

# Project Documentation

## 1. Introduction

- **PROJECT TITLE:** COOKBOOK: YOUR VIRTUAL KITCHEN ASSISTANT
- **Team ID :** NM2025TMID40287
- **Team Leader:** MAGESHWARAN S [mageshwaran03037@gmail.com]
- **Team Members:** YOGA PRASANNA J [yogap2232006@gmail.com]  
SANTHOSH S [santhoselva007@gmail.com]  
VIGNESH KUMAR E R [vvigneshkumar972@gmail.com]

## 2. Project Overview

- **Purpose:** Team-Sanmayog is a dynamic and intuitive real-time chat application. It is designed to provide users with a seamless and interactive communication experience. The platform facilitates instant messaging, allowing users to connect and chat with each other efficiently and securely.

### Features:

- **User Authentication:** Secure user registration and login functionality.
- **Real-Time Messaging:** Instantaneous sending and receiving of messages with other users.
- **Online User Status:** View a list of currently online users.
- **Responsive Design:** A user-friendly interface that adapts to various screen sizes.

## 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

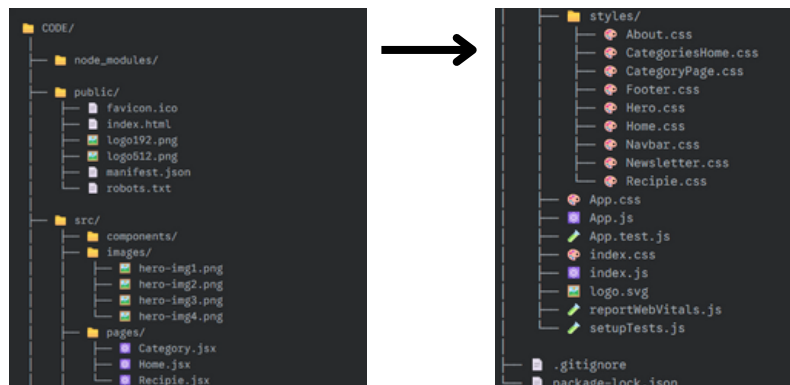
### • Installation Steps:

```
# Clone the repository git clone

# Install client dependencies cd
client npm install

# Install server dependencies cd
../server npm install
```

## 5. Folder Structure



## 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

The user interface is built with

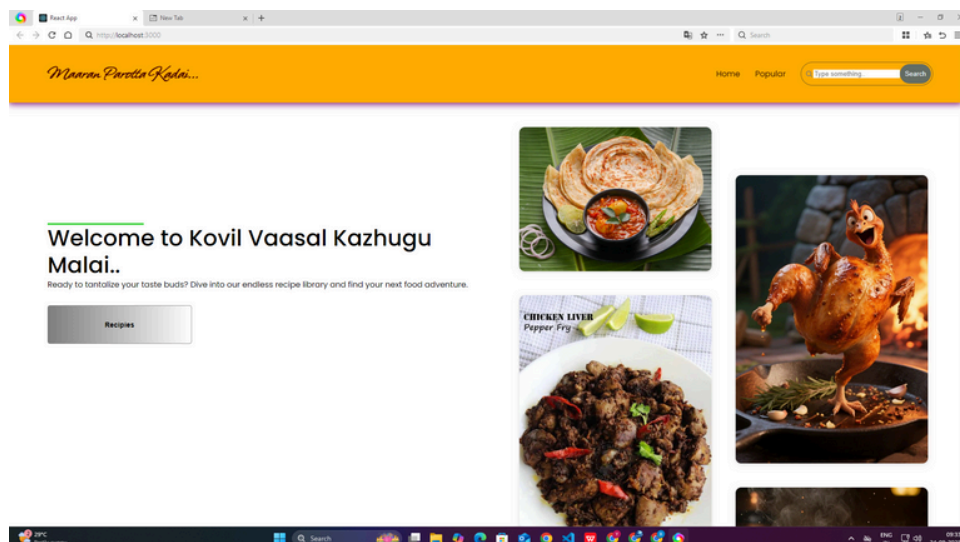
React.js and is designed to be clean and intuitive. The primary components of the UI include:

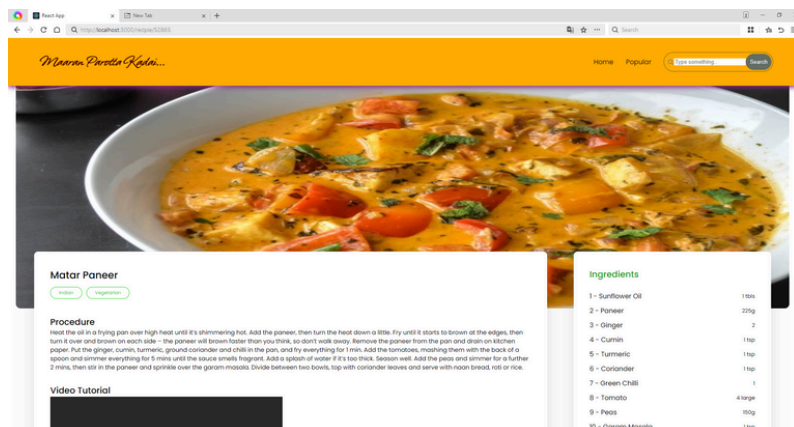
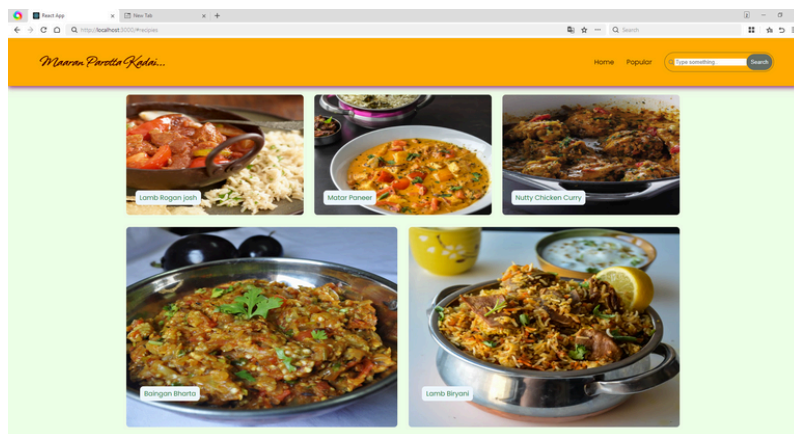
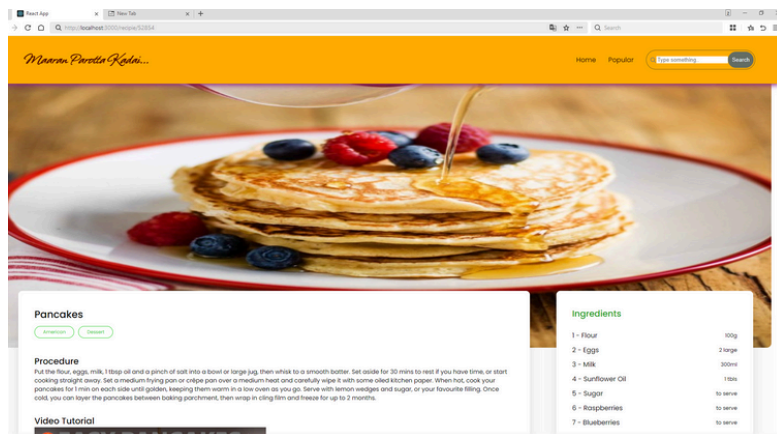
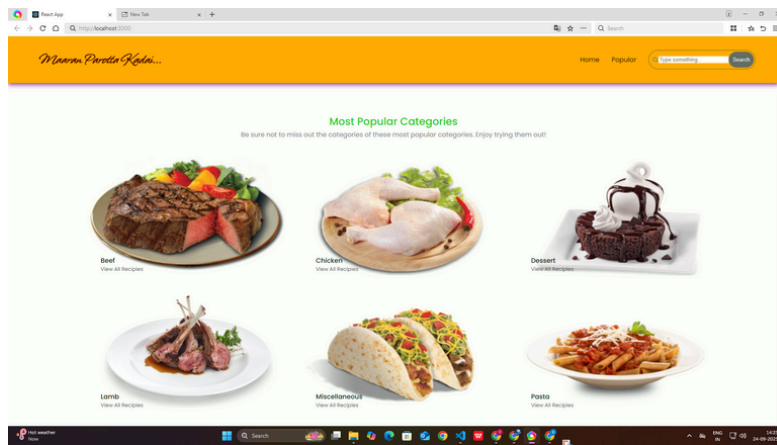
- Login/Registration Pages: Forms for new user signup and existing user sign-in.
- Chat Interface: The main screen where users can see messages in real-time, view a list of online users, and send new messages.
- Responsive Design: The layout is built to be functional and visually appealing across different devices, such as desktops and mobile phones.

## 10. Testing

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

## 11. Screenshots or Demo





## **12. Known Issues :**

There are no major known issues or bugs documented for the application at this time. The core functionality for user authentication and real-time messaging operates as expected.

## **13. Future Enhancements**

**Several features are being considered for future development to expand the application's capabilities. 🚀 Potential enhancements include:**

- **Group Chat:** Implementing functionality for users to create or join multi-user chat rooms.
- **File Sharing:** Allowing users to send images, documents, and other media files within chats.
- **Push Notifications:** Adding notifications to alert users of new messages when they are not actively using the application.
- **User Profiles:** Creating customizable user profiles with avatars and status messages.
- **Typing Indicators:** Showing when another user is actively typing a message in the chat.