CookBook: Your Virtual Kitchen Assistant

(React Application)

Team Leader : Nivetha. R (Nivetharajesh0624@gmail.com)

Team member : Gayathri. M (gayathriammu3097@gmail.com)

Team member : Gayathri. P (gayathri.perumal.789@gmail.com)

Team member : Rubavarshini. K (rubavarshinikumar@gmail.com)

Introduction

CookBook: Your Virtual Kitchen Assistant is a revolutionary web application designed to change the way you discover, organize, and create recipes. It caters to both novice and professional chefs, offering a user-friendly interface, robust features, and a vast collection of inspiring recipes.

Project Overview

Purpose

The primary goal of CookBook is to provide a user-friendly platform that caters to individuals passionate about cooking, baking, and exploring new culinary horizons. Our objectives include creating an interface that is easy to navigate, ensuring users can effortlessly discover, save, and share their favorite recipes. It offers robust features for organizing and managing recipes, including advanced search options. Leveraging modern web development technologies such as React.js ensures an efficient and enjoyable user experience.

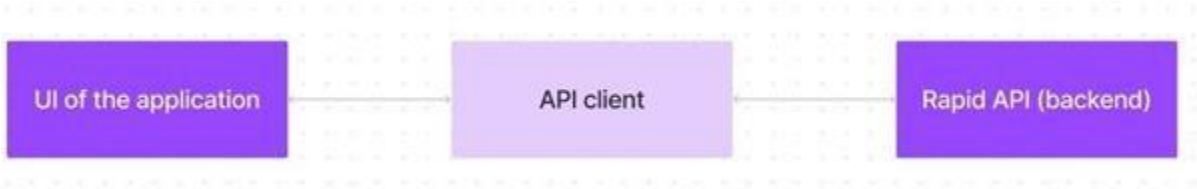
Features

CookBook includes recipes from the Meals DB API, providing a vast library of international recipes spanning diverse cuisines and dietary needs. Users can visually browse recipes, explore categories, and discover new dishes through curated image galleries. The application features an intuitive and user-friendly design, allowing seamless navigation with a clean, modern interface and clear navigation. The search feature ensures that various dishes can be accessed easily.

Architecture

Component Structure

The user experience starts with the CookBook web application's UI, likely built with a framework like React or Vue.js for a smooth, single-page experience. This UI interacts with an API client specifically designed for CookBook. This client handles communication with the backend while leveraging Rapid API to provide access to various external APIs. This suggests that CookBook integrates external data feeds or functionalities through Rapid API, enriching the user experience without building everything from scratch.



Project Structure

The project structure consists of three major folders: Components, Pages, and Styles. The Pages folder stores files that act as pages at different URLs in the application. The Components folder contains files that return small components within the application. All styling CSS files are stored in the Styles folder.

Setup Instructions

Prerequisites

The key prerequisites for developing a frontend application using CookBook include Node.js and npm. Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code locally, providing a scalable and efficient platform for building network applications. React.js is a popular JavaScript library for building user interfaces, enabling developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications. Basic knowledge of HTML for structuring the app, CSS for styling, and JavaScript for client-side interactivity is essential.

Installation

To set up the application, install Node.js and npm on your development machine. Then install React.js and create a new React app using the command:

npx create-react-app my-react-app Navigate

to the project directory using:

cd my-react-app

To start the development server and see your React application in action, use the command:

npm start

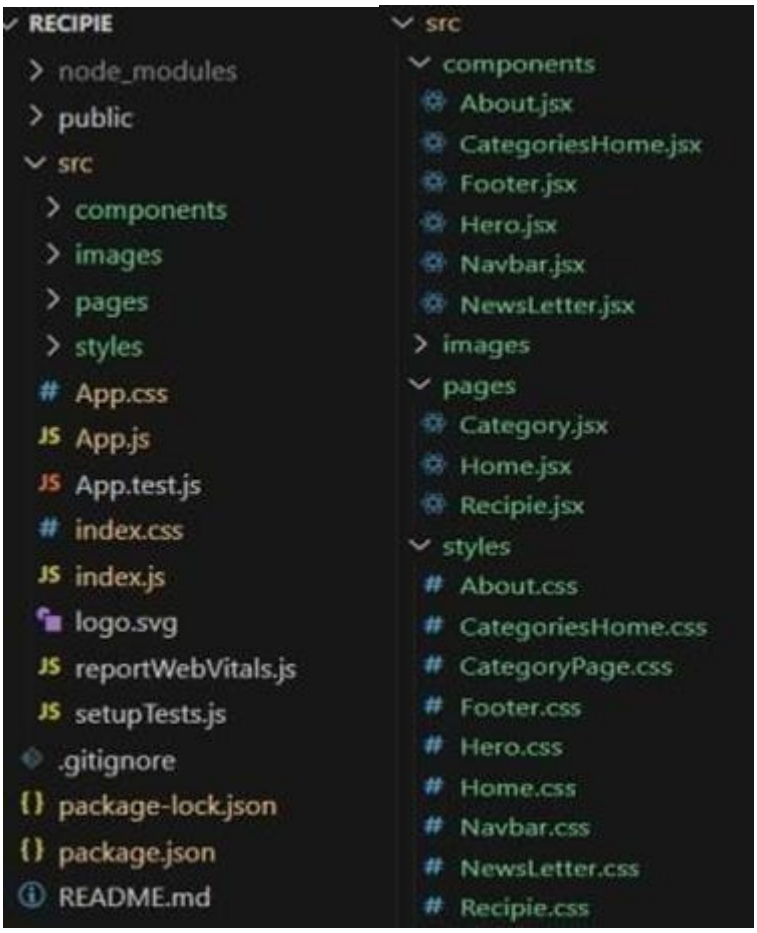
Folder Structure

Client

In this project, the files are split into three major folders: Components, Pages, and Styles. The Pages folder contains files that serve as pages at different URLs in the application. The Components folder stores all the files that return small components in the application. The Styles folder contains all the styling CSS files.

Utilities

Helper functions, custom hooks, and other utility files are stored separately to maintain modularity and code reusability.



Running the Application

To start the frontend server locally, navigate to the client directory and run the command:

npm start

Component Documentation

Key Components

The hero component provides a brief description of the application and a button to view more recipes. The popular categories component contains all the popular categories of recipes. The trending dishes component showcases some of the trending dishes in the application. The newsletter component provides an email input to subscribe to recipe newsletters.

Reusable Components

Reusable UI components such as buttons, search bars, and input fields ensure consistency across the application.

State Management

Global State

CookBook utilizes React's useState hook for managing component states and the useEffect hook for handling side effects such as data fetching. The global state is managed through the API interaction, while local states handle individual component updates.

Local State

Components maintain their own local state using the useState hook to handle user interactions and UI updates dynamically**.**

User Interface

The user interface consists of various components, including the Hero component, which introduces the application. The Popular Categories component displays different recipe categories, and the Trending Dishes component lists trending meals. The Newsletter component allows users to subscribe for updates. The Category Dishes page contains a list of dishes under a certain category, while the Recipe page includes images, recipe instructions, ingredients, and even a tutorial video.

Styling

CSS Frameworks/Libraries

CookBook employs CSS modules for styling, ensuring modular and reusable styles across components**.**

Theming

The application follows a theming system that maintains consistency in design and layout.

Testing

Testing Strategy

The testing strategy includes using Jest and React Testing Library for unit and integration testing.

Code Coverage

Code coverage tools are implemented to ensure adequate testing across components.

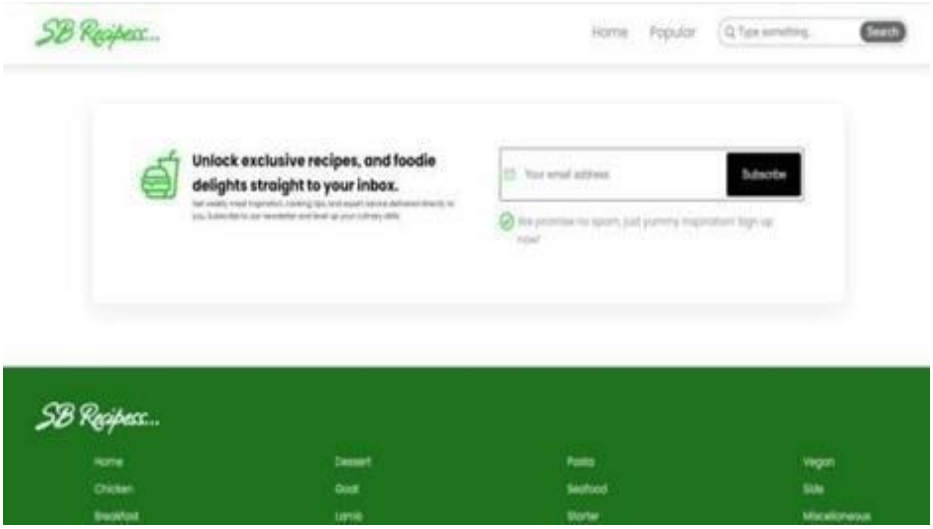
```
const [categories, setCategories] = React.useState([])

useEffect(() => {
  fetchCategories()
}, [])

const fetchCategories = async () => {
  await axios.get('https://www.themealdb.com/api/json/v1/1/categories.php')
    .then(response => {
      setCategories(response.data.categories)
      console.log(response.data.categories)
    })
    .catch(error => console.error(error));

}
```

