Project Documentation

CookBook – Your Virtual Kitchen Assistant

1. Introduction

Background

Cooking is an essential life skill, yet many people struggle with discovering new recipes, learning how to prepare meals, or finding inspiration in their daily cooking routine. In the digital age, food-related applications play a major role in helping people explore, learn, and create meals with confidence.

CookBook: *Your Virtual Kitchen Assistant* is designed as a modern web application that allows users to discover recipes through category-based browsing. By leveraging **TheMealDB API**, CookBook provides quick access to thousands of meal ideas without the complexity of setting up or managing databases manually.

Team Information

- **Project Title:** CookBook – Your Virtual Kitchen Assistant
- **Team ID:** [NM2025TMID37859]
- **Team Leader:** [Naveena.G - naveenagandhi20077@gmail.com]
- **Team Members:**
    - [Karuna Choyal.G - karunachoyal22@gmai.com]
    - [Mahalakshmi.K - mahalakshmi2006lav@gmail.com]
    - [Renganayagi.T - renganayagi865@gmail.com]

2. Problem Statement

Food lovers and beginners alike often face these problems:

1. Difficulty in deciding “What to cook today?”
2. Lack of a categorized meal database.
3. No personalized recommendations or virtual assistance.
4. Time-consuming searches across the internet for reliable recipes.

CookBook aims to solve these problems by:

- Providing **organized recipe categories**.
- Offering **simple navigation** for quick access.
- Allowing users to **explore meals visually** with pictures and details.

3. Objectives

- To create a **web-based virtual kitchen assistant** using React.js.
- To implement **dynamic fetching of meals** from TheMealDB API.
- To design a **user-friendly, visually appealing interface**.
- To ensure the project is **scalable** for future features such as authentication and meal planning.

4. Project Overview

CookBook is an intuitive, category-driven recipe discovery platform. The primary aim is to make the cooking experience engaging and efficient. Users can:

- Explore meals under categories such as *Chicken, Vegetarian, Dessert, Seafood, Starter*.
- View details such as meal name, thumbnail, and recipe details.
- Navigate seamlessly between categories using **React Router DOM**.
- Gain inspiration for daily meals with minimal effort.

5. Features in Detail

Category-Based Recipe Browsing

Users can select a category (e.g., Chicken, Vegetarian) to view meals belonging to it.

API-Driven Dynamic Content

Meals are fetched directly from TheMealDB API in real time.

Recipe Details

Clicking on a meal navigates the user to a detailed recipe page with:

- Ingredients
- Instructions
- Meal Image

Popular Category Shortcuts

The homepage provides **quick navigation buttons** for trending categories.

Scalability

Additional features like login, meal planner, and favorites can be added.

6. System Architecture

High-Level View

[User] <----> [React Frontend] <----> [TheMealDB API]

Explanation

- **User Interface:** Built using React.js components with CSS styling.
- **Frontend Logic:** Handles navigation, state management, and API calls.
- **API Layer:** Fetches meals and recipes dynamically.
- **Database:** Not required for current version (future: MongoDB for user data).

7. Technology Stack

- **Frontend Framework:** React.js
- **Language:** JavaScript (ES6+), HTML5, CSS3
- **Styling:** Custom CSS (CategoryPage.css)
- **API:** TheMealDB
- **Navigation:** React Router DOM
- **Tools:** Visual Studio Code, GitHub, npm, Postman

8. Setup Instructions

Prerequisites

- Install **Node.js** (LTS version recommended).

- Install **Git** for version control.
- Install **Visual Studio Code** (IDE).

Steps

```
# Clone the repository
git clone <repo-link>

# Navigate into the project folder
cd cookbook-frontend

# Install dependencies
npm install

# Start the application
npm start
```

Access

- Frontend: http://localhost:3000

9. Folder Structure

```
CookBook/
│-- public/
│-- src/
│   │-- components/     # Reusable components
│   │-- pages/          # Pages such as Category.jsx
│   │   └── Category.jsx
│   │-- styles/
│   │   └── CategoryPage.css
│   │-- App.js          # Main application logic
│-- package.json
```

Explanation:

- **components/** → For small reusable UI parts.
- **pages/** → Full-page React components.
- **styles/** → CSS stylesheets.
- **App.js** → Entry point of the app.

10. API Documentation

Fetch Meals by Category

`GET https://www.themealdb.com/api/json/v1/1/filter.php?c={category}`

Response Example:

```
{
  "meals": [
    {
      "strMeal": "Grilled Chicken",
      "strMealThumb": "https://www.themealdb.com/images/media/meals/123.jpg",
      "idMeal": "52772"
    }
  ]
}
```

Fetch Recipe Details by ID

```
GET https://www.themealdb.com/api/json/v1/1/lookup.php?i={mealId}
```

11. User Interface Design

- **Category Page:** Displays all meals under selected category.
- **Recipe Page:** Shows meal image, recipe instructions, and ingredients.
- **Navigation Buttons:** Allow switching categories quickly.

12. Authentication (Future Scope)

- Planned JWT authentication.
- Features to be added:
    - Save favorite meals.
    - Personalized recommendations.
    - User profile dashboard.

13. Testing

Manual Test Cases

Test Case	Action	Expected Result	Status
Load Category	Click “Chicken” button	List of chicken meals appears	✅
Fetch Recipe	Click meal card	Detailed recipe loads	✅
Invalid Category	Enter random category	Error handled gracefully	✅

Tools Used

- **Postman** – API Testing
- **Chrome Dev Tools** – Inspect UI and console logs

14. Screenshots (Demo)

- Homepage Screenshot
- Category Page Screenshot
- Recipe Page Screenshot
- API Response Screenshot

15. Known Issues

1. Dependency on external API → App won't load if TheMealDB API is down.
2. No offline support.
3. Limited features (no favorites, no meal planner).

16. Future Enhancements

- User authentication and personalized dashboards.
- Meal planning system (weekly/monthly planners).
- Shopping list generation.
- Offline caching.
- Integration with smart home/kitchen devices.
- Multi-language support.

17. Conclusion

CookBook represents a **practical and scalable web solution** for modern cooking needs. With its clean UI, easy navigation, and API-driven design, it provides a foundation for future growth into a full-fledged virtual assistant.

This project demonstrates:

- Strong application of **React.js concepts**.
- Effective use of **REST APIs**.
- Potential for expansion into advanced cooking assistance.