

VALLIAMMAL COLLEGE FOR WOMEN

Department Of Computer Applications

Project Documentation For CookBook

1. Introduction

❖ **Project Title:** CookBook

❖ **Team Id** : 147615

❖ **Team Size** : 4

❖ **Team Members:**

❖ **Sujeetha D(Team Leader)**

[Email Id:devarajsujeetha@gmail.com]

 **Anne Libo S**

[Email Id:sannelibo@gmail.com]

 **Inbamathi E**

[Email Id:inbamathi65@gmail.com]

 **Harini T**

[Email Id:hariniharini37704@gmail.com]

2. Project Overview

□ **Purpose:**

A Cookbook Recipe Application is a digital platform that allows users to explore, save, and manage various recipes. Here are a few key points about such an application:

□ **Features:**

- Recipe Library.
 - Ingredient List.
 - User authentication.
 - Step by Step Cooking Instruction.
 - Detailed Procedure.
-

3. Architecture

□ **Component Structure:**

The application is built using React.js with a component-based architecture. Major components include:

- **Header:** Contains the navigation bar and search bar.
- **Sidebar:** Displays user playlists and navigation links.
- **HomePage:** Most Popular Categories, and Trending dishes.
- **SearchPage:** Allows users to search for recipe, ingredient, and video.
- **State Management:**

The application uses **Redux** for global state management. The Redux store manages user authentication, recipe, ingredients, and search results.
- **Routing:**

The application uses **React Router** for navigation. Routes include:

 - `/`: Home page
 - `/search`: Search page
 - `/login`: User login page

4. Setup Instructions

- **Prerequisites:**
 - Node.js (v16 or higher)
 - npm (v8 or higher)
 - Git
- **Installation:**
 1. Clone the repository: git clone :
https://github.com/Syed1284/CookBook_SyedAliAasin
 2. Install dependencies: npm install
 3. Configure environment variables: Create a .env file in the client directory and add the necessary variables (e.g., API keys).
 4. Start the development server: npm start

5. Folder Structure

□ Client:

- **src/components:** # Reusable components (Header, Player, etc.)
- **src/pages:** # Page components (HomePage, SearchPage, etc.)
- **src/assets:** # Images, icons, and other static files
- **src/redux:** # Redux store, actions, and reducers
- **src/utls:** # Utility functions and helpers
- **App.js:** # Main application component
- **index.js:** # Entry point

□ Utilities:

- **api.js:** Handles API requests to the backend.
- **auth.js:** Manages user authentication and token storage.
- **hooks/usePlayer.js:** Custom hook for managing the music player state.

6. Running the Application

Frontend:

- To start the frontend server, run the following command in the client directory:
npm start
- npm install
- npx json-server ./db/db.json
- npm run dev
- The application will be available at <http://localhost:3000>

7. Component Documentation

□ Key Components:

- **Header:** Displays the navigation bar and search bar.
 - Props: `onSearch` (function to handle search queries).
- **Player:** Controls the video.
 - Props: `currentTrack` (object containing track details), `onPlay`, `onPause`, `onSkip`.

- **Playlist:** Displays a playlist with its name and cover image.
 - ▢ Props: playlist (object containing playlist details), onClick (function to handle playlist selection).
 - ▢ **Reusable Components:**
 - **Button:** A customizable button component.
 - ▢ Props: text, onClick, disabled.
 - **Input:** A reusable input field for forms and search.
 - ▢ Props: type, placeholder, value, onChange.
-

8. State Management

- ▢ **Global State:**

The Redux store manages the following global states:

 - **user:** Current authenticated user.
 - **player:** Current playing track, playback status (playing/paused), and volume.
 - **searchResults:** Results from the search functionality.
 - ▢ **Local State:**

Local state is managed using React's useState hook within components. For example, the SearchPage component manages the search query input locally.
-

9. User Interface

- ▢ **Screenshots**
 - **Home Page:** Display featured tracks and recommended playlists.

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](#)

- **Most popular Categories:** This component contains all the popular categories of recipes.

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](#)

- **Trending Dishes:** this components contains some of the trending dishes in the application.



Lamb Rogan josh



Matar Paneer



Nutty Chicken Curry



Baingan Bharta



Lamb Biryani

10. Styling

□ CSS Frameworks/Libraries:

The application uses **Styled-Components** for styling. This allows for modular and scoped CSS within components.

□ Theming:

A custom theme is implemented using Styled-Components, with support for light and dark modes.

11. Testing

□ Testing Strategy:

- **Unit Testing:** Using **Jest** and **React Testing Library**.
- **Integration Testing:** Is performed to ensure that components work together as expected.
- **End-to-End Testing:** **Cypress** is used for end-to-end testing of user flows.

□ Code Coverage:

- Code coverage is monitored using Jest's built in coverage tool. The current coverage is 85%.
-

12. Screenshots or Demo

- **Demo Link:**
https://drive.google.com/drive/folders/1vGGiA_IzhwOfI6lWK-e3QNSfVSslzmlY
- **Screenshots:** See section 9 for UI screenshots.

13. Known Issues

- **Issue 1:** The search functionality is slow with large datasets.

14. Future Enhancements

- **Future Features:**
 - Add support for user profiles and social sharing.
 - Implement a recommendation engine for personalized music suggestions.
 - Add animations and transitions for a smoother user experience.
-

This documentation provides a comprehensive overview of the **CookBook** project, including its architecture, setup instructions, and future plans.