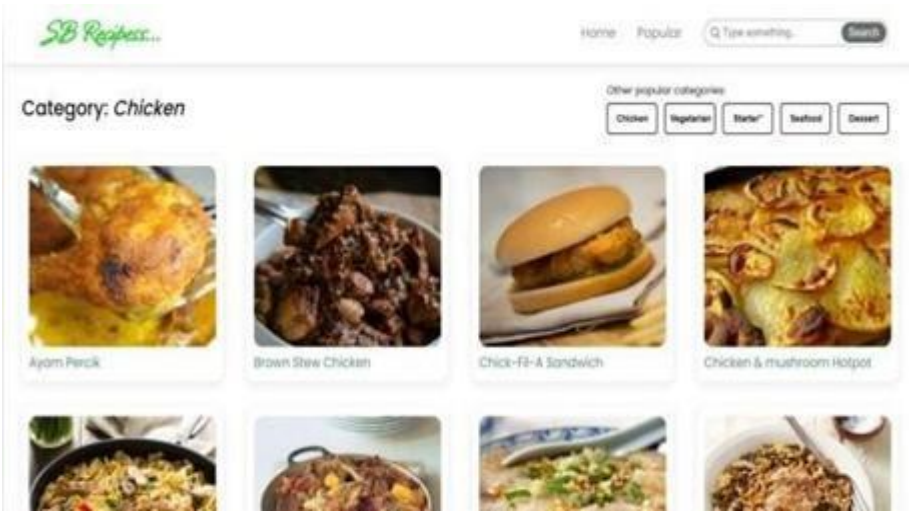
Screenshots or Demo

The application includes visual representations of different pages and components, showcasing its features and design. A demo link provides an interactive preview of the application's functionalities.

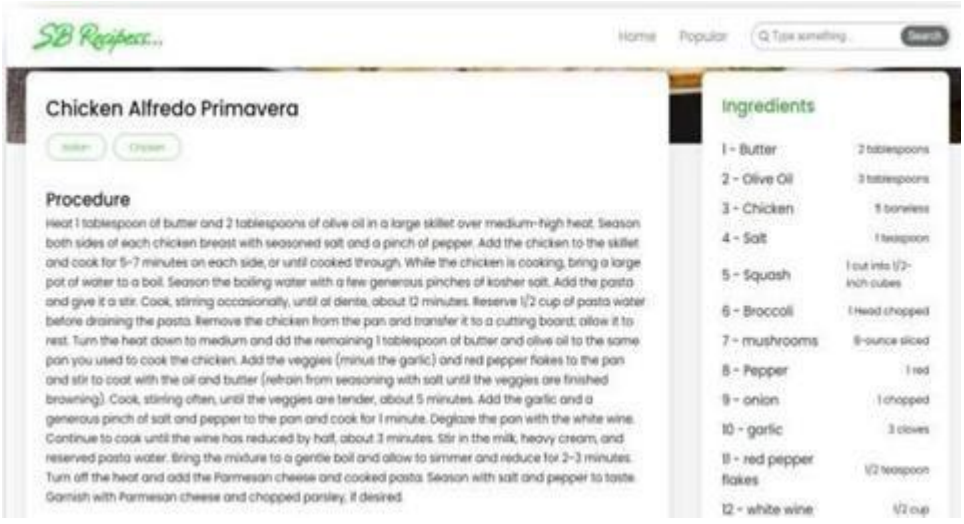

SB Recipess...
Home
Popular
Search
Trending Dishes
Lamb Rogan josh
Matar Paneer
Nutty Chicken Curry

