**PROJECT REPORT**

**CookBook: Your Virtual Kitchen Assistant**

YEAR : **2024 – 2025**

COLLEGE NAME : **Quaid-E Millath Government College for Woman**

CODE : UNM203

DEPARTMENT : **COMPUTER SCIENCE**

PROGRAM : **B.Sc Computer Science**

SEMESTER : **VI**

PROJECT SUBMITTED TO: UNIVERSITY OF MADRAS / NAAN MUDALVAN

Course Name : **Front End Development and Database Administration**

**TEAM LEADER:**

KAVYA K

**MEMBERS:**

1.MABU NISHA M

2.DIVYA DHARSHINI V

3,RAMYA I

**GUIDED BY: SUJATHA D**

**INTRODUCTION**

The **CookBook** is an advanced React.js application designed to provide culinary enthusiasts with a platform to explore, search, and learn recipes from various cuisines. Featuring a clean interface with the iconic React logo as a favicon and a **"Village Cooking Theme,"** the dashboard offers a suite of tools: recipe browsing with lazy loading, category navigation, search functionality, detailed recipe views with YouTube videos, and a responsive layout. Powered by React.js, react-router-dom for navigation, axios for API integration, and APIs from https://recipe-backend-tgsm.onrender.com/recipes and YouTube, the CookBook Dashboard empowers users to discover culinary delights while serving as an educational resource.

**DESCRIPTION**

The **CookBook** is a comprehensive recipe exploration platform built with React.js. It integrates a navigation bar, a hero section, a recipe list with lazy loading, category pages, search results, detailed recipe views with video integration, and a footer. The application leverages react-router-dom for seamless navigation, axios for fetching data, and a custom useRecipes hook for managing recipe data. Its design, styled with a warm, earthy "Village Cooking Theme" using custom CSS variables in App.css and global styles in index.css, ensures an engaging and accessible experience across devices.

**SCENARIO**

Sarah, a home cook, wants to explore new recipes and learn cooking techniques.

1. **Objective:** Sarah aims to find Indian recipes and watch a tutorial video.
2. **Using CookBook :** Sarah opens the app on her tablet.
3. **Navigation:** She uses the navbar to go to the "Indian" category.
4. **Overview:** On the category page, Sarah browses recipe cards and loads more.
5. **Exploration:** She searches for a dish, clicks a card, and views details with a video.
6. **Analysis:** Sarah watches the YouTube video and reviews ingredients/instructions.
7. **Additional Features:** She navigates back to search for another cuisine.
8. **Outcome:** With new skills, Sarah plans her next meal.

**PRE-REQUISITES**

* **Node.js and npm**
* **Download:** <https://nodejs.org/en/download/>
* **Install:** <https://nodejs.org/en/download/package-manager/>
* **React.js:** Install: npx create-react-app cookbook-dashboard
* **Libraries (as specified in package.json, inferred):**
  + **react**: "^19.0.0"
  + **react-dom:** "^19.0.0"
  + **react-router-dom:** "^7.3.0"
  + **react-scripts:** "5.0.1"
  + **axios:** "^1.8.1"
  + **react-icons**: "^5.3.0"
  + **web-vitals:** "^2.1.0"
  + **@testing-library/jest-dom:** "^5.17.0"
  + **Install:** npm install
  + HTML, CSS, JavaScript: Basic knowledge required.
* **Editor**: Visual Studio Code recommended (<https://code.visualstudio.com/>).
* **YouTube API Key:** Required for video integration (replace 'YOUR\_YOUTUBE\_API\_KEY' in code).

**PROJECT SETUP**

**Install Dependencies**

* **Command**: npm install (Installs dependencies listed in package.json)

**Start the Development Server**

* **Command**: npm start (As defined in package.json scripts).
* **URL**: <http://localhost:3000>

**Verify Setup**

* Open <http://localhost:3000> in your browser to see the dashboard with the React logo in the favicon.

