

NAAN MUDHALVAN PROJECT

COOKBOOK-YOUR VITRUAL KITCHEN ASSISTANT

1. Introduction

Project Title: Cook book

Team ID: NM2025TMID47083

Team Leader: C.Kaviya

Email Id: pioc67789@gmail.com

Team members:

1.M.Nirmala-nishanirmal1121@gmail.com-Document making

2.S.Dharshini-adharshini740@gmail.com –Document making

3.V.Kaviya-kaviyav031@gmail.com -Demo video maker

4.S.Bharathi-bharathivel132007@gmail.com -Coding developer

2. Project Overview

Purpose:

- SmartCook is a React.js-based online cookbook application that allows users to browse,
- save, and share recipes. It aims to make cooking easier by providing detailed recipes,
- step-by-step instructions, and personalized recommendations.

Features:

- Browse recipes by category, cuisine, or ingredients
- Save favourite recipes to personal profile
- Add and edit user-submitted recipes
- Responsive UI for mobile and desktop

3. Architecture

Component Structure:

- Header (navigation bar)
- RecipeList (list of recipes)
- RecipeDetail (individual recipe page)
- Favourites (user's saved recipes)
- AddRecipeForm (form to add new recipes)
- Footer

State Management:

- Using React Context API to manage global state for recipes and user favourites.
- Routing:
- React Router v6 handles navigation between Home, Recipe Detail, favourites, and Add Recipe
- Pages

