****

**PROJECT REPORT**

**COOKBOOK**

YEAR : **2024 – 2025**

COLLEGE NAME : **K.C.S KASI NADAR COLLEGE OF ARTS & SCIENCE**

CODE : UNM203

DEPARTMENT : **COMPUTER Science**

PROGRAM : **BCA Computer Application**

SEMESTER : **VI**

PROJECT SUBMITTED TO: UNIVERSITY OF MADRAS / NAAN MUDALVAN

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

**TEAM LEADER: Nivedha.T**

**MEMBERS:**

1. Samrin.A

2. Sharmila Begum.I

3. .Yazhini.M

**GUIDED BY: Mrs.M.GINITHA**

**SPOC NAME: Dr.K. LALITHAKAMESWARI**

**CookBook: Your Virtual Kitchen Assistant**

**Introduction**

CookBook is a revolutionary web application designed to enhance the way users discover, organize, and create recipes. Catering to both novice and professional chefs, it offers a user-friendly interface, robust features, and an extensive collection of inspiring recipes.

**Description**

Welcome to the future of culinary exploration with CookBook! This cutting-edge web application is meticulously designed to transcend traditional recipe management by integrating a seamless and interactive experience for users of all skill levels.

CookBook provides an intuitive interface, making it easy to navigate through a vast collection of culinary inspirations. With advanced search functionality, dynamic recipe discovery, and organizational tools, the platform fosters a thriving community of food enthusiasts who can share and collaborate on their favorite recipes.

By blending innovation with tradition, CookBook transforms every cooking experience into an adventure. Whether you’re a beginner looking for step-by-step guidance or a professional chef seeking fresh inspiration, CookBook is your ultimate virtual kitchen assistant.

**Scenario-Based Introduction**

Sarah rummaged through her fridge, staring at the dwindling ingredients while her teenage son, Ethan, sighed in frustration. “What are we even going to eat?” he groaned, glued to his phone.

Suddenly, Sarah recalled her friend Maya raving about a new recipe platform called CookBook. Intrigued by its promise of “elevating culinary endeavors”, she grabbed her laptop.

“Hold that thought, Ethan,” she said with newfound excitement. “We might be about to embark on a delicious adventure.” With just a few clicks, a world of culinary possibilities unfolded before them.

**Technical Architecture**

CookBook is built using React.js for a smooth and dynamic single-page application experience. The application interacts with an API client that communicates with the backend and integrates external functionalities using Rapid API. This enables access to extensive recipe databases and additional features without the need for a custom-built backend.

**Technology Stack**

* Frontend: React.js, HTML, CSS, JavaScript
* API Integration: Rapid API (for fetching external recipe data)
* Development Tools: Node.js, npm, Visual Studio Code

**Project Goals and Objectives**

CookBook aims to provide a seamless, user-friendly platform for individuals passionate about cooking, baking, and culinary exploration.

**Key Objectives**

* User-Friendly Experience: Ensure smooth navigation and easy access to recipes.
* Comprehensive Recipe Management: Provide advanced features for organizing, saving, and sharing recipes.
* Seamless Integration: Leverage modern web development technologies for efficiency and performance.

**Features**

* Recipes from the MealsDB API: Access a vast collection of international recipes catering to diverse dietary needs.
* Visual Recipe Browsing: Explore categorized recipes with image galleries for enhanced discovery.
* Intuitive UI: Clean, modern design for easy navigation and interaction.
* Search Feature: Quickly find recipes based on ingredients, cuisine, or meal type.

**Prerequisites**

To develop CookBook, the following tools and knowledge are required:

Software Requirements

1. **Node.js and npm:**
   * Node.js provides a scalable JavaScript runtime environment.
   * Install Node.js and npm from: Node.js Download
   * Installation instructions: Node.js Package Manager
2. **React.js:**
   * Install React.js using the following command:
   * npx create-react-app cookbook-app
   * Navigate to the project directory:
   * cd cookbook-app
   * Start the development server:
   * npm start
   * The app will be accessible at <http://localhost:3000>.

**Coding page**

**Index.html:**

<!DOCTYPE html>

<html lang=”en”>

<head>

<meta charset=”utf-8” />

<link rel=”icon” href=”%PUBLIC\_URL%/favicon.ico” />

<meta name=”viewport” content=”width=device-width, initial-scale=1” />

<meta name=”theme-color” content=”#000000” />

<meta

Name=”description”

Content=”Web site created using create-react-app”

/>

<link rel=”apple-touch-icon” href=”%PUBLIC\_URL%/logo192.png” />

<!—

Manifest.json provides metadata used when your web app is installed on a

User’s mobile device or desktop. See <https://developers.google.com/web/fundamentals/web-app-manifest/>

🡪

<link rel=”manifest” href=”%PUBLIC\_URL%/manifest.json” />

<!—

Notice the use of %PUBLIC\_URL% in the tags above.

