Frontend Development with React.js

Project Documentation Format

1.Introduction

Project Title:CookBook: Your Virtual Kitchen Assistant

Team Lead:Vanmathi .S

Team Id: Team-159059

**Team Members Details:**

|  |  |
| --- | --- |
| **Team Members Name** | **Email id** |
| Vanmathi **.**S**(Leader)** | sm.madhi004@gmail.com |
| Ragavi .A | ragavi2k4@gmail.com |
| Tharani .N | ntharani168@gmail.com |
| Priyatharshini .V | priyatharshinivinayagam@gmail.com |

2.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.

StepbyStep Instructions: Clear, concise cooking instructions guide users through each recipe.

Favorites and Personalization: Users can save their favorite recipes and get personalized recommendations.

Ingredient Checker: The system can check available ingredients and suggest recipes accordingly.

User Authentication: Secure login and registration system to maintain user preferences and saved recipes.

Responsive Design: Fully optimized for various devices, including desktops, tablets, and smartphones.

3. 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.

FavoritesList: Displays the user’s saved 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.

Favorites Page: Stores user selected favorite recipes.

Login/Signup Page: Manages user authentication and account creation.

4.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. Clone the repository:

sh

git clone https://github.com/yourrepo/cookbook.git

2. Navigate to the project directory:

sh

cd cookbook

3. Install dependencies:

sh

npm install

4. Run the application:

sh

npm start

5.Folder Structure

Client:

/src

/components Reusable React components

/pages Pagelevel 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.

6.Running the Application

Run the frontend server with:

sh

npm start

This will launch the application at http://localhost:3000/

7.Component Documentation

Key Components:

RecipeCard: Displays a summary of a recipe.

RecipeDetails: Provides indepth details about a recipe.

FavoritesList: Shows the user’s saved recipes.

SearchBar: Enables filtering of recipes.

Reusable Components:

Button: Customstyled button component.

Modal: Used for popup dialogs.

FormField: Handles user input in forms.

8.State Management

Global State:

User Authentication State: Manages user login and session details.

Recipe Data State: Stores fetched recipe information.

Local State:

Search Input: Stores user input for filtering recipes.

Toggle States: Manages modal visibility and dropdowns.

9.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.

10.Styling

CSS Frameworks/Libraries:

Tailwind CSS for styling.

StyledComponents for scoped component styling.

Theming:

A light/dark mode toggle is implemented to enhance user experience.

11.Testing

Testing Strategy:

Unit Testing: Performed using Jest and React Testing Library.

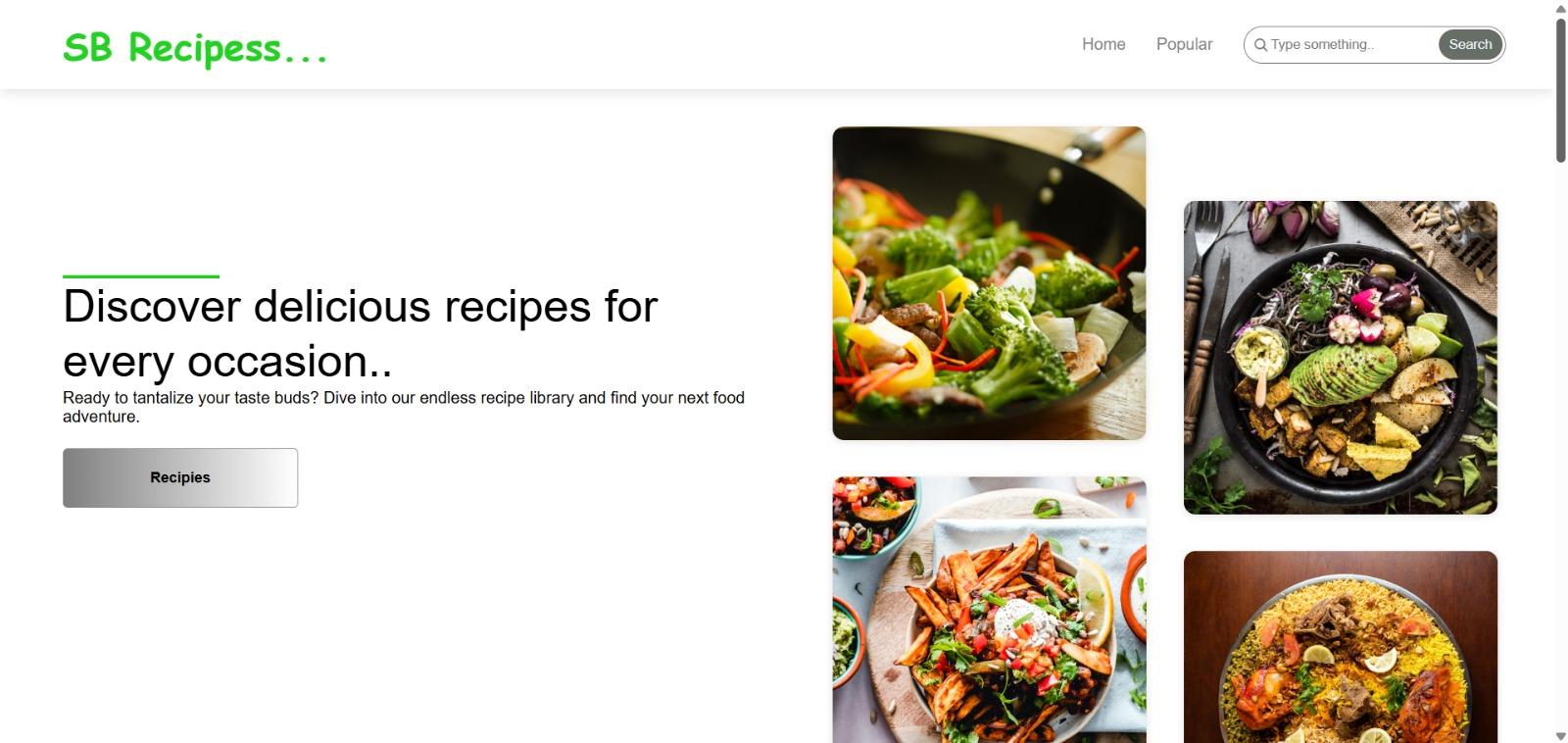
Integration Testing: Ensures component interactions work correctly.

