Project Documentation

MOM'S DRIP COOKBOOK

Introduction

- **Project Title:** cookbook-your virtual kitchen assistant
- **Team ID: NM2025TMID37874**
- **Team Members:**

>Madhumitha J (madhumithajai2006@gmail.com) - Team leader

>Jaina S (sanchetiijaina@gmail.com) - Team member

>Shalini G (shalinigovindhan06@gmail.com) - Team member

>Sivatmika Anand (sivatmikaanand@gmail.com) - Team member

➢ Acknowledgment:

We would like to express our sincere gratitude to Naan Mudhalvan and SmartInternz for providing us with this wonderful opportunity to work on the Cookbook Web Application Project. We are deeply thankful to our mentor/faculty for giving this opportunity.

➢ Overview of Project:

This Cookbook Project is a web-based application designed to provide users with an easy and interactive way to explore, organize, and share recipes. In today's digital age, but many platforms lack personalization and user-friendly design, our project aims to build a modern, responsive, and dynamic cookbook application.

Cooking is one of the oldest and most essential human activities, and recipes have been shared across generations through handwritten notes, cookbooks, and oral traditions. In today's digital era, the way people discover, store, and share recipes has transformed significantly. With the growth of the internet and web applications, people now prefer quick, accessible, and organized platforms to explore new dishes. However, most recipe-sharing platforms are either cluttered, difficult to navigate, or lack personalization.

Whether you're taking your very first steps in the kitchen or mastering complex dishes, CookBook embraces a diverse community of food lovers. It fosters collaboration, sharing, and creativity— bringing together everyone who shares a passion for the art of cooking.

➢ Project Objective:

CookBook revolutionizes the way recipes are discovered, organized, and created. Crafted with user-friendly aesthetics in mind, CookBook immerses you in an unparalleled culinary journey. Effortlessly navigate through a

vast library of inspiration with dynamic search, smart filters. The Cookbook application allows users to browse recipes, search for specific dishes, and view detailed instructions along with ingredients. By combining front-end technologies with a structured design, the project not only demonstrates technical skills but also offers a practical solution that can be expanded into a real-world application. This Cookbook Project emerges as a modern solution for food enthusiasts, home cooks, and culinary explorers.It is used to learn the recipies in upgradable form of technology.

This web-based application is designed to be a dynamic, interactive, and visually appealing platform. This Cookbook is more than just a recipie repositary—it's a celebration of food culture, a tool for learning, and a digital space for creativity. Whether you're a beginner looking for simple meals or a seasoned chef experimenting with fusion cuisine, this platform offers something for everyone. This Cookbook project is built using HTML, CSS, JavaScript, and the powerful React framework, this project showcases how front-end technologies can be harnessed to create a seamless and engaging user experience. Recipes are grouped by category (e.g., breakfast, mains,desserts) or theme (e.g., seasonal, regional, dietary). Each recipe includes prep time, cook time,serving size, and difficulty level.

   *User-Friendly Interface: To create a clean, intuitive layout that allows users to navigate recipies.

   *Responsive Design: To ensure the application works seamlessly across devices—desktops,tablets, and smartphones.

   *Recipe Cards: Each recipe is displayed in a visually appealing card format with images,ingredients, and instructions.

   *Search & Filter: Users can search recipes by name or filter by category (e.g., vegetarian, desserts,quick meals).

   *View Recipe Details – See ingredients, steps, and cooking time.

   ➢ **Technologies Used:**

   *HTML (HyperText Markup Language)Provides the structural foundation of the application. Every recipe card and form is crafted using semantic HTML elements to ensure accessibility and clarity**.**

   *CSS (Cascading Style Sheets) Brings the visual flair. From elegant typography to responsive layouts and animations, CSS transforms the raw HTML into a polished, user-friendly interface.

   *JavaScript Powers the interactivity. JavaScript handles dynamic behaviors such as filtering recipes and managing user inputs, making the application responsive and intuitive.

   *React is the heart of the application. React's component-based architecture allows for modular development, efficient rendering, and state management. Features like live search, recipe filtering, and form validation are implemented using React hooks and props.

