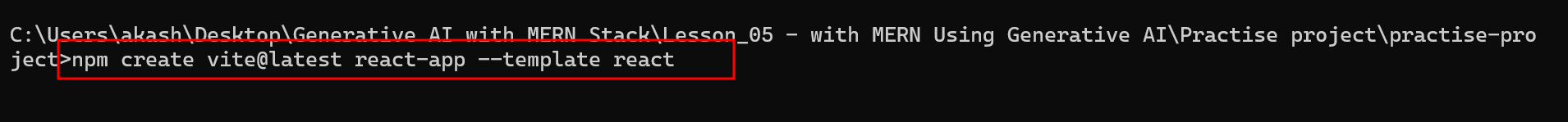
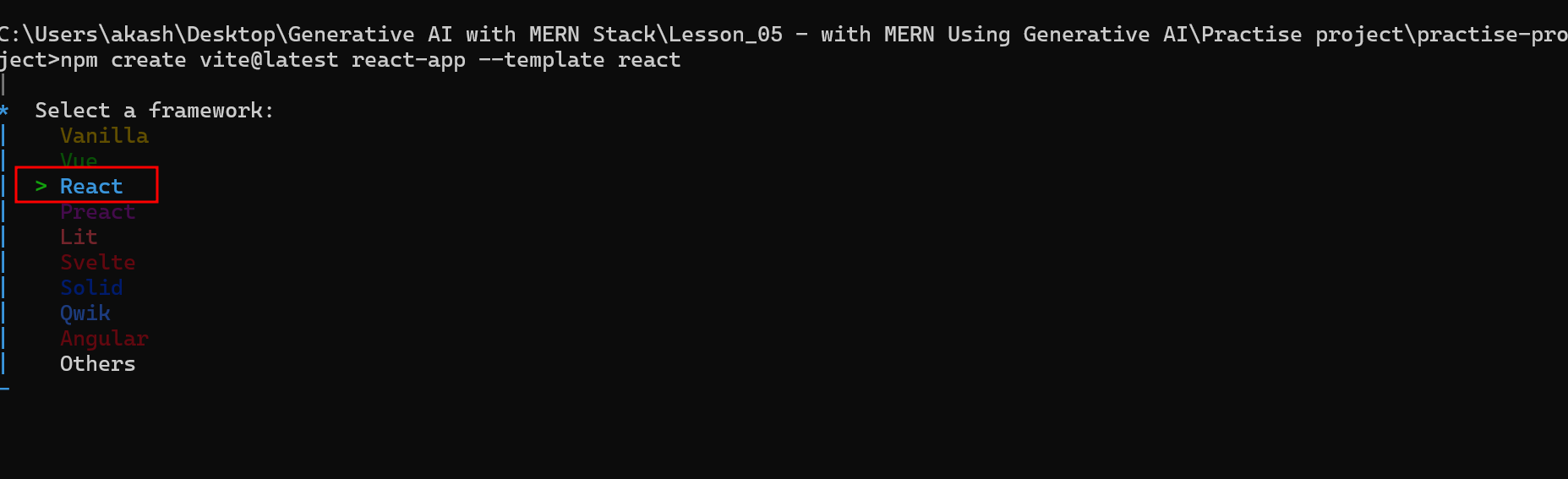
**Answer Key**

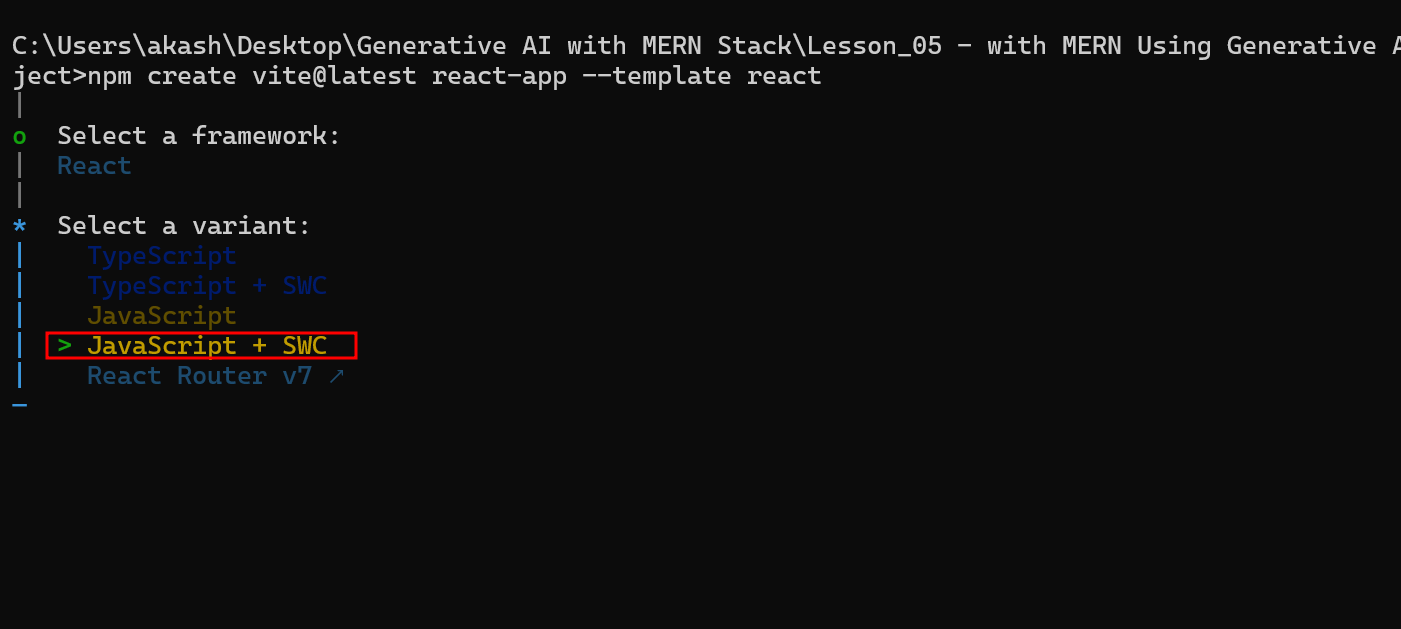
1. **Set up the project in Visual Studio Code and GitHub utilizing npx and GitHub Copilot for code generation**

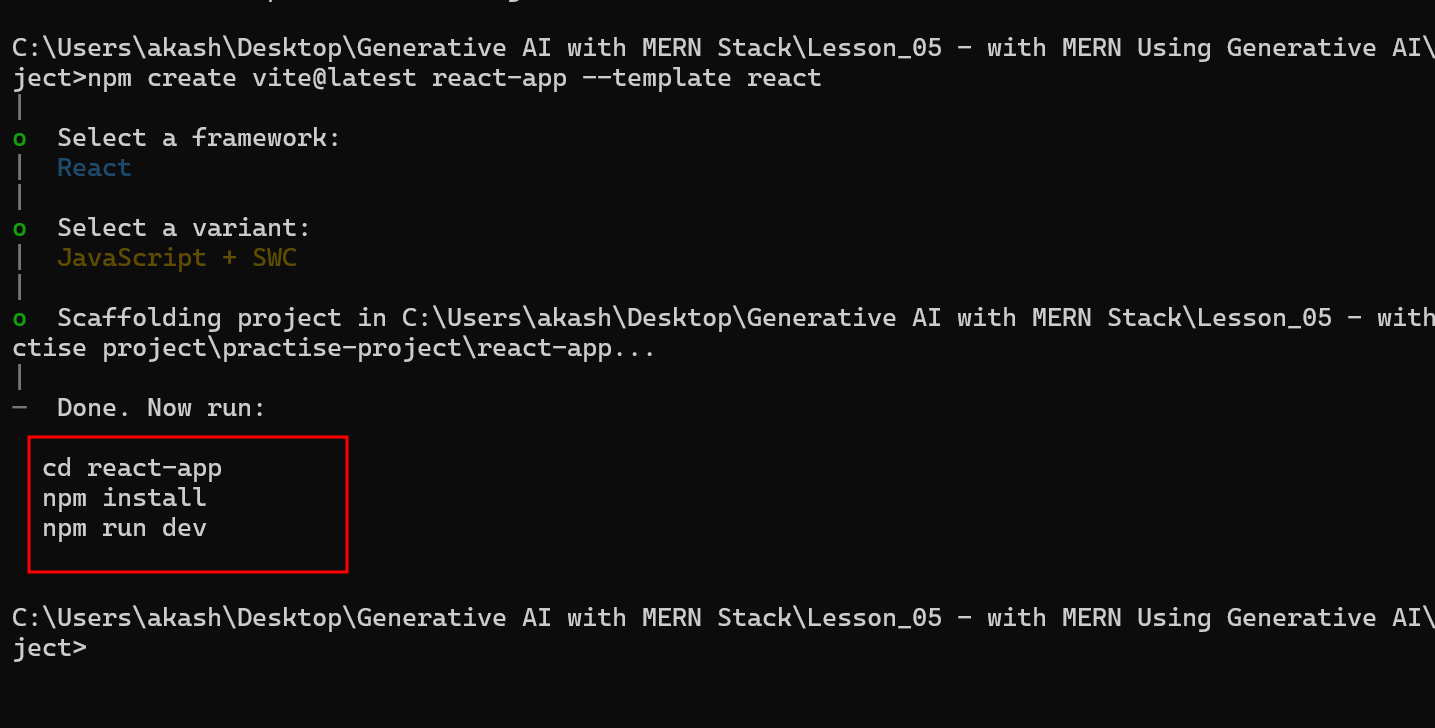
* 1. Set up a new React.js project with the default project structure and configuration using the following command:

