

Frontend Development with React.js

Great Recipes Documentation .

1. Introduction

- Project Title: Great Recipes
- Team Members:
 - Dhanush Kumar M - Team Leader
 - Ajay Daniel B
 - Gokulan S
 - Danush SV
 - Prince Martin S

2. Project Overview

- Purpose: The Great Recipes project aims to provide users with a platform to discover, share, and manage a variety of recipes. The goal is to create an engaging and user-friendly experience for cooking enthusiasts.
- Features:
 - User authentication and profile management
 - Recipe search and filtering options
 - User-generated content for sharing recipes
 - Rating and commenting system for recipes
 - Responsive design for mobile and desktop users

3. Architecture

- Component Structure:
 - Major components include:
 - App
 - Navbar
 - RecipeList
 - RecipeDetail
 - UserProfile

- Components interact through props and state management.
- State Management:
 - The project uses Redux for global state management to handle user authentication and recipe data.
- Routing:
 - React Router is used to manage navigation between different pages such as Home, Recipe Detail, and User Profile.

4. Setup Instructions

- Prerequisites:
 1. Node.js (version 14 or higher)
 2. npm (Node package manager)
- Installation:
 1. Clone the repository: `git clone https://github.com/Gokulan08/Great-Recipes.git`
 2. Navigate to the project directory: `cd Great-Recipes`
 3. Install dependencies: `npm install`
 4. Configure environment variables as needed.

5. Folder Structure

- Client:
 - The React application is organized as follows:
 - `src/components`: Contains reusable components.
 - `src/pages`: Contains page components.
 - `src/styles`: Contains CSS files for styling.
 - `src/images`: Contains image assets.
- Utilities:
 - Helper functions and custom hooks are located in `src/utils`.

6. Running the Application

- Commands:
 - To start the frontend server locally, run: `npm start` in the client directory.

7. Component Documentation

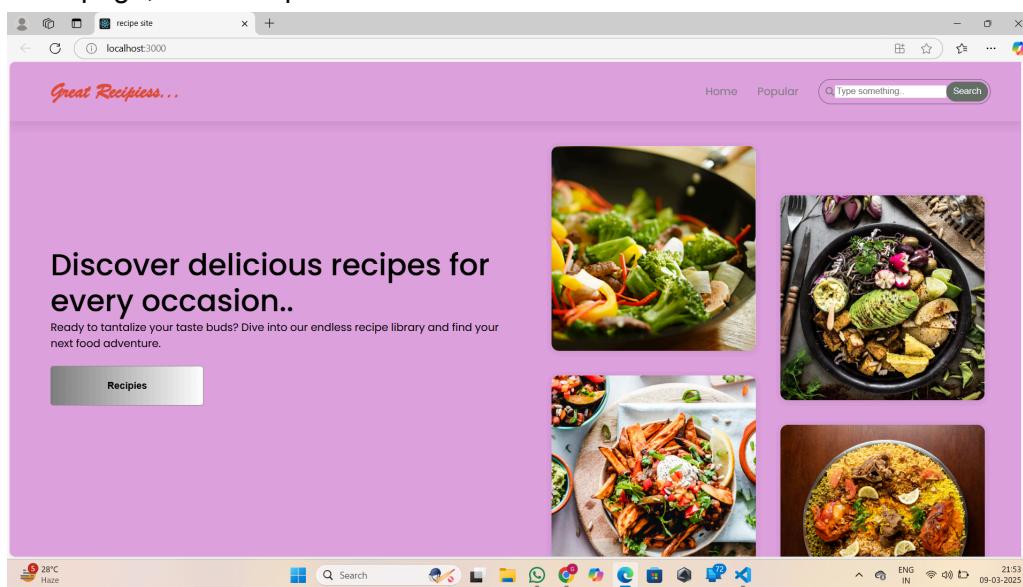
- Key Components:
 - App: The main component that renders the application.
 - Navbar: Navigation bar for routing.
 - RecipeList: Displays a list of recipes.
 - RecipeDetail: Shows detailed information about a selected recipe.
- Reusable Components:
 - Button: A reusable button component that accepts props for customization.