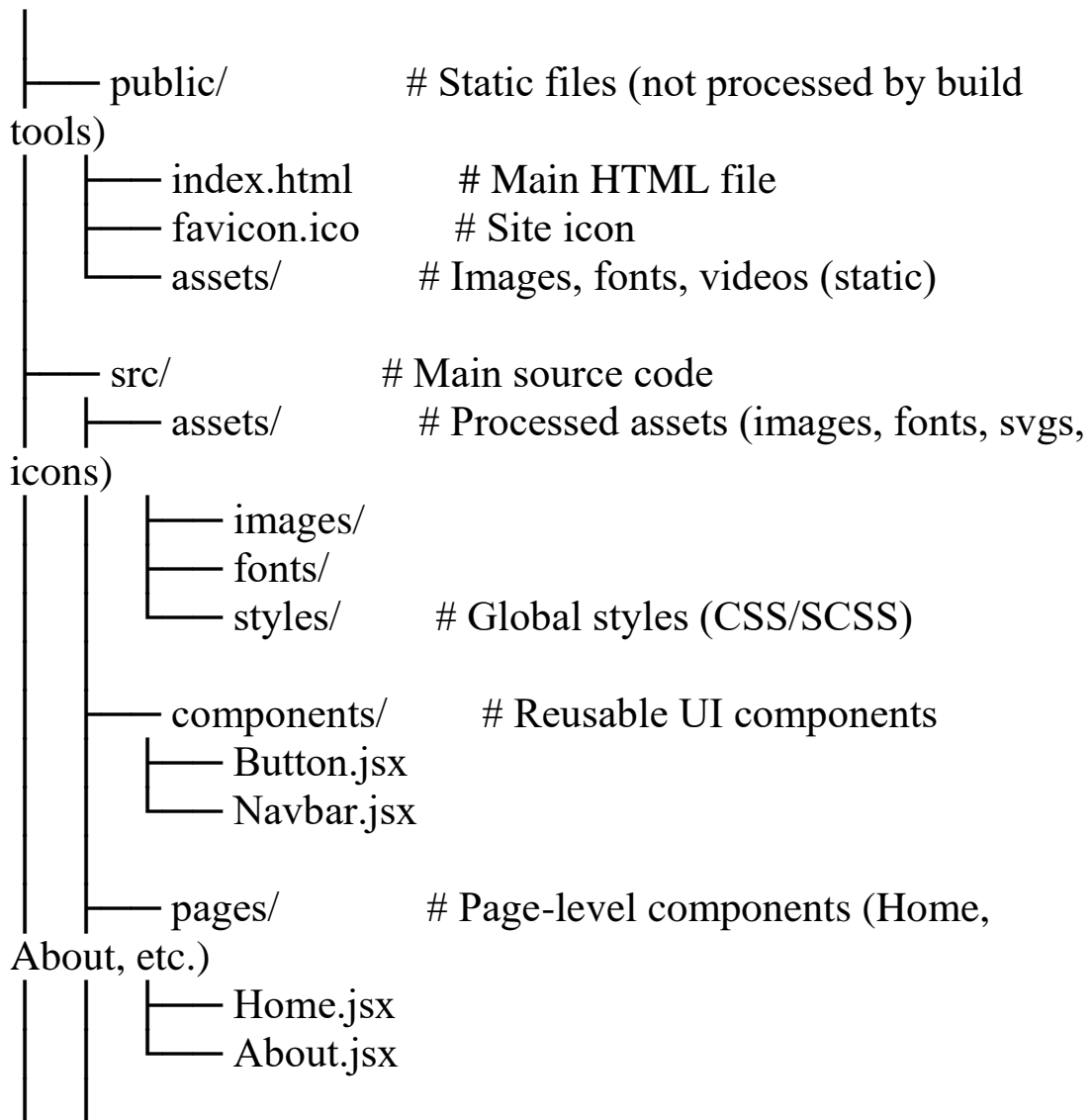
4. Setup Instructions

