Frontend Development with React.js

Project Documentation [Cookbook - Recipe Management App]

Introduction

Project Title: ARAHA Cookbook -
Recipe Management App

Team Members:

- Ashifa Nasreen N– Team Lead, Frontend Developer

- nasreenashifa07@gmail.com

- Asna Bai – Ideation, Documentation, Demonstration

- asnabai205@gmail.com

- Rakshatha B - API Integration Specialist

- rakshathab07@gmail.com

- Abinaya N – Execution Specialist, Branch Management.

- johnnaresh2531@gmail.com

- Haritha E – Environment supervisor.

- bindukutty331@gmail.com

Project Overview

Purpose:

- The purpose of a recipe is to provide a   structured guideline for preparing a dish, combining culinary science, creativity, and cultural context to ensure consistency, efficiency, and reproducibility in cooking.

- Ultimately, a recipe aims to communicate the culinary vision and intent of the recipe creator, conveying the desired flavors, textures, and presentation of a dish.

Features:

- Clear Title: A concise and descriptive title that indicates the name of the dish.
- Ingredient List: A detailed list of ingredients, including quantities and measurements.
- Instructions: A step-by-step guide to preparing and cooking the dish.
- Cooking Techniques: Specific cooking methods, such as baking, grilling, or sautéing.
- Serving Suggestions: Ideas for presenting and serving the finished dish.

Architecture

Component Structure:

1. Recipe Header: Title, serving size, and cooking time.

2. Ingredient List: Quantities and measurements of ingredients.

3. Instructions: Step-by-step cooking instructions.

4. Cooking Methods: Specific techniques, such as baking or grilling.

5. Nutritional Information: Calories, macronutrients, and other dietary details.

6. Additional Information: Variations, substitutions, and serving suggestions.

State Management: The application uses React Context API to manage global state, particularly for user preferences (such as dark mode) and API responses.

Routing: React Router is implemented for seamless navigation between different recipe categories and art details.

