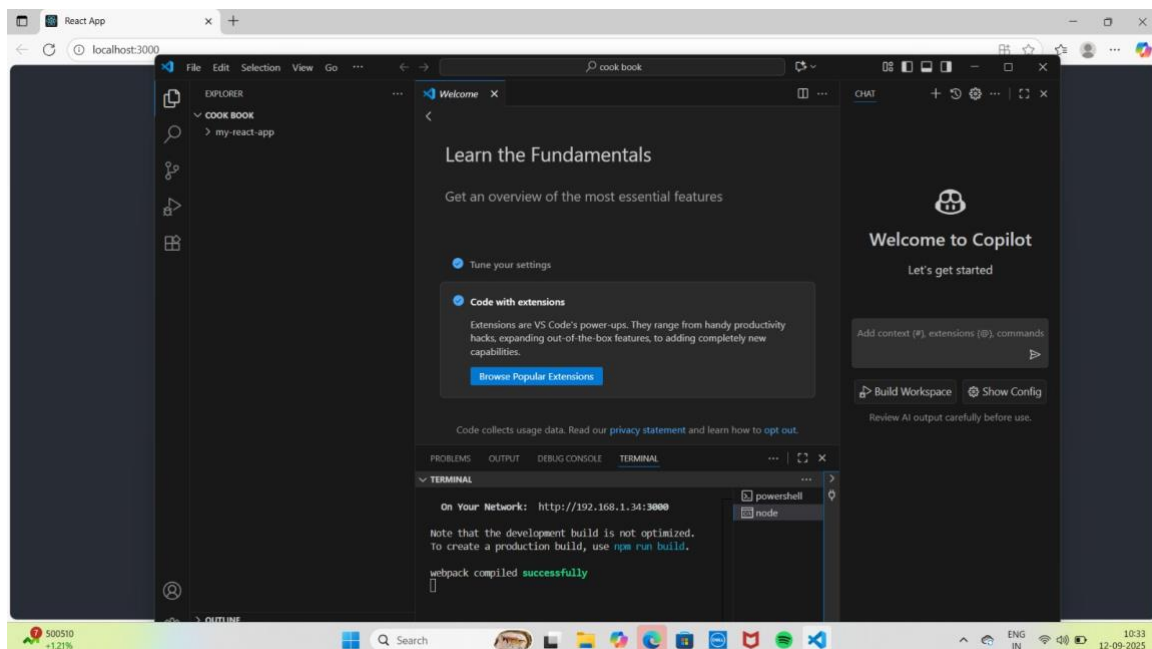
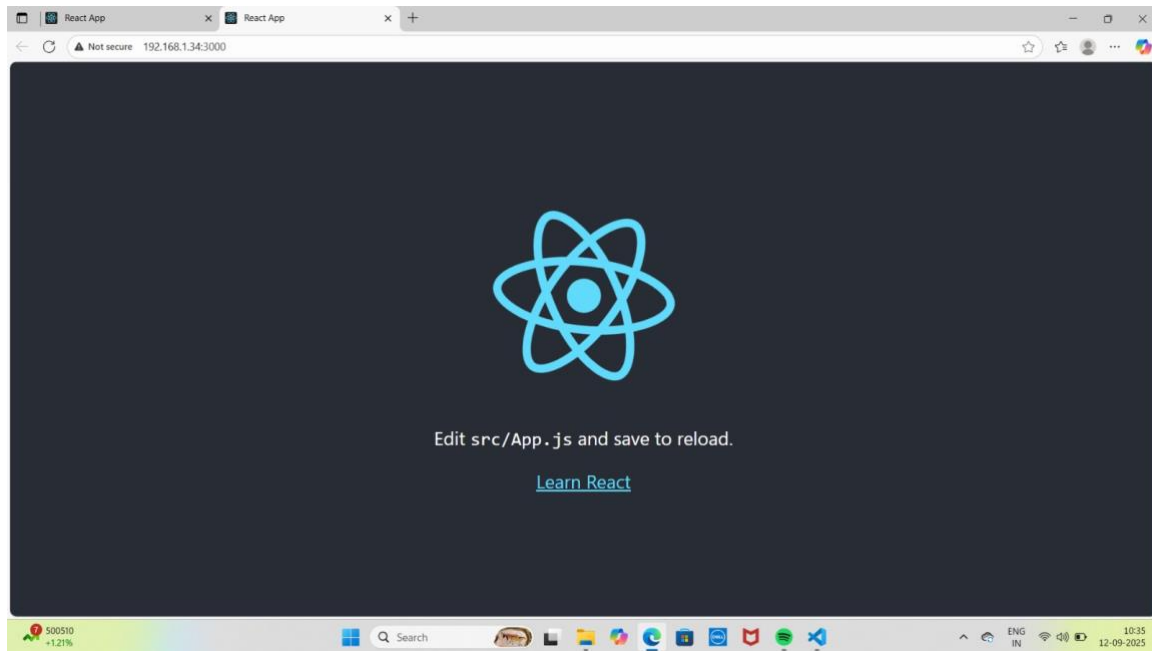
- ***Admin Panel***