**npm create vite@latest react-app --template react**





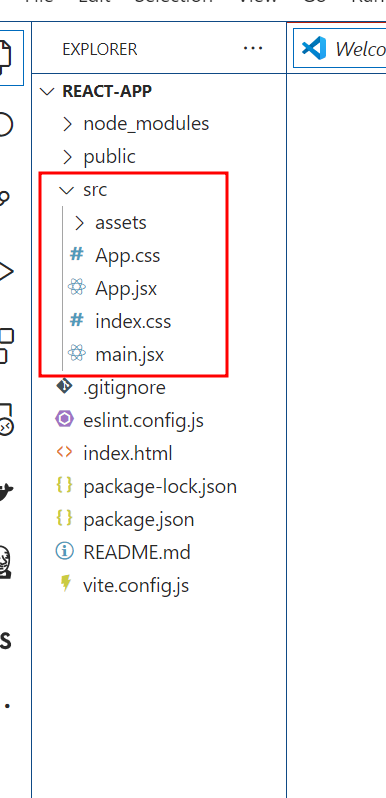




**cd react-app** move inside a project

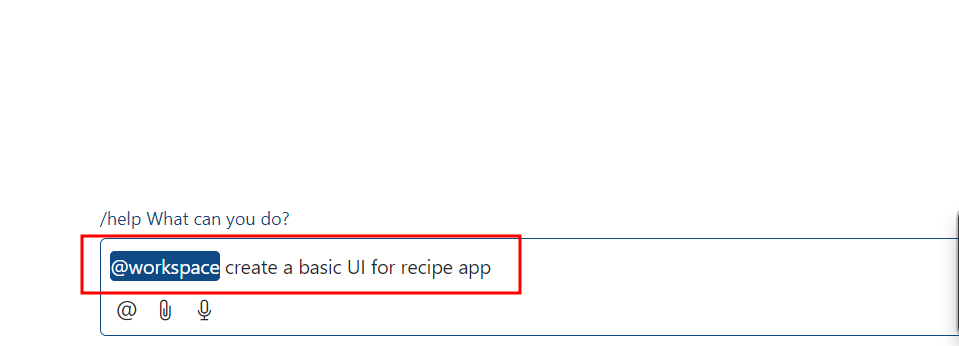
**npm install** installed required dependencies to run the react js project

The basic React.js structure appears as follows:



1. **Generate a basic user interface for recipe data using GitHub Copilot**
   1. Enhance the project by prompting GitHub Copilot to generate React.js code for a basic UI for the recipe app

**@workspace create a basic UI for recipe app**

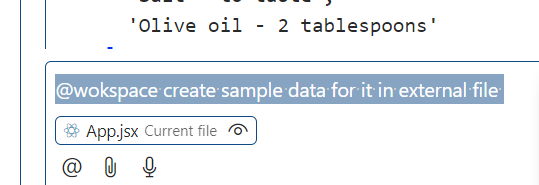


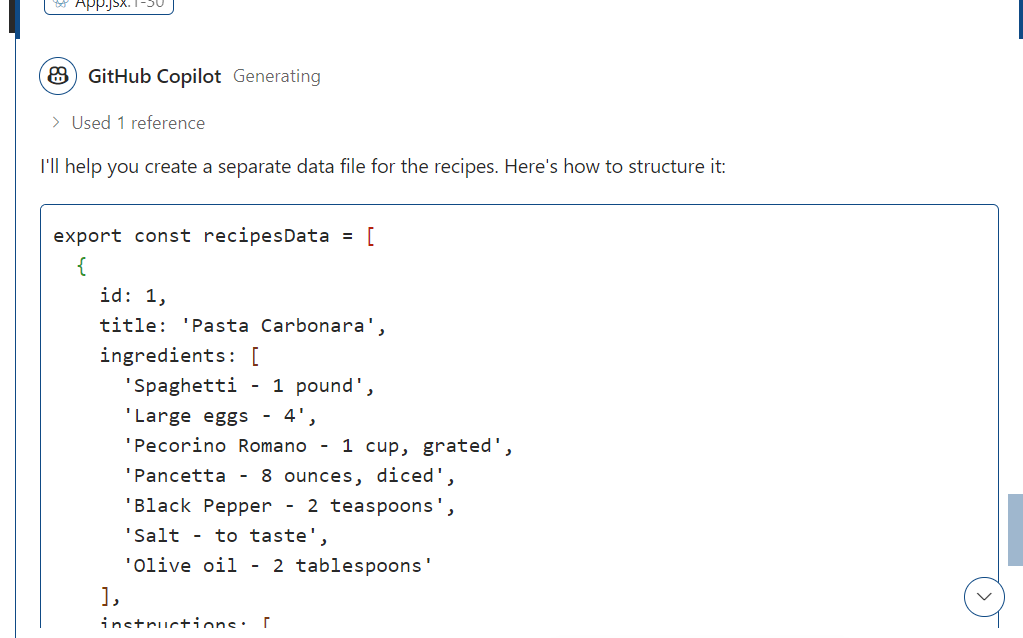
**Note:** Copy the contents of the generated script and paste them into the existing **src\App.jsx** file



1. **Generate sample data using GitHub Copilot**
   1. Prompt GitHub Copilot to create sample data for the existing recipe app

**@wokspace create sample data for it in external file**





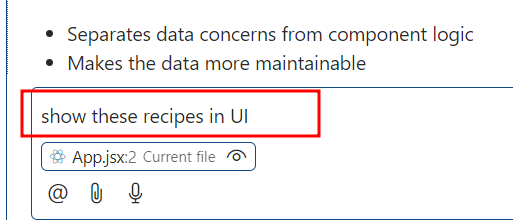
Create the folder with name as data and inside that create the recipes.js file and copy/paste the generate code.

* 1. Open **App.jsx** and import the newly created **data/recipe.js** by adding the following script:

import { recipesData } from './data/recipes'

1. **Generate React code to display the recipes using the user interface**
   1. Prompt GitHub Copilot to generate code to display the recipe data using the   
      user interface

**show these recipes in UI**





**App.jsx**

import { useState } from 'react'

import { recipesData } from './data/recipes'

import './App.css'

function App() {

  const [recipes, setRecipes] = useState(recipesData)

  return (

    <div className="app">

      <h1>My Recipe Book</h1>

      <div className="recipes-container">

        {recipes.map(recipe => (

          <div key={recipe.id} className="recipe-card">

            <h2>{recipe.title}</h2>

            <div className="recipe-meta">

              <span>Category: {recipe.category}</span>

              <span>Prep: {recipe.prepTime}</span>

              <span>Cook: {recipe.cookTime}</span>

              <span>Servings: {recipe.servings}</span>

            </div>

            <div className="recipe-content">

              <div className="ingredients">

                <h3>Ingredients:</h3>

                <ul>

                  {recipe.ingredients.map((ingredient, index) => (

                    <li key={index}>{ingredient}</li>

                  ))}

                </ul>

              </div>

              <div className="instructions">

                <h3>Instructions:</h3>

                <ol>

                  {recipe.instructions.map((step, index) => (

                    <li key={index}>{step}</li>

                  ))}

                </ol>

              </div>

            </div>

          </div>

        ))}

      </div>

    </div>

  )

}

export default App

Add new styling rule in App.css file

**App.css**

#root {

  max-width: 1280px;

  margin: 0 auto;

  padding: 2rem;

  text-align: center;

}

.logo {

  height: 6em;

  padding: 1.5em;

  will-change: filter;

  transition: filter 300ms;

}

.logo:hover {

  filter: drop-shadow(0 0 2em #646cffaa);

}

.logo.react:hover {

  filter: drop-shadow(0 0 2em #61dafbaa);

}

@keyframes logo-spin {

  from {

    transform: rotate(0deg);

  }

  to {

    transform: rotate(360deg);

  }

}

@media (prefers-reduced-motion: no-preference) {

  a:nth-of-type(2) .logo {

    animation: logo-spin infinite 20s linear;

  }

}

.card {

  padding: 2em;

}

.read-the-docs {

  color: #888;

}

.app {

  max-width: 1200px;

  margin: 0 auto;

  padding: 2rem;

}

h1 {

  text-align: center;

  color: #333;

  margin-bottom: 2rem;

}

.recipes-container {

  display: grid;

  grid-template-columns: repeat(auto-fit, minmax(350px, 1fr));

  gap: 2rem;

}

.recipe-card {

  background: white;

  border-radius: 10px;

  padding: 1.5rem;

  box-shadow: 0 2px 5px rgba(0,0,0,0.1);

}

.recipe-card h2 {

  color: #2c3e50;

  margin-bottom: 1rem;

  border-bottom: 2px solid #3498db;

  padding-bottom: 0.5rem;