End toEnd Testing: Cypress is used for full application testing.

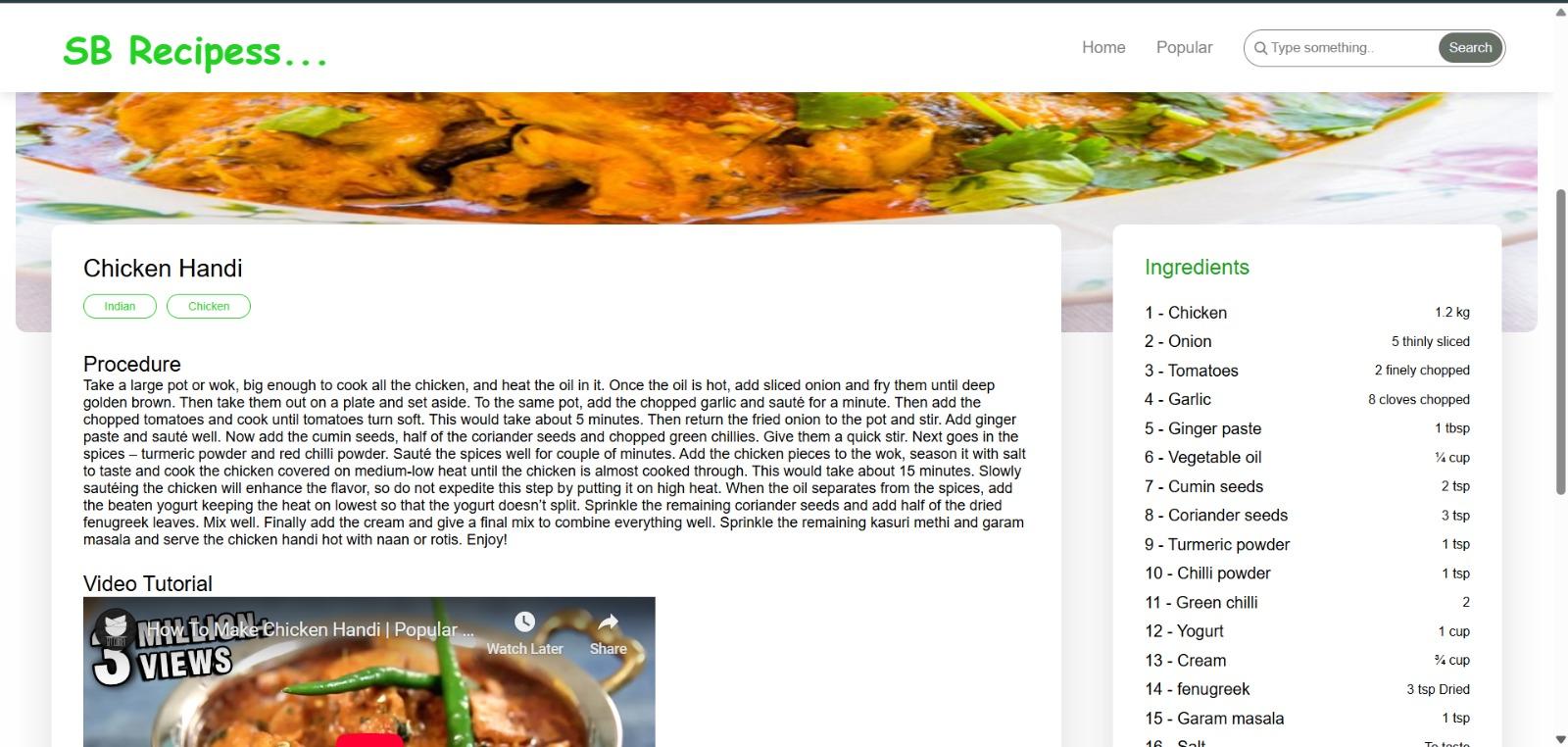
Code Coverage:

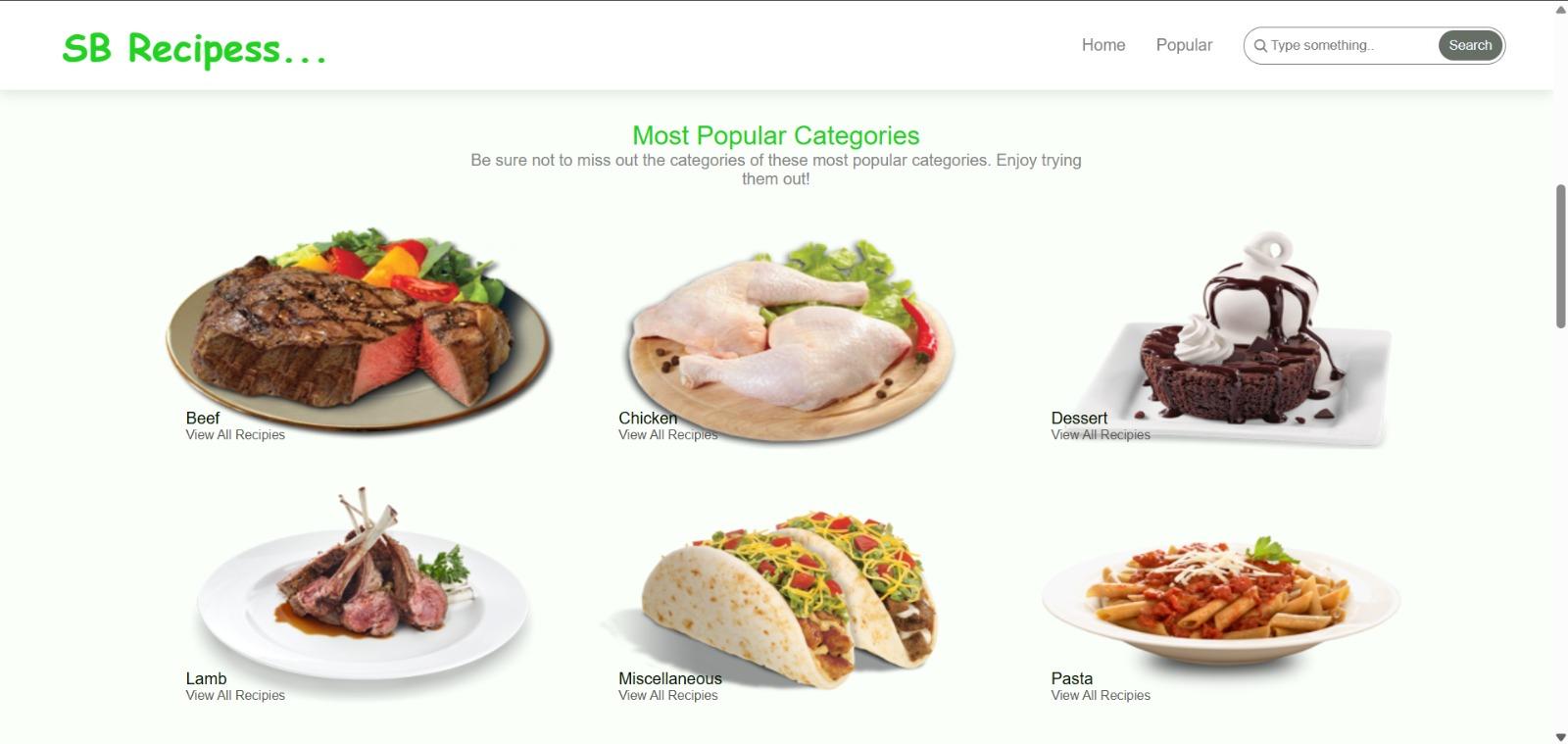
Coverage reports generated using Jest ensure comprehensive test validation.