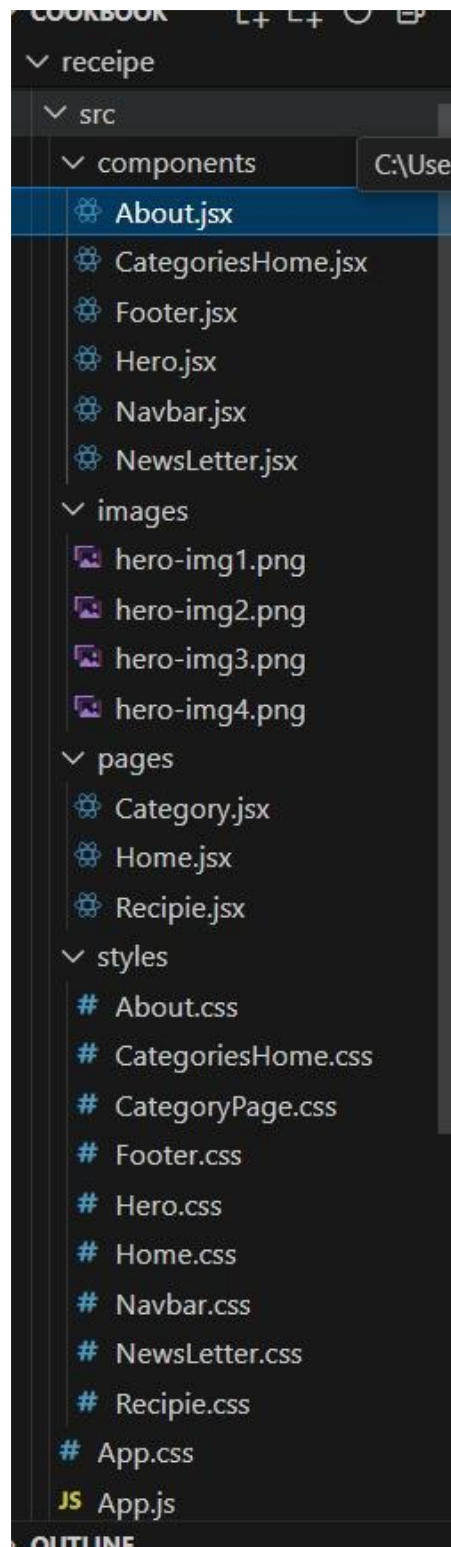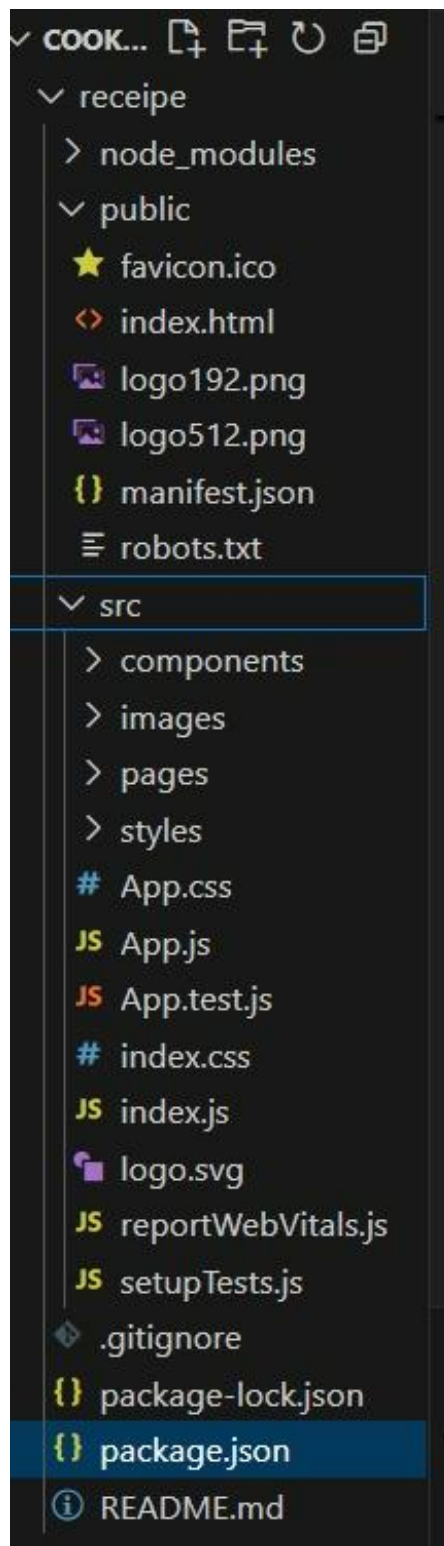Setup Instructions

Prerequisites:

Node.js (>=14.x)

npm or yarn package manager

Basic knowledge of React.js Installation:

```
# Clone the repository git clone
https://github.com/Ashifa007/cook_book.Ashifa/tree/main

# Navigate to the project directory cd SB Recipes

# Install dependencies npm install
```

∨ COOK... 🗋 🗂 ↺ 🗗

∨ receipe
  > node_modules
  ∨ public
    ★ favicon.ico
    <> index.html
    🖼 logo192.png
    🖼 logo512.png
    {} manifest.json
    ≡ robots.txt
  ∨ src
    > components
    > images
    > pages
    > styles
    # App.css
    JS App.js
    JS App.test.js
    # index.css
    JS index.js
    🔷 logo.svg
    JS reportWebVitals.js
    JS setupTests.js
  ◈ .gitignore
  {} package-lock.json
  {} package.json
  ⓘ README.md

COOKBOOK   🗋 🗋 ↺ 🗗

∨ receipe
  ∨ src
    ∨ components   C:\Use
      ⚛ About.jsx
      ⚛ CategoriesHome.jsx
      ⚛ Footer.jsx
      ⚛ Hero.jsx
      ⚛ Navbar.jsx
      ⚛ NewsLetter.jsx
    ∨ images
      🖼 hero-img1.png
      🖼 hero-img2.png
      🖼 hero-img3.png
      🖼 hero-img4.png
    ∨ pages
      ⚛ Category.jsx
      ⚛ Home.jsx
      ⚛ Recipie.jsx
    ∨ styles
      # About.css
      # CategoriesHome.css
      # CategoryPage.css
      # Footer.css
      # Hero.css
      # Home.css
      # Navbar.css
      # NewsLetter.css
      # Recipie.css
    # App.css
    JS App.js

> OUTLINE

Running the Application

To start the frontend server locally, use:

```
npm start
```

This will launch the development server at **http://localhost:3000**.

