Cookbook Documentation

📝 Introduction

Project Title**:**

**cook book**

Team Leader : J.AKSHAYA

Team member 1: S.DEVI PRIYA

Team member 2: S.ATCHAYA

Team member 3: J.INIYA

Team member 4: S.KANIMOZHI

---

🎯 Project Overview

**Purpose:**

A web-based cookbook where users can browse, search, and add recipes. “The primary aim of this project is to provide a single platform containing various recipes and cooking methods.

It helps users quickly find dishes based on ingredients, cuisine type, or occasion, making cooking simple, fun, and accessible to everyone.”

**Goals:**

1. Provide an easy-to-use recipe search and filter system. The importance of the Cook Book project lies in its ability to promote healthy eating, time management, and culinary creativity.

In today’s busy world, many people struggle to plan meals or try new dishes.

This cookbook serves as a reliable source of recipes that are easy to follow, helping both beginners and experienced cooks prepare tasty meals.”

2. Allow users to contribute new recipes.

This project is extremely useful because it:

Offers step-by-step guidance to prepare different dishes.

Encourages users to try new cuisines and flavors.

Reduces dependency on expensive cookbooks or random online searches.

Supports healthy and budget-friendly meal planning.

Whether you are a beginner learning to cook or an expert looking for new ideas, this cookbook is a helpful companion.”

3. Create a responsive, visually appealing UI.

**Key Features**:

1. Recipe listing with images

2.Search & filter by category (e.g., breakfast, dinner, desserts)

3.Add/Edit recipes (form)

4.Favorite recipes list

5.Responsive design

---

🏗️ Architecture

Component Structure:

App.js – Main entry point

Navbar – Navigation bar

RecipeList – Displays all recipes

RecipeCard – Individual recipe display

RecipeForm – Add/Edit recipes

Favorites – User’s saved recipes

State Management (Context/Redux):

Context API for global recipe and favorites state

Routing:

React Router (/, /recipes/:id, /add, /favorites)

---

⚙️ Setup Instructions

Prerequisites (Node.js, npm, etc.):

Node.js >= 16

npm or yarn

1. Install dependencies

cd recipe-book-app

npm install

2. Configure environment variables

Create .env file

Add API\_URL if using backend

---

📂 Folder Structure

Client (components, pages, assets):

src/

├── components/

│ ├── Navbar.js

│ ├── RecipeCard.js

│ └── RecipeForm.js

├── pages/

│ ├── Home.js

│ ├── Favorites.js

│ └── RecipeDetail.js

├── assets/

│ ├── images/

│ └── styles/

├── context/

│ └── RecipeContext.js

Utilities (hooks, helpers):

useFetch.js for API calls

formatDate.js helper

---

▶️ Running the Application

Start frontend:

npm start

---

🧩 Component Documentation

Key Components (with props):

RecipeCard({title, image, ingredients, steps}) → Displays recipe info

RecipeForm({onSubmit}) → Adds new recipes

Reusable Components:

Button, Input, Modal

---

🔄 State Management

Global State:

Recipes, Favorites handled by RecipeContext

Local State:

Form input states inside RecipeForm

UI states (modal open/close, search query)

---

🎨 User Interface & Styling

Screenshots / UI examples: (Insert app screenshots here)

CSS/Styling Libraries:

Tailwind CSS

Styled-Components for custom

Theming:

Light/Dark mode toggle

---

🧪 Testing

Strategy (Unit, Integration, E2E):

Unit tests for components

Integration tests for context and routing

E2E with Cypress

Tools (Jest, RTL, Cypress, etc.):