12.Screenshots or Demo

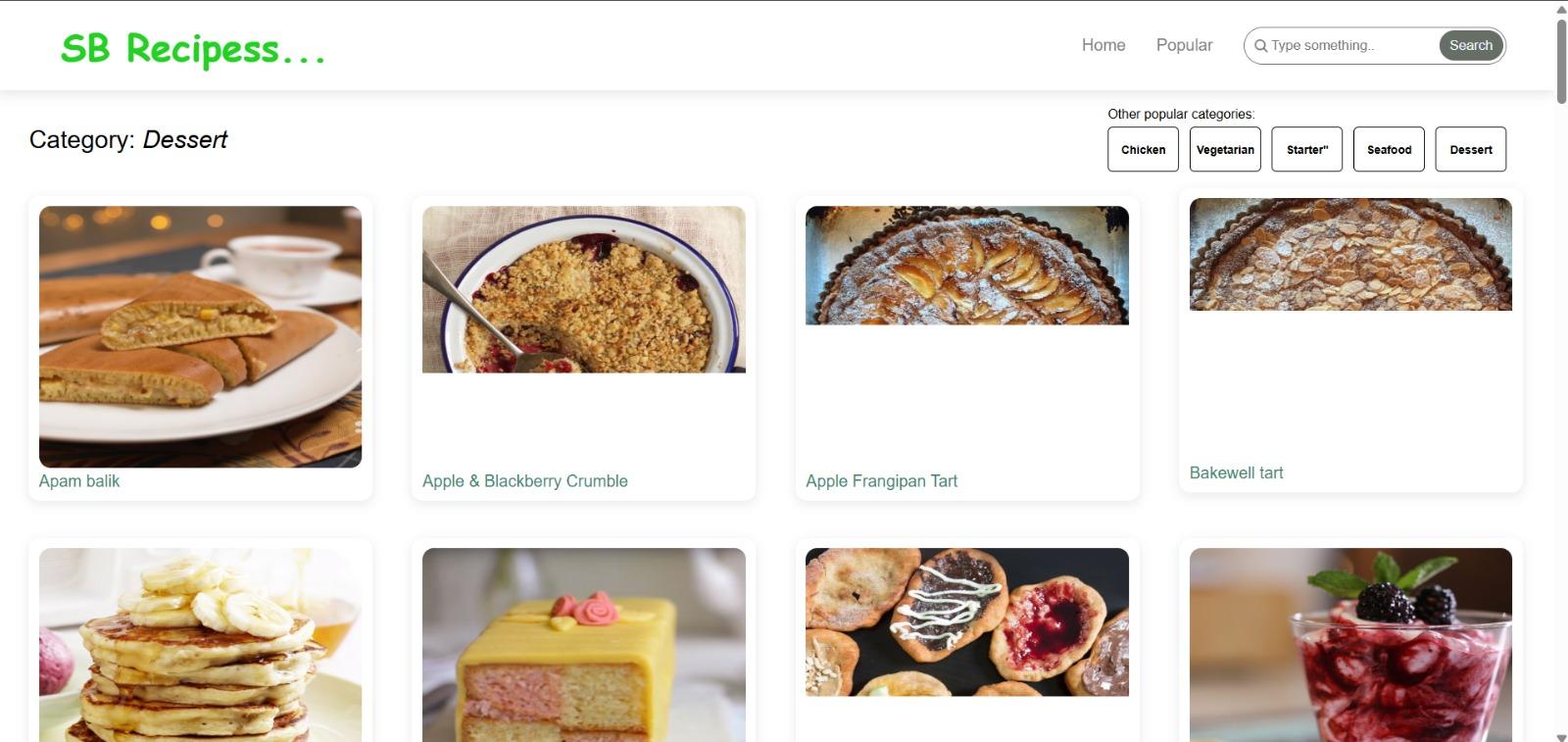
Home Page:

Step By Step Instructions:



Popular Categories:

Categories:



13.Known Issues

Some API responses may have delayed loading times.

UI glitches in some mobile resolutions.

14.Future Enhancements

AI Powered Recipe Suggestions: Personalized recommendations based on user behavior.

Community Recipes: Users can upload and share their own recipes.

VoiceGuided Cooking: Step By Step instructions read aloud.

Advanced Filtering: Filters for dietary preferences (vegan, gluten free, etc.).