It will be replaced with the URL of the public folder during the build.

Only files inside the public folder can be referenced from the HTML.

Unlike “/favicon.ico” or “favicon.ico”, “%PUBLIC\_URL%/favicon.ico” will

Work correctly both with client-side routing and a non-root public URL.

Learn how to configure a non-root public URL by running npm run build.

🡪

<title>React App</title>

</head>

<body>

<noscript>You need to enable JavaScript to run this app.</noscript>

<div id=”root”></div>

<!—

This HTML file is a template.

If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.

The build step will place the bundled scripts into the <body> tag.

To begin the development, run npm start or yarn start.

To create a production bundle, use npm run build or yarn build.

🡪

</body>

</html>

**categoriesHome.jsx:**

import React, { useEffect } from ‘react’

import ‘../styles/CategoriesHome.css’

import heroImg1 from ‘../images/hero-img1.png’

import heroImg2 from ‘../images/hero-img2.png’

import heroImg3 from ‘../images/hero-img3.png’

import heroImg4 from ‘../images/hero-img4.png’

import axios from ‘axios’

import { useNavigate } from ‘react-router-dom’

const CategoriesHome = () => {

const navigate = useNavigate() const [items, setItems] = React.useState([])

const [categories, setCategories] = React.useState([])

useEffect(() => {

fetchCategories()

fetchItems()

}, [])

Const fetchCategories = async () => {

Await axios.get(‘https://www.themealdb.com/api/json/v1/1/categories.php’)

.then(response => {

setCategories(response.data.categories)

console.log(response.data.categories)

})

.catch(error => console.error(error));

}

Const fetchItems = async () => {

Await axios.get(‘https://www.themealdb.com/api/json/v1/1/filter.php?a=Indian’)

.then(response => {

setItems(response.data.meals)

console.log(response.data.meals)

})

.catch(error => console.error(error));

}

Return (

<div className=’home-categories-container’ id=’popular’>

<div className=”popular-categories-body”>

<h3>Most Popular Categories</h3>

<p>Be sure not to miss out the categories of these most popular categories. Enjoy trying them out!</p>

{categories.length > 0 ?

<>

<div className=”popular-categories”>

{categories.map((category, index)=>{

Return (

Index < 9 &&

<div className=”popular-category” key={index} onClick={()=>navigate(/category/${category.strCategory})} >

<img src={category.strCategoryThumb} alt=”” />

<span>

<h4>{category.strCategory}</h4>

<p>View All Recipies</p>

</span>

</div>

)

}

</div>

</>

:”Loading”}

</div>

<div className=”popular-dishes-body” id=’recipies’>

<h3>Trending Dishes</h3>

{items.length > 0 ?

<div className=”popular-dishes”>

<span className=’dishes-small’>

<div className=”popular-dish” onClick={()=> navigate(/recipie/${items[6].idMeal})}>

<img src={items[6].strMealThumb} alt=”” />

<p>{items[6].strMeal}</p>

</div>

<div className=”popular-dish” onClick={()=> navigate(/recipie/${items[7].idMeal})}>

<img src={items[7].strMealThumb} alt=”” />

<p>{items[7].strMeal}</p>

</div>

<div className=”popular-dish” onClick={()=> navigate(/recipie/${items[8].idMeal})}>

<img src={items[8].strMealThumb} alt=”” />

<p>{items[8].strMeal}</p>

</div>

</span>

<span className=’dishes-big’>

<div className=”popular-dish” onClick={()=> navigate(/recipie/${items[0].idMeal})} >

<img src={items[0].strMealThumb} alt=”” />

<p>{items[0].strMeal}</p>

</div>

<div className=”popular-dish” onClick={()=> navigate(/recipie/${items[5].idMeal})}>

<img src={items[5].strMealThumb} alt=”” />

<p>{items[5].strMeal}</p>

</div>

</span>

<span className=’dishes-small’>

<div className=”popular-dish” onClick={()=> navigate(/recipie/${items[2].idMeal})}>

<img src={items[2].strMealThumb} alt=”” />

<p>{items[2].strMeal}</p>

</div>

<div className=”popular-dish” onClick={()=> navigate(/recipie/${items[3].idMeal})}>

