



Frontend Development with React.js


Project Documentation for cook book


1. Introduction

- **Project Title:** Cook book
- **Team Members:**

 **Chandru S (Team Leader)** [Email Id: mschandrau@gmail.com]

 **Harisiva L** [Email Id: lharisiva1010@gmail.com]

 **Lakshmikanthan M** [Email Id: lakshmikanthanm275@gmail.com]

 **Kishon raj L** [Email Id: kishonraj087@gmail.com]



2. Project Overview

Purpose

The Recipe Application serves as a comprehensive platform for food enthusiasts to:

- Discover new recipes across various categories
- Search for specific recipes
- Save favorite recipes
- Access detailed cooking instructions
- Share recipes with others

Key Features:

- Recipe categorization and browsing
- Advanced search functionality
- User authentication and profiles
- Favorite recipe management
- Responsive design with 3D animations
- Dark/Light theme support
- Category-based recipe filtering
- Recipe detail views with ingredients and instructions
- Loading states and error handling
- Smooth scrolling and animatio

3. Architecture

Technology Stack

- Frontend: React.js
- Routing: React Router v6
- State Management: React Hooks + Context API
- API Integration: Axios
- Styling: CSS Modules + Custom Animations
- Authentication: JWT
- API: TheMealDB + Spoonacular

Routing Structure

javascript

Routes:

- / (Home)
- /categories (Categories Overview)
- /category/:id (Category Detail)
- /recipe/:id (Recipe Detail)
- /search (Search Results)
- /profile (User Profile)
- /saved (Saved Recipes)
- /settings (User Settings)

4. Setup Instructions

Prerequisites

- Node.js (v14 or higher)
- npm or yarn
- Git

Installation

bash

Clone the repository

git clone [repository-url]

Navigate to project directory

cd recipe-app

Install dependencies

npm install

Create environment file

cp .env.example .env

Add your API keys

REACT_APP_SPOONACULAR_API_KEY=your_api_key

5. Folder Structure

recipe-app/

├─ public/

├─ src/

| └─ components/

| | └─ CategoriesHome/

| | └─ Navbar/

| | └─ RecipeCard/

| | └─ ...

| └─ hooks/

```
| | └─ useScrollAnimation.js
| | └─ ...
| └─ services/
| | └─ api.js
| | └─ recipeApi.js
| └─ context/
| └─ styles/
| └─ utils/
| └─ pages/
└─ tests/
└─ docs/
```

6. Running the Application

Development Mode

bash

Start development server

npm start

Access the application

Open <http://localhost:3000>

Production Build

bash

Create production build

npm run build

Serve production build

npm run serve

7. State Management

Global State

- User authentication state
- Theme preferences
- Saved recipes
- Search history

Local State Management

- Component-specific states using `useState`
- Custom hooks for reusable state logic
- Context API for theme and auth management

8. Component Documentation

CategoriesHome Component

javascript

/**

* Displays a grid of recipe categories with 3D animations

* Props: None

* State:

* - categories: Array of category objects

* - loading: Boolean loading state

* - error: Error state object

*/

RecipeCard Component

javascript

/**

* Displays individual recipe information

* Props:

* - recipe: Recipe object

* - onSave: Function to handle saving

* - isSaved: Boolean saved state

*/

9. User Interface

Design Principles

- Modern and clean interface
- Responsive design for all devices
- Intuitive navigation
- Smooth animations and transitions
- Accessible color schemes
- Clear visual hierarchy

Key UI Elements

- Navigation bar with search
- Category cards with 3D effects
- Recipe detail layouts
- Loading states and error messages
- User profile interface
- Settings panel

10. Styling

CSS Architecture

- CSS Modules for component isolation
- Global variables for theming
- Responsive breakpoints
- Animation utilities
- Dark mode support

Animation System

css

/* Base animations */

```
@keyframes slideInUp {  
  from {  
    transform: translateY(50px) translateZ(-20px);  
    opacity: 0;  
  }  
  to {  
    transform: translateY(0) translateZ(0);  
    opacity: 1;  
  }  
}
```

11. Testing

Test Structure

bash

tests/

├─ unit/

├─ integration/

└─ e2e/

Running Tests

bash

Run all tests

npm test

Run specific test suite

npm test -- -t "CategoriesHome"

Generate coverage report

npm test -- --coverage

12. Known Issues

Current Limitations

- API rate limiting with Spoonacular
- Image loading optimization needed
- Mobile performance optimization
- Search functionality refinement

Workarounds

- MealDB API fallback
- Lazy loading for images
- Debounced search implementation