   *Node.js is the engine behind a digital cookbook project, especially if we are building a web or mobile app. Here's how it powers things under the hood.

   ➢ **Pre-Requisites:**

   This PRE-REQUISITES section is telling the tools, software, and basic knowledge must have before

developing this CookBook frontend project using React.js.

1. Node.js and npm:

   * Node.js is a JavaScript runtime environment.

   *npm comes with Node.js (installed together).

   *npm helps to download and manage libraries/tools for the project.

    download link

2. Steps to create cookbook app:

   *Open Command Prompt (cmd)

   *Run the commands one by one:

    Command 1: npx create-cookbook-app

              my-cookbook-app

   This will create a new React folder named my-cookbook-app.

     Command 2: cd my-cookbook-app

    This moves you inside the project folder.

3. Running the React app:

     Command 3: npm start

    This starts the development server and automatically opens your app in a browser at

     open on a localhost

4. Basic Knowledge Required:

   *HTML → For structure of the app.

   *CSS → For styling.

   *JavaScript → For client-side interactivity.

5. Version Control:

*Using Git to track changes and collaborate.

*Hosting repositories on GitHub.

   Git Installation:

  https://git-scm.com/downloads

6.Development Environment:

Visual Studio Code:

> ## **Project Structure:**

A project structure is the organized layout of folders and files in the project. It's like the blueprint or skeleton. In this project, the files are organized into three main folders:

*Pages:

Contains the files that act as application pages.

*Components:

Contains reusable UI components used across multiple pages.

*Styles:

Contains all the CSS files that define the styling of the application.

> ## **Folders and Files in VS Code:**

```
my-react-app
├── .gitignore
├── README.md
├── package.json
├── package-lock.json
├── node_modules/
│
└── src/
     ├── public/
     ├── images/
     ├── pages/
     ├── styles/
     ├── components/
     │    ├── App.js
     │    ├── App.css
     │    └── index.css
     │
     ├── App.js
     ├── App.css
     ├── index.css
     ├── reportWebVitals.js
     ├── App.test.js
     ├── setupTests.js
     └── index.js
```

