

Naan Mudhalvan

Cook book: your virtual kitchen assistant

Introduction project title: cook book your virtual kitchen assistant

Team ID: NM2025TMID41463

Team members:

Team leader name: K.YASHINIDEVI

Team member name : M SANTHIYA

Team member name: VIJAYALAKSHMI V

Team member name: KAVIYA R

Team member name : MONIKA P|

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 favorite 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)
- Favorites (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 favorites.

Routing:

React Router v6 handles navigation between Home, Recipe Detail, Favorites, and Add Recipe pages.

4. Setup Instructions

Prerequisites:

- Node.js
- MongoDB
- Git
- React.js
- Express.js - Mongoose - Visual Studio Code

Installation Steps

1.Clone the Repository

5. Folder Structure

Client:

/src

 /components

 Header.js

 RecipeList.js

 RecipeDetail.js

 AddRecipeForm.js

```
/pages
  Home.js
  Favorites.js
/context
  RecipesContext.js
/assets
  images/
```

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 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://smartcook-demo.netlify.app>

