

# FRONTEND DEVELOPMENT WITH REACT.JS – PROJECT DOCUMENTATION

## 1. INTRODUCTION

**Project Title:** Cook Book – Your Virtual Kitchen Assistant

**Team Members:**

[P. Paveethra] – Frontend Developer

[Y. Gayathri] – UI/UX Designer

[V. Kamali] – State Management & Integration

[S. Parameshwari] – Testing & Deployment

## 2. Project Overview

**Purpose:**

The project aims to provide a virtual kitchen assistant that helps users explore, save, and manage recipes. It focuses on delivering an intuitive UI and seamless user experience.

**Features:**

- Recipe browsing with categories & filters.
- Search functionality with suggestions.
- User authentication (login/signup).
- Recipe saving and favourites list.
- Responsive UI for mobile and desktop

## 3. Architecture

**Component Structure:**

**App.js:** Root component with routing

**Header, Footer, Navbar:** Layout components

**Recipe List, Recipe Detail:** Feature components

**Auth:** Login/Signup components

**Favourites:** User's saved recipes

**State Management:**

- Implemented with Context API for global state (user, theme, favorites).
- Local state handled using React's useState and useReducer.

**Routing:**

react-router-dom for page navigation.

Routes: /, /recipes, /recipe/:id, /login, /favorites.

## 4. Setup Instructions

**Prerequisites:**

Node.js (v16 or later)

npm or yarn

**Installation:**

git clone <https://github.com/your-repo/cook-book.git>

cd cook-book/client

npm install

**Environment Variables:**

**Create a .env file in the root:**

REACT\_APP\_API\_URL=http://localhost:5000/api

REACT\_APP\_AUTH\_KEY=your-secret-key

## 5. Folder Structure

```
client/  
  public/  
  src/  
    assets/      # Images, icons, styles  
    components/  # Reusable UI components  
    pages/       # Page-level components  
    context/     # Context API files  
    hooks/       # Custom hooks  
    utils/       # Helper functions  
  App.js  
  index.js  
  package.json
```

### Utilities:

**api.js:** API call handler

**useAuth.js:** Custom hook for authentication

**useFetch.js:** Data fetching hook

## 6. Running the Application

### Frontend (React):

```
cd client
```

```
npm start
```

## 7. Component Documentation

### Key Components:

**Recipe Card:** Displays recipe preview (props: title, image, time).

**Recipe Detail:** Shows recipe details (props: id).

**Auth Form:** Handles login & signup.

### Reusable Components:

**Button:** Configurable button with variants.

**Input:** Reusable input with validation props.

## 8. State Management

### Global State:


Managed with Context API (AuthContext, RecipeContext).

### Local State:

useState for UI-level states (modals, toggles).

useReducer for managing complex form states.

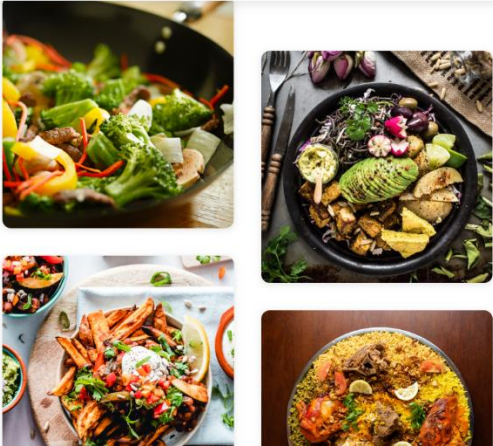
## 9. User Interface


Home PopularSearch


### Discover delicious recipes for every occasion..

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

[Recipes](#)



Home PopularSearch



#### White chocolate creme brulee

[French](#)[Dessert](#)


**Procedure**


Heat the cream, chocolate and vanilla pod in a pan until the chocolate has melted. Take off the heat and allow to infuse for 10 mins, scraping the pod seeds into the cream. If using the vanilla extract, add straight away. Heat oven to 160C/fan 140C/gas 3. Beat yolks and sugar until pale, stir in the chocolate cream. Strain into a jug and pour into ramekins. Place in a deep roasting tray and pour boiling water halfway up the sides. Bake for 15-20 mins until just set with a wobbly centre. Chill in the fridge for at least 4 hrs. To serve, sprinkle some sugar on top of the brulees and caramelise with a blowtorch or briefly under a hot grill. Leave caramel to harden, then serve.

**Ingredients**

1 - Double Cream	568ml
2 - White Chocolate Chips	100g
3 - Vanilla	Pod of
4 - Egg Yolks	6
5 - Caster Sugar	2 tbs
6 - Caster Sugar	Top


**Video Tutorial**

How to make White Chocolate Creme B...Watch laterShare


Home PopularSearch

### Most Popular Categories


Be sure not to miss out the categories of these most popular categories.  
Enjoy trying them out!