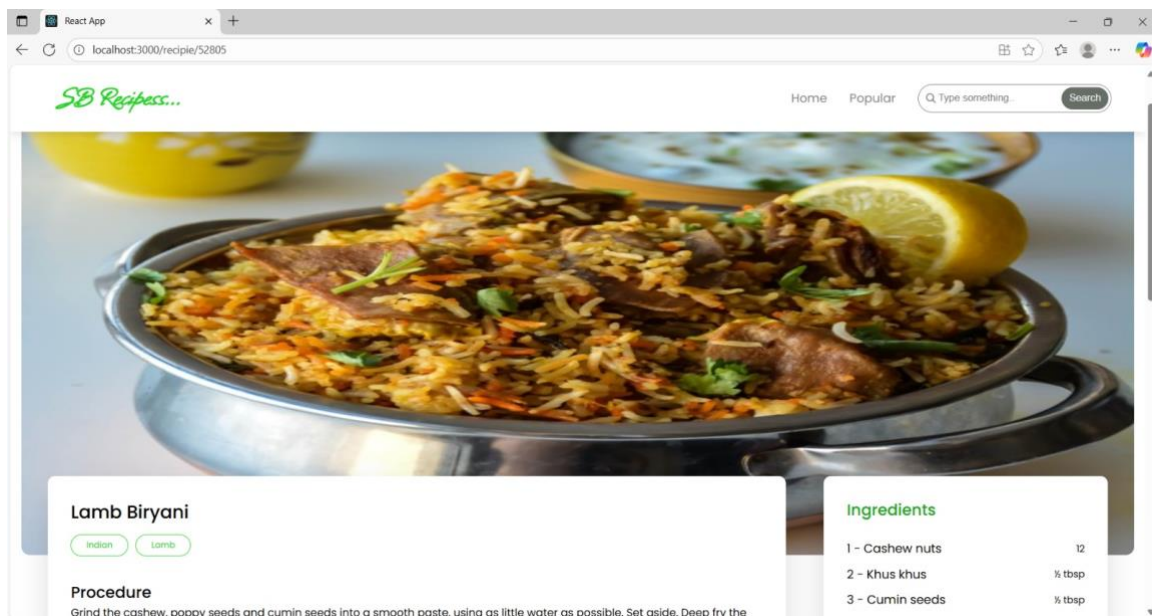
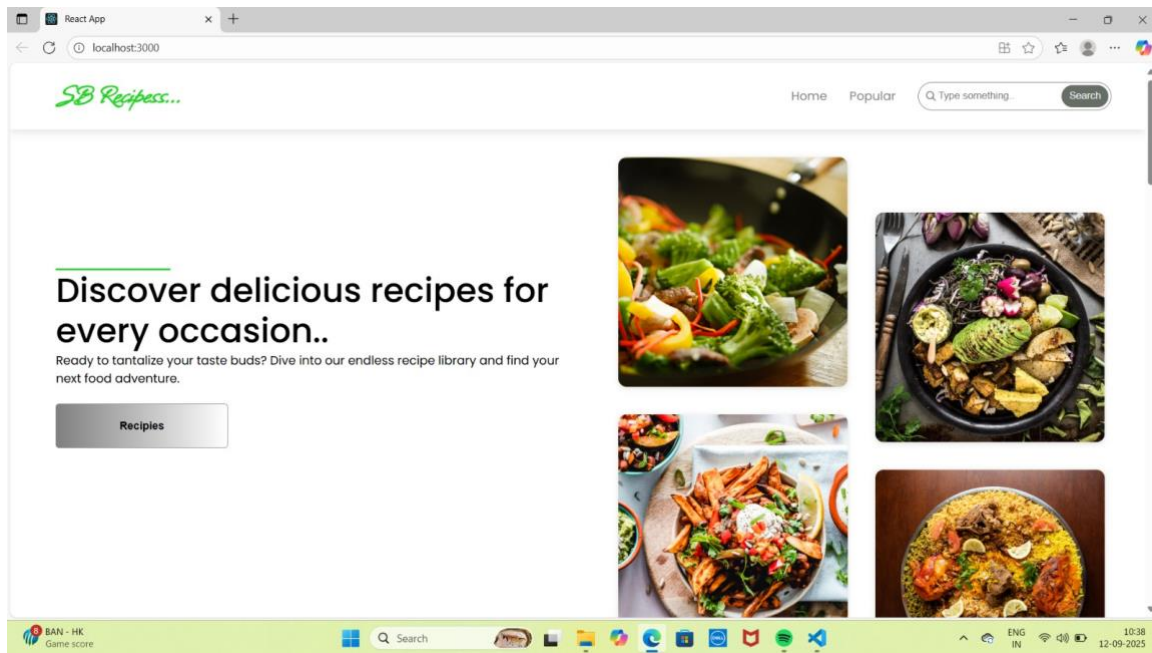
- ***Project Details Page***