➢ **Public Folder**
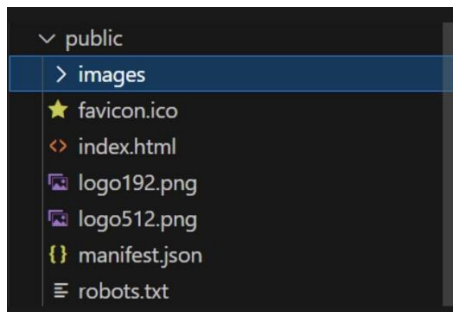
➢ **Image Folder:**
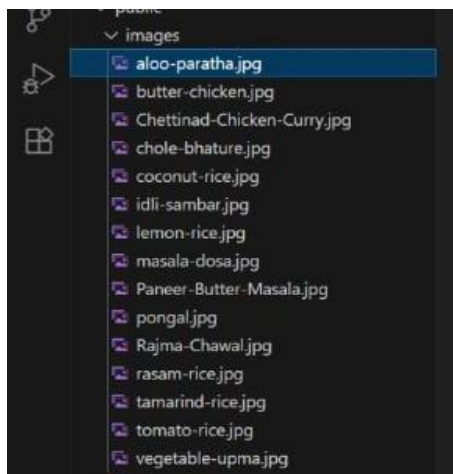
➢ **Page Folder:**
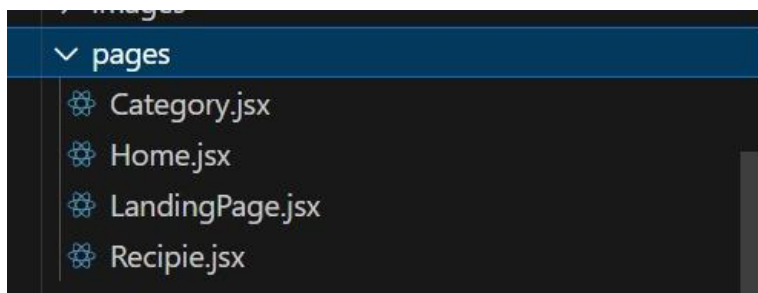
➤ **Styles Folder:**
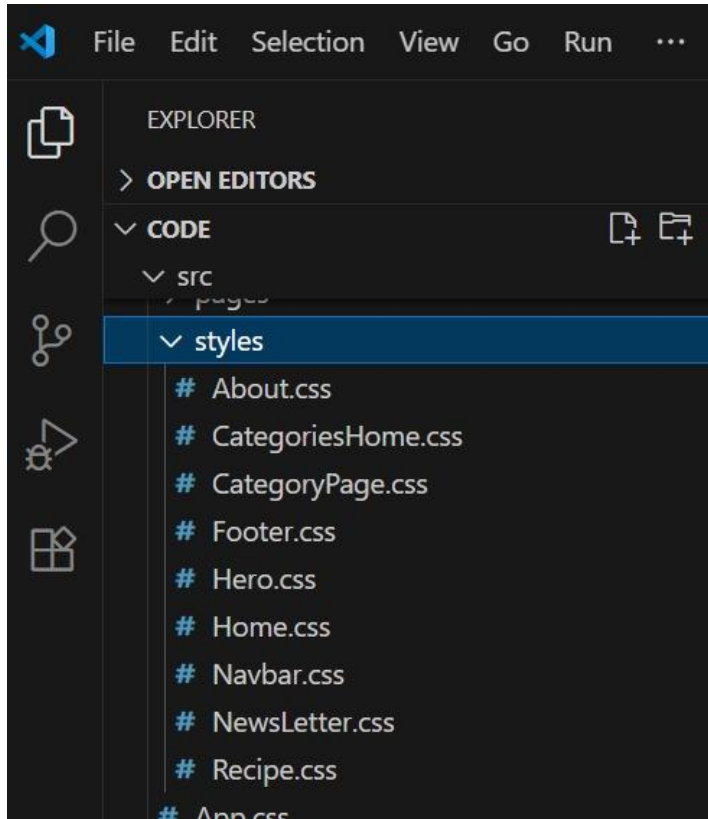
➤ **Insert src folder:**
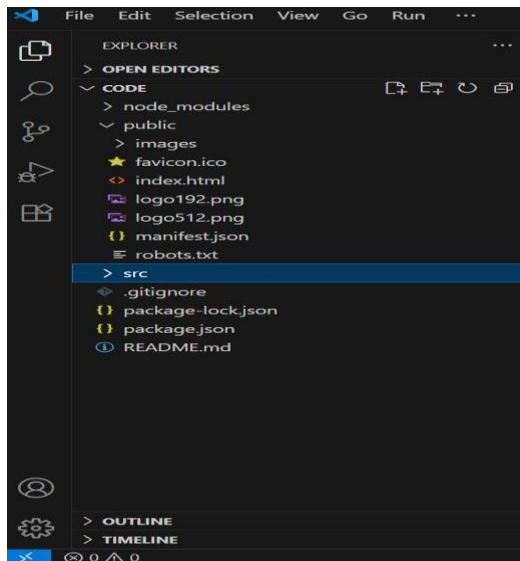
➤ src/**index.js:**

```
src > JS index.js > ...
 1   import React from "react";
 2   import ReactDOM from "react-dom/client";
 3   import { BrowserRouter } from "react-router-dom";
 4   import App from "./App";
 5
 6   const root = ReactDOM.createRoot(document.getElementById("root"));
 7   root.render(
 8     <BrowserRouter>
 9       <App />
10     </BrowserRouter>
11   );
12
```