}

.recipe-meta {

  display: flex;

  flex-wrap: wrap;

  gap: 1rem;

  margin-bottom: 1.5rem;

  font-size: 0.9rem;

}

.recipe-meta span {

  background: #f0f0f0;

  padding: 0.3rem 0.8rem;

  border-radius: 15px;

  color: #666;

}

.recipe-content {

  display: flex;

  flex-direction: column;

  gap: 1.5rem;

}

.ingredients ul {

  list-style-type: disc;

  padding-left: 1.5rem;

}

.instructions ol {

  padding-left: 1.5rem;

  line-height: 1.6;

}

.instructions li {

  margin-bottom: 0.5rem;

}

h3 {

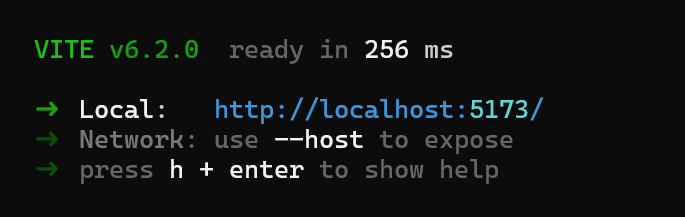
  color: #2c3e50;

  margin-bottom: 0.8rem;

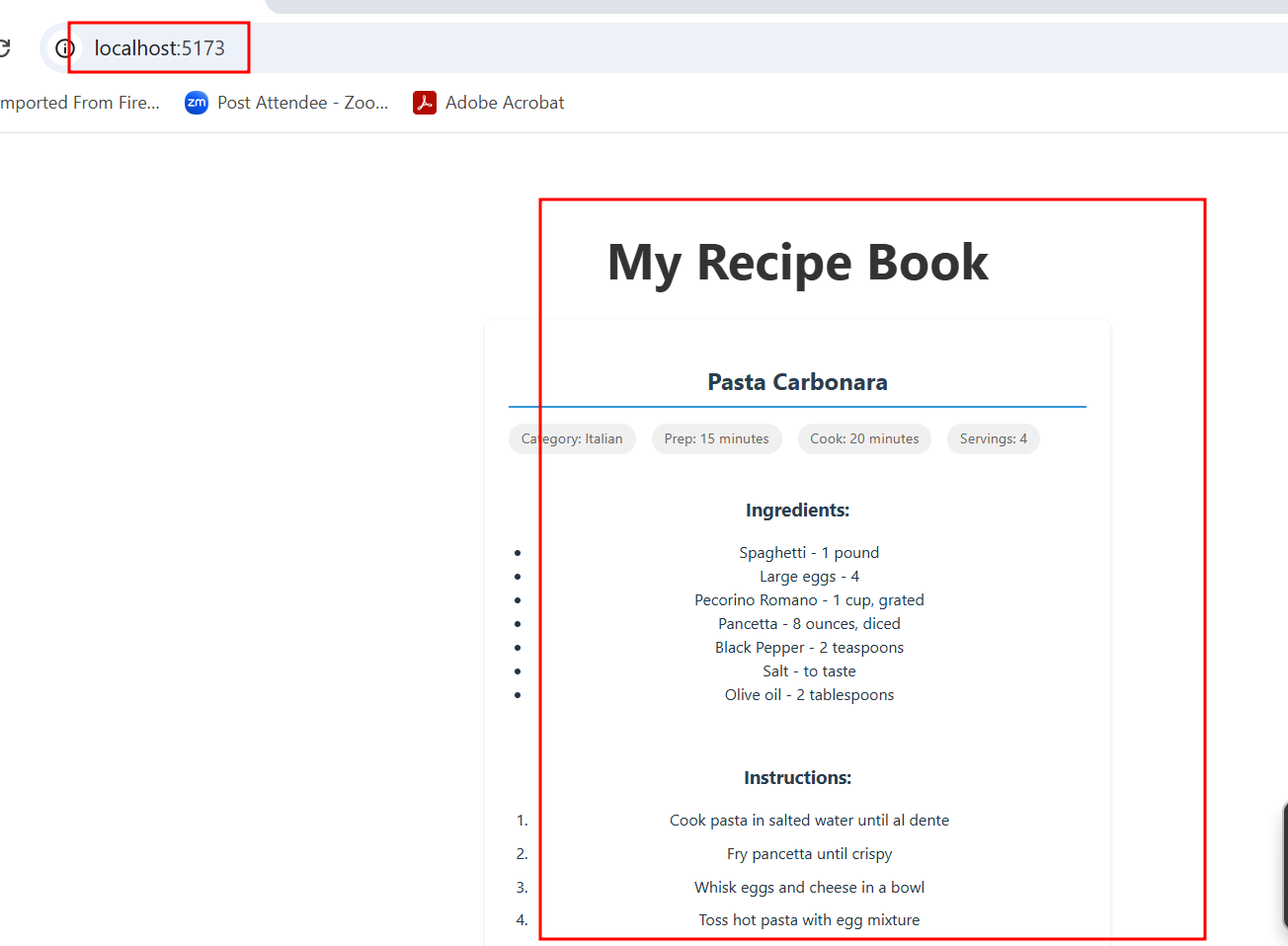
}

* 1. Run the react project after updating App.js

**npm run dev**

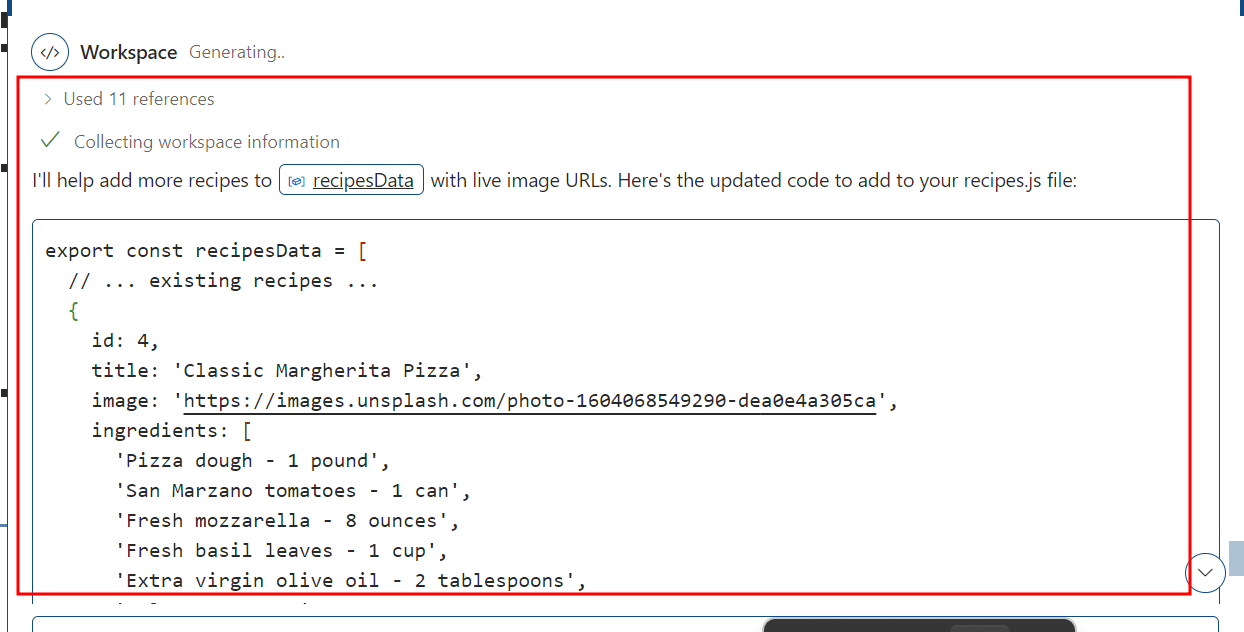


**Output on browser as**



1. **Generate a React code to add more recipes to the recipe app**
   1. Improve the recipe app by adding more recipes using GitHub Copilot prompts

**@workspace add more recipes with images with live image url**



The following React.js code will be generated by GitHub Copilot:

**Update the recipes.js file code**

export const recipesData = [

  {

    id: 4,

    title: 'Classic Margherita Pizza',

    image: 'https://images.unsplash.com/photo-1604068549290-dea0e4a305ca',

    ingredients: [

      'Pizza dough - 1 pound',

      'San Marzano tomatoes - 1 can',

      'Fresh mozzarella - 8 ounces',

      'Fresh basil leaves - 1 cup',

      'Extra virgin olive oil - 2 tablespoons',

      'Salt - to taste'

    ],

    instructions: [

      'Preheat oven to 500°F with pizza stone',

      'Stretch dough into 12-inch circle',

      'Top with crushed tomatoes',

      'Add torn mozzarella pieces',

      'Bake for 12-15 minutes',

      'Garnish with fresh basil and olive oil'

    ],

    category: 'Italian',

    prepTime: '25 minutes',

    cookTime: '15 minutes',

    servings: 4

  },

  {

    id: 5,

    title: 'Sushi Roll',

    image: 'https://images.unsplash.com/photo-1579871494447-9811cf80d66c',

    ingredients: [

      'Sushi rice - 2 cups',

      'Nori sheets - 4',

      'Fresh salmon - 8 ounces',

      'Cucumber - 1, julienned',

      'Avocado - 1, sliced',

      'Rice vinegar - 1/4 cup',

      'Wasabi and soy sauce - to serve'