<img src={items[3].strMealThumb} alt=”” />

<p>{items[3].strMeal}</p>

</div>

<div className=”popular-dish” onClick={()=> navigate(/recipie/${items[4].idMeal})}>

<img src={items[4].strMealThumb} alt=”” />

<p>{items[4].strMeal}</p>

</div>

</span>

</div>

:””}

{/\*

<button >View more</button> \*/}

</div>

</div>

)

}

Export default CategoriesHome

**Navbar.jsx:**

Import React from ‘react’

Import ‘../styles/Navbar.css’

Import { IoSearch } from “react-icons/io5”;

Import { Link, useNavigate } from ‘react-router-dom’

Import axios from ‘axios’;

Const Navbar = () => {

Const navigate = useNavigate()

Const [search, setSearch] = React.useState(‘’);

Const handleSearch = async () => {

Await axios.get([https://www.themealdb.com/api/json/v1/1/search.php?s=${search}](https://www.themealdb.com/api/json/v1/1/search.php?s=$%7bsearch%7d))

.then(response => {

// console.log(response.data.meals[0].idMeal)

setSearch(‘’);

navigate(/recipie/${response.data.meals[0].idMeal})

})

.catch(error => {

Navigate(‘/’)

setSearch(‘’);

alert(‘No such recipe found!!’)

}

);

}

(

<div className=’Navbar’>

<h3 onClick={()=>navigate(‘/’)}>SB Recipess…</h3>

<div className=’nav-options’>

<ul>

<li onClick={()=>navigate(‘/’)}>Home</li>

{(window.location.href === <http://localhost:3000/> || window.location.href === <http://localhost:3000/#popular>) ?

<a href=”#popular”><li>Popular</li></a>

:

<Link to=’/#popular’><li>Popular</li></Link>

}

</ul>

<div className=”nav-search”>

<span>

<IoSearch className=’nav-search-icon’ />

<input type=”text” placeholder=”Type something..” onChange={€=>setSearch(e.target.value)} value={search} />

</span>

<button onClick={handleSearch} >Search</button>

</div>

</div>

</div>

)

}

Export default Navbar

**Home.jsx:**

Import React from ‘react’

Import Hero from ‘../components/Hero’

Import CategoriesHome from ‘../components/CategoriesHome’

Import About from ‘../components/About’

Import NewsLetter from ‘..

/components/NewsLetter’

Import ‘../styles/Home.css’Const Home = () => {

Return (

<div className=’homepage’>

<Hero />

<CategoriesHome />

<NewsLetter />

</div>

)

}

Export default Home

**Recipe.css:**

.Recipie-page{

Padding-top: 12vh;

Min-height: 60vh;

}

.recipie-img{

Width: 100%;

Padding-top: 2vh;

}

.recipie-img img{

Width: 98%;

Margin-left: 1%;

Height: 85vh;

Border-radius: 0.7rem;

}

.recipie-data-container{

Display: grid;

Grid-template-columns: 65% 25%;

Justify-content: space-evenly;

}

.recipie-data{

Box-shadow: rgba(0, 0, 0, 0.2) 0px 18px 50px -10px;

Position: relative;

Top: -15vh;

Background-color: #fff;

Padding: 4vh 2vw;

Border-radius: 0.6rem;

}

.recipie-header{

}

.recipie-header h4{

Margin: 0;

Font-size: 1.5rem;

Font-weight: 500;

}

.recipie-header .recipie-specials{

Display: flex;

Gap: 10px;

}

.recipie-header .recipie-specials p{

Border: 1px solid rgb(35, 209, 35);

Color: rgb(35, 209, 35);

Padding: 5px 20px;

Border-radius: 2rem;

Font-size: 0.7rem;

}

.recipie-data .procedure{

}

.recipie-data .procedure h5{

Font-size: 1.3rem;

Font-weight: 500;

Margin: 0;

Padding-top: 3vh;

}

.recipie-data .procedure p{

Margin: 0;

Font-size: 0.9rem;

}

.youtube-video-container{

}

.youtube-video-container h5{

Font-size: 1.3rem;

Font-weight: 500;

Margin: 0;

Padding-top: 3vh;

}

.ingredients-container{

Box-shadow: rgba(0, 0, 0, 0.2) 0px 18px 50px -10px;

Position: relative;

Top: -15vh;

Background-color: #fff;

Padding: 4vh 2vw;

Border-radius: 0.6rem;

Height: fit-content;

}

.ingredients-container h3{

Margin: 0;

Padding-bottom: 1vh;

Font-size: 1.3rem;

Font-weight: 500;