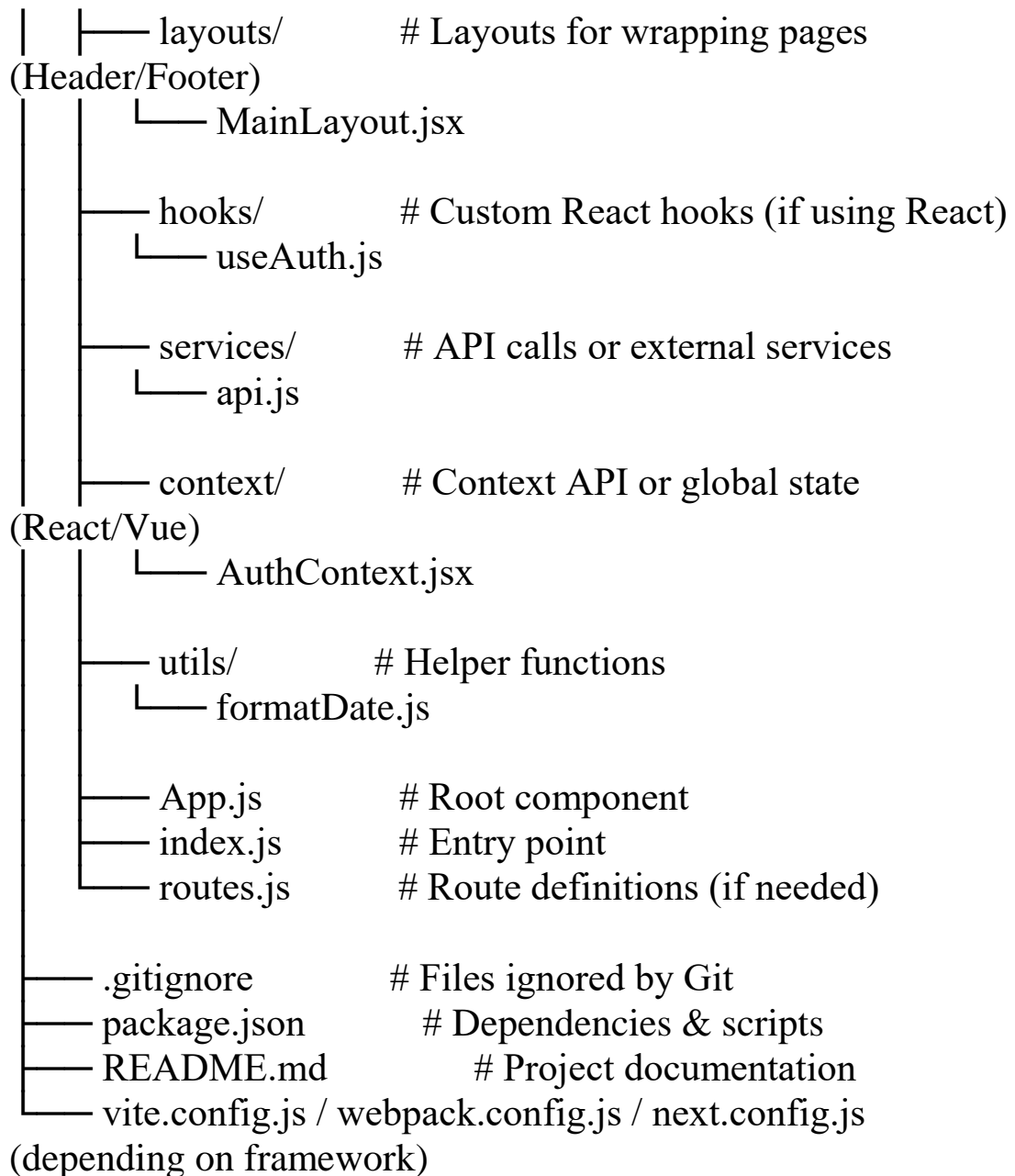
Prerequisites:

- Node.js
- MongoDB
- Git
- React.js
- Express.js – Mongoose – Visual Studio Code
- Installation Steps
- Clone the Repository

5. Folder Structure

Cookbook:virtual kitchen Assistant





6. Running the Application

- Frontend:cd client
- npm start
- Backend:cd server npm start
- Access: <http://localhost:3000>

7. Component Documentation

Key Components:

- RecipeList: Displays recipe cards fetched from API. Props: recipes[].
- RecipeDetail: Shows full recipe details. Props: recipeId.

Reusable Components:

- RecipeCard – used in RecipeList and Favorites.
- Button – styled button component used throughout the app

8. State Management

Global State:

- Managed by RecipesContext for recipes, favorites, and user login status.

Local State:

- Form input states managed inside AddRecipeForm.

9. User Interface

Include screenshots or GIFs of:

- Home page showing recipes
- Recipe detail page
- Adding a recipe

10. Styling

CSS Frameworks/Libraries:

- Tailwind CSS for styling; Styled Components for scoped styles.
- Theming:
- Dark and light mode toggle implemented via context.

11. Testing

Testing Strategy:

- Jest + React Testing Library for unit and integration tests.

Code Coverage:

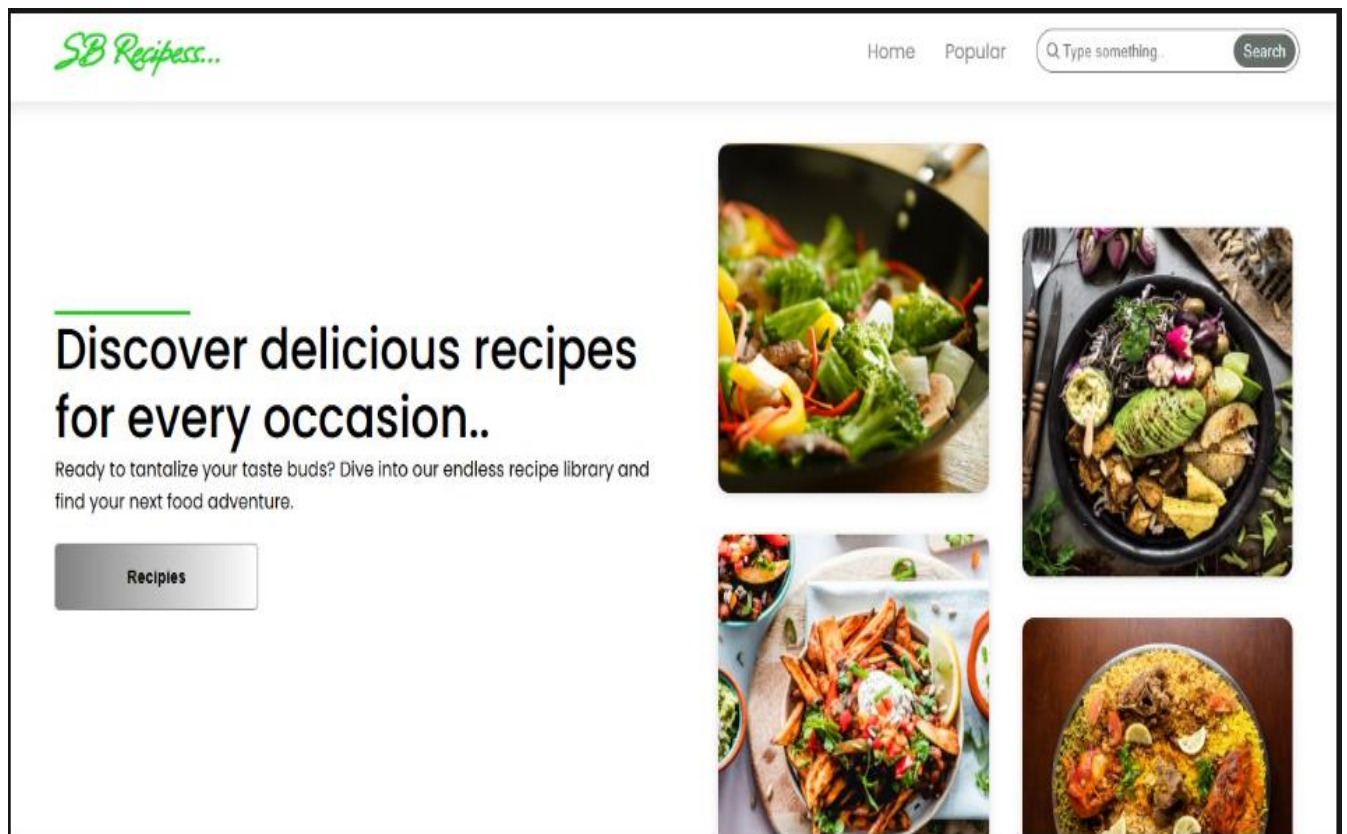
- Using jest --coverage to ensure 80%+ coverage