    ],

    instructions: [

      'Cook and season rice with vinegar',

      'Place nori on bamboo mat',

      'Spread rice on nori',

      'Add fish and vegetables',

      'Roll tightly using mat',

      'Cut into 8 pieces'

    ],

    category: 'Japanese',

    prepTime: '30 minutes',

    cookTime: '20 minutes',

    servings: 4

  },

  {

    id: 6,

    title: 'Thai Green Curry',

    image: 'https://images.unsplash.com/photo-1455619452474-d2be8b1e70cd',

    ingredients: [

      'Coconut milk - 2 cans',

      'Green curry paste - 4 tablespoons',

      'Chicken breast - 1 pound, sliced',

      'Thai eggplants - 4, quartered',

      'Bamboo shoots - 1 cup',

      'Thai basil - 1 cup',

      'Fish sauce - 2 tablespoons',

      'Palm sugar - 1 tablespoon'

    ],

    instructions: [

      'Simmer coconut milk until oil separates',

      'Stir in curry paste and cook until fragrant',

      'Add chicken and cook until nearly done',

      'Add vegetables and remaining coconut milk',

      'Season with fish sauce and palm sugar',

      'Garnish with Thai basil'

    ],

    category: 'Thai',

    prepTime: '20 minutes',

    cookTime: '25 minutes',

    servings: 4

  }

]

**Update new content for App.jsx as well as App.css file**

**App.jsx**

import { useState } from 'react'

import { recipesData } from './data/recipes'

import './App.css'

function App() {

  const [recipes, setRecipes] = useState(recipesData)

  return (

    <div className="app">

      <h1>My Recipe Book</h1>

      <div className="recipes-container">

        {recipes.map(recipe => (

          <div key={recipe.id} className="recipe-card">

            {recipe.image && (

              <div className="recipe-image">

                <img src={recipe.image} alt={recipe.title} />

              </div>

            )}

            <h2>{recipe.title}</h2>

            <div className="recipe-meta">

              <span>Category: {recipe.category}</span>

              <span>Prep: {recipe.prepTime}</span>

              <span>Cook: {recipe.cookTime}</span>

              <span>Servings: {recipe.servings}</span>

            </div>

            <div className="recipe-content">

              <div className="ingredients">

                <h3>Ingredients:</h3>

                <ul>

                  {recipe.ingredients.map((ingredient, index) => (

                    <li key={index}>{ingredient}</li>

                  ))}

                </ul>

              </div>

              <div className="instructions">

                <h3>Instructions:</h3>

                <ol>

                  {recipe.instructions.map((step, index) => (

                    <li key={index}>{step}</li>

                  ))}

                </ol>

              </div>

            </div>

          </div>

        ))}

      </div>

    </div>

  )

}

export default App

**App.css**

#root {

  max-width: 1280px;

  margin: 0 auto;

  padding: 2rem;

  text-align: center;

}

.logo {

  height: 6em;

  padding: 1.5em;

  will-change: filter;

  transition: filter 300ms;

}

.logo:hover {

  filter: drop-shadow(0 0 2em #646cffaa);

}

.logo.react:hover {

  filter: drop-shadow(0 0 2em #61dafbaa);

}

@keyframes logo-spin {

  from {

    transform: rotate(0deg);

  }

  to {

    transform: rotate(360deg);

  }

}

@media (prefers-reduced-motion: no-preference) {

  a:nth-of-type(2) .logo {

    animation: logo-spin infinite 20s linear;

  }

}

.card {

  padding: 2em;

}

.read-the-docs {

  color: #888;

}

.app {

  max-width: 1200px;

  margin: 0 auto;

  padding: 2rem;

}

h1 {

  text-align: center;

  color: #333;

  margin-bottom: 2rem;

}

.recipes-container {

  display: grid;

  grid-template-columns: repeat(auto-fit, minmax(350px, 1fr));

  gap: 2rem;

}

.recipe-card {

  background: white;

  border-radius: 10px;

  padding: 1.5rem;

  box-shadow: 0 2px 5px rgba(0,0,0,0.1);

  display: flex;

  flex-direction: column;

  overflow: hidden;

}

.recipe-card h2 {

  color: #2c3e50;

  margin-bottom: 1rem;

  border-bottom: 2px solid #3498db;

  padding-bottom: 0.5rem;

}

.recipe-meta {

  display: flex;

  flex-wrap: wrap;

  gap: 1rem;

  margin-bottom: 1.5rem;

  font-size: 0.9rem;

}

.recipe-meta span {

  background: #f0f0f0;

  padding: 0.3rem 0.8rem;

  border-radius: 15px;

  color: #666;

}

.recipe-content {

  display: flex;

  flex-direction: column;

  gap: 1.5rem;

}

.recipe-image {

  width: 100%;

  height: 200px;

  overflow: hidden;

  border-radius: 8px 8px 0 0;

  margin-bottom: 1rem;

}

.recipe-image img {

  width: 100%;

  height: 100%;

  object-fit: cover;

  transition: transform 0.3s ease;

}

.recipe-image img:hover {

  transform: scale(1.05);

}

.ingredients ul {

  list-style-type: disc;

  padding-left: 1.5rem;

}

.instructions ol {

  padding-left: 1.5rem;

  line-height: 1.6;

}

.instructions li {

  margin-bottom: 0.5rem;

}

h3 {

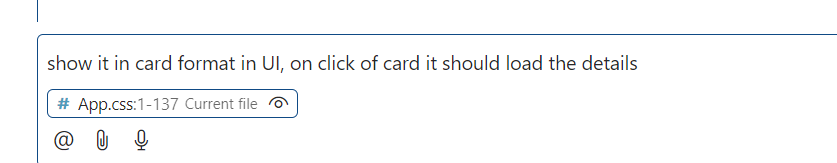
  color: #2c3e50;

  margin-bottom: 0.8rem;

}

1. **Create the Recipe Detail Component to display detailed information about each recipe**
   1. Prompt GitHub Copilot to generate a React code for displaying details of each recipe

**show all receipts in card format in UI, on click particular image display the recipe details apply good look and feel using CSS**



**Note:** Copy the contents of the generated script and paste them into the existing **src\App.jsx** as well as App.css file

**Apply new updated in App.js and App.css file**

**App.jsx**

import { useState } from 'react'

import { recipesData } from './data/recipes'

import './App.css'

function App() {

  const [recipes, setRecipes] = useState(recipesData)

  const [selectedRecipe, setSelectedRecipe] = useState(null)

  const handleRecipeClick = (recipe) => {

    setSelectedRecipe(recipe)

  }

  const handleClose = () => {

    setSelectedRecipe(null)

  }

  return (

    <div className="app">

      <h1>My Recipe Book</h1>

      <div className="recipes-container">

        {recipes.map(recipe => (

          <div

            key={recipe.id}

            className="recipe-card"

            onClick={() => handleRecipeClick(recipe)}

          >

            <div className="recipe-image">

              <img src={recipe.image} alt={recipe.title} />

            </div>

            <div className="recipe-card-content">

              <h2>{recipe.title}</h2>

              <div className="recipe-meta">

                <span>{recipe.category}</span>

                <span>{recipe.prepTime}</span>

              </div>

            </div>

          </div>

        ))}

      </div>

      {selectedRecipe && (

        <div className="recipe-modal-overlay" onClick={handleClose}>

          <div className="recipe-modal" onClick={e => e.stopPropagation()}>

            <button className="close-button" onClick={handleClose}>&times;</button>

            <div className="recipe-detail">

              <div className="recipe-detail-image">

                <img src={selectedRecipe.image} alt={selectedRecipe.title} />

              </div>

              <h2>{selectedRecipe.title}</h2>

              <div className="recipe-meta">

                <span>Category: {selectedRecipe.category}</span>

                <span>Prep: {selectedRecipe.prepTime}</span>

                <span>Cook: {selectedRecipe.cookTime}</span>

                <span>Servings: {selectedRecipe.servings}</span>

              </div>

              <div className="recipe-content">

                <div className="ingredients">

                  <h3>Ingredients</h3>

                  <ul>

                    {selectedRecipe.ingredients.map((ingredient, index) => (

                      <li key={index}>{ingredient}</li>

                    ))}

                  </ul>

                </div>

                <div className="instructions">

                  <h3>Instructions</h3>

                  <ol>

                    {selectedRecipe.instructions.map((step, index) => (

                      <li key={index}>{step}</li>

                    ))}

                  </ol>

                </div>

              </div>

            </div>

          </div>

        </div>

      )}

    </div>

  )

}

export default App

**App.css file**

#root {

  max-width: 1280px;

  margin: 0 auto;

  padding: 2rem;

  text-align: center;

}

.logo {

  height: 6em;

  padding: 1.5em;

  will-change: filter;

  transition: filter 300ms;

}

.logo:hover {

  filter: drop-shadow(0 0 2em #646cffaa);

}

.logo.react:hover {

  filter: drop-shadow(0 0 2em #61dafbaa);