Component Documentation

Key Components:

1. Title: Name of the dish

2. Ingredients: List of required ingredients

3. Instructions: Step-by-step cooking instructions

4. Servings: Number of people the recipe serves

Reusable Components:

- Sauces and marinades
- Seasoning blends
- Dressings and vinaigrettes
- Garnishes and toppings
- Basic cooking techniques (e.g., roasting, grilling, sautéing)

State Management

Global State:

Recipes Context API stores fetched new dishes, user preferences, and API responses.

Local State:

use State hooks manage component-level UI interactions, such as loading states and search queries.

User Interface

1. **Search bar:** Easy search for specific recipes or ingredients.

2. **Recipe categories:** Organization by course, cuisine, dietary restriction, etc.

3**. Clear recipe formatting**: Easy-to-read instructions and ingredient lists.

4. **High-quality images**: Visual representation of the finished dish.

Styling

CSS Frameworks/Libraries:

- Tailwind CSS for rapid UI development.

- React Icons for enhanced UI elements.

Theming:

- Dark mode toggle using Context API.

- Responsive design using CSS media queries.

Testing

Testing Strategy:

- **Unit Testing:** Using Jest for testing individual components.

- **Integration Testing:** React Testing Library for simulating UI interactions.

Code Coverage:

- Ensuring 80%+ test coverage with Jest and React Testing Library.

Screenshots or Demo

https://drive.google.com/drive/folders/1uKk9OUWZFeLWNjddWFSOlihJWg2vbXm1
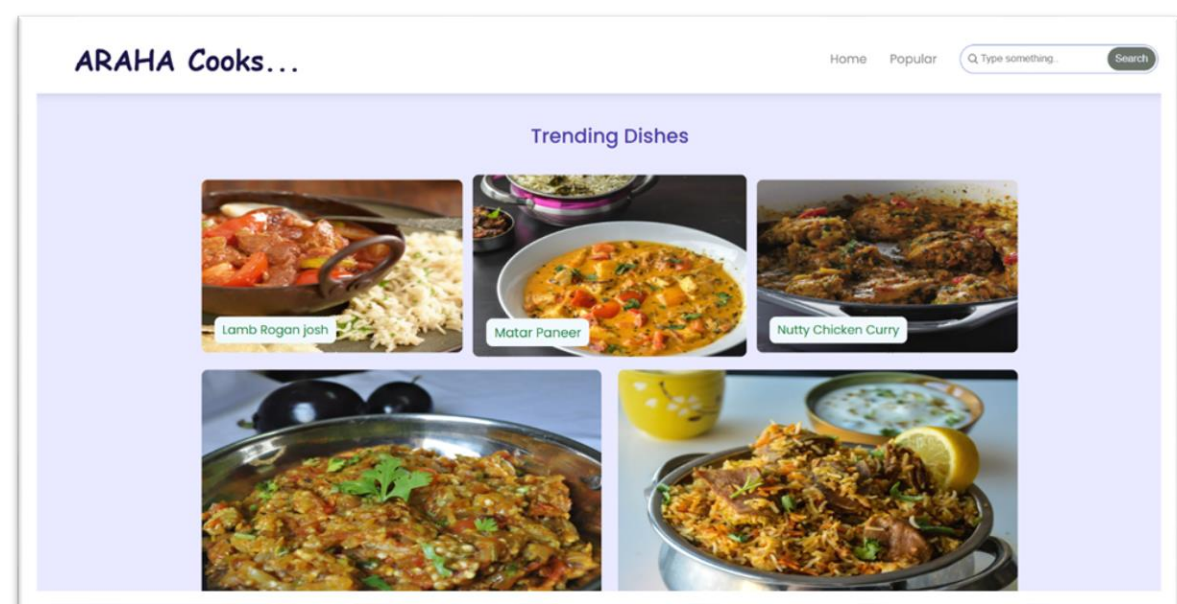
Discover delicious recipes for every occasion..

