**Frontend Development with React.js**

**Project Documentation format**

**Introduction**

**Project Title**: CookBook: Your Virtual Kitchen Assisstent

**Team Id:** Team-159066

**Team Details:**

|  |  |
| --- | --- |
| **Team Member** | **Team Member Email ID** |
| **Shiflina Nilofar P(Lead)** | Shiflinanilofar22@gmail.com |
| Shamitha V | Shamitha1108@gmail.com |
| Harini E | Elumalaiharini43@gmail.com |
| Madhu Mitha B | Madhu2303mitha@gmail.com |

**Project Overview**

**Purpose:**

The **CookBook** project aims to serve as a virtual kitchen assistant, helping users find and manage recipes efficiently. The platform allows users to browse a vast collection of recipes, save their favorites, and get step-by-step cooking instructions. The primary goal is to simplify meal planning and encourage home cooking by providing an intuitive and interactive UI.

**Features:**

* **Recipe Discovery:** Users can explore a vast collection of recipes based on categories, ingredients, and user preferences.
* **Search and Filter:** Advanced search capabilities allow users to find specific recipes quickly.
* **Step-by-Step Instructions:** Clear, concise cooking instructions guide users through each recipe.
* **Personalization:** Users can save their favorite recipes and get personalized recommendations.
* **Ingredient Checker:** The system can check available ingredients and suggest recipes accordingly.

**Architecture**

**Component Structure:**

The project follows a modular component-based structure, breaking the UI into reusable elements. Major components include:

* **Navbar:** Handles navigation and user authentication status.
* **RecipeCard:** Displays a summary of recipes in a visually appealing manner.
* **RecipeDetails:** Provides detailed information about a selected recipe, including ingredients and instructions.
* **SearchBar:** Allows users to search and filter recipes.

**State Management:**

The application employs **Context API** for global state management, ensuring a seamless flow of data across components. It helps maintain user authentication status, recipe preferences, and UI interactions.

**Routing:**

Using **React Router**, the project implements client-side navigation, allowing seamless transitions between pages such as:

* **Home Page:** Displays trending and recommended recipes.
* **Recipe Details Page:** Provides detailed information about selected recipes.

**Setup Instructions**

**Prerequisites:**

Ensure the following dependencies are installed:

* **Node.js** (v14 or later)
* **npm** or **yarn**
* **React.js** (latest stable version)
* **React Router**

**Installation:**

1. Navigate to the project directory:
2. cd cookbook
3. Install dependencies:
4. npm install
5. Run the application:
6. npm start

**Folder Structure**

**Client:**

* **/src**
  + **/components** - Reusable React components
  + **/pages** - Page-level components
  + **/assets** - Images, icons, and static files
  + **/services** - API calls and backend interactions
  + **/context** - Context providers for state management

**Utilities:**

* **Custom Hooks:** Handles API requests, local storage, and user preferences.
* **Helper Functions:** Includes functions for data formatting, filtering, and validation.

**Running the Application**

Run the frontend server with:

npm start

This will launch the application at [**http://localhost:3000/**](http://localhost:3000/)

**Component Documentation**

**Key Components:**

* **RecipeCard:** Displays a summary of a recipe.
* **RecipeDetails:** Provides in-depth details about a recipe.
* **SearchBar:** Enables filtering of recipes.

**Reusable Components:**

* **Button:** Custom-styled button component.
* **Modal:** Used for pop-up dialogs.
* **FormField:** Handles user input in forms.

**State Management**

**Global State:**

* **Recipe Data State:** Stores fetched recipe information.

**Local State:**

* **Search Input:** Stores user input for filtering recipes.

**User Interface**

Screenshots and GIFs of the UI, showcasing features like:

* Homepage with trending recipes.
* Recipe details page with step-by-step instructions.
* Favorites section displaying saved recipes.

**Styling**

**CSS Frameworks/Libraries:**

* **Tailwind CSS** for styling.
* **Styled-Components** for scoped component styling.

**Theming:**

A **light** is implemented to enhance user experience.

**Testing Strategy:**

* **Unit Testing:** Performed using Jest and React Testing Library.
* **Integration Testing:** Ensures component interactions work correctly.
* **End-to-End Testing:** Cypress is used for full application testing.

**Code Coverage:**

Coverage reports generated using Jest ensure comprehensive test validation.

**Screenshots or Demo**

A link to the live application or recorded demo showcasing:

* Recipe search and filtering.
* Procedure, Incredients and youtube video for references.
* Step-by-step cooking instructions.

**Home Module:**

A screenshot of a recipe

AI-generated content may be incorrect.

**Category Module:**

A screenshot of a menu

AI-generated content may be incorrect.

**Popular / Trending Dishes Module:**

A screenshot of a food menu

AI-generated content may be incorrect.

**Home Module:**

A screenshot of a computer

AI-generated content may be incorrect.

**Recipe Module:**

A screenshot of a food

AI-generated content may be incorrect.

**Known Issues**

* Some API responses may have delayed loading times.
* UI glitches in some mobile resolutions.

**Future Enhancements**

* **AI-Powered Recipe Suggestions:** Personalized recommendations based on user behavior.
* **Community Recipes:** Users can upload and share their own recipes.
* **Voice-Guided Cooking:** Step-by-step instructions read aloud.
* **Advanced Filtering:** Filters for dietary preferences (vegan, gluten-free, etc.).