

CookBook: Your Virtual Kitchen Assistant

Introduction

Project Title: CookBook – Your Virtual Kitchen Assistant

Team Members:

Leader: Kanagassakthivel B

Members: Girish Kumar S, Gokulavarman V, Gopinath K, Murali N

Project Overview

Purpose:

CookBook is a React.js-based recipe management and discovery web application. It allows users to explore, organize, and share recipes with an intuitive and modern interface. The project aims to simplify recipe discovery, provide dynamic search, and enable users to maintain their personal recipe collections online.

Features:

Dynamic search for quick recipe discovery

Categorized recipe browsing

Save and organize favorite recipes

Intuitive and user-friendly interface

Responsive design for desktop and mobile

MongoDB integration for recipe and user data management

Architecture

Component Structure:

Navbar Component – Provides navigation across pages.

Search Component – Enables dynamic search for recipes.

RecipeList Component – Displays categorized recipes.

RecipeDetail Component – Shows full recipe details.

Favorites Component – Allows saving and managing favorite recipes.

Footer Component – Contains additional links and information.

State Management:

Local state with React Hooks (useState, useEffect) for handling recipe data and UI states.

Context API for managing global states such as user favorites and recipe collections.

Routing:

Implemented using React Router for smooth navigation between pages (Home, Categories, Recipe Details, Favorites).

Setup Instructions

Prerequisites:

Node.js

npm

MongoDB (for backend database support)

Installation:

1. Install Node.js and npm

2. Clone the repository:

```
git clone <repo-link>
```

3. Navigate to the project folder:

```
cd cookbook
```

4. Install dependencies:

```
npm install
```

5. Run the app:

```
npm start
```

Folder Structure

Client (React App):

src/components – Reusable components (Navbar, Footer, RecipeCard, etc.)

src/pages – Page-level components (Home, Categories, Favorites, RecipeDetails)

src/assets – Images, icons, and static files

src/context – Context API setup for global state

src/utils – Utility functions and custom hooks

Running the Application

Start the frontend server locally:

npm start

Component Documentation

Key Components:

Navbar – Navigation bar with links to main pages.

Search – Input field with dynamic recipe filtering.

RecipeList – Displays recipe cards with category filters.

RecipeDetail – Shows complete recipe instructions and ingredients.

Favorites – Displays recipes saved by the user.

Reusable Components:

RecipeCard – Displays recipe thumbnail, name, and quick actions.

Button – Custom styled button for various actions.

State Management

Global State:

Managed with Context API to handle user favorites and recipe collections.

Local State:

Managed with React Hooks for search queries, recipe details, and UI interactions.

User Interface

Modern, responsive design for desktop and mobile.

Intuitive navigation with clear recipe categories.

Simple layout for beginners and professionals alike.

Styling

CSS Frameworks/Libraries: Tailwind CSS for styling.

Theming: Default light mode; future support for dark mode planned.

Testing

Testing Strategy:

Unit tests for major components using Jest and React Testing Library.

Integration tests for search and favorites features.

Code Coverage:

Jest configuration for code coverage reports.

Screenshots or Demo

(To be added once project is live – UI mockups or demo link can be inserted here.)

Known Issues

No authentication implemented yet (recipes not tied to specific users).

No review or rating system currently available.

Future Enhancements

User authentication and personalized recipe collections.

Recipe rating and reviews.

Meal planning and shopping list integration.

AI-powered recipe recommendations.

Dark mode option.