

Answer Key

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

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

npx create-react-app@latest rec-app

```
C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_05 - with MERN Using Generative AI\Practise project
\Lesson_05\practise-project>npx create-react-app@latest rec-app

Creating a new React app in C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_05 - with MERN Using Ge
nerative AI\Practise project\Lesson_05\practise-project\rec-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

[Progress Bar] / idealTree:@babel/core: sill fetch manifest klona@^2.0.4
```

Npx : node package extension which help to create the take the help of external model js module without downloading explicitly and do the task for that module.

The React.js project is successfully created, showing the following after creation:

```

Run 'npm audit' for details.

Created git commit.

Success! Created rec-app at C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demo
nerative AI\Practise project\Lesson_05\practise-project\rec-app
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd rec-app
  npm start

Happy hacking!

C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_05 - with MERN Usi
\Lesson_05\practise-project>

```

1.2 Execute the following command to navigate to the folder where it is created and to open that project in another instance of Visual Studio:

cd .\rec-app it is use to move inside a project directory
code . it is one of the way to open the current directory in VSCode

```

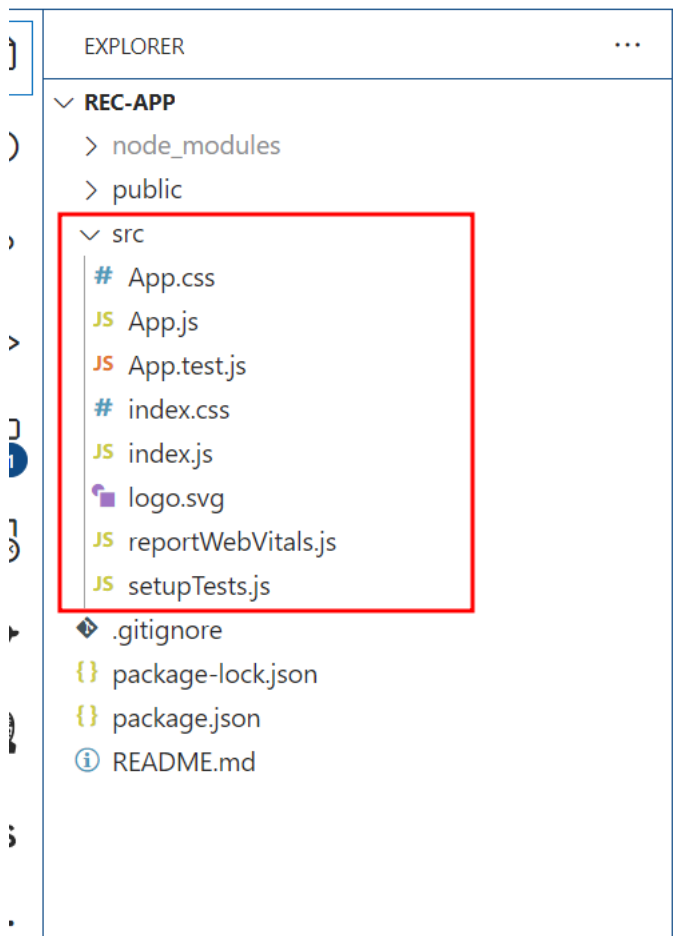
C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_05 - with MERN Using Generati
\Lesson_05\practise-project>cd rec-app

C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_05 - with MERN Using Generati
\Lesson_05\practise-project\rec-app>code .

C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_05 - with MERN Using Generati
\Lesson_05\practise-project\rec-app>

```

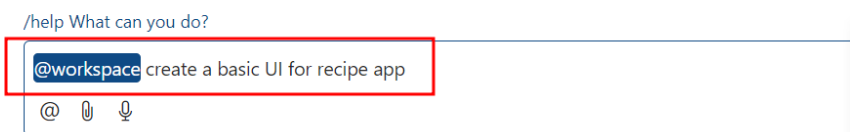
The basic React.js structure appears as follows:



2. Generate a basic user interface for recipe data using GitHub Copilot

2.1 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.js` file

 **Workspace**

> Used 13 references

✓ Collecting workspace information

To create a basic UI for a recipe app, you can start by modifying the `JS App.js` file to include a simple layout with a header, a list of recipes, and a form to add new recipes. Here's an example:

Update `JS App.js`

```
import React, { useState } from 'react';
import './App.css';

function App() {
  const [recipes, setRecipes] = useState([
    { name: 'Spaghetti Bolognese', ingredients: 'Spaghetti, Meat, Tomato Sauce' },
    { name: 'Chicken Curry', ingredients: 'Chicken, Curry Powder, Coconut Milk' },
  ]);
  const [newRecipe, setNewRecipe] = useState({ name: '', ingredients: '' });

  const handleInputChange = (e) => {
    const { name, value } = e.target;
```

What features should we add next?