Ready to tantalize your taste buds? Dive into our endless recipe library and find your next food adventure.

Recipies

ARAHA Cooks...

Trending Dishes

Lamb Rogan josh

Matar Paneer

Nutty Chicken Curry

ARAHA Cooks...

Lamb Rogan josh

Indian   Lamb

Procedure

Put the onions in a food processor and whizz until very finely chopped. Heat the oil in a large heavy-based pan, then fry the onion with the lid on, stirring every now and then, until it is really golden and soft. Add the garlic and ginger, then fry for 5 mins more. Tip the curry paste, all the spices and the bay leaves into the pan, with the tomato purée. Stir well over the heat for about 30 secs, then add the meat and 300ml water. Stir to mix, turn down the heat, then add the yogurt. Cover the pan, then gently simmer for 40-60 mins until the meat is tender and the sauce nice and thick. Serve scattered with coriander, with plain basmati or pilau rice.

Video Tutorial

Lamb Rogan Josh recipe | Morrisons

Share

Ingredients

1 - Onion		2 quartered
2 - Sunflower Oil		4 tbsp
3 - Garlic		4 cloves
4 - Ginger		Thumb sized peeled and very finely grated
5 - Madras Paste		2 tbsp
6 - Paprika		2 tsp
7 - cinnamon stick		1
8 - Cardamom		6 bashed to break shells
9 - Cloves		4
10 - Bay Leaf		2
11 - Tomato Purée		1 tbsp
12 - Lamb		1kg cubed

ARAHA Cooks...

Unlock exclusive recipes, and foodie delights straight to your inbox.

Get weekly meal inspiration, cooking tips, and expert advice delivered directly to you. Subscribe to our newsletter and level up your culinary skills

✉ Your email address   Subscribe

⊘ We promise no spam, just yummy inspiration! Sign up now to Araha!

ARAHA Cooks...

Home	Dessert	Pasta	Vegan
Chicken	Goat	Seafood	Side
Breakfast	Lamb	Starter	Miscellaneous

```
<Routes>

  <Route path="/" element={<Home />} />
  <Route path="/category/:id" element={<Category />} />
  <Route path="/recipie/:id" element={<Recipie />} />
</Routes>
```

```
const [categories, setCategories] = React.useState([])

useEffect(() => {
  fetchCategories()
}, [])

const fetchCategories = async () => {
  await axios.get('https://www.themealdb.com/api/json/v1/1/categories.php')
    .then(response => {
      setCategories(response.data.categories)
      console.log(response.data.categories)
    })
    .catch(error => console.error(error));

}
```

```
const {id} = useParams();

const [items, setItems] = React.useState([])

useEffect(() => {
  fetchItems(id)
}, [window.location.href])

const fetchItems = async (idd) => {
  await axios.get(`https://www.themealdb.com/api/json/v1/1/filter.php?c=${idd}`)
    .then(response => {
      setItems(response.data.meals)
      console.log(response.data.meals)
    })
    .catch(error => console.error(error));

}
```

```
const {id} = useParams();

const [recipie, setRecipie] = React.useState()


useEffect(() => {
  fetchRecipie()
}, [])

const fetchRecipie = async () => {
  await axios.get(`https://www.themealdb.com/api/json/v1/1/lookup.php?i=${id}`)
    .then(response => {
      setRecipie(response.data.meals[0])
      console.log(response.data.meals[0])
    })
    .catch(error => console.error(error));
}
```

Known Issues

- API rate limits from recipeAPI.org can sometimes block requests.

- Occasional delays when fetching large datasets.

Future Enhancements

- Implement a bookmarking feature for saving favourite recipes.

- Add user authentication for personalized recipe feeds.

- Introduce push notifications for new recipes.

- Expand API sources for a richer new cooking experience.