Frontend Development with React.js - Project Documentation

# 1. Introduction

### Project Title: Your Virtual Kitchen Assistant.

Team Leader:AKSHIYA G R

Team Members:

ABINAYA E

ANITHA KUMARI R

SUPRIYA S

VENNILA S

# 2. Project Overview

Purpose: The Recipe Finder Web Application is designed to help users explore, search, and view different recipes based on categories. It provides an interactive interface for food enthusiasts.

Features:   
- Home page with featured recipes  
- Category-wise recipe browsing  
- Detailed recipe view  
- Responsive navigation and footer  
- Newsletter subscription form

# 3. Architecture

Component Structure: The application is divided into multiple React components such as Navbar, Hero, Footer, About, CategoriesHome, and Recipie Page components. These components interact to provide a modular UI.

State Management: Local state is managed using React's useState and props drilling. Global state is minimal and handled directly in pages and components.

Routing: Implemented using React Router for navigation between Home, Category, and Recipie pages.

# 4. Setup Instructions

Prerequisites: Node.js and npm must be installed.

Installation Steps:  
1. Clone the repository.  
2. Navigate to the project folder.  
3. Run 'npm install' to install dependencies.  
4. Run 'npm start' to start the development server.

# 5. Folder Structure

Client:  
- src/components: Reusable UI components  
- src/pages: Page-level components (Home, Category, Recipie)  
- src/styles: CSS styling files  
- src/assets: Static assets (images, icons)

Utilities: reportWebVitals.js and setupTests.js for performance and testing.

# 6. Running the Application

Command: npm start

# 7. Component Documentation

- Navbar: Provides site navigation.  
- Hero: Displays the landing section with featured visuals.  
- Footer: Contains links and site information.  
- About: Provides information about the application.  
- CategoriesHome: Displays recipe categories on home page.  
- Recipie: Displays detailed information about a selected recipe.  
- NewsLetter: Subscription form component.

# 8. State Management

Global State: Minimal usage, most state handled locally.  
Local State: Each component manages its UI state using useState and props.

# 9. User Interface

The application features a clean and responsive UI with a Hero section, category listings, and recipe detail pages.

# 10. Styling

CSS Frameworks/Libraries: Plain CSS with modular styles in 'src/styles/'.  
Theming: Light theming with consistent color scheme applied across components.

# 11. Testing

Testing Strategy: Basic unit tests are included using Jest and React Testing Library.  
Code Coverage: Tests focus on rendering of components and UI behavior.

# 12. Screenshots or Demo

Screenshots of the application can be added here.  
Demo can be deployed using Netlify or Vercel.

# 13. Known Issues

- Limited global state management.  
- No backend integration for dynamic recipe fetching.  
- Newsletter form is not connected to a backend.

# 14. Future Enhancements

- Integration with a recipe API for dynamic content.  
- Improved state management using Redux or Context API.  
- Enhanced UI animations.  
- User authentication and profile features.