**PROJECT STRUCTURE**

**Cookbook**

cookbook-dashboard/

├── public/

│ ├── index.html

│ ├── favicon.ico

│ ├── logo192.png

│ ├── logo512.png

│ ├── manifest.json

│ └── robots.txt

└── src/

├── components/

│ ├── Hero.jsx

│ ├── RecipeList.jsx

│ ├── RecipeCard.jsx

│ ├── Navbar.jsx

│ └── Footer.jsx

├── features/

│ └── recipes/

│ ├── RecipeDetails.jsx

│ ├── SearchResults.jsx

│ ├── Home.jsx

│ └── Category.jsx

├── hooks/

│ └── useRecipes.js

├── styles/

│ ├── App.css

│ └── index.css

├── App.js

├── index.js

├── setupTests.js

└── reportWebVitals.js

**PROJECT DEVELOPMENT**

**Root-Level Files**

* .**gitignore**:
  + Ignores: node\_modules, /build, .env files, logs, etc.
* **package.json:** 
  + Project metadata: Name (cookbook-dashboard), version (0.1.0), private (true).
  + Dependencies: Includes react, react-dom, react-router-dom, axios, react-icons, react-scripts, web-vitals, @testing-library/jest-dom.
  + Scripts: start, build, test, eject (standard Create React App scripts).
  + ESLint configuration: Extends react-app and react-app/jest.
* **package-lock.json:** 
  + Locks dependency versions (e.g., react@19.0.0, axios@1.8.1).

**App Component (App.js)**

* **Purpose**: Root component with routing and feature integration.
* **Features**:

1. **Routing**: Uses react-router-dom to define paths: Home (/), Category (/category/:id), Search (/search), and Recipe Details (/recipe/:id).
2. **Component Integration:** Renders Navbar, main content (Routes), and Footer within a div with class app.

**Index Entry (index.js)**

* **Purpose**: Entry point of the application.
* **Features**:

1. **Rendering**: Uses createRoot from react-dom/client to render App in strict mode into the #root div.
2. **Styling**: Imports App.css for the application’s theme.

**Global Styling (index.css)**

* **Purpose**: Defines global styles for the application.
* **Features**:

1. **Body**: Sets margin to 0, uses system fonts (e.g., 'Segoe UI', 'Roboto'), and enables font smoothing.
2. **Code**: Uses monospace fonts for code elements (e.g., 'source-code-pro', 'Monaco').

**Testing Setup (setupTests.js)**

* **Purpose**: Configures testing environment.
* **Features**:

1. Imports @testing-library/jest-dom for custom Jest matchers (e.g., toHaveTextContent).

**Web Vitals (reportWebVitals.js)**

* **Purpose**: Monitors web performance metrics.
* **Features**:

1. **Metrics**: Tracks CLS, FID, FCP, LCP, and TTFB using the web-vitals library.
2. **Usage**: Accepts an onPerfEntry callback to report metrics. </CONTENT>

**UseRecipes Hook (useRecipes.js)**

* **Purpose**: Fetches and manages recipe data from the backend API.
* **Features**:

1. **Data Fetching:** Uses axios to fetch recipes from https://recipe-backend-tgsm.onrender.com/recipes.
2. **State Management:** Utilizes useState and useEffect to manage recipes, loading, and error states.
3. **Return Values:** Provides recipes (array), loading (boolean), and error (string/null).

**Navbar Component (Navbar.jsx)**

* **Purpose:** Provides navigation and search functionality.
* **Features:**

1. **Search:** Uses useState and useNavigate to handle recipe searches.
2. **Navigation:** Includes links to Home, Indian, and Continental with a hamburger menu.
3. **Icons**: Utilizes FaSearch, FaBars, and FaTimes from react-icons.

**Hero Component (Hero.jsx)**

* **Purpose**: Displays a welcoming section with a call to action.
* **Features**: Call to Action: Triggers onGetStarted to load more recipes.

**RecipeList Component (RecipeList.jsx)**

* **Purpose:** Renders a list of recipe cards.
* **Features:** Mapping: Iterates over recipes prop to display RecipeCard components.

**RecipeCard Component (RecipeCard.jsx)**

* **Purpose**: Displays individual recipe previews.
* **Features**: Dynamic Content: Shows image, name, cuisine, and a link to recipe details.

**Footer Component (Footer.jsx)**

* **Purpose**: Provides a simple footer with copyright.

**Home Component (Home.jsx)**

* **Purpose:** Displays the hero and initial recipe list with lazy loading.
* **Features:**

1. Lazy Loading: Loads 8 recipes initially with a "Load More" button.
2. Data: Uses useRecipes hook to fetch recipes.

**Category Component (Category.jsx)**

* **Purpose:** Filters recipes by cuisine.
* **Features:** Filtering: Displays recipes matching the URL parameter (id).

