
Cook Book – Project Documentation

1. Introduction

- **Project Title:** Cook Book
- **Team ID :** NM2025TMID38007
- **Team Leader :** MONISHA D – dmonisha1120@gmail.com
- **Team Members:**
 - Mohana priya .R – Priarmohana56@gmail.com
 - Nivetha .M – niviinivetha235@gmail.com
 - Jayamalini .A – malinimunimathi@gmail.com

2. Project Overview

- **Purpose:**

Cook Book is a web-based application built using React.js to help users explore, manage, and share cooking recipes. The goal is to provide a simple, interactive, and visually appealing way to browse recipes by category, ingredients, and cuisine.
- **Features:**
 - Recipe listing with images and categories
 - Detailed recipe view with ingredients and cooking steps
 - Search and filter functionality
 - Add to favorites / bookmark recipes
 - Responsive design for mobile and desktop

3. Architecture

- **Component Structure:**
 - App – Root component
 - Header – Navigation bar with search
 - RecipeList – Displays all recipes
 - RecipeCard – Single recipe preview
 - RecipeDetail – Detailed recipe view
 - Favorites – User's saved recipes

- Footer – Basic app footer
 - **State Management:**
 - Using **React Context API** for managing global state (favorites, user preferences).
 - Local component states used for form inputs, filters, and UI toggles.
 - **Routing:**
 - Implemented using **react-router-dom** for navigation:
 - / – Home (all recipes)
 - /recipe/:id – Recipe details
 - /favorites – Saved recipes
-

4. Setup Instructions

- **Prerequisites:**
 - Node.js (v16 or higher)
 - npm or yarn
 - **Installation:**
 - git clone https://github.com/yourusername/cook-book.git
 - cd cook-book
 - npm install
-

5. Folder Structure

```
cook-book/
├── public/           # Static assets
├── src/
│   ├── assets/      # Images, icons
│   ├── components/  # Reusable components
│   ├── pages/       # Page-level components
│   ├── context/     # Context API setup
│   ├── hooks/       # Custom hooks
│   ├── App.js       # Main app component
│   └── index.js     # Entry point
```

6. Running the Application

```
npm start
```

This will start the frontend server at `http://localhost:3000`.

7. Component Documentation

- **Key Components:**
 - `RecipeCard`: Displays recipe image, title, and short description.
 - `RecipeDetail`: Shows cooking steps, ingredients, and preparation time.
 - `SearchBar`: Allows filtering recipes by keyword or category.
 - **Reusable Components:**
 - `Button`, `Card`, `Modal` – Custom reusable UI elements.
-

8. State Management

- **Global State:**
 - Favorites stored using Context API across the application.
 - **Local State:**
 - Search input, filter selections, and modal visibility handled with `React useState`.
-

9. User Interface

- Clean UI with recipe images, categorized layouts, and easy navigation.
 - **Example Screens:**
 - Home page with recipe list
 - Recipe detail page with cooking instructions
 - Favorites page with saved recipes
-

10. Styling

- **CSS Frameworks/Libraries:** Tailwind CSS for styling.
 - **Theming:** Light/Dark mode toggle using CSS variables.
-

11. Testing

- **Testing Strategy:**
 - Unit testing using Jest & React Testing Library.
 - Integration tests for navigation and API calls.

- **Code Coverage:** Jest coverage reports.
-

12. Screenshots or Demo

- Screenshot examples (to be added after UI build).
 - Live Demo: [Deployment link if hosted on Vercel/Netlify]
-

13. Known Issues

- Search filter not optimized for large datasets.
 - Recipe API may return incomplete data.
-

14. Future Enhancements

- User authentication and profile management.
 - Ability to upload and share custom recipes.
 - Offline mode with local storage support.
 - Advanced filters (by calories, preparation time, etc.).
 - Voice-guided cooking assistant.
-