INTRODUCTION

PROJECT TITLE:

CookBook: Your Virtual Kitchen Assistant

Team

Team Leader : Nandhana Das D(CODE EXECUTION AND GIT HUB LINK)

Team member : Vishalini S(CODE EXECUTION AND GIT HUB LINK)

Team member : Saranya S(CODE EXECUTION AND GIT HUB LINK)

Team member : Pavithra R(DOCUMENTATION)

Team member : Uvasri S(VOICE OVER)

Mail IDs

Team Leader : Nandhana Das D – devadasnandhana@gmail.com

Team member : Vishalini S – svishalini04@gmail.com

Team member : Saranya S – saranyasrinivasan2004.12.17@gmail.com

Team member : Pavithra R – pavithra.r13122004@gmail.com

Team member : Uvasri S – Hannayuvasri@gmail.com

PROJECT OVERVIEW

Purpose:

CookBook is a dynamic web application designed to revolutionize recipe discovery, organization, and creation. It caters to both home cooks and professional chefs, providing a seamless user experience with powerful features like advanced search, recipe categorization, and interactive recipe pages.

Features:

Recipe Discovery: Access a vast library of international recipes via the MealsDB API.

Search Functionality: Find dishes by name, ingredients, or category.

Visual Browsing: Explore recipes through curated image galleries.

User-Friendly Design: Clean and intuitive interface with clear navigation.

Recipe Details: View ingredients, cooking instructions, and video tutorials.

Trending Dishes: Stay updated with popular and trending recipes.

Newsletter Subscription: Receive recipe updates and culinary tips.

ARCHITECTURE

COMPONENT STRUCTURE:

Navbar Component: Handles site-wide navigation.

Hero Section: Showcases the app's purpose with call-to-action buttons.

Category & Recipe Components: Fetch and display recipe categories and details.

Trending Dishes & Popular Categories: Dynamically load content via API calls.

Newsletter Component: Allows users to subscribe to newsletters.

State Management:

State management is handled using React's useState and useEffect hooks. Data is fetched and updated through Axios API calls, ensuring real-time updates.

Routing:

React Router Dom is used to manage page navigation:

/ → Home Page

/category/:id → Category-specific recipes

/recipe/:id → Detailed recipe view

SETUP INSTRUCTIONS

Prerequisites:

Node.js & npm: For package management and running the development server.

React.js: JavaScript library for building the UI.

Code Editor: Visual Studio Code, WebStorm, or your preferred IDE.

INSTALLATION:

1. Clone the Repository:

```
git clone [repository-url]
cd recipe-app-react
```

2. Install Dependencies:

npm install

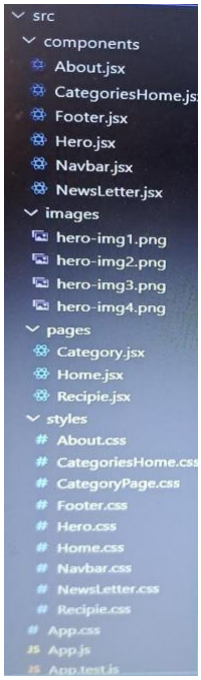
3.Start the Development Server:

npm start

4.Access the Application:

Open your browser and navigate to**:**
http://localhost:3000

FOLDER STRUCTURE:



Running the Application

Start the app locally:

```
npm start
```

It will automatically open in your default browser at:

```
http://localhost:3000
```

Component Documentation

Key Components:

Navbar: Manages site navigation.

Hero Section: Welcomes users and highlights app features.

Recipe List: Displays a list of recipes within a category.

Recipe Detail: Shows full recipe details, including ingredients, instructions, and a tutorial video.

Newsletter Signup: Collects user emails for subscriptions.

Reusable Components:

Button: Customizable button with variant options.

Card: Displays recipe previews with images and titles.

State Management

Global State: Managed via React's Context API to store user preferences and favorites.

Local State: Managed with useState hooks for component-level interactions, like toggling recipe details or handling form inputs.

User Interface

The UI is designed for simplicity and efficiency:

Homepage: Showcases featured categories and trending dishes.

Category Page: Lists all recipes in a selected category.

Recipe Page: Displays full recipe details with images, ingredients, instructions, and a video tutorial.

Styling

CSS Frameworks:

Tailwind CSS: For modern, responsive design.

React Icons: For intuitive iconography.

Custom CSS: For unique component styling.

Theming:
The app follows a light and dark theme switcher to enhance user experience.

Testing

Testing Strategy:

Unit Testing: Using Jest for testing individual components.

Integration Testing: Using React Testing Library for testing component interactions.

End-to-End Testing: Using Cypress for simulating user flows.

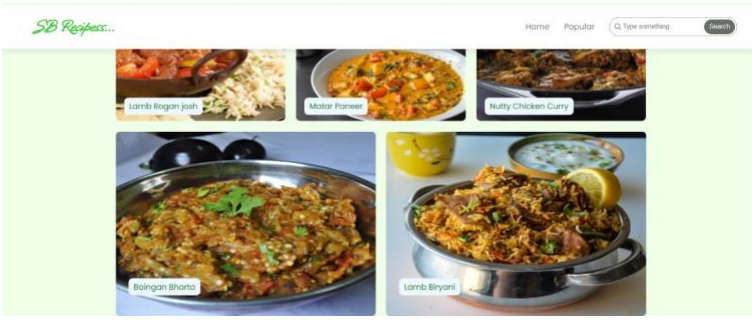
Code Coverage:

Ensure at least 80% coverage for critical components.

Use tools like Istanbul for generating coverage reports.

Screenshots or Demo

Screenshots



Drive link :https://drive.google.com/drive/folders/1s97-R2xmgUwOuaXqHRUNsFD5LQ-OBIkr?usp=drive_link

Video link:
https://drive.google.com/file/d/1n_nABLNCFeQOxXOYuM-5RwM_ROGMsRTG/view?usp=drivesdk

Known Issues

API rate limits might affect data loading.

Missing images for certain recipes from the MealsDB API.

Future Enhancements

User Authentication: Save and manage personal recipe collections.

Custom Recipe Uploads: Let users contribute their own recipes.

Meal Planning Feature: Generate weekly meal plans based on user preferences.

Ingredient Substitutions: Suggest alternatives for unavailable ingredients.

Voice-Guided Cooking: Step-by-step audio guidance for hands-free cooking.