CookBook.docx

CookBook: Your Visual Kitchen Assistant

Introduction

Project Title: CookBook: Your Visual Kitchen Assistant

Team Members:

• Santhiya – Team Leader

• Zeba Fathima – Team Member

• Parameshwari – Team Member

• Raja Lakshmi – Team Member

Project Overview

Purpose: The purpose of CookBook is to provide users with an interactive and visual kitchen assistant. It helps users explore recipes, follow step-by-step cooking guides, and enhance their cooking experience through a visually engaging interface.

Features:

• Recipe discovery with visual guidance

• Step-by-step cooking instructions with images/videos

• Search functionality for recipes

• Favorites/saved recipes option

• User-friendly and responsive design

• Easy navigation across recipe categories

Architecture

Component Structure: The application is built using React components such as Navbar, Home, RecipeList, RecipeDetails, and Favorites. Each component is modular and reusable, making the project scalable.

State Management: Local state is managed using React’s useState and useEffect hooks. For shared states, Context API is used to provide data across components.

Routing: React Router is used to handle navigation between pages such as Home, Recipes, and Favorites.

Setup Instructions

Prerequisites: Node.js and npm should be installed.

Installation Steps:

1. Clone the repository from GitHub

2. Navigate to the project directory

3. Run `npm install` to install dependencies

4. Run `npm start` to launch the application

Folder Structure

Client:

• src/components – Contains reusable React components

• src/pages – Contains page-level components like Home, Recipes, Favorites

• src/assets – Contains images, icons, and other static files

• src/utils – Contains helper functions and custom hooks

Running the Application

Frontend: Run the following command in the client directory:

`npm start`

Component Documentation

Key Components:

• Navbar – Provides navigation across pages

• Home – Displays featured recipes and categories

• RecipeList – Lists all recipes with filters

• RecipeDetails – Shows ingredients, steps, and visuals

• Favorites – Displays saved recipes

State Management

Global State: Managed using Context API to share user preferences and favorites.

Local State: Managed using useState and useEffect within components for handling search, filters, and UI interactions.

User Interface

The UI is designed with simplicity and clarity. It provides intuitive navigation, clear visuals of recipes, and responsive layouts for mobile and desktop views.

Styling

CSS Frameworks/Libraries: Tailwind CSS and custom CSS are used to style components.

Theming: A light theme with clean visuals is used for better user experience.

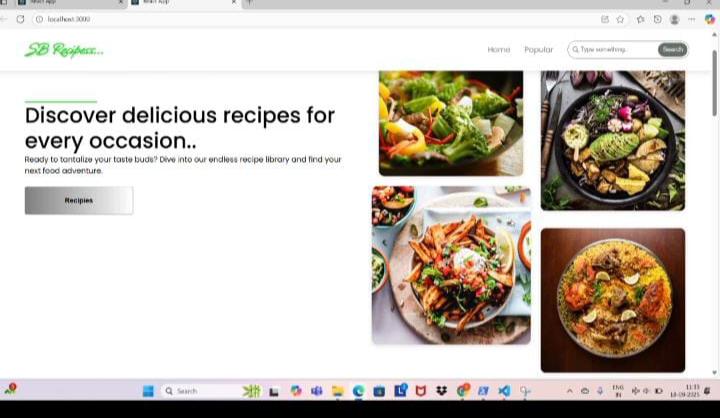
Testing

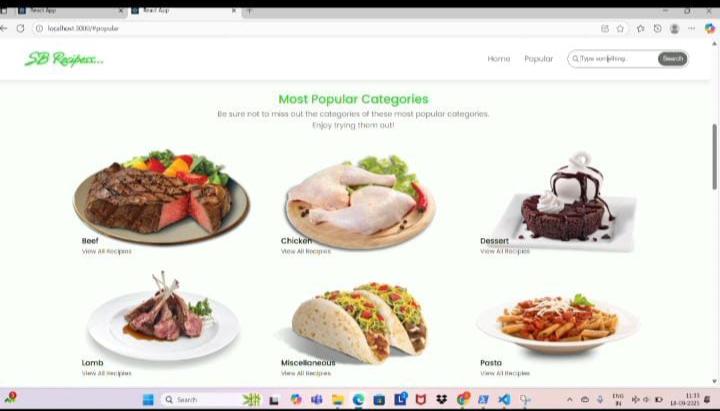
Testing Strategy: Components are tested using Jest and React Testing Library. The application is verified for rendering, navigation, and user interaction.

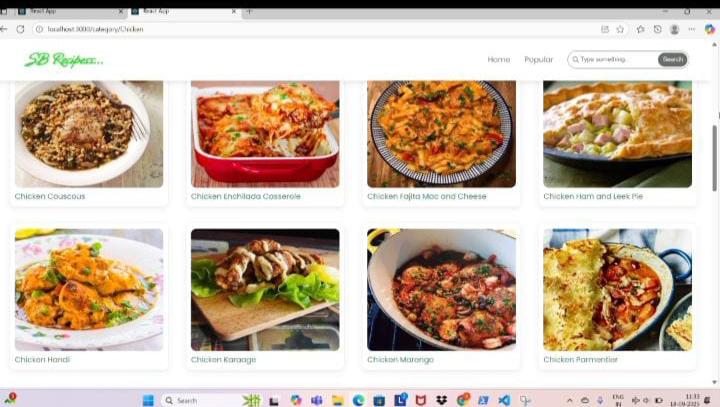
Code Coverage: Achieved using Jest coverage reports.

Screenshots or Demo

A demo video showcasing the application’s features is available. The video highlights recipe browsing, detailed instructions, and the favorites feature.







Known Issues

• Limited number of recipes in the current version

• No user authentication system implemented

Future Enhancements

• Add user authentication and profiles

• Enable recipe uploads by users

• Include video tutorials for recipes

• Enhance UI with animations and dark mode