➤ src/components/**CategoriesHome.jsx:**

```
CategoriesHome.jsx ×    HomePage.jsx      LandingPage.jsx    {} recipe.json      butter-chicken.jpg      Pa
src > components > CategoriesHome.jsx > ...
 1   import React, { useState } from "react";
 2   import { useNavigate } from "react-router-dom";
 3   import recipes from "../data/recipe.json";
 4
 5   function CategoriesHome() {
 6     const navigate = useNavigate();
 7     const [search, setSearch] = useState("");
 8     const [cuisine, setCuisine] = useState("All");
 9
10     // Filter recipes by search + cuisine
11     const filteredRecipes = recipes.filter((recipe) => {
12       const matchesSearch = recipe.title.toLowerCase().includes(search.toLowerCase());
13       const matchesCuisine = cuisine === "All" || recipe.cuisine === cuisine;
14       return matchesSearch && matchesCuisine;
15     });
16
17     return (
18       <div style={{ fontFamily: "'Poppins', sans-serif", padding: "40px" }}>
19         <h1 style={{ textAlign: "center", marginBottom: "30px" }}>  MOM'S DRIP COOKBOOK</h1>
20
21         {/* Search + Dropdown Controls */}
22         <div
23           style={{
24             display: "flex",
25             justifyContent: "center",
26             gap: "15px",
27             marginBottom: "30px"
28           }}
29         >
```

```
30         <input
31           type="text"
32           placeholder="  Search recipes..."
33           value={search}
34           onChange={(e) => setSearch(e.target.value)}
35           style={{
36             padding: "10px",
37             width: "250px",
38             borderRadius: "8px",
39             border: "1px solid  #ccc",
40             fontSize: "15px"
41           }}
42         />
43
44         <select
45           value={cuisine}
46           onChange={(e) => setCuisine(e.target.value)}
47           style={{
48             padding: "10px",
49             borderRadius: "8px",
50             border: "1px solid  #ccc",
51             fontSize: "15px"
52           }}
53         >
54           <option value="All">All Cuisines</option>
55           <option value="South Indian">South Indian</option>
56           <option value="North Indian">North Indian</option>
57         </select>
```

```jsx
 5    function CategoriesHome() {
60          {/* Recipe Grid */}
61          <div
62            style={{
63              display: "grid",
64              gridTemplateColumns: "repeat(auto-fill, minmax(250px, 1fr))",
65              gap: "25px"
66            }}
67          >
68            {filteredRecipes.length > 0 ? (
69              filteredRecipes.map((recipe) => (
70                <div
71                  key={recipe.id}
72                  onClick={() => navigate(`/recipe/${recipe.id}`)}
73                  style={{
74                    backgroundColor: "white",
75                    borderRadius: "12px",
76                    boxShadow: "0 4px 8px rgba(0,0,0,0.1)",
77                    cursor: "pointer",
78                    transition: "transform 0.2s, box-shadow 0.2s",
79                    overflow: "hidden",
80                    textAlign: "center"
81                  }}
82                  onMouseEnter={(e) => {
83                    e.currentTarget.style.transform = "scale(1.05)";
84                    e.currentTarget.style.boxShadow = "0 8px 16px rgba(0,0,0,0.2)";
85                  }}
86                  onMouseLeave={(e) => {
87                    e.currentTarget.style.transform = "scale(1)";
```

```jsx
 5    function CategoriesHome() {
69              filteredRecipes.map((recipe) => (
90                >
91                  <img
92                    src={recipe.image}
93                    alt={recipe.title}
94                    style={{
95                      width: "100%",
96                      height: "180px",
97                      objectFit: "cover"
98                    }}
99                  />
100                 <div style={{ padding: "15px" }}>
101                   <h3 style={{ margin: "10px 0", fontSize: "18px", color: "#2c3e50" }}>
102                     {recipe.title}
103                   </h3>
104                   <p style={{ fontSize: "14px", color: "#777" }}>{recipe.cuisine}</p>
105                 </div>
106               </div>
107             ))
108           ) : (
109             <p style={{ textAlign: "center", gridColumn: "1 / -1" }}>No recipes found.</p>
110           )}
111         </div>
112       </div>
113     );
114   }
115
116   export default CategoriesHome;
```

