Cookbook Documentation (React Project)

📝 Introduction

Project Title:COOKBOOK

Team Members & Roles:

Team leader : C.Archana

Member 1: N.Indra devi

Member 2: R.Jeevitha

Member 3: R.Gobika

---

🎯 Project Overview

Purpose:

A web-based cookbook where users can browse, search, and add recipes.

Goals:

1. Provide an easy-to-use recipe search and filter system.

2. Allow users to contribute new recipes.

3. Create a responsive, visually appealing UI.

Key Features:

Recipe listing with images

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

Add/Edit recipes (form)

Favorite recipes list

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

Installation Steps:

1. Clone repo

git clone <https://github.com/archanasarjana/cook_book/settings/pages>

2. Install dependencies

cd recipe-book-app

npm install

3. 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)

Tailwind CSS

Styled-Components for custom UI

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

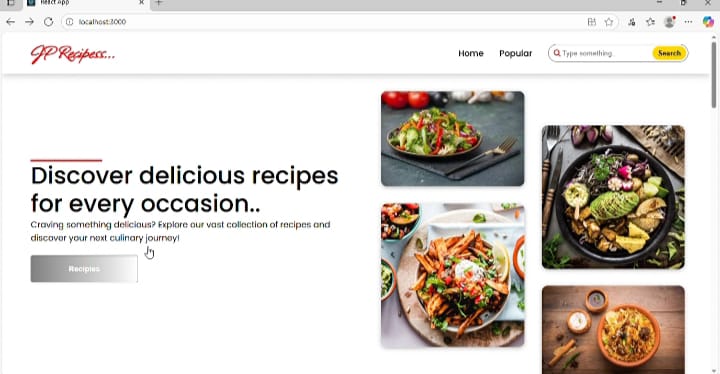
Cypress

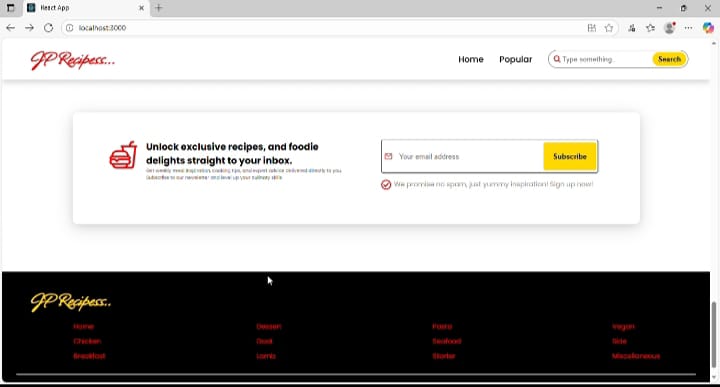
---

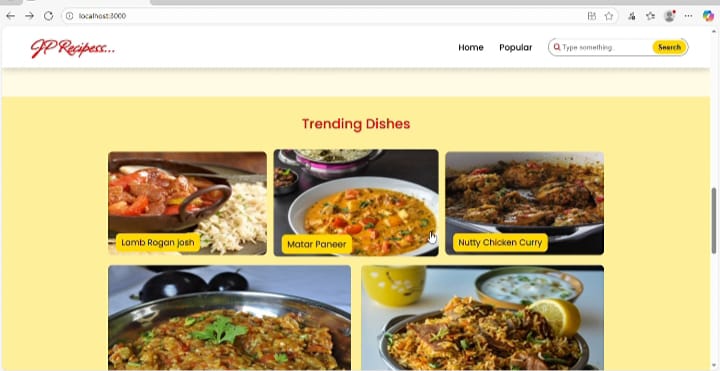
📸 Screenshots or Demo

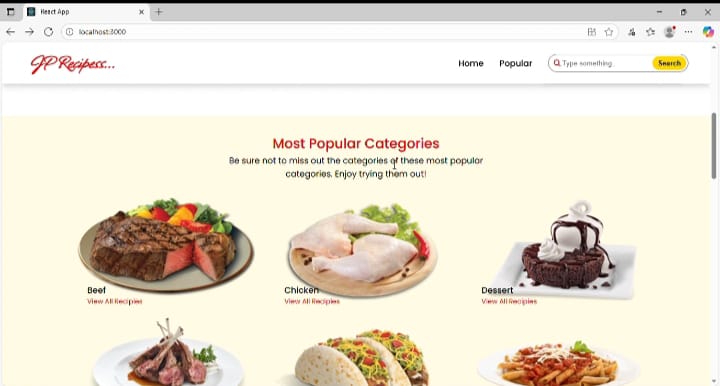
Live Demo: https://drive.google.com/file/d/1aGLhUCO\_uJK9MJ2ruW768O3QjiWlT9HF/view?usp=sharing

Screenshots:









---

🐞 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