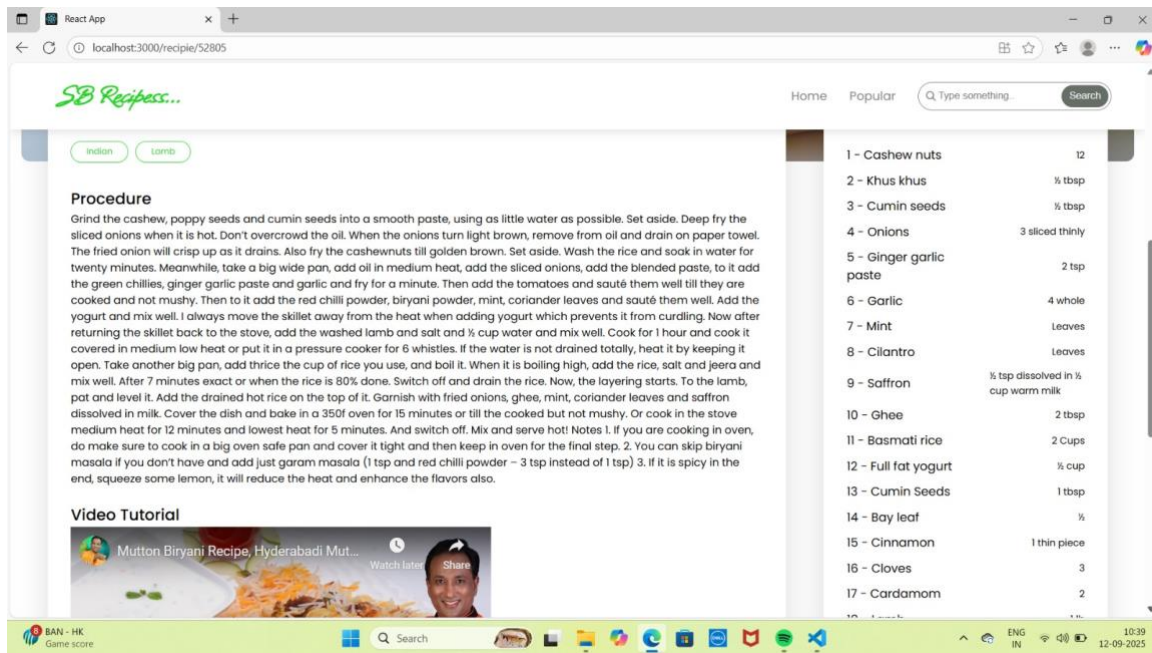
10. Testing

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

11. Screenshots or demo







12. Known issues:

- ***State Management***: Efficiently handling state changes and props in React components to ensure seamless user experience.
- ***API Integration***: Managing API requests and responses using Axios, handling errors, and optimizing data fetching for recipes.
- ***Responsive Design***: Ensuring the application is visually appealing and user-friendly across various devices and screen sizes using Bootstrap or Tailwind CSS.
- **Error*Handling***: Implementing robust error handling mechanisms to handle API failures and response errors efficiently.
- ***Navigation and Routing***: Seamlessly navigating between pages using React router

13.Future Enhancement

Ingredient-Based Recipe Search:

Allow users to input available ingredients to receive recipe suggestions, making it easier to plan meals based on what they have on hand.

Multimedia Integration:

Incorporate embedded videos or augmented reality overlays for visual guidance on complex steps, providing a richer and more interactive cooking experience.

Personalized Meal Planning:

Develop features for users to save favorite recipes, create meal plans, and generate customized weekly menus based on dietary preferences and available ingredients.

User Experience Enhancements

Integrated Timers and Notifications:

Embed interactive timers within recipe steps that provide both visual and audible alerts when a step requires precise timing.