➢ src/components/**RecipeDetail.jsx:**

```jsx
import React, { useState } from "react";
import { useParams, useNavigate } from "react-router-dom";
import recipes from "../data/recipe.json";

function RecipeDetail() {
  const { id } = useParams();
  const navigate = useNavigate();
  const recipe = recipes.find((r) => r.id === parseInt(id));

  const [showIngredients, setShowIngredients] = useState(true);

  if (!recipe) return <h2>Recipe not found</h2>;

  return (
    <div style={{ fontFamily: "'Poppins', sans-serif", paddingBottom: "50px" }}>
      {/* Hero Section with Image + Title Overlay */}
      <div
        style={{
          position: "relative",
          textAlign: "center",
          marginBottom: "40px"
        }}
      >
        <img
          src={recipe.image}
          alt={recipe.title}
          style={{
            width: "100%",
            maxHeight: "400px",
```

```jsx
function RecipeDetail() {
            objectFit: "cover",
            borderBottom: "5px solid #eee"
          }}
        />
        <h1
          style={{
            position: "absolute",
            bottom: "20px",
            left: "50%",
            transform: "translateX(-50%)",
            color: "white",
            backgroundColor: "rgba(0,0,0,0.6)",
            padding: "10px 20px",
            borderRadius: "8px",
            fontSize: "28px"
          }}
        >
          {recipe.title}
        </h1>
      </div>

      {/* Content Layout: Steps + Ingredients */}
      <div
        style={{
          display: "flex",
          justifyContent: "center",
          alignItems: "flex-start",
          gap: "40px",
```

```jsx
          gap: "40px",
          padding: "0 40px"
        }}
      >
        {/* Steps Timeline */}
        <div style={{ flex: "2", textAlign: "left" }}>
          <h3 style={{ color: "#2c3e50", marginBottom: "20px" }}>📋 Steps</h3>
          <ol style={{ listStyle: "none", paddingLeft: "0" }}>
            {recipe.steps?.map((step, index) => (
              <li
                key={index}
                style={{
                  marginBottom: "20px",
                  position: "relative",
                  paddingLeft: "30px"
                }}
              >
                <span
                  style={{
                    position: "absolute",
                    left: "0",
                    top: "0",
                    backgroundColor: "#3498db",
                    color: "white",
                    borderRadius: "50%",
                    width: "22px",
                    height: "22px",
                    display: "flex",
```

```jsx
                    display: "flex",
                    justifyContent: "center",
                    alignItems: "center",
                    fontSize: "14px"
                  }}
                >
                    {index + 1}
                  </span>
                  <span style={{ fontSize: "16px", color: "#444" }}>{step}</span>
                </li>
            ))}
          </ol>
        </div>

        {/* Ingredients Box with Collapse */}
        <div
          style={{
            flex: "1",
            position: "sticky",
            top: "20px",
            border: "1px solid #ddd",
            borderRadius: "12px",
            padding: "20px",
            backgroundColor: "#fff",
            boxShadow: "0 6px 12px rgba(0,0,0,0.1)"
          }}
        >
```

```jsx
          <button
            onClick={() => setShowIngredients(!showIngredients)}
            style={{
              width: "100%",
              padding: "10px",
              backgroundColor: "#e67e22",
              color: "white",
              border: "none",
              borderRadius: "6px",
              cursor: "pointer",
              marginBottom: "15px",
              fontSize: "15px"
            }}
          >
            {showIngredients ? "Hide Ingredients ▲" : "Show Ingredients ▼"}
          </button>

          <div
            style={{
              maxHeight: showIngredients ? "500px" : "0px",
              overflow: "hidden",
              transition: "max-height 0.4s ease, opacity 0.3s ease",
              opacity: showIngredients ? 1 : 0
            }}
          >
            <h3
              style={{
                color: "#2c3e50",
```

```jsx
                textAlign: "center"
              }}
            >
              🍽 Ingredients
            </h3>
            <ul style={{ paddingLeft: "20px", lineHeight: "1.8" }}>
              {recipe.ingredients?.map((item, index) => (
                <li key={index} style={{ fontSize: "16px", marginBottom: "6px" }}>
                  {item}
                </li>
              ))}
            </ul>
          </div>
        </div>
      </div>

      {/* Back Button */}
      <div style={{ marginTop: "50px", textAlign: "center" }}>
        <button
          onClick={() => navigate("/")}
          style={{
            padding: "12px 25px",
            backgroundColor: "#3498db",
            color: "white",
            border: "none",
            borderRadius: "8px",
            cursor: "pointer",
            fontSize: "16px",
```

```
165              borderRadius: "8px",
166              cursor: "pointer",
167              fontSize: "16px",
168              boxShadow: "0 4px 8px □rgba(0,0,0,0.1)",
169              transition: "background 0.2s"
170          }}
171          onMouseEnter={(e) => (e.target.style.backgroundColor = "□#2980b9")}
172          onMouseLeave={(e) => (e.target.style.backgroundColor = "□#3498db")}
173        >
174          ← Back to Recipes
175        </button>
176      </div>
177    </div>
178  );
179 }
180
181 export default RecipeDetail;
182
```