3. Generate sample data using GitHub Copilot

3.1 Prompt GitHub Copilot to create sample data for the existing recipe app **@workspace create sample data for it**

```
|      const { name, value } = e.target;
What features should we add next?
```

@workspace create sample data for it

`JS App.js` Current file 

@  

This generates the sample data for the recipe application. Create a file named **sampleData.js**, copy the sample data generated into this file, and save it in the **rec-app\src** folder

 **Workspace** (rerun without)

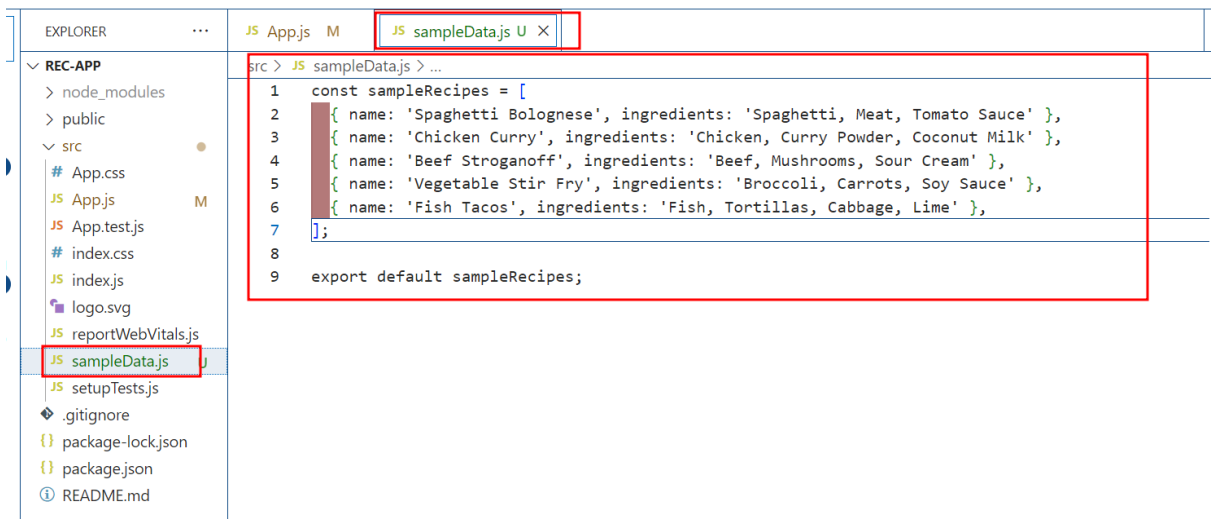
> Used 13 references

✓ Collecting workspace information

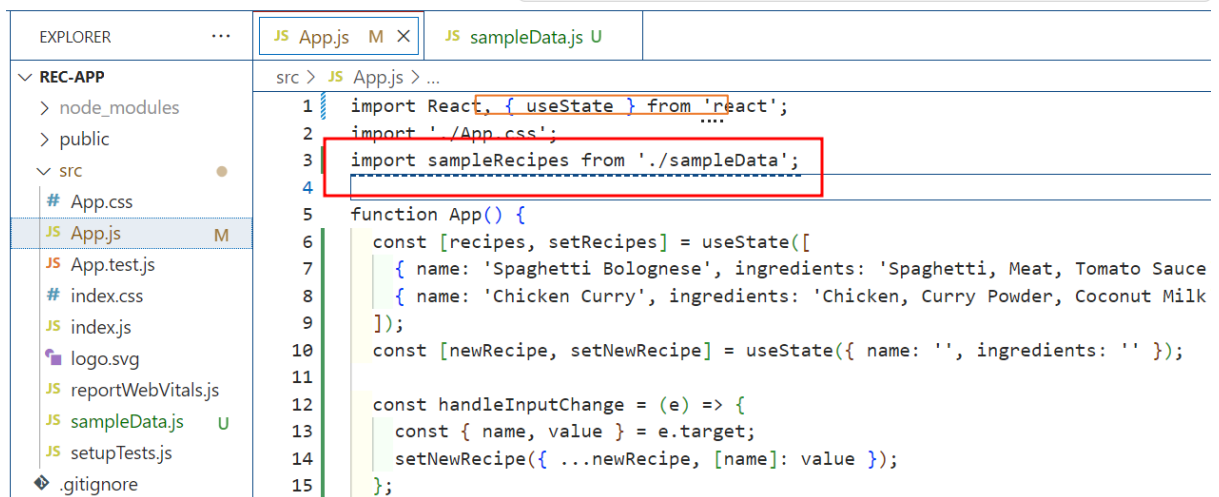
You can create a separate file to store sample data for your recipes. Let's create a new file called `sampleData.js` in the `src` directory and export the sample data from there.