}

@keyframes logo-spin {

  from {

    transform: rotate(0deg);

  }

  to {

    transform: rotate(360deg);

  }

}

@media (prefers-reduced-motion: no-preference) {

  a:nth-of-type(2) .logo {

    animation: logo-spin infinite 20s linear;

  }

}

.card {

  padding: 2em;

}

.read-the-docs {

  color: #888;

}

.app {

  max-width: 1200px;

  margin: 0 auto;

  padding: 2rem;

}

h1 {

  text-align: center;

  color: #2c3e50;

  margin-bottom: 3rem;

  font-size: 2.5rem;

}

.recipes-container {

  display: grid;

  grid-template-columns: repeat(auto-fill, minmax(280px, 1fr));

  gap: 2rem;

  padding: 1rem;

}

.recipe-card {

  background: white;

  border-radius: 12px;

  overflow: hidden;

  box-shadow: 0 4px 15px rgba(0,0,0,0.1);

  transition: transform 0.3s ease, box-shadow 0.3s ease;

  cursor: pointer;

}

.recipe-card:hover {

  transform: translateY(-5px);

  box-shadow: 0 8px 20px rgba(0,0,0,0.15);

}

.recipe-card h2 {

  margin: 0;

  color: #2c3e50;

  font-size: 1.25rem;

  margin-bottom: 0.5rem;

}

.recipe-meta {

  display: flex;

  gap: 0.8rem;

  flex-wrap: wrap;

}

.recipe-meta span {

  background: #f0f2f5;

  padding: 0.4rem 0.8rem;

  border-radius: 20px;

  font-size: 0.875rem;

  color: #666;

}

.recipe-content {

  display: grid;

  grid-template-columns: 1fr 1fr;

  gap: 2rem;

}

.recipe-image {

  width: 100%;

  height: 200px;

  overflow: hidden;

}

.recipe-image img {

  width: 100%;

  height: 100%;

  object-fit: cover;

  transition: transform 0.3s ease;

}

.recipe-card:hover .recipe-image img {

  transform: scale(1.05);

}

.recipe-card-content {

  padding: 1.5rem;

}

.ingredients ul {

  list-style-type: disc;

  padding-left: 1.5rem;

}

.instructions ol {

  padding-left: 1.5rem;

  line-height: 1.6;

}

.instructions li {

  margin-bottom: 0.5rem;

}

h3 {

  color: #2c3e50;

  margin-bottom: 1rem;

  font-size: 1.25rem;

}

.recipe-modal-overlay {

  position: fixed;

  top: 0;

  left: 0;

  right: 0;

  bottom: 0;

  background: rgba(0, 0, 0, 0.75);

  display: flex;

  justify-content: center;

  align-items: center;

  padding: 2rem;

  z-index: 1000;

}

.recipe-modal {

  background: white;

  border-radius: 12px;

  padding: 2rem;

  max-width: 800px;

  width: 90%;

  max-height: 90vh;

  overflow-y: auto;

  position: relative;

}

.close-button {

  position: absolute;

  top: 1rem;

  right: 1rem;

  background: none;

  border: none;

  font-size: 2rem;

  cursor: pointer;

  color: #666;

  transition: color 0.3s ease;

}

.close-button:hover {

  color: #333;

}

.recipe-detail-image {

  width: 100%;

  height: 300px;

  border-radius: 8px;

  overflow: hidden;

  margin-bottom: 2rem;

}

.recipe-detail-image img {

  width: 100%;

  height: 100%;

  object-fit: cover;

}

.recipe-detail h2 {

  color: #2c3e50;

  margin-bottom: 1rem;

  font-size: 1.75rem;

}

.recipe-detail .recipe-meta {

  margin-bottom: 2rem;

}

@media (max-width: 768px) {

  .recipe-content {

    grid-template-columns: 1fr;

  }

}

.ingredients ul {

  list-style-type: disc;

  padding-left: 1.5rem;

  margin: 0;

}

.instructions ol {

  padding-left: 1.5rem;

  margin: 0;

}

.ingredients li,

.instructions li {

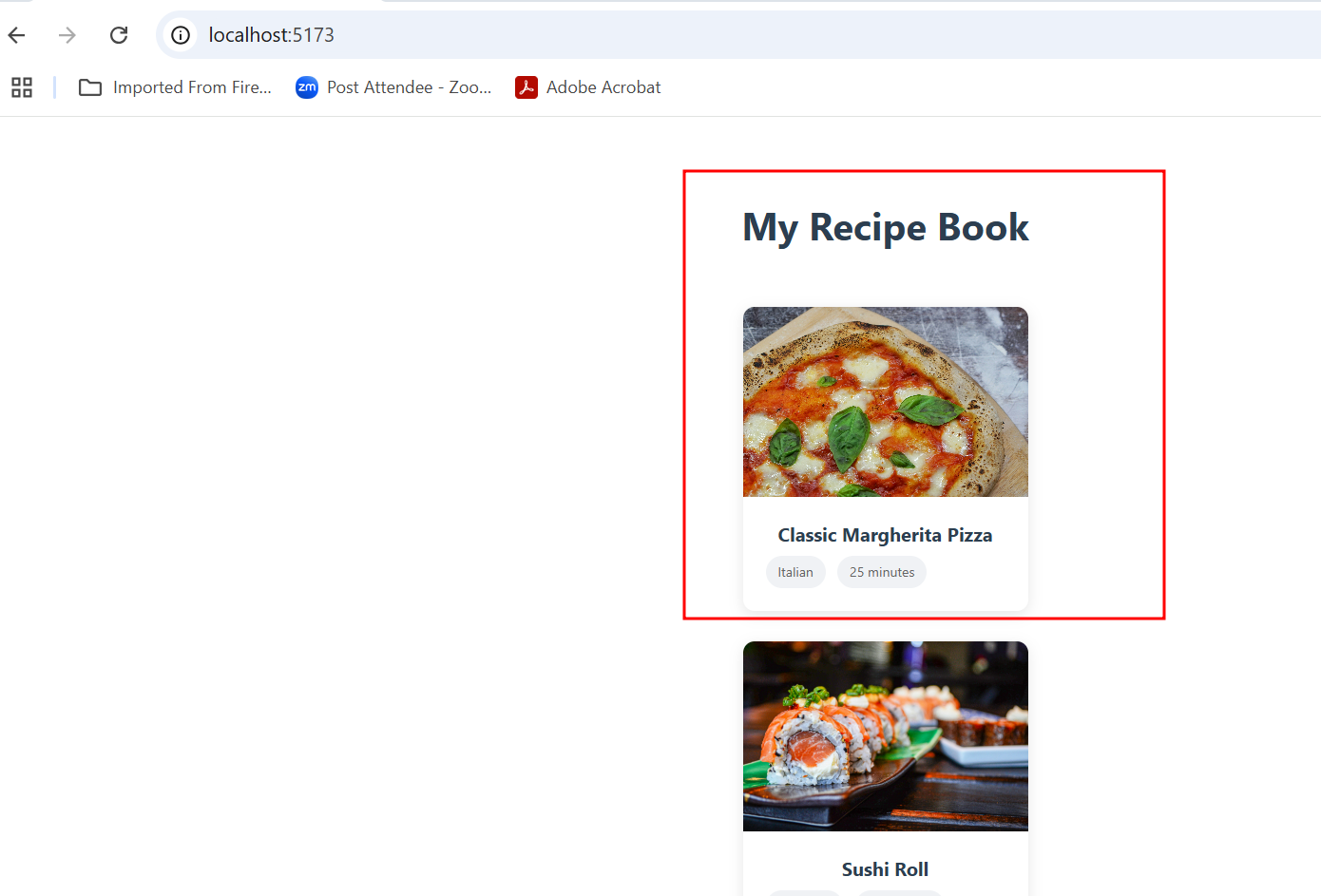
  margin-bottom: 0.5rem;

  line-height: 1.6;

