**Frontend Development with React.js**

# Project Documentation

## Introduction

Project Title: CookBook – Your Virtual Kitchen Assistant

Team Members:

• Team Leader: Praveen M

• Member: Manjunath C

• Member: Ragunathan C

• Member: Praveen S

## Project Overview

Purpose: CookBook is a web application designed to act as a virtual kitchen assistant. It helps users explore, manage, and interact with recipes in an intuitive and user-friendly interface. The project aims to provide a simple way for users to search, view, and manage cooking instructions, while also ensuring the app runs smoothly in a modern React.js environment.

Features:

* User-friendly interface for recipe exploration.
* Organized navigation for easy browsing.
* Fast and responsive frontend built using React.js.
* Localhost deployment for quick testing and demonstration.

## Architecture

Component Structure:

* App.js – Root component that manages routing and global structure.
* Header / Navbar – Provides navigation across different sections.
* Home Page – Displays welcome content or featured recipes.
* Recipe Components – Displays detailed recipe steps and ingredients.
* Footer – Contains credits and useful links.

State Management: Local state handled with React’s useState and useEffect. No external state management library (Redux/Context) indicated.

Routing: React Router can be used for navigation between Home, Recipes, and About pages. Default route points to / (Home), project runs on http://localhost:3000.

## Setup Instructions

Prerequisites: Node.js (LTS version), npm (comes with Node.js), Visual Studio Code

Installation Steps:

1. Download Node.js LTS version from nodejs.org.
2. Install Node.js following the OS-specific installer steps.
3. Open Windows PowerShell as Administrator and run: set-executionPolicy unrestricted.
4. Press Y and hit Enter.
5. From SmartInternz portal, download the project code folder.
6. Extract the downloaded zip file.
7. Open the extracted folder in VS Code.
8. Open a terminal in VS Code and run: npm install
9. Run npm start
10. Application runs on: http://localhost:3000

## Folder Structure

CookBook/  
│── public/ # Static files (index.html, favicon)  
│── src/  
│ ├── assets/ # Images, icons  
│ ├── components/ # Reusable UI components (Navbar, Footer, etc.)  
│ ├── pages/ # Page-level components (Home, Recipes, About)  
│ ├── utils/ # Helper functions (if any)  
│ ├── App.js # Root component  
│ ├── index.js # Entry point  
│── package.json # Dependencies and scripts

## Running the Application

To start the frontend server locally:

npm start

The app runs at http://localhost:3000

## Component Documentation

Key Components:

* Navbar: Provides app-wide navigation. Props: menuItems.
* RecipeCard: Displays a recipe with name, ingredients, and description. Props: title, image, ingredients.
* Footer: Displays project info and credits.

Reusable Components: Button, Card, Loader, etc.

## State Management

Global State: Minimal global state, primarily managed at the App.js level. Could be extended with Context API if project scales.

Local State: Managed using useState within components. Example: handling input fields, search filters, or UI toggles.

## User Interface

UI opens in browser via localhost:3000. Recipe browsing and viewing interface is provided. (Screenshots can be added after deployment or demo recording.)

## Styling

CSS Frameworks/Libraries: Plain CSS and React inline styles (can be extended with Tailwind, Bootstrap, or Material-UI).

Theming: Default theme with scope for future light/dark mode implementation.

## Testing

Testing Strategy: Manual testing of components during development. Can be extended with:  
- Unit Testing: Jest  
- Component Testing: React Testing Library

Code Coverage: Not explicitly included, but Jest provides built-in coverage reports if implemented.

## Screenshots or Demo

The app runs at: http://localhost:3000  
Demo video link available via SmartInternz portal.

## Known Issues

* Limited to local environment (requires Node.js setup).
* No backend integration (currently frontend only).
* State management limited to local state (no global state persistence).

## Future Enhancements

* Add search functionality for recipes.
* Integrate with a backend or recipe API for real data.
* Add user authentication (login, profile).
* Improve UI/UX with Material UI / Tailwind CSS.
* Add dark mode and advanced theming.