Create `src/sampleData.js`

```
const sampleRecipes = [  
  { name: 'Spaghetti Bolognese', ingredients: 'Spaghetti, Meat, Tomato Sauce' },  
  { name: 'Chicken Curry', ingredients: 'Chicken, Curry Powder, Coconut Milk' },  
  { name: 'Beef Stroganoff', ingredients: 'Beef, Mushrooms, Sour Cream' },  
  { name: 'Vegetable Stir Fry', ingredients: 'Broccoli, Carrots, Soy Sauce' },  
  { name: 'Fish Tacos', ingredients: 'Fish, Tortillas, Cabbage, Lime' },  
];  
  
export default sampleRecipes;
```



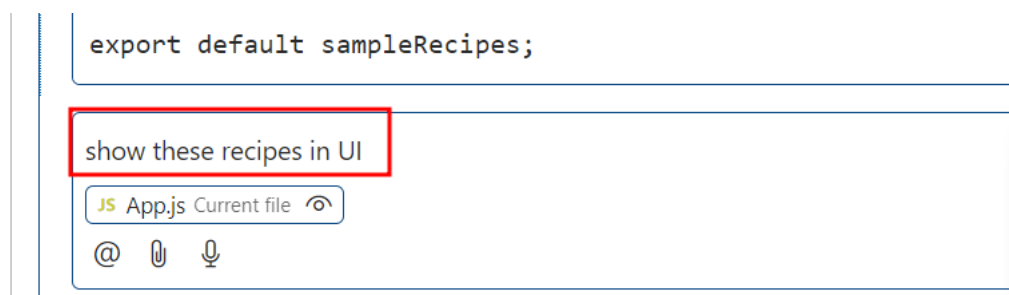
3.2 Open **App.js** and import the newly created **recipe.js** by adding the following script:
import sampleRecipes from './sampleData';