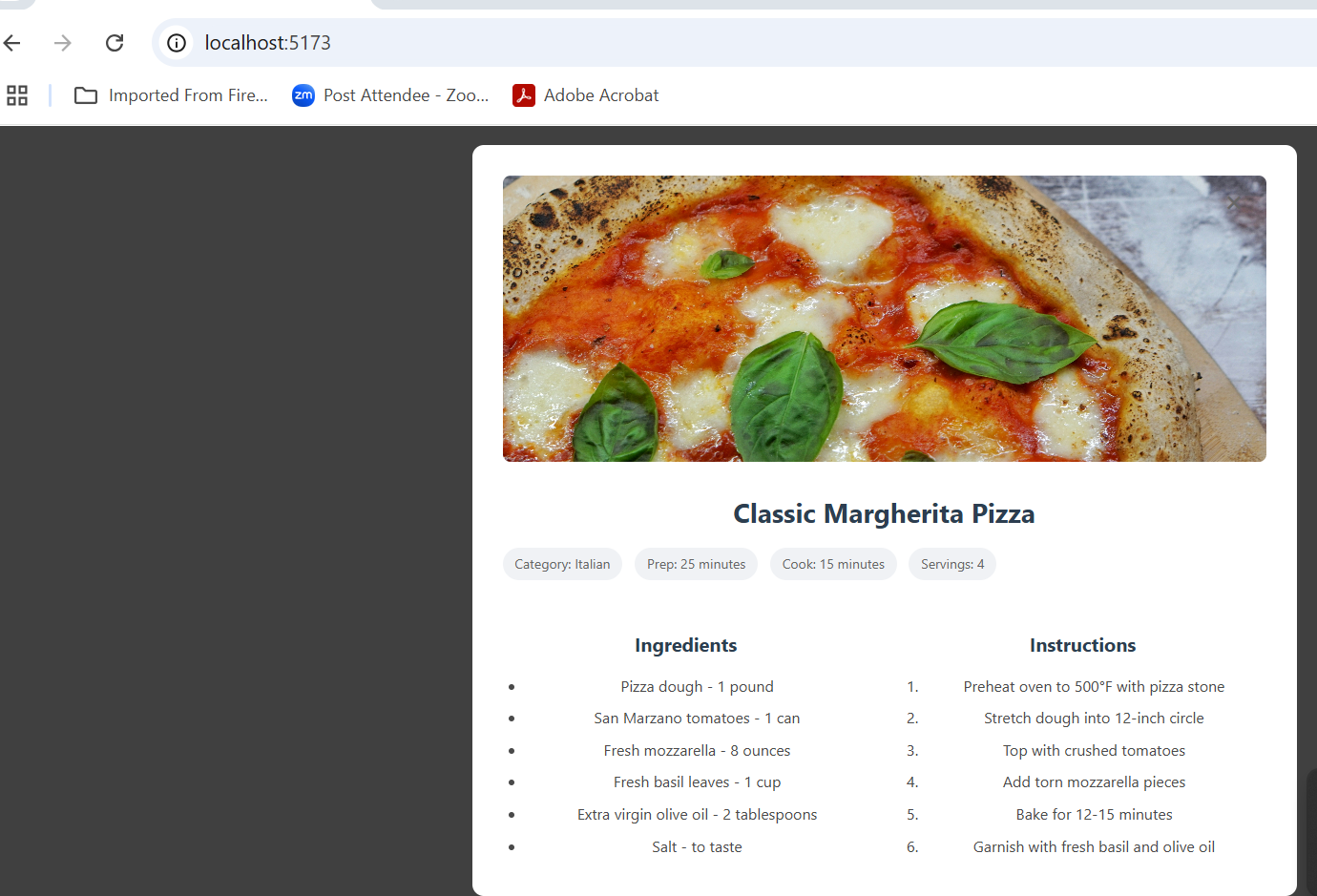
  color: #4a4a4a;

}

Now check updated output



When you click on particular image you get that image details



1. **Implement search functionality to enable users to search for   
   specific recipes**

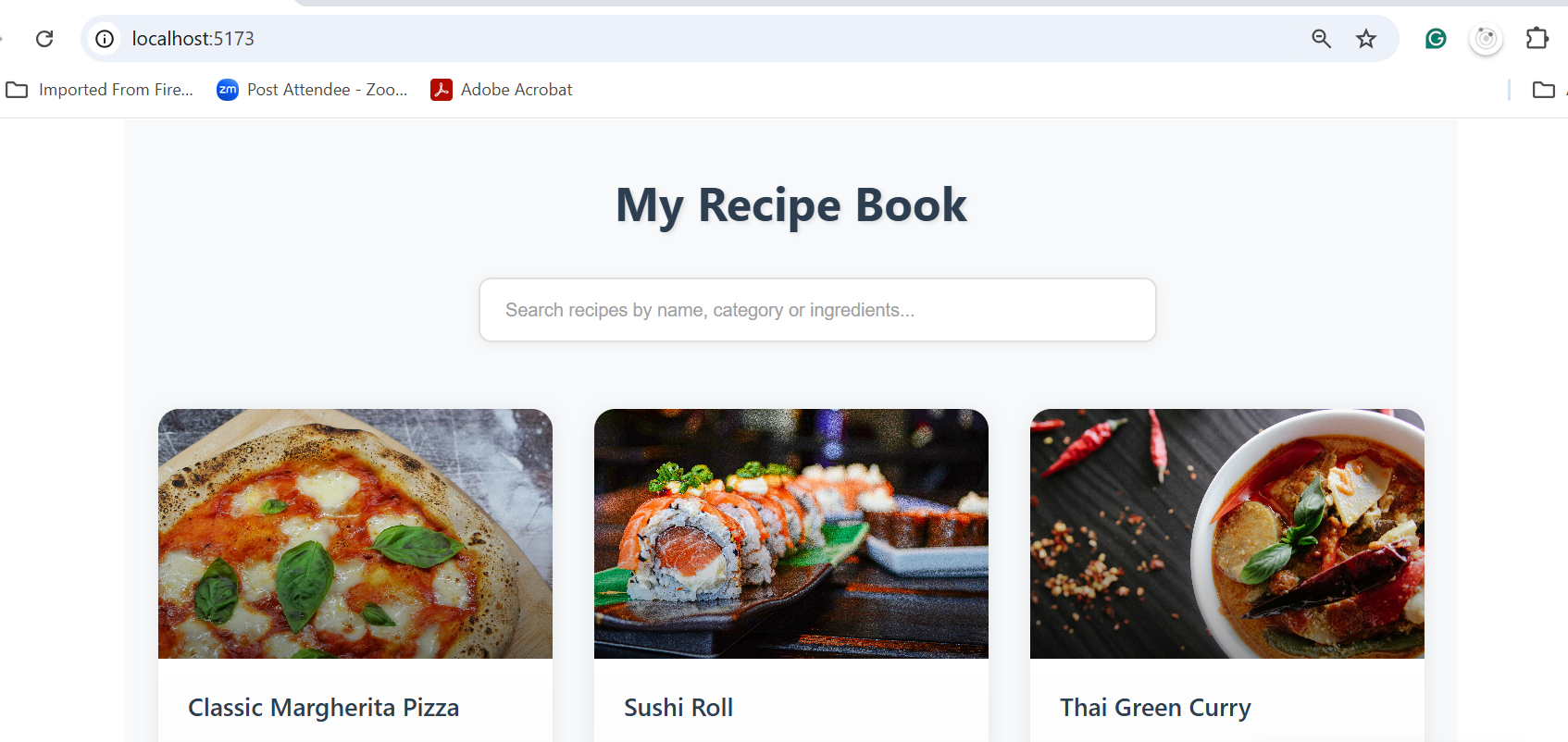
7.1 Design a search feature by prompting GitHub Copilot

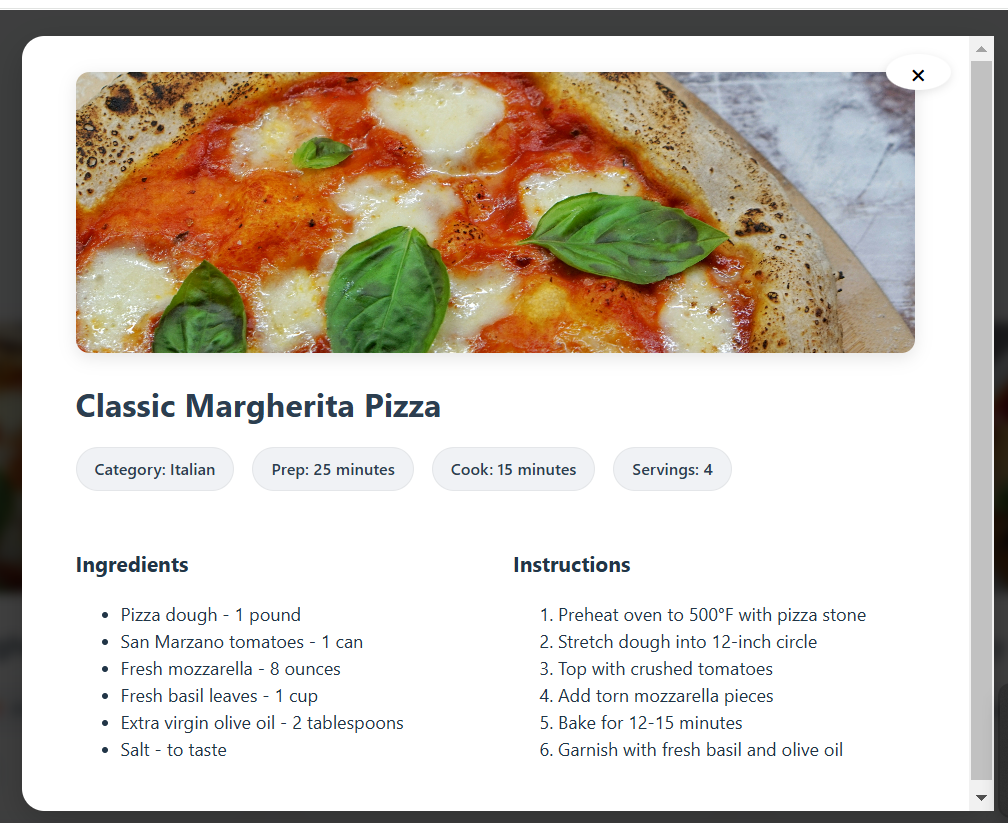
**@workspace implement search recipe functionality**