Jest

React Testing Library

Cypress

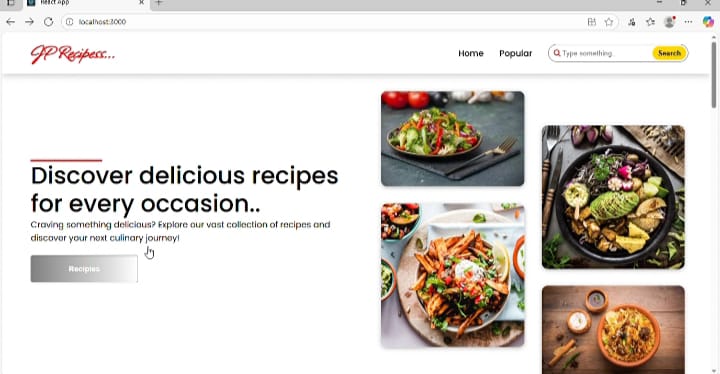
---

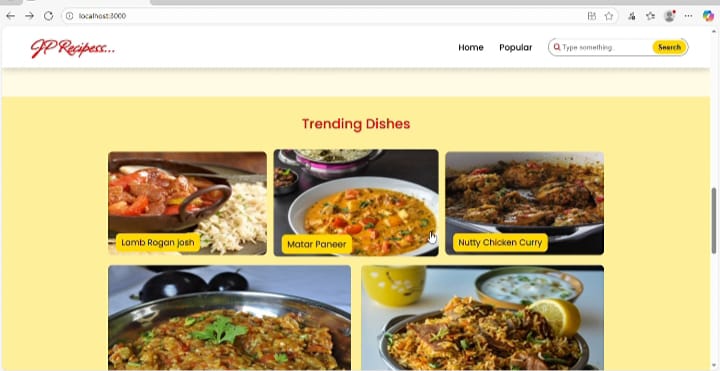
📸 Screenshots or Demo

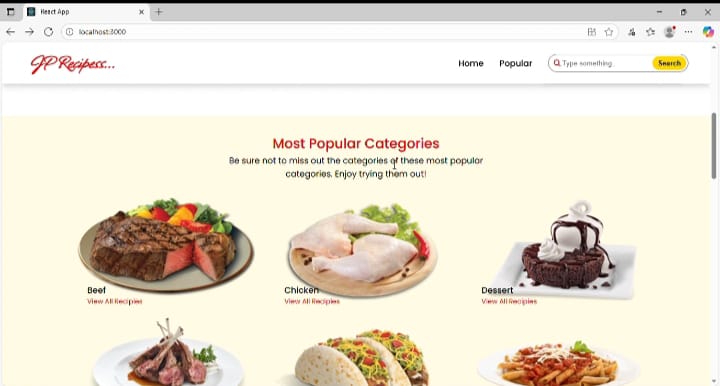
Live Demo:

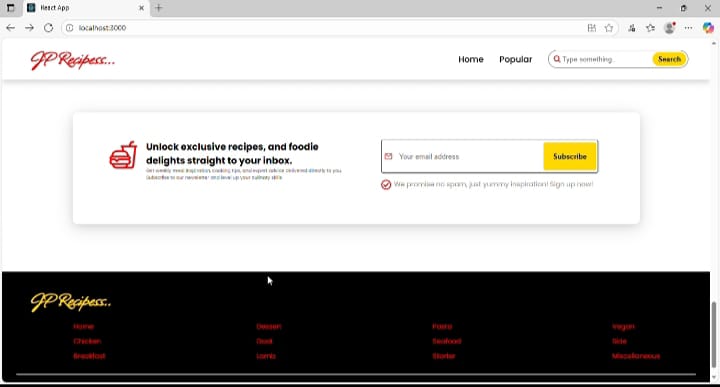
<https://drive.google.com/file/d/1IcrCIoaUhhKp54E290eihFjpnxOXvhTw/view?usp=sharing>

Screenshots: (insert images)









---

🐞 Known Issues

Recipes reset on page refresh (no backend persistence yet)

Limited search functionality

---

🚀 Future Enhancements

User authentication (login/signup)

Backend integration (Node.js/Express + MongoDB)

Advanced filtering (dietary preferences, cooking time)

Recipe rating & comments