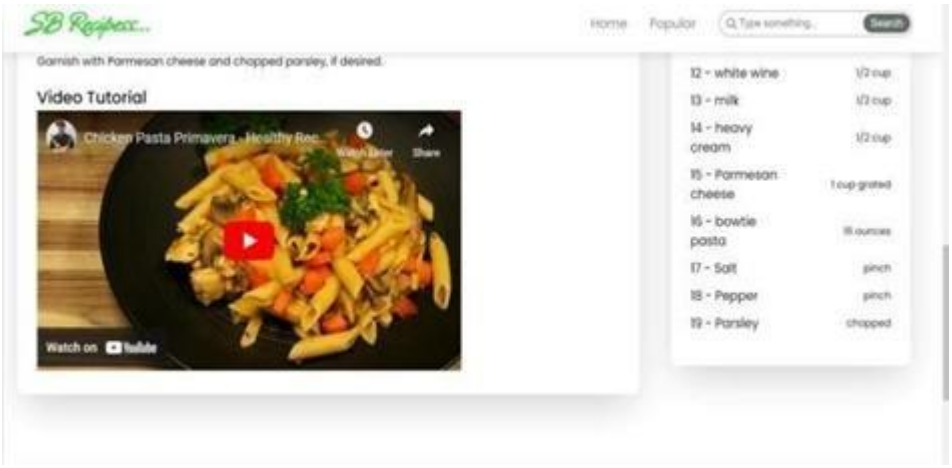

SB Recipess...
Most Popular Categories
Beef
Dessert
Lamb
Pasta


SB Recipess...
Home
Popular
Search
Unlock exclusive recipes, and foodie delights straight to your inbox.
Subscribe
SB Recipess...
Home
Chicken
Breakfast
Dessert
Goat
Lamb
Pasta
Seafood
Starter
Vegan
Side
Miscellaneous


SB Recipes...
Home
Popular
Search
Category: Chicken
Other popular categories
Chicken
Vegetarian
Starter
Seafood
Dessert
Ayam Percik
Brown Stew Chicken
Chick-Fil-A Sandwich
Chicken & mushroom Hotpot


SB Recipes...
Home
Popular
Search
Chicken Alfredo Primavera
Ingredients


SB Recipes...
Home
Popular
Search
Chicken Alfredo Primavera
Procedure
Heat 1 tablespoon of butter and 2 tablespoons of olive oil in a large skillet over medium-high heat. Season both sides of each chicken breast with seasoned salt and a pinch of pepper. Add the chicken to the skillet and cook for 5-7 minutes on each side, or until cooked through. While the chicken is cooking, bring a large pot of water to a boil. Season the boiling water with a few generous pinches of kosher salt. Add the pasta and give it a stir. Cook, stirring occasionally, until al dente, about 12 minutes. Reserve 1/2 cup of pasta water before draining the pasta. Remove the chicken from the pan and transfer it to a cutting board; allow it to rest. Turn the heat down to medium and dd the remaining 1 tablespoon of butter and olive oil to the same pan you used to cook the chicken. Add the veggies (minus the garlic) and red pepper flakes to the pan and stir to coat with the oil and butter (refrain from seasoning with salt until the veggies are finished browning). Cook, stirring often, until the veggies are tender, about 5 minutes. Add the garlic and a generous pinch of salt and pepper to the pan and cook for 1 minute. Deglaze the pan with the white wine. Continue to cook until the wine has reduced by half, about 3 minutes. Stir in the milk, heavy cream, and reserved pasta water. Bring the mixture to a gentle boil and allow to simmer and reduce for 2-3 minutes. Turn off the heat and add the Parmesan cheese and cooked pasta. Season with salt and pepper to taste. Garnish with Parmesan cheese and chopped parsley, if desired.
Ingredients
1 - Butter
2 tablespoons
2 - Olive Oil
3 tablespoons
3 - Chicken
5 boneless
4 - Salt
1 teaspoon
5 - Squash
1 cut into 1/2-inch cubes
6 - Broccoli
1 Head chopped
7 - mushrooms
8-ounce sliced
8 - Pepper
1 red
9 - onion
1 chopped
10 - garlic
3 cloves
11 - red pepper flakes
1/2 teaspoon
12 - white wine
1/2 cup



Known Issues

There are a few known limitations, such as API rate limits that may affect data fetching, and some UI responsiveness issues on smaller screens. These issues are being addressed in future updates.

Future Enhancements

Future plans include adding user authentication, implementing dark mode, and enhancing animations and transitions to improve the user experience.

Project Demo link

https://drive.google.com/file/d/1bphk8zG29ZHocViCBauUxl3Zij8sojsn/view?usp=sharing