Frontend Development with React.js

Project Documentation format

# Introduction

Project Title: Cookbook – Your Virtual Kitchen Assistant

Team ID: NM2025TMID48239

Team Leader: G. Mohana (Email: gmohana235@gmail.com)

Team Members:

• Dharshini M – Email: dharshinim8977@gmail.com

• Sathya P – Email: srinivasanmaster0@gmail.com

• Siva Sankari S – Email: sankarisuresh434@gmail.com

# Project Overview

Purpose: The purpose of this project is to simplify meal planning and cooking by offering a user-friendly interface to search, save, and organize recipes while providing helpful kitchen tips and recommendations.

Features:

* Recipe search by ingredients, cuisine, or dietary preferences
* Interactive recipe instructions with step-by-step guidance
* Personalized recommendations based on user preferences
* Saved recipes and favorites management
* Ingredient substitution suggestions
* Responsive and visually appealing UI for both desktop and mobile

# Architecture

Component Structure:

- App.js – Root component managing routing and global layout  
- Header & Footer – Common UI components across all pages  
- RecipeSearch – Allows users to search for recipes by keywords or ingredients  
- RecipeList – Displays recipe results from search queries  
- RecipeDetails – Shows step-by-step instructions and ingredients for a selected recipe  
- Favorites – Stores user’s saved recipes  
- UserProfile – Handles user preferences and personalized suggestions  
- UI Components – Buttons, Cards, Modals, Inputs for reusability

State Management: Managed using React Context API for global state (user preferences, favorites). Local state handled within components.

Routing: Implemented using React Router DOM with routes for Home, Search, Recipe Details, Favorites, and Profile.

# Setup Instructions

Prerequisites: Node.js (v14 or later), npm/yarn, Git.

Installation:

1. Clone the repository: git clone <repo-url>

2. Navigate to project folder: cd cookbook-frontend

3. Install dependencies: npm install

4. Start development server: npm start

# Folder Structure

client/  
 ├── public/ # Static assets  
 ├── src/  
 │ ├── components/ # Reusable components  
 │ ├── pages/ # Page-level components  
 │ ├── assets/ # Images, icons, styles  
 │ ├── hooks/ # Custom hooks  
 │ ├── utils/ # Helper functions  
 │ └── App.js # Root component

# Running the Application

Frontend: npm start in the client directory.

# Component Documentation

Key Components: RecipeSearch, RecipeDetails, Favorites, UserProfile.

Reusable Components: Button, Card, Modal for UI consistency.

# State Management

Global State: Managed via Context API for authentication, favorites, and preferences.

Local State: Managed within individual components for temporary data.

# User Interface

Responsive and modern UI built with Tailwind CSS, ensuring seamless usage on mobile and desktop.

# Styling

CSS Frameworks/Libraries: Tailwind CSS and CSS Modules.

Theming: Custom theme system for consistent colors and typography.

# Testing

Testing Strategy: Unit testing with Jest, component testing with React Testing Library, and end-to-end testing with Cypress.

Code Coverage: Ensured through Jest coverage reports.

# Screenshots or Demo

Provide screenshots of Home Page, Recipe Search, Recipe Details, and Favorites.

# Known Issues

• Limited recipe API data may restrict results.  
• Offline functionality not yet supported.

# Future Enhancements

• Add meal planning calendar  
• Support voice-based recipe navigation  
• Enable grocery list generation  
• Integrate with smart kitchen devices