**Beef**  
[View All Recipes](#)




**Chicken**  
[View All Recipes](#)




**Dessert**  
[View All Recipes](#)



**Lamb**  
[View All Recipes](#)



**Miscellaneous**  
[View All Recipes](#)



**Pasta**  
[View All Recipes](#)

## 10. Styling

- CSS Frameworks/Libraries:
- Tailwind CSS for utility-first styling.
- Styled Components for dynamic styles.

### Theming:

Light/Dark mode toggle implemented using Context.

## 11. Testing

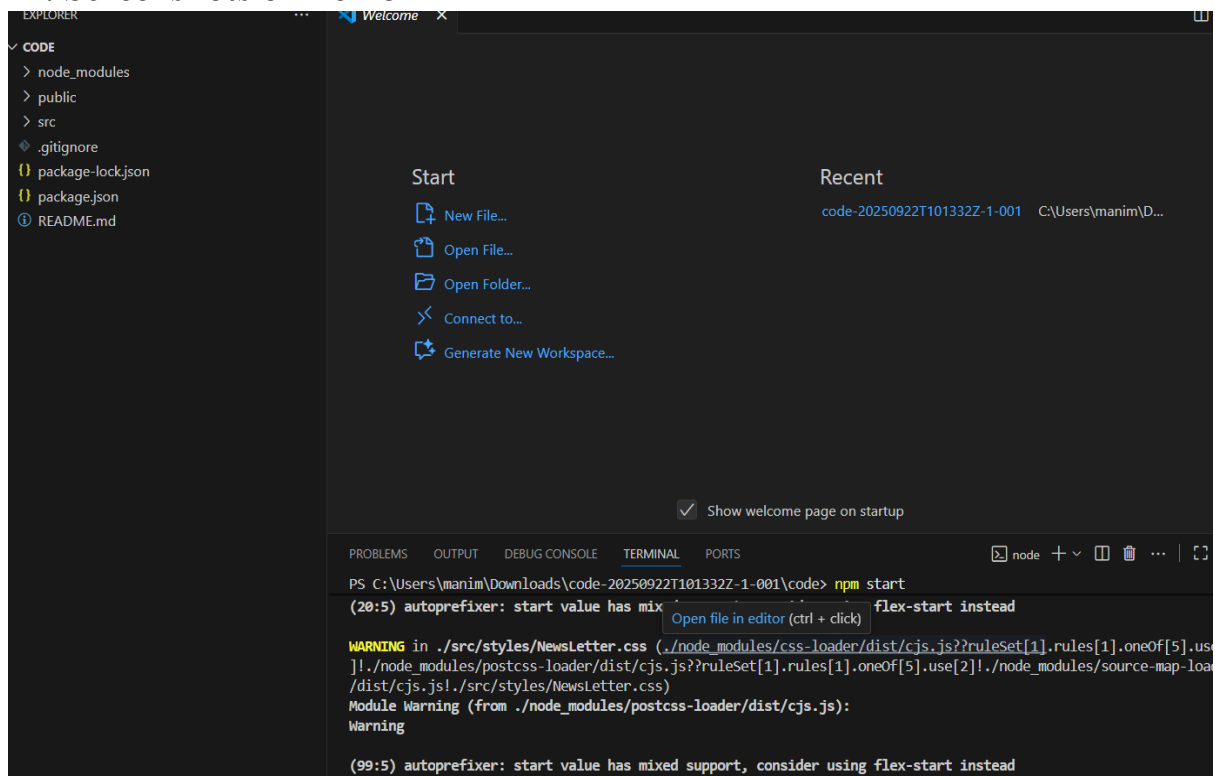
### Testing Strategy:

- Unit Testing with Jest + React Testing Library
- Snapshot Testing for components
- End-to-End Testing with Cypress

### Code Coverage:

Configured with Jest coverage reports.

## 12. Screenshots or Demo



Demo Link - [https://drive.google.com/file/d/11vqyKD66vQ3Vnkd\\_HvRJ6ybX-2gUCTx8/view?usp=drivesdk](https://drive.google.com/file/d/11vqyKD66vQ3Vnkd_HvRJ6ybX-2gUCTx8/view?usp=drivesdk)

### **13. Known Issues**

- Recipe API sometimes returns incomplete data.
- Favourites list not persisting after refresh without backend integration.

### **14. Future Enhancements**

- Add voice-based recipe assistant.
- Integration with meal planners & shopping lists.
- Push notifications for saved recipes.
- Advanced filtering (by calories, cuisine, etc.).