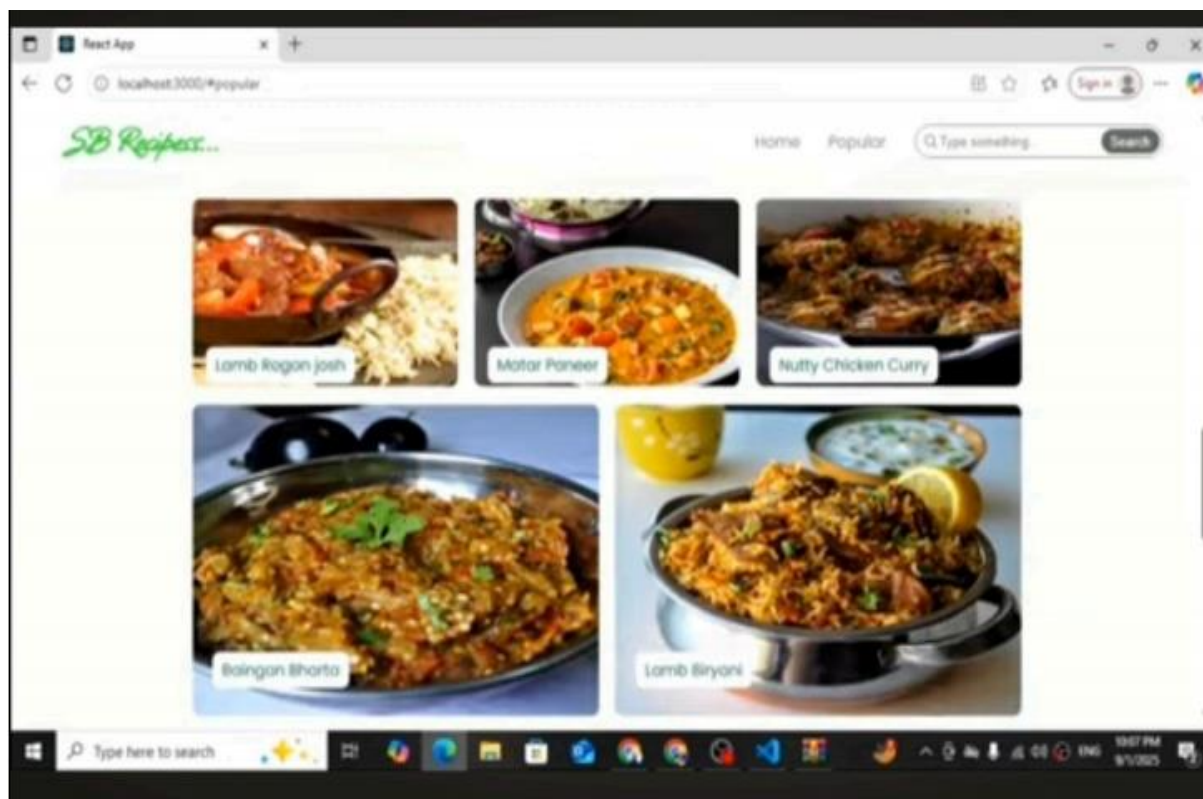
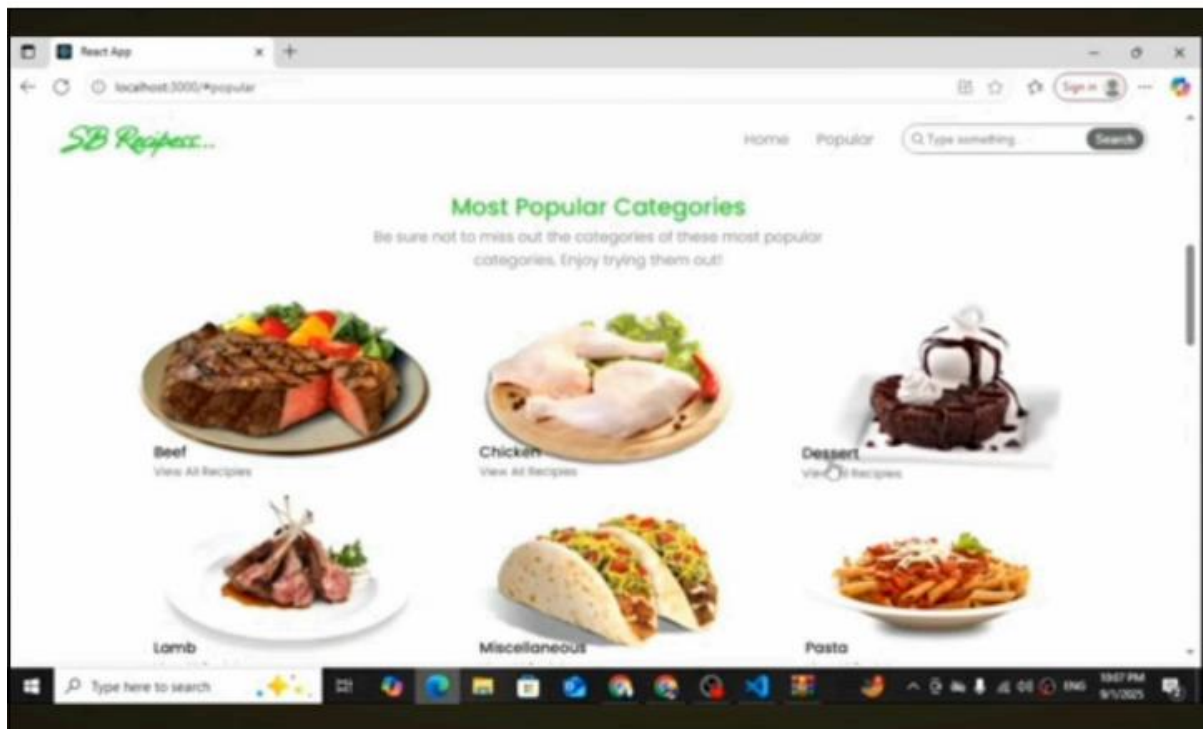
12. Screenshots or Demo

Add actual screenshots or a demo link:

- Demo Link:

https://drive.google.com/file/d/10Yb6uK8hyTln7H6gjF_OTA8pF9BXHY4F/view?usp=drivesdk





13. Known Issues

- Search filtering may be slow with very large recipe lists.
- Image upload sometimes fails on slow networks.

14. Future Enhancements

- Add shopping list generation from recipe ingredients
- Introduce push notifications for new recipes
- Offline mode for saved recipes