➢ **Components Folder:**
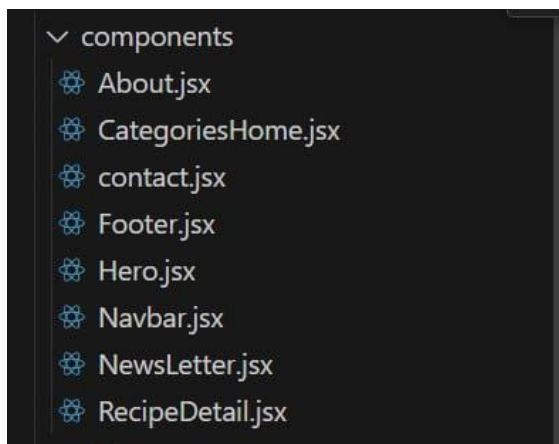
- src/pages/**Category.jsx:**

```
src > pages > ⚙ Category.jsx > ...
  1   import React from "react";
  2   import { useNavigate } from "react-router-dom";
  3   import recipes from "../data/recipe.json";
  4
  5   function CategoriesHome() {
  6     const navigate = useNavigate();
  7
  8     return (
  9       <div style={{ display: "flex", gap: "20px", flexWrap: "wrap", padding: "20px" }}>
 10         {recipes.map((recipe) => (
 11           <div
 12             key={recipe.id}
 13             style={{
 14               border: "1px solid ▨#ccc",
 15               borderRadius: "8px",
 16               padding: "10px",
 17               width: "250px",
 18               textAlign: "center",
 19               cursor: "pointer"
 20             }}
 21             onClick={() => navigate(`/recipe/${recipe.id}`)}
 22           >
 23             <img
 24               src={recipe.image}
 25               alt={recipe.title}
 26               style={{ width: "200px", height: "150px", objectFit: "cover" }}
 27             />
 28             <h3>{recipe.title}</h3>
 29           </div>
```

```
 23             <img
 24               src={recipe.image}
 25               alt={recipe.title}
 26               style={{ width: "200px", height: "150px", objectFit: "cover" }}
 27             />
 28             <h3>{recipe.title}</h3>
 29           </div>
 30         ))}
 31       </div>
 32     );
 33   }
 34
 35   export default CategoriesHome;
 36
```

- src/pages/**Recipe.jsx:**

```jsx
import React, { useEffect } from 'react'
import YouTube from 'react-youtube'
import '../styles/Recipie.css'
import { useNavigate, useParams } from 'react-router-dom'
import axios from 'axios'

const Recipie = () => {

  const navigate = useNavigate()

  const {id} = useParams();

  const [recipie, setRecipie] = React.useState()


  useEffect(() => {
    fetchRecipie()
  }, [])

  const fetchRecipie = async () => {
    await axios.get(`https://www.themealdb.com/api/json/v1/1/lookup.php?i=${id}`)
      .then(response => {
        setRecipie(response.data.meals[0])
        console.log(response.data.meals[0])
      })
      .catch(error => console.error(error));
  }
```

```jsx
  return (

    <>


        <div className='Recipie-page'>

      {recipie ?
          <>


              <div className="recipie-img">
                <img src={recipie.strMealThumb} alt="food-item" />
              </div>


              <div className="recipie-data-container">

                <div className="recipie-data">
                    <div className="recipie-header">

                      <h4>{recipie.strMeal}</h4>
                      <div className="recipie-specials">
                          <p>{recipie.strArea && recipie.strArea}</p>
                          <p>{recipie.strCategory && recipie.strCategory}</p>
                      </div>
                    </div>


                    <div className="procedure">
```

```
58                         <h5>Procedure</h5>
59                         <p>{recipie.strInstructions}</p>
60                     </div>

62                     {
63                       recipie.strYoutube !== "" &&

65                         <div className="youtube-video-container">
66                           <h5>Video Tutorial</h5>
67                           <YouTube className='youtube-video'
68                             videoId={recipie.strYoutube.slice(32)}
69                             opts={{
70                               height: '315',
71                               width: '560',
72                             }}
73                           />
74                         </div>
75                     }

78 ...................</div>

80                   <div className="ingredients-container">

82                     <h3>Ingredients</h3>

84                       <ul className="ingredients">
85
```

```
86                       {Object.entries(recipie).map(([key, value]) => {
87                           if (key.startsWith("strIngredient") && value) {
88                             const ingredientNumber = key.slice("strIngredient".length);
89                             const measure = recipie[`strMeasure${ingredientNumber}`] || "";

91                             return (
92                               <li key={key} className="ingredient">
93                                 <h5>{ingredientNumber} - {value}</h5>
94                                 <p>{measure}</p>
95                               </li>
96                             );
97                           }
98                           return null;
99                       })}

101                     </ul>

103                 </div>

105               </div>

109         </>

111 :"Loading..."}

113       </div>
```

