

# **CHEFMATE**

## **YOUR VIRTUAL KITCHEN ASSISTANT**

### **THE FLAVOURS OF GREEN AND FIRE GRILL**

## **DOCUMENTATION:**

### **I. INTRODUCTION**

PROJECT TITLE:

➤ YOUR VIRTUAL KITCHEN ASSISTANT

TEAM MEMBERS:

- Yuvarani S- Team Leader
- Roshini S-Team Member
- Rithika D-Team Member
- Ridhiksha V-Team Member
- Yuvashree K-Team Member

### **II. PROJECT OVERVIEW**

PURPOSE:

This project is a React-based application designed to provide an interactive and dynamic user experience. It utilizes Create React App as its foundation and incorporates various dependencies for enhanced functionality.

FEATURES:

- Search and save recipes
- Generate grocery lists based on recipes
- AI-powered cooking assistant
- Step-by-step cooking instructions
- Personalized meal planning

### III. ARCHITECTURE

The application follows a component-based architecture, with reusable UI elements managed through React components. It consists of:

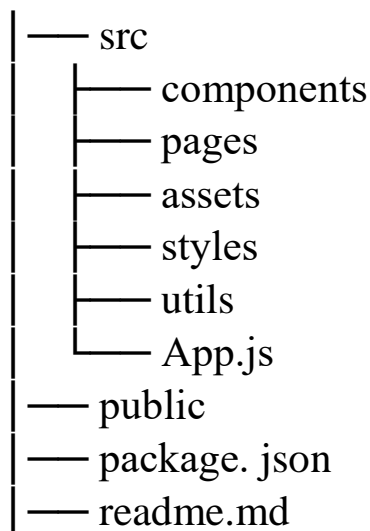
- A structured folder hierarchy for maintainability.
- State management for efficient data flow.
- API integrations for dynamic content.
- Modular styling for a clean UI.

### IV. SETUP INSTRUCTIONS

To set up the project locally, follow these steps:

1. Clone the repository:  
`git clone <repository-URL>`
2. Navigate to the project directory:  
`cd Recipes-application-main`
3. Install dependencies:  
`NPM install`
4. Start the development server:  
`NPM start`

### V. FOLDER STRUCTURE



## **VI. RUNNING THE APPLICATION**

Running the following command to start the development server:

➤ NPM start

## **VII. COMPONENTS DOCUMENTATION**

- Recipe List: Displays a list of recipes.
- Recipe Detail: Shows details of a selected recipe.
- Search Bar: Allows users to search for recipes.
- Favourites: Displays favourite recipes saved by users.
- Components are stored in /src/components and follow a modular structure:
- Header.js - Navigation bar
- Footer.js - Footer section
- Button.js - Reusable button component
- Form.js - Form component for user input

## **VIII. STATE MANAGEMENT**

- The application manages state using Context API/Redux to store and retrieve recipe data efficiently.

## **IX. USER INTERFACE**

- The UI is designed for a smooth user experience, featuring:
- A clean and minimalistic design
  - Responsive layout for mobile and desktop
  - Search functionality for recipes

## **X. STYLING**

- The application uses CSS Modules/Tailwind CSS/Styled Components for styling.

## **XI. TESTING**

- Unit and integration tests can be run using:

NPM test

## **XII. SCREENSHOT OR DEMO**

- [INSERT SCREENSHOT OR A LINK TO THE LIVE DEMO]

## **XIII. KNOWN ISSUES**

- Some API calls may fail due to rate limits.
- UI responsiveness needs improvements on smaller screens.

## **XIV. FUTURE ENHANCEMENTS**

- Add user authentication
- Implement recipe categories
- Improve performance with caching
- Enhance UI with animations