Naan Mudhalvan Project

Cook book -your virtual kitchen assistant

1. Introduction:

Project title: cook book

Team id :NM2025TMID41854

Team leader name:Balaji M

Mail id : [balajiboo132@gmail.com](mailto:balajiboo132@gmail.com)

Team members:

* Balaji M [–balajiboo132@gmail.com](mailto:–balajiboo132@gmail.com) - demo video making
* Maqbool ali [M –makbool940@gmail.com](mailto:M%20–makbool940@gmail.com) - code developer
* Kanakaraj [N–gnkraj06@gmail.com](mailto:N–gnkraj06@gmail.com) –documentation
* Selvakumar E – [selvakumarem2006@gmail.com](mailto:selvakumarem2006@gmail.com) --documentation

2.Project overview:

1. Purpose:

Smartcook is a react is-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 recommondations.

1. Features:

Browse recipes by category, cuisine,or ingredients

Save favorite recipes to personal profile

Add and edit uset -submitted recipes

Responsive UI for mobile and desktop

3.Architecure:

* 1. Component structure:
     + Header(navigation bar)
     + recipeList(list of recipes)
     + recipeDetail(individual recipe page)
     + favorites(user’s saved recipes)
     + addRecipeForm(form to add new recipes)
     + footer
  2. State Management:
     + using react context API to manage global state forrecipes and user favorites
  3. Routing:
     + react router v6 handles navigation betweenhome,recipe detail,favorites, and add recipe pages .

4.SetupInstructions:

* Prerequisites:
* Node.js
* MongoDB
* Git
* React.js
* Express.js-mongoose-visual studio code
* Installation steps
* clone the repository

5.Folder Structure :

1. Client:

* /src
* /components
* Header,js
* recipeList.js
* recipeDetail.js
* addRecipeFoem,js
* /pages
* Home.js
* Favorites.js
* /context
* Recipescontext.js
* /assets
* Images/

1. Utilities:
   * + - Api.js for fetching recipes from api
       - Helpers.js for common formatting functions

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 throught the app

8. State Management :

Global state:

* Managed by recipes Context for recipes favorites and user

login state:

* Form input states managed inside AddRecipeForm.

9. User Interface :

* Homepage showing recipes
* Recipe detailed 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:

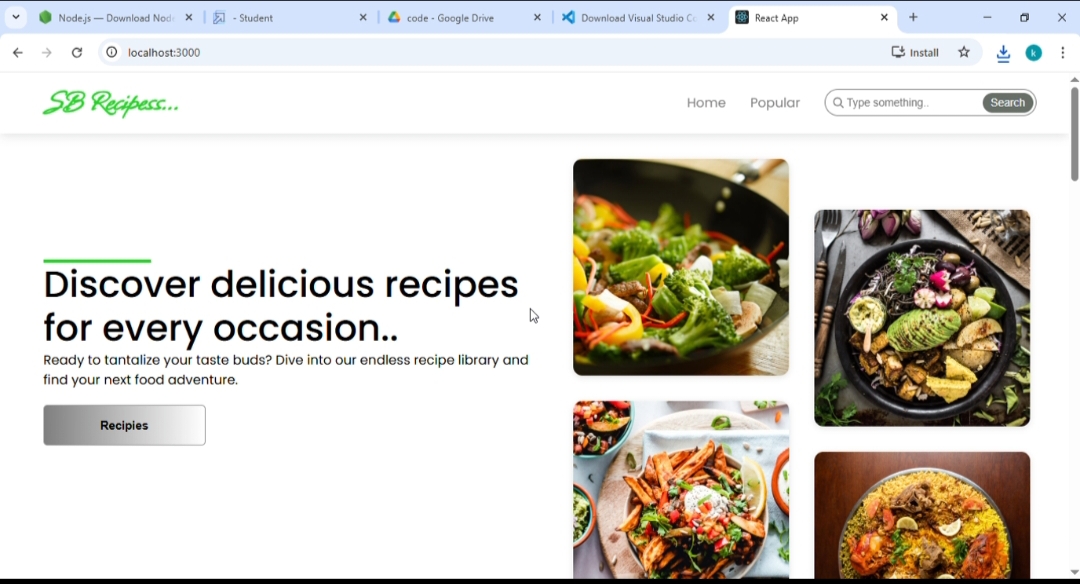
* jest+React Testing Library for unit and integration tests.