➢ src/**App.js:**

```js
JS App.js          ✕

src > JS App.js > ...
    1    import React from "react";
    2    import { Routes, Route } from "react-router-dom";
    3    import CategoriesHome from "./components/CategoriesHome";
    4    import RecipeDetail from "./components/RecipeDetail";
    5
    6    function App() {
    7      return (
    8        <Routes>
    9          <Route path="/" element={<CategoriesHome />} />
   10          <Route path="/recipe/:id" element={<RecipeDetail />} />
   11        </Routes>
   12      );
   13    }
   14
   15    export default App;
   16
```

➢ **Project Setup and Configuration:**

Installation of Required Tools:

To build the Cookbook Web App, we installed modern toolkit. In This project we uses React.js for building the interactive interface, React Router DOM for navigation, and Axios for fetching recipe data from external APIs. For visual styling, we use Bootstrap, along with React Icons for intuitive UI elements.

Technologies Used:

• React.js – Core framework for building the UI

• React Router DOM – Enables client-side routine

• Axios – Handles API requests and data fetching

• React Icons – Provides scalable vector icons

• Bootstrap – For responsive and elegant styling

we get started with react using

https://react.dev/learn/installation

we learn how to use bootstrap components using
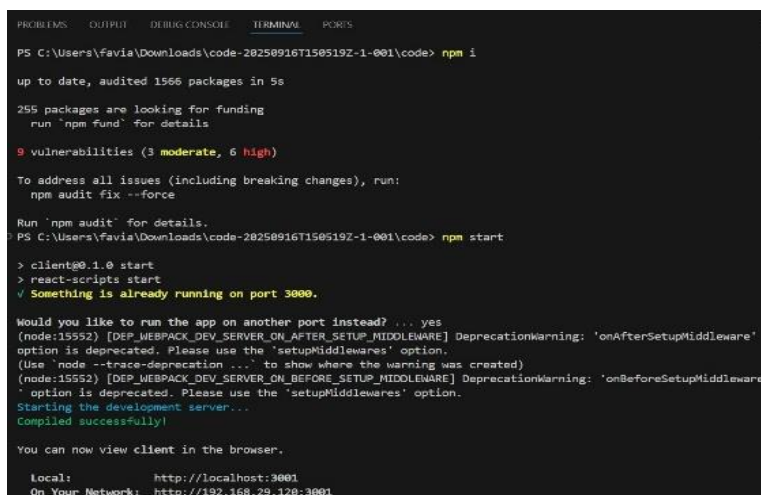
https://react-bootstrap-v4.netlify.app/getting-started/introduction/

Axios documentation teaches us how to make, customize, and manage HTTP requests efficiently in JavaScript and Node.js using

https://axios-http.com/docs/intro

➢ **Project Execution:**

   Project Execution shows the output of the coding they are home page, login page, dashboard

page, recipe page.

➢ **Demo link:**
   https://drive.google.com/file/d/15eTeQRDO3fa2mErrQWsXdJ5jrBHgOgAU/view?usp=drivesdk