Color: rgb(23, 162, 23);

}

.ingredients{

Display: flex;

Flex-direction: column;

Gap: 10px;

Padding: 0;

}

.ingredient{

Display: flex;

Align-items: center;

Justify-content: space-between;

}

.ingredient h5{

Font-size: 1rem;

Margin: 0;

Font-weight: 400;

Max-width: 50%;

}

.ingredient p{

Margin: 0;

Font-size: 0.8rem;

Max-width: 40%;}

**Package.json:**

{

“name”: “client”,

“version”: “0.1.0”,

“private”: true,

“dependencies”: {

“@testing-library/jest-dom”: “^5.17.0”,

“@testing-library/react”: “^13.4.0”,

“@testing-library/user-event”: “^13.5.0”,

“axios”: “^1.6.2”,

“react”: “^18.2.0”,

“react-dom”: “^18.2.0”,

“react-icons”: “^4.12.0”,

“react-router-dom”: “^6.21.0”,

“react-scripts”: “5.0.1”,

“react-youtube”: “^10.1.0”,

“web-vitals”: “^2.1.4”

},

“scripts”: {

“start”: “react-scripts start”,

“build”: “react-scripts build”,

“test”: “react-scripts test”,

“eject”: “react-scripts eject”

},

“eslintConfig”: {

“extends”: [

“react-app”,

“react-app/jest”

]

},

“browserslist”: {

“production”: [

“>0.2%”,

“not dead”,

“not op\_mini all”

],

“development”: [

“last 1 chrome version”,

“last 1 firefox version”,

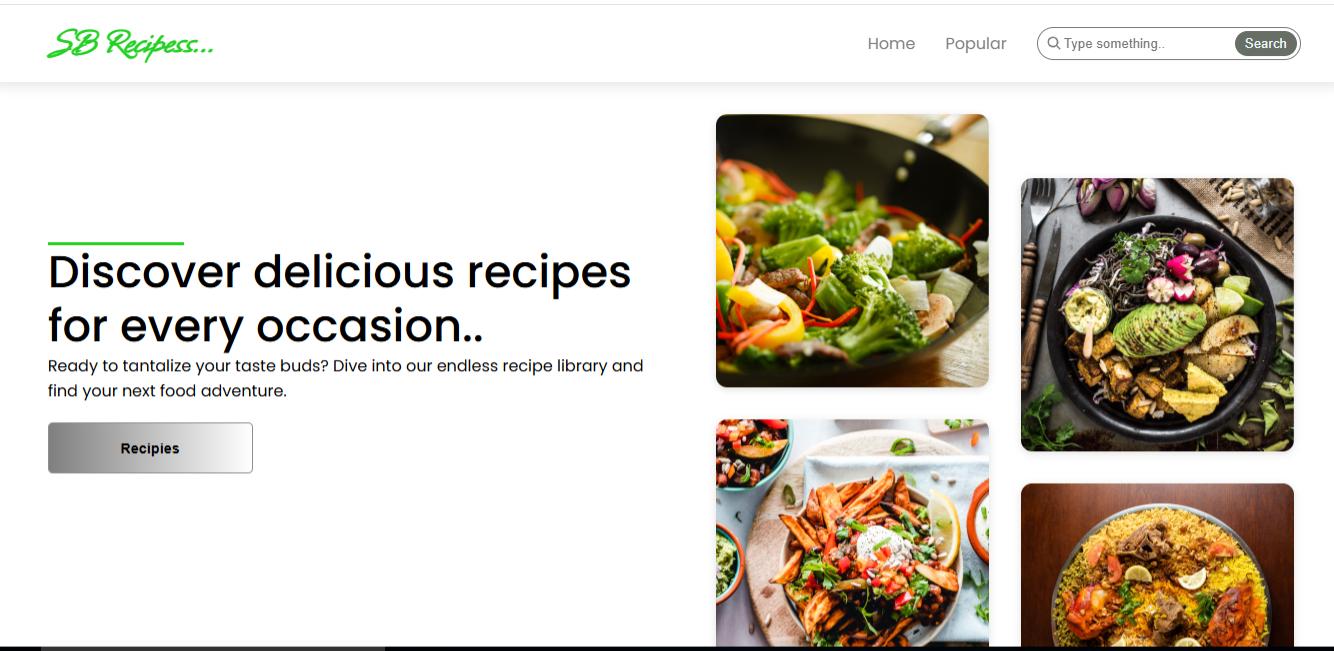
“last 1 safari version”

]

}

}

**Project Overview**

**Project Execution & Structure**