**SearchResults Component (SearchResults.jsx)**

* **Purpose:** Shows search results based on query.
* **Features:** Filtering: Matches recipe name or cuisine with the search query.

**RecipeDetails Component (RecipeDetails.jsx)**

* **Purpose**: Displays detailed recipe information and video.
* **Features**:

1. Data Fetching: Uses axios to fetch recipe and YouTube video data.
2. Layout: Includes image, ingredients, instructions, and an embedded YouTube video.

**Styling (App.css)**

* **Purpose**: Defines the "Village Cooking Theme" with a warm, earthy aesthetic using custom CSS variables.
* **Highlights**:
  + **Variables**: --primary-color: #6b4e31, --secondary-color: #d97706, --background-color: #f5e8c7, --card-bg: #fff7e6, --text-color: #3c2f2f, --light-text: #6b4e31, --accent-color: #2e7d32, --shadow: 0 4px 8px rgba(0, 0, 0, 0.2), --transition: all 0.3s ease.
  + **Navbar**: Fixed position with var(--primary-color) background, hamburger menu for mobile, and hover effects with var(--accent-color).
  + **Main Content**: Flex column layout with max-width: 1280px, padded to avoid navbar overlap.
  + **Button (btn)**: Styled with var(--secondary-color), hover effect with var(--accent-color) and scale transform.
  + **Card**: var(--card-bg) background, hover lift effect, and var(--primary-color) border.
  + **Row**: Grid layout with 4 columns (responsive to 3 at 1024px, 2 at 768px, 1 at 480px).
  + **Hero**: Gradient background with var(--secondary-color) and var(--primary-color), centered text.
  + **Footer**: var(--primary-color) background, centered text.
  + **Recipe Details**: Grid layout with three rows (image, info/ingredients, instructions/video), responsive stacking at 768px.

**Responsive Design**: Breakpoints at 1024px, 768px, and 480px adjust grid columns, font sizes, and element dimensions.

**PROJECT FLOW**

1. **Setup:**

* **Create React app:** npx create-react-app cookbook-dashboard.
* **Install dependencies:** npm install (including react-router-dom, axios, react-icons, web-vitals).
* Configure routing and hooks.

1. **Design UI:**

* Build navbar, hero, recipe list, and footer in App.js.
* Style with App.css for the "Village Cooking Theme" and index.css for global styles.

1. **Implement Logic:**

* Fetch recipe data via useRecipes hook and axios.
* Integrate YouTube API in RecipeDetails.
* Enable navigation, search, and lazy loading.

**NOTES**

* API Details: Uses https://recipe-backend-tgsm.onrender.com/recipes for recipes and YouTube API for videos. Replace YOUR\_YOUTUBE\_API\_KEY with a valid key.
* Testing: Includes @testing-library/jest-dom for DOM assertions (npm test).
* Performance: Uses web-vitals for monitoring performance metrics.
* Enhancements: Potential additions include user accounts, favorite recipes, or offline support.

**REFERENCES**

* Create React App Documentation: <https://facebook.github.io/create-react-app/docs/getting-started>
* React Documentation: <https://reactjs.org/>
* React Router Documentation: <https://reactrouter.com/>
* Axios Documentation: <https://axios-http.com/docs/intro>
* YouTube API Documentation: <https://developers.google.com/youtube/v3>
* React Icons Documentation: <https://react-icons.github.io/react-icons/>
* Web Vitals Documentation: <https://github.com/GoogleChrome/web-vitals>
* Jest-DOM Documentation: <https://github.com/testing-library/jest-dom>

**CONCLUSION**

The CookBook Dashboard is a robust and interactive solution for culinary enthusiasts, leveraging React.js, `react-router-dom`, `axios`, and a custom `useRecipes` hook to deliver a feature-rich recipe exploration experience. Through its integration of recipe and YouTube APIs, lazy loading, and detailed views styled with the "Village Cooking Theme," the application provides a scalable foundation for features like user customization or advanced filtering. The project’s responsive design, performance monitoring with `web-vitals`, and testing setup with `@testing-library/jest-dom` ensure accessibility and maintainability. With the planned GitHub repository, demo, and video, users can explore and extend this project to suit their culinary needs.