NAAN MUDHALVAN PROJECT

COOKBOOK-YOUR VITRUAL KITCHEN ASSISTANT

1. Introduction

Project Title:Cook book

Team ID: NM2025TMID47124

Team Leader: M.Devi

Email Id: devimanokaran991@gmail.com

Team members:

1.S. Jeevitha - jeevitha 97891@gmail.com - Document Making

2. M.Devi - devimanokaran991@gmail.com -Document Making

3. S.Monisha - monishaselvam39@gmail.com -Demo Video Maker

4. M.Kanagavalli- kkanaga472@gmail.com -Coding Developer

2. Project Overview

Purpose:

- Smart Cook 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)
- Recipe List (list of recipes)
- Recipe Detail (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

```
public/
                  # Static files (not processed by build tools)
    index.html
                     # Main HTML file
                     # Site icon
   - favicon.ico
                   # Images, fonts, videos (static)
   - assets/
                 # Main source code
- src/
                   # Processed assets (images, fonts, svgs, icons)
    · assets/
       images/
       fonts/
       - styles/
                   # Global styles (CSS/SCSS)
                      # Reusable UI components
    components/
       - Button.jsx
       Navbar.jsx
```

```
pages/
                   # Page-level components (Home, About, etc.)
      - Home.jsx
       - About.jsx
   - layouts/
                   # Layouts for wrapping pages (Header/Footer)
    — MainLayout.jsx
   - hooks/
                   # Custom React hooks (if using React)
       - useAuth.js
                   # API calls or external services
   - services/
  L—api.js
   - context/
                   # Context API or global state (React/Vue)
  AuthContext.jsx
                 # Helper functions
   - utils/
  formatDate.js
                   # Root component
    App.js
   - index.js
                   # Entry point
    – routes.js
                    # Route definitions (if needed)
.gitignore
                  # Files ignored by Git
package.json
                    # Dependencies & scripts
                       # Project documentation
README.md
- vite.config.js / webpack.config.js / next.config.js (depending on framework)
```

6. Running the Application

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

7. Component Documentation

Key Components:

- Recipe List: Displays recipe cards fetched from API. Props: recipes [].
- Recipe Detail: Shows full recipe details. Props: recipe Id.

Reusable Components:

- Recipe Card used in Recipe List and Favourites.
- Button styled button component used throughout the app

8. State Management

Global State:

• Managed by Recipes Context for recipes, favourites, and user login status.

Local State:

• Form input states managed inside Add Recipe Form.

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.

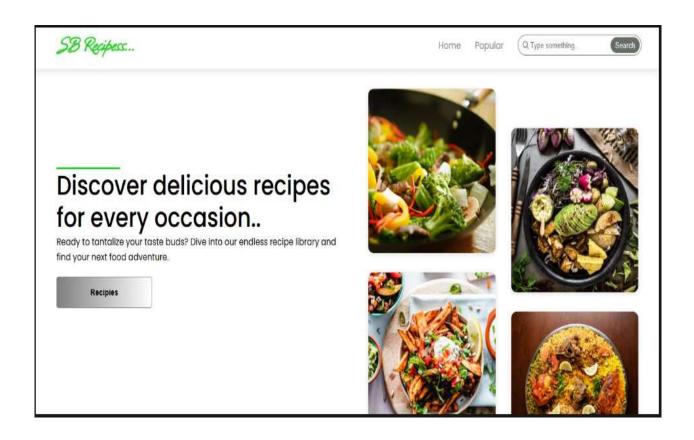
Code Coverage:

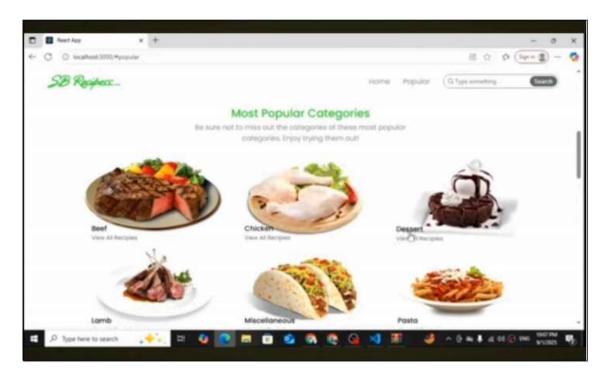
• Using jest --coverage to ensure 80%+ coverage

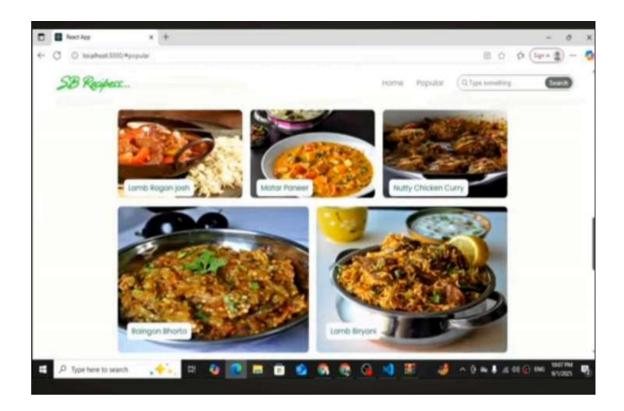
12. Screenshots or Demo

Add actual screenshots or a demo link:

• Demo Link: https://smartcook-demo.netlify.app







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