13. Future Enhancements

Planned Features

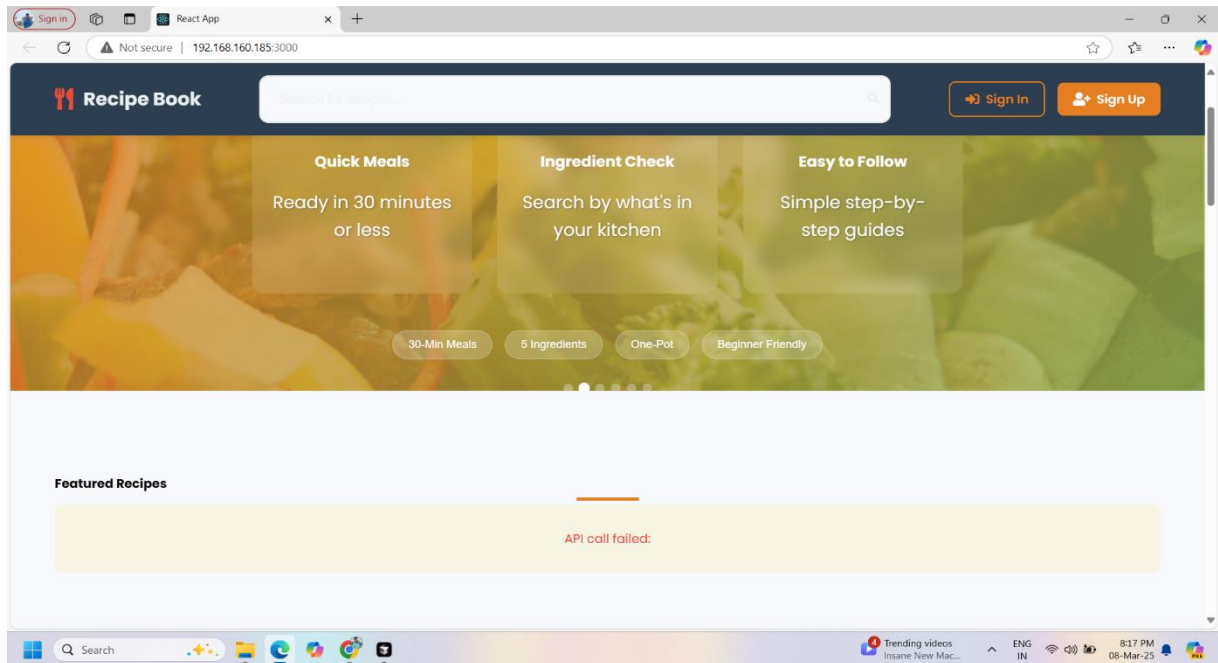
- Recipe creation and sharing
- Social features and comments
- Advanced filtering options
- Nutrition information
- Shopping list generation
- Meal planning calendar
- Recipe scaling functionality
- Print-friendly recipe views

Technical Improvements

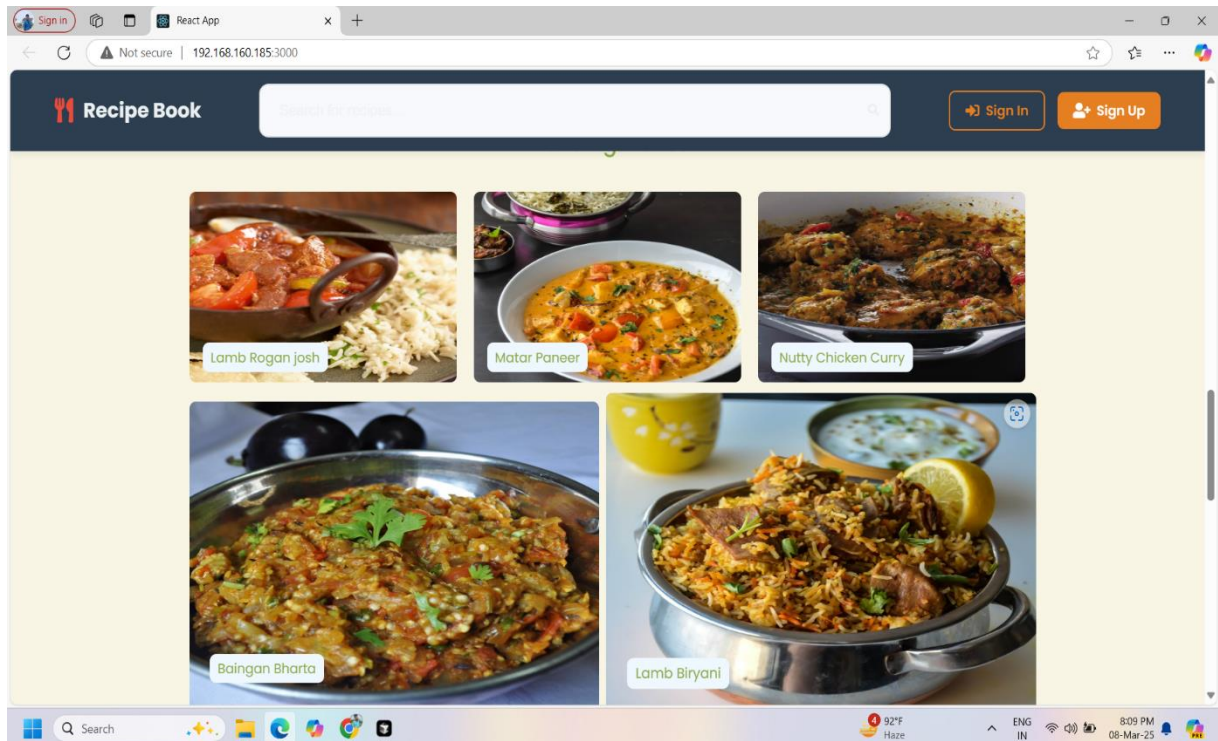
- PWA implementation
- Performance optimization
- Offline support
- Image optimization
- Advanced caching
- Analytics integration
- Accessibility improvements

14.Screenshot & Demo link

HOME PAGE



CATEGORYS PAGE



GOOGLE DRIVE LINK

<https://drive.google.com/drive/folders/14iCGhmZtklOL4EU9epygicW0Jl2CNVwQ?usp=sharing>

VIDEO LINK:

<https://drive.google.com/file/d/15-6Rl8nug89PfavfG6OCkVScjXi2m0j/view?usp=sharing>