After get new code update the code in App.jsx and App.css file

The output will appear as follows:





The final code of all the file as below

**App.css**

/\* Root and Global Styles \*/

#root {

  max-width: 1280px;

  margin: 0 auto;

  padding: 2rem;

}

.app {

  background-color: #f8f9fa;

  min-height: 100vh;

  padding: 2rem 1rem;

}

/\* Header Styles \*/

h1 {

  text-align: center;

  color: #2c3e50;

  margin-bottom: 3rem;

  font-size: 2.8rem;

  font-weight: 700;

  text-shadow: 2px 2px 4px rgba(0,0,0,0.1);

}

/\* Search Styles \*/

.search-container {

  margin: 2rem auto 3rem;

  max-width: 600px;

  padding: 0 1rem;

  position: relative;

}

.search-input {

  width: 100%;

  padding: 1.2rem 1.5rem;

  font-size: 1.1rem;

  border: 2px solid #e0e0e0;

  border-radius: 12px;

  background-color: white;

  color: #2c3e50;

  transition: all 0.3s ease;

  box-shadow: 0 2px 10px rgba(0,0,0,0.05);

}

.search-input:focus {

  outline: none;

  border-color: #646cff;

  box-shadow: 0 4px 15px rgba(100,108,255,0.1);

}

.search-input::placeholder {

  color: #a0a0a0;

}

.logo {

  height: 6em;

  padding: 1.5em;

  will-change: filter;

  transition: filter 300ms;

}

.logo:hover {

  filter: drop-shadow(0 0 2em #646cffaa);

}

.logo.react:hover {

  filter: drop-shadow(0 0 2em #61dafbaa);

}

@keyframes logo-spin {

  from {

    transform: rotate(0deg);

  }

  to {

    transform: rotate(360deg);

  }

}

@media (prefers-reduced-motion: no-preference) {

  a:nth-of-type(2) .logo {

    animation: logo-spin infinite 20s linear;

  }

}

.card {

  padding: 2em;

}

.read-the-docs {

  color: #888;

}

/\* Recipe Grid Container \*/

.recipes-container {

  display: grid;

  grid-template-columns: repeat(auto-fill, minmax(320px, 1fr));

  gap: 2.5rem;

  padding: 1rem;

  max-width: 1400px;

  margin: 0 auto;

}

/\* Recipe Card Styles \*/

.recipe-card {

  background: white;

  border-radius: 20px;

  overflow: hidden;

  box-shadow: 0 4px 20px rgba(0,0,0,0.08);

  transition: all 0.4s ease;

  display: flex;

  flex-direction: column;

  height: 100%;

  cursor: pointer;

}

.recipe-card:hover {

  transform: translateY(-8px) scale(1.01);

  box-shadow: 0 12px 30px rgba(0,0,0,0.12);

}

/\* Recipe Image Styles \*/

.recipe-image {

  position: relative;

  height: 240px;

  overflow: hidden;

}

.recipe-image::after {

  content: '';

  position: absolute;

  bottom: 0;

  left: 0;

  right: 0;

  height: 60%;

  background: linear-gradient(to top, rgba(0,0,0,0.4), transparent);

  transition: opacity 0.3s ease;

}

.recipe-image img {

  width: 100%;

  height: 100%;

  object-fit: cover;

  transition: transform 0.6s ease;

}

.recipe-card:hover .recipe-image img {

  transform: scale(1.1);

}

/\* Recipe Content Styles \*/

.recipe-card-content {

  padding: 1.8rem;

  display: flex;

  flex-direction: column;

  gap: 1.2rem;

  flex: 1;

  background: linear-gradient(to bottom, #ffffff, #f8f9fa);

}

.recipe-card h2 {

  color: #2c3e50;

  font-size: 1.5rem;

  font-weight: 600;

  margin: 0;

  line-height: 1.4;

}

/\* Meta Information Styles \*/

.recipe-meta {

  display: flex;

  flex-wrap: wrap;

  gap: 0.8rem;

  margin-top: auto;

}

.recipe-meta span {

  background: #f0f2f5;

  color: #2c3e50;

  padding: 0.6rem 1.2rem;

  border-radius: 25px;

  font-size: 0.9rem;

  font-weight: 500;

  border: 1px solid rgba(0,0,0,0.05);

  transition: all 0.3s ease;

}

.recipe-meta span:hover {

  background: #e8eaf0;

  transform: translateY(-2px);

}

/\* Modal Overlay and Container \*/

.recipe-modal-overlay {

  position: fixed;

  top: 0;

  left: 0;

  right: 0;

  bottom: 0;

  background: rgba(0, 0, 0, 0.75);

  display: flex;

  justify-content: center;

  align-items: center;

  padding: 1rem;

  z-index: 1000;

  backdrop-filter: blur(5px);

}

.recipe-modal {

  background: white;

  border-radius: 20px;

  padding: 2rem;

  width: 90%;

  max-width: 800px;

  max-height: 85vh;

  overflow-y: auto;

  position: relative;

  box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2);

}

/\* Recipe Detail Image \*/

.recipe-detail-image {

  width: 100%;

  height: 250px;

  border-radius: 12px;

  overflow: hidden;

  margin-bottom: 1.5rem;

  box-shadow: 0 4px 15px rgba(0, 0, 0, 0.1);

}

.recipe-detail-image img {

  width: 100%;

  height: 100%;

  object-fit: cover;

}

/\* Recipe Detail Content \*/

.recipe-detail {

  padding: 0 1rem;

}

.recipe-detail h2 {

  color: #2c3e50;

  font-size: 1.8rem;

  margin-bottom: 1rem;

}

.recipe-content {

  display: grid;

  grid-template-columns: 1fr 1fr;

  gap: 2rem;

  margin-top: 2rem;

}

/\* Meta Information in Modal \*/

.recipe-detail .recipe-meta {

  display: flex;

  flex-wrap: wrap;

  gap: 1rem;

  margin-bottom: 1.5rem;

}

.recipe-detail .recipe-meta span {

  background: #f0f2f5;

  padding: 0.5rem 1rem;

  border-radius: 20px;

  font-size: 0.9rem;

  color: #2c3e50;

}

/\* Close Button \*/

.close-button {

  position: absolute;

  top: 1rem;

  right: 1rem;

  background: white;

  border: none;

  width: 32px;

  height: 32px;

  border-radius: 50%;

  font-size: 1.5rem;

  cursor: pointer;

  display: flex;

  align-items: center;

  justify-content: center;

  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);

  transition: all 0.3s ease;

}

.close-button:hover {

  background: #f0f2f5;

  transform: scale(1.1);

}

/\* Responsive adjustments \*/

@media (max-width: 768px) {

  .recipes-container {

    grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));

    gap: 1.5rem;

    padding: 1rem;

  }

  .recipe-image {

  align-items: center;

  padding: 2rem;

  z-index: 1000;

}

.recipe-modal {

  background: white;

  border-radius: 12px;

  padding: 2rem;

  max-width: 800px;

  width: 90%;

  max-height: 90vh;

  overflow-y: auto;

  position: relative;

}

.close-button {

  position: absolute;

  top: 1rem;

  right: 1rem;

  background: none;

  border: none;

  font-size: 2rem;

  cursor: pointer;

  color: #666;

  transition: color 0.3s ease;

}

.close-button:hover {

  color: #333;

}

.recipe-detail-image {

  width: 100%;

  height: 300px;

  border-radius: 8px;

  overflow: hidden;

  margin-bottom: 2rem;

}

.recipe-detail-image img {

  width: 100%;

  height: 100%;

  object-fit: cover;

}

.recipe-detail h2 {

  color: #2c3e50;

  margin-bottom: 1rem;

  font-size: 1.75rem;

}

.recipe-detail .recipe-meta {

  margin-bottom: 2rem;

}

@media (max-width: 768px) {

  .recipe-content {

    grid-template-columns: 1fr;

  }

}

.ingredients ul {

  list-style-type: disc;

  padding-left: 1.5rem;

  margin: 0;

}

.instructions ol {

  padding-left: 1.5rem;

  margin: 0;

}

.ingredients li,

.instructions li {

  margin-bottom: 0.5rem;

  line-height: 1.6;

  color: #4a4a4a;

}

.search-container {

  margin: 2rem auto 3rem;

  max-width: 600px;

  padding: 0 1rem;

  position: relative;

}

.search-input {

  width: 100%;

  padding: 1.2rem 1.5rem;

  font-size: 1.1rem;

  border: 2px solid #e0e0e0;

  border-radius: 12px;

  background-color: white;

  color: #2c3e50;

  transition: all 0.3s ease;

  box-shadow: 0 2px 10px rgba(0,0,0,0.05);

}

.search-input:focus {

  outline: none;

  border-color: #646cff;

  box-shadow: 0 4px 15px rgba(100,108,255,0.1);

}

.search-input::placeholder {

  color: #a0a0a0;

}

  .recipe-modal {

    padding: 1.5rem;

    width: 95%;

  }

  .recipe-detail-image {

    height: 200px;

  }

  .recipe-content {

    grid-template-columns: 1fr;

    gap: 1.5rem;

