**PROJECT DOCUMENTATION**

**Introduction**

**Project Title:**

CookBook: Your Virtual Kitchen Assistant

**Team Members:**

* **[R.DHARSHINI DEVI]** - Frontend Developer, Project Manager
* **[M.SHANMUGA PRIYA,P.ROOPA]** - Backend Developer
* **[V.ASHWINI PRIYA,R.MONISHA]** - UI/UX Designer

**Project Goals:**

The primary goal of CookBook is to revolutionize the way people access, organize, and utilize recipes. By providing an aesthetically pleasing and highly functional platform, the project aims to cater to food enthusiasts of all skill levels, bridging the gap between traditional recipe books and modern technology.

**User Stories:**

1. **As a user**, I want to search for recipes by ingredient or cuisine type so that I can find something specific to cook.
2. **As a user**, I want to save my favorite recipes so that I can easily access them later.
3. **As a user**, I want a visually appealing interface that makes exploring recipes enjoyable.

**Project Overview**

**Purpose:**

CookBook is designed to simplify the culinary journey by offering a robust platform that merges functionality with creativity. The purpose of the application is to:

* Provide users with a seamless way to discover recipes.
* Enable organization of favorite recipes for quick access.
* Create an engaging experience with detailed instructions and visuals.

**Features:**

1. **Search Functionality**: Users can search recipes by ingredients, cuisine, or keywords.
2. **MealsDB API Integration**: Leverages a comprehensive API for a diverse recipe database.
3. **Favorites**: Allows users to save recipes for future use.
4. **Responsive Design**: Optimized for desktops, tablets, and mobile devices.
5. **Dark Mode**: Offers better visibility for users browsing in low-light environments.
6. **Recipe Details**: Detailed pages for each recipe with step-by-step instructions and images.
7. **Animations**: Smooth transitions and interactive elements for enhanced user engagement.

**Architecture**

**Technical Stack:**

* **Frontend**: React.js, Tailwind CSS
* **Backend**: Node.js (if applicable)
* **API**: MealsDB API for recipes
* **Deployment**: Vercel for hosting the frontend

**Component Structure:**

**Core Components:**

1. **Navbar**: Handles navigation across pages, includes links to trending recipes, categories, and saved items.
2. **Hero Section**: A visually impactful introductory section with an inviting tagline and call-to-action.
3. **Trending Recipes**: Highlights popular recipes fetched dynamically via the API.
4. **Category Page**: Displays recipes categorized by cuisine type.
5. **Recipe Page**: Provides comprehensive information about selected recipes.
6. **Footer**: Contains additional navigation links, copyright information, and social media icons.

**Component Tree Diagram:**

App

├── Navbar

├── HomePage

│ ├── HeroSection

│ ├── TrendingRecipes

├── CategoriesPage

│ ├── CategoryList

│ ├── CategoryCard

├── RecipePage

│ ├── RecipeDetails

│ ├── IngredientsList

│ ├── Instructions

├── Footer

**Setup Instructions**

**Prerequisites:**

* **Node.js and npm**: [Download Node.js](https://nodejs.org/en/download/)
* **Code Editor**: Install Visual Studio Code for an optimized development experience: [Download VS Code](https://code.visualstudio.com/download)

**Installation:**

1. Clone the repository:
2. git clone [repository-url]
3. Navigate to the project directory:
4. cd recipe-app-react
5. Install project dependencies:
6. npm install
7. Start the development server:
8. npm start
9. Open [http://localhost:3000](http://localhost:3000/) in your browser to view the application.

**Deployment:**

To deploy the application, follow these steps:

1. Build the project:
2. npm run build
3. Use a hosting platform like **Vercel** or **Netlify** and upload the build/ directory.

**Folder Structure**

**Frontend:**

* **/components**: Houses reusable UI components (e.g., Navbar, Cards).
* **/pages**: Contains page-level components (e.g., HomePage, RecipePage).
* **/assets**: Stores static resources like images and icons.
* **/styles**: Contains global and component-specific CSS files.

**Utilities:**

* **api.js**: Manages API calls to MealsDB.
* **hooks/**: Contains custom React hooks for fetching and managing data.
* **helpers/**: Utility functions for formatting data and handling errors.

**Running the Application**

**Local Development:**

To test the app locally, use:

npm start

**Production Testing:**

1. Build the project for production:
2. npm run build
3. Use a local server to serve the build:
4. npm install -g serve
5. serve -s build
6. Open the provided localhost URL in your browser.

**Component Documentation**

**Key Components:**

**Navbar:**

* **Props**: None
* **Description**: Provides app-wide navigation links.

**HeroSection:**

* **Props**: title (string), subtitle (string)
* **Description**: Displays an introductory banner with a call-to-action button.

**TrendingRecipes:**

* **Props**: recipes (array)
* **Description**: Displays a carousel of popular recipes.

**RecipeCard:**

* **Props**: image (string), title (string), onClick (function)
* **Description**: Represents a single recipe with a thumbnail and title.

**Styling**

**Themes:**

* **Light and Dark Modes**: Offers better readability and user comfort.
* **Color Palette**: Includes soft, warm tones for an inviting aesthetic.

**Animations:**

* Hover effects for buttons and cards.
* Smooth page transitions using CSS and React animations.
* Interactive loading spinners for better user feedback.

**Testing**

**Tools:**

* **Jest**: For unit tests.
* **React Testing Library**: For component interaction tests.
* **Cypress**: For end-to-end testing.

**Strategy:**

1. Write unit tests for core components (e.g., Navbar, RecipeCard).
2. Test API calls and ensure proper rendering of data.
3. Simulate user workflows to validate app functionality.

**Future Enhancements**

1. **User Authentication**: Allow users to create accounts and save personalized data.
2. **Advanced Search Filters**: Include filters for dietary preferences and cooking difficulty.
3. **Meal Planner**: Enable users to create weekly meal plans.
4. **Voice Commands**: Implement voice search for hands-free navigation.
5. **Offline Mode**: Cache popular recipes for offline access.

**FAQ**

1. **What is CookBook?** CookBook is a virtual assistant for discovering and managing recipes.
2. **Is CookBook mobile-friendly?** Yes, it is designed to work seamlessly on all devices.
3. **Can I save my favorite recipes?** This feature will be added in future updates.
4. **Does it support multiple languages?** Multilingual support is planned for upcoming releases.

**Glossary**

* **API**: Application Programming Interface used to fetch data.
* **Component**: A reusable piece of UI in React.
* **Context API**: A React feature for global state management.
* **Tailwind CSS**: A utility-first CSS framework for styling.
* **MealsDB API**: An external API providing recipe data.