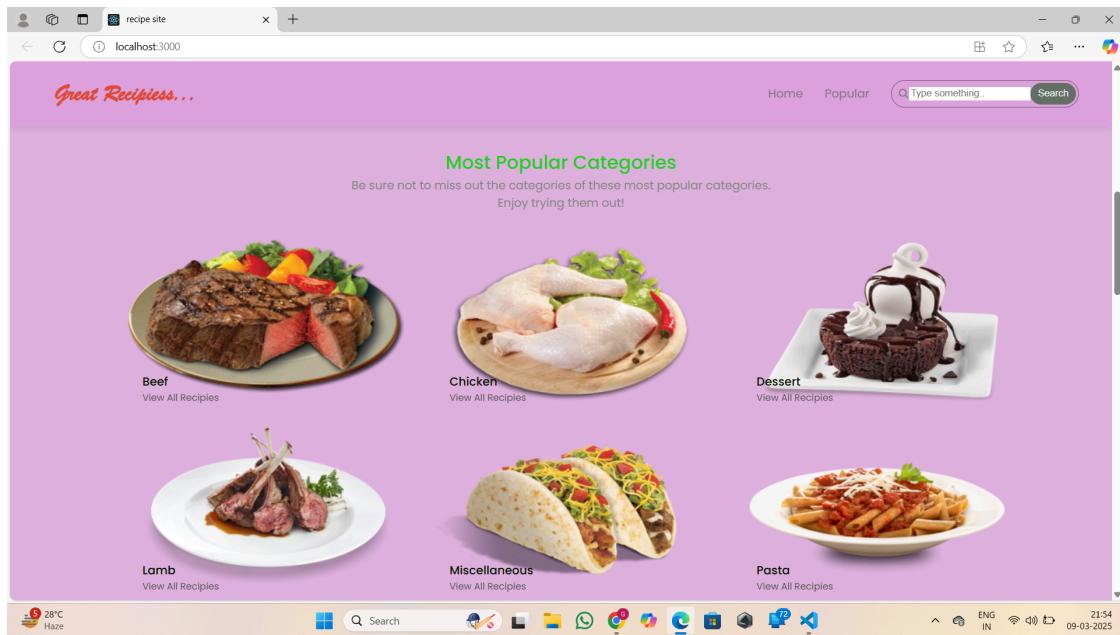
8. State Management

- Global State:
 - Redux is used to manage the global state, allowing for easy access to user and recipe data across components.
- Local State:
 - Local state is managed using React's useState hook within individual components.

9. User Interface

- Visuals: Screenshots of showcasing the UI features, such as the homepage, recipe detail page, and user profile.