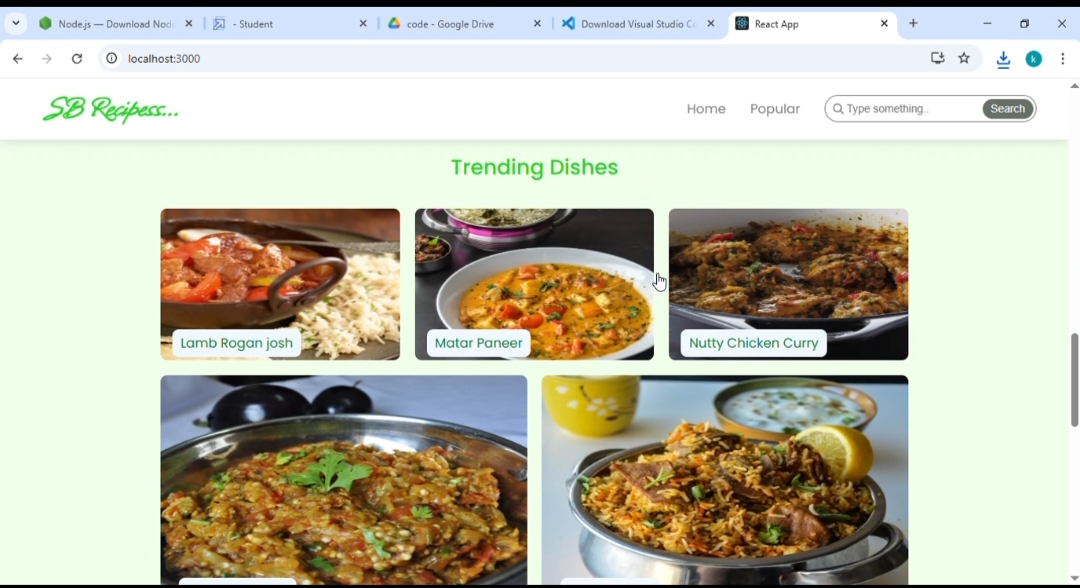
Code coverage:

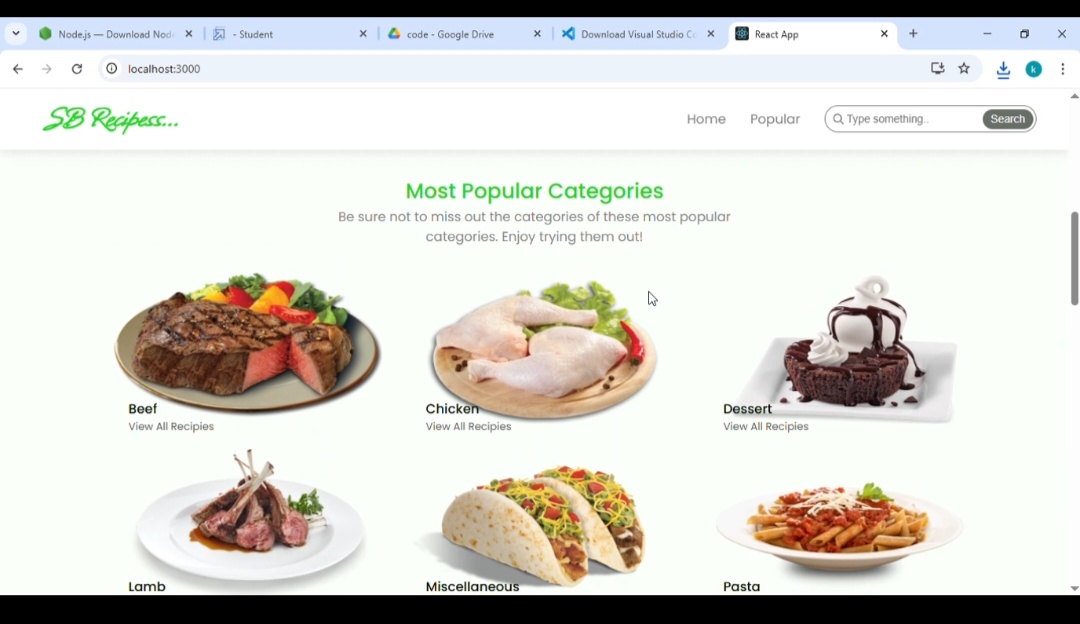
* Using jest – coverage to ensure B0%+coverage

12.Screenshots Or Demo:

* Add actual screenshots or a demolink:
* -demo link:https://drive.google .com/file/d/1akNbr4EqVQ9-Kcdfjn6Lbcxz1Q3Ga\_3m/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
* Ofline mode for saved recipes