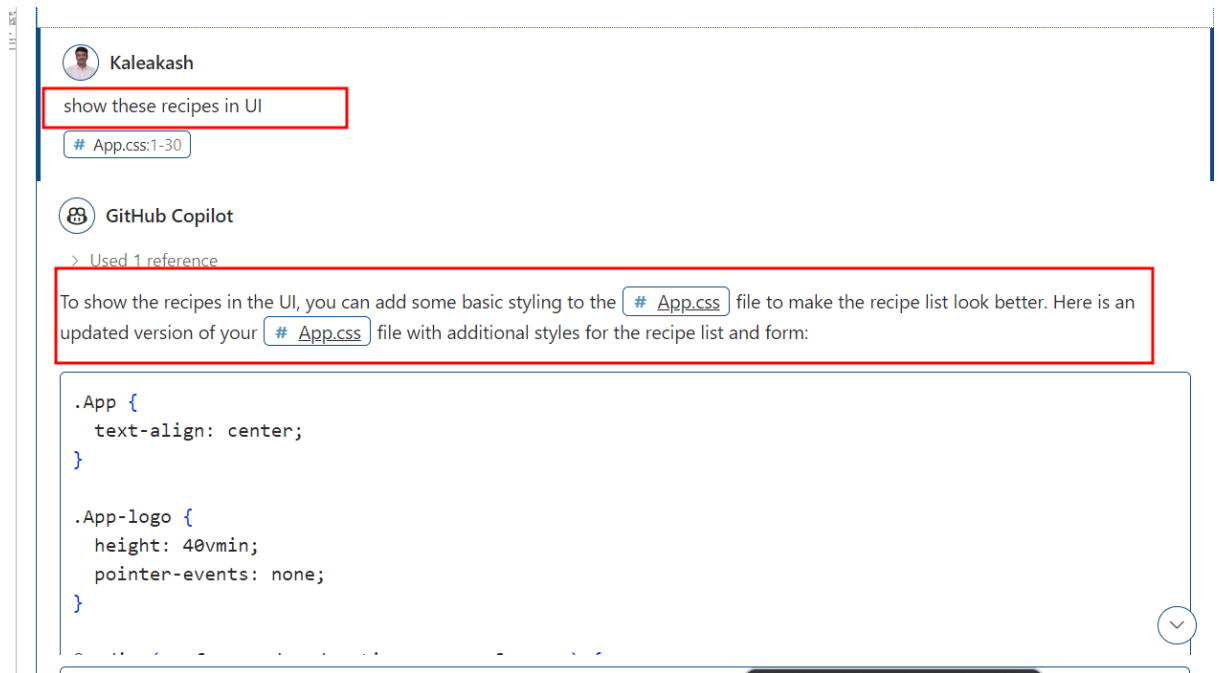


4. Generate React code to display the recipes using the user interface

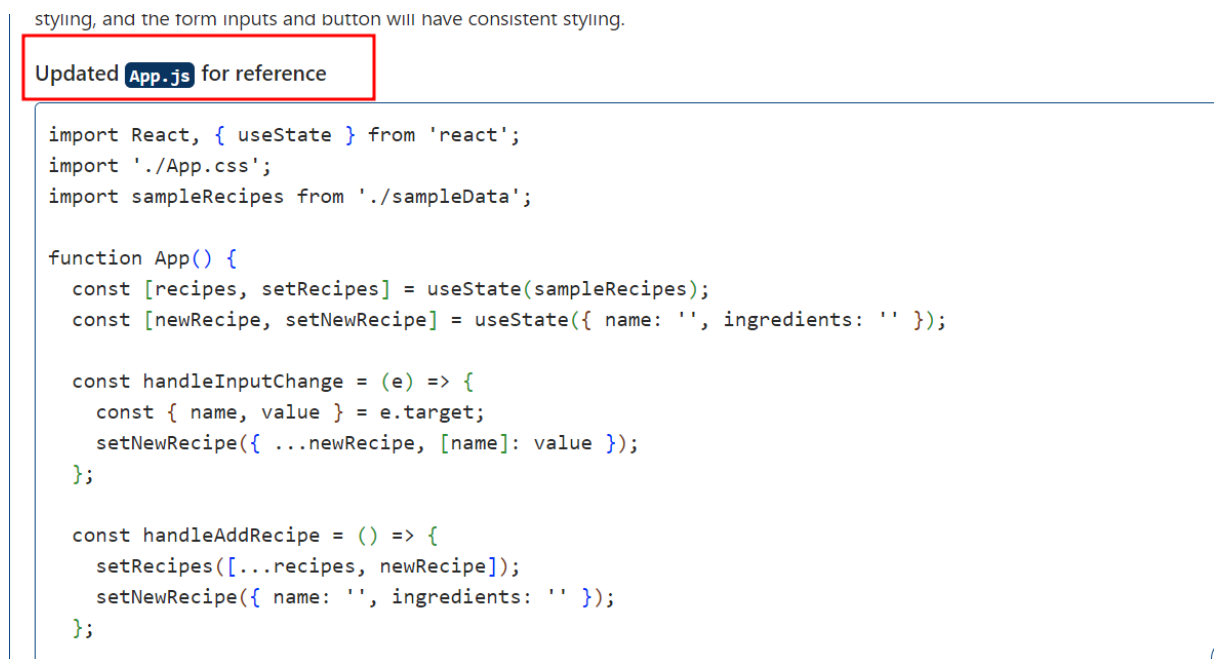
4.1 Prompt GitHub Copilot to generate code to display the recipe data using the user interface

show these recipes in UI





Copy basic CSS rules in App.css file



Update new generated content in App.js file

4.2 Run the react project after updating App.js

npm start

```
C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_05 - with MERN Using Generative AI\Practise project
\Lesson_05\practise-project\rec-app>npm start
```

The following message will be displayed after successful execution:

```
Compiled successfully!