10. Styling

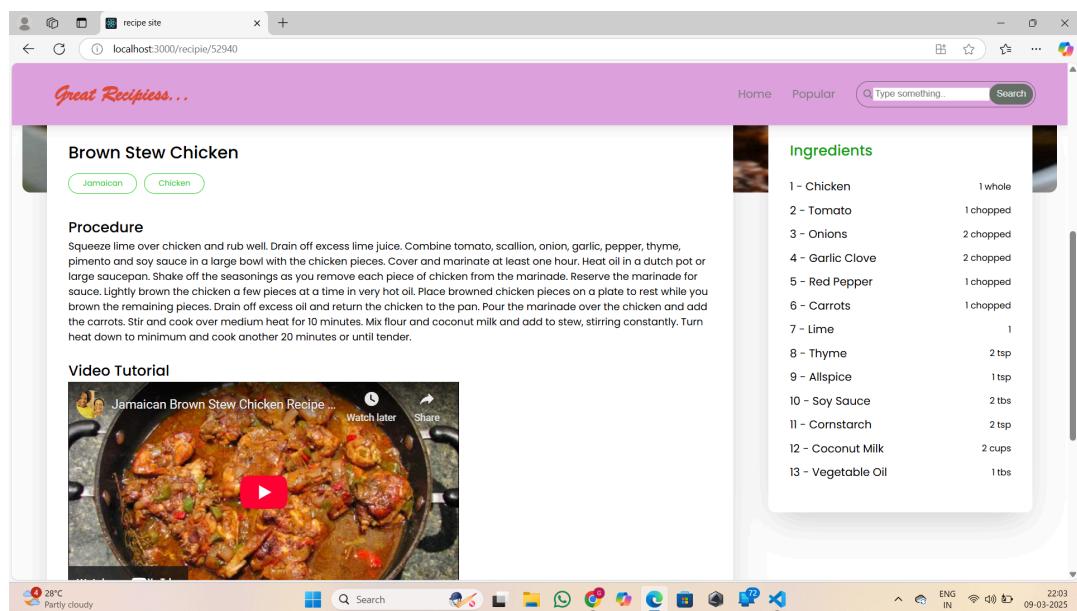
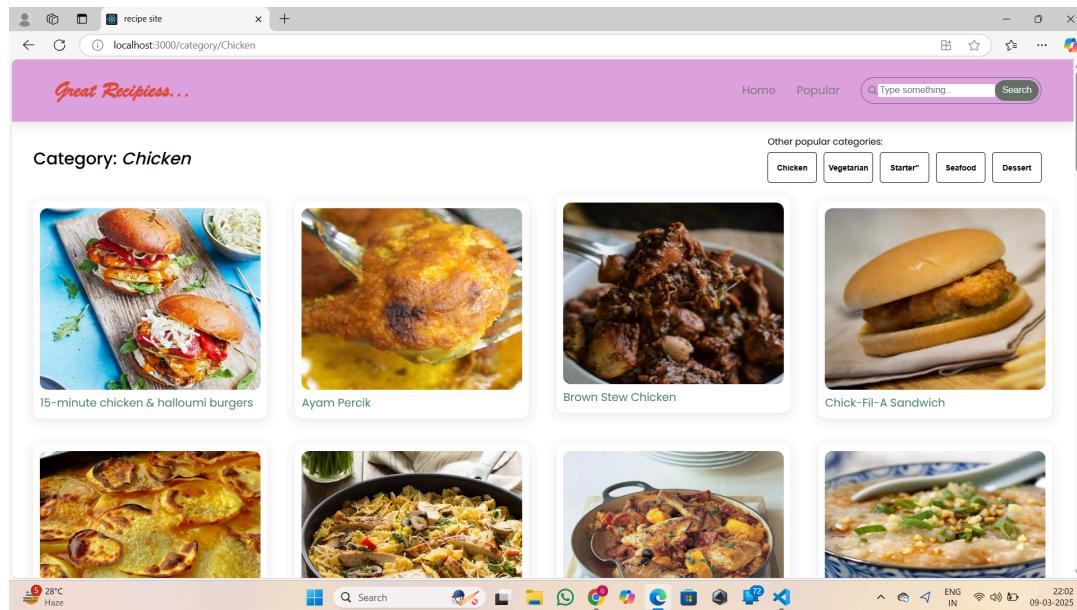
- CSS Frameworks/Libraries:
 - The project uses Styled-Components for styling components.
- Theming:
 - A custom design system is implemented to maintain consistency across the application.

11. Testing

- Testing Strategy:
 - The project employs unit testing using Jest and React Testing Library for component testing.
- Code Coverage:
 - Code coverage is monitored using Jest's built-in coverage tool.

12. Screenshots

- Visuals: Provide screenshots or a link to a demo showcasing the application's features and design.



13. Known Issues

- Bugs:
 - Document any known bugs or issues that users or developers should be aware of, such as:
 - Recipe search may return incomplete results.
 - User profile updates may not reflect immediately.

14. Future Enhancements

- Improvements:
 - Potential future features include:
 - Adding a meal planning feature.
 - Implementing a shopping list generator based on selected recipes.
 - Enhancing the user interface with animations and transitions.