  }

  .recipe-detail h2 {

    font-size: 1.5rem;

  }

  .recipe-detail {

    padding: 0;

  }

}

**recipes.js**

export const recipesData = [

  {

    id: 4,

    title: 'Classic Margherita Pizza',

    image: 'https://images.unsplash.com/photo-1604068549290-dea0e4a305ca',

    ingredients: [

      'Pizza dough - 1 pound',

      'San Marzano tomatoes - 1 can',

      'Fresh mozzarella - 8 ounces',

      'Fresh basil leaves - 1 cup',

      'Extra virgin olive oil - 2 tablespoons',

      'Salt - to taste'

    ],

    instructions: [

      'Preheat oven to 500°F with pizza stone',

      'Stretch dough into 12-inch circle',

      'Top with crushed tomatoes',

      'Add torn mozzarella pieces',

      'Bake for 12-15 minutes',

      'Garnish with fresh basil and olive oil'

    ],

    category: 'Italian',

    prepTime: '25 minutes',

    cookTime: '15 minutes',

    servings: 4

  },

  {

    id: 5,

    title: 'Sushi Roll',

    image: 'https://images.unsplash.com/photo-1579871494447-9811cf80d66c',

    ingredients: [

      'Sushi rice - 2 cups',

      'Nori sheets - 4',

      'Fresh salmon - 8 ounces',

      'Cucumber - 1, julienned',

      'Avocado - 1, sliced',

      'Rice vinegar - 1/4 cup',

      'Wasabi and soy sauce - to serve'

    ],

    instructions: [

      'Cook and season rice with vinegar',

      'Place nori on bamboo mat',

      'Spread rice on nori',

      'Add fish and vegetables',

      'Roll tightly using mat',

      'Cut into 8 pieces'

    ],

    category: 'Japanese',

    prepTime: '30 minutes',

    cookTime: '20 minutes',

    servings: 4

  },

  {

    id: 6,

    title: 'Thai Green Curry',

    image: 'https://images.unsplash.com/photo-1455619452474-d2be8b1e70cd',

    ingredients: [

      'Coconut milk - 2 cans',

      'Green curry paste - 4 tablespoons',

      'Chicken breast - 1 pound, sliced',

      'Thai eggplants - 4, quartered',

      'Bamboo shoots - 1 cup',

      'Thai basil - 1 cup',

      'Fish sauce - 2 tablespoons',

      'Palm sugar - 1 tablespoon'

    ],

    instructions: [

      'Simmer coconut milk until oil separates',

      'Stir in curry paste and cook until fragrant',

      'Add chicken and cook until nearly done',

      'Add vegetables and remaining coconut milk',

      'Season with fish sauce and palm sugar',

      'Garnish with Thai basil'

    ],

    category: 'Thai',

    prepTime: '20 minutes',

    cookTime: '25 minutes',

    servings: 4

  },

  {

    id: 7,

    title: 'Beef Burger',

    image: 'https://images.unsplash.com/photo-1568901346375-23c9450c58cd',

    ingredients: [

      'Ground beef (80/20) - 2 pounds',

      'Burger buns - 6',

      'Onion - 1 medium, diced',

      'Garlic powder - 1 teaspoon',

      'Worcestershire sauce - 1 tablespoon',

      'Salt and pepper - to taste',

      'Cheddar cheese - 6 slices',

      'Lettuce leaves - 6',

      'Tomato slices - 12',

      'Red onion - 1, sliced'

    ],

    instructions: [

      'Mix ground beef with diced onion, garlic powder, and Worcestershire sauce',

      'Form into 6 equal-sized patties',

      'Season both sides with salt and pepper',

      'Grill for 4-5 minutes per side for medium',

      'Add cheese in the last minute',

      'Toast the buns lightly',

      'Assemble with lettuce, tomato, and onion'

    ],

    category: 'American',

    prepTime: '15 minutes',

    cookTime: '10 minutes',

    servings: 6

  },

  {

    id: 8,

    title: 'Chocolate Chip Cookies',

    image: 'https://images.unsplash.com/photo-1499636136210-6f4ee915583e',

    ingredients: [

      'All-purpose flour - 2¼ cups',

      'Butter - 1 cup, softened',

      'Brown sugar - 3/4 cup',

      'White sugar - 3/4 cup',

      'Large eggs - 2',

      'Vanilla extract - 1 teaspoon',

      'Baking soda - 1 teaspoon',

      'Salt - 1/2 teaspoon',

      'Chocolate chips - 2 cups'

    ],

    instructions: [

      'Preheat oven to 375°F (190°C)',

      'Cream together butter and sugars until smooth',

      'Beat in eggs and vanilla',

      'Mix in flour, baking soda, and salt',

      'Stir in chocolate chips',

      'Drop rounded tablespoons onto ungreased baking sheets',

      'Bake for 10-12 minutes until edges are lightly browned'

    ],

    category: 'Dessert',

    prepTime: '20 minutes',

    cookTime: '12 minutes',

    servings: 24

  },

  {

    id: 9,

    title: 'Greek Salad',

    image: 'https://images.unsplash.com/photo-1540189549336-e6e99c3679fe',

    ingredients: [

      'Cucumber - 1 large, chunked',

      'Cherry tomatoes - 2 cups, halved',

      'Red onion - 1 medium, sliced',

      'Green bell pepper - 1, chunked',

      'Kalamata olives - 1 cup',

      'Feta cheese - 200g, cubed',

      'Extra virgin olive oil - 1/4 cup',

      'Red wine vinegar - 2 tablespoons',

      'Dried oregano - 1 teaspoon',

      'Salt and pepper - to taste'

    ],

    instructions: [

      'Combine cucumber, tomatoes, onion, and pepper in a large bowl',

      'Add olives and feta cheese',

      'Whisk together olive oil, vinegar, and oregano',

      'Pour dressing over salad',

      'Season with salt and pepper',

      'Toss gently and serve immediately'

    ],

    category: 'Mediterranean',

    prepTime: '15 minutes',

    cookTime: '0 minutes',

    servings: 4

  }

]

**App.jsx**

import { useState } from 'react'

import { recipesData } from './data/recipes'

import './App.css'

function App() {

  const [recipes, setRecipes] = useState(recipesData)

  const [selectedRecipe, setSelectedRecipe] = useState(null)

  const [searchTerm, setSearchTerm] = useState('')

  const handleRecipeClick = (recipe) => {

    setSelectedRecipe(recipe)

  }

  const handleClose = () => {

    setSelectedRecipe(null)

  }

  const handleSearch = (event) => {

    const term = event.target.value.toLowerCase()

    setSearchTerm(term)

    if (term === '') {

      setRecipes(recipesData)

    } else {

      const filteredRecipes = recipesData.filter(recipe =>

        recipe.title.toLowerCase().includes(term) ||

        recipe.category.toLowerCase().includes(term) ||

        recipe.ingredients.some(ingredient =>

          ingredient.toLowerCase().includes(term)

        )

      )

      setRecipes(filteredRecipes)

    }

  }

  return (

    <div className="app">

      <h1>My Recipe Book</h1>

      <div className="search-container">

        <input

          type="text"

          placeholder="Search recipes by name, category or ingredients..."

          value={searchTerm}

          onChange={handleSearch}

          className="search-input"

        />

      </div>

      <div className="recipes-container">

        {recipes.map(recipe => (

          <div

            key={recipe.id}

            className="recipe-card"

            onClick={() => handleRecipeClick(recipe)}

          >

            <div className="recipe-image">

              <img src={recipe.image} alt={recipe.title} />

            </div>

            <div className="recipe-card-content">

              <h2>{recipe.title}</h2>

              <div className="recipe-meta">

                <span>{recipe.category}</span>

                <span>⏰ {recipe.prepTime}</span>

                <span>👥 {recipe.servings} servings</span>

              </div>

            </div>

          </div>

        ))}

      </div>

      {selectedRecipe && (

        <div className="recipe-modal-overlay" onClick={handleClose}>

          <div className="recipe-modal" onClick={e => e.stopPropagation()}>

            <button className="close-button" onClick={handleClose}>&times;</button>

            <div className="recipe-detail">

              <div className="recipe-detail-image">

                <img src={selectedRecipe.image} alt={selectedRecipe.title} />

              </div>

              <h2>{selectedRecipe.title}</h2>

              <div className="recipe-meta">

                <span>Category: {selectedRecipe.category}</span>

                <span>Prep: {selectedRecipe.prepTime}</span>

                <span>Cook: {selectedRecipe.cookTime}</span>

                <span>Servings: {selectedRecipe.servings}</span>

              </div>

              <div className="recipe-content">

                <div className="ingredients">

                  <h3>Ingredients</h3>

                  <ul>

                    {selectedRecipe.ingredients.map((ingredient, index) => (

                      <li key={index}>{ingredient}</li>

                    ))}

                  </ul>

                </div>

                <div className="instructions">

                  <h3>Instructions</h3>

                  <ol>

                    {selectedRecipe.instructions.map((step, index) => (

                      <li key={index}>{step}</li>

                    ))}

                  </ol>

                </div>

              </div>

            </div>

          </div>

        </div>

      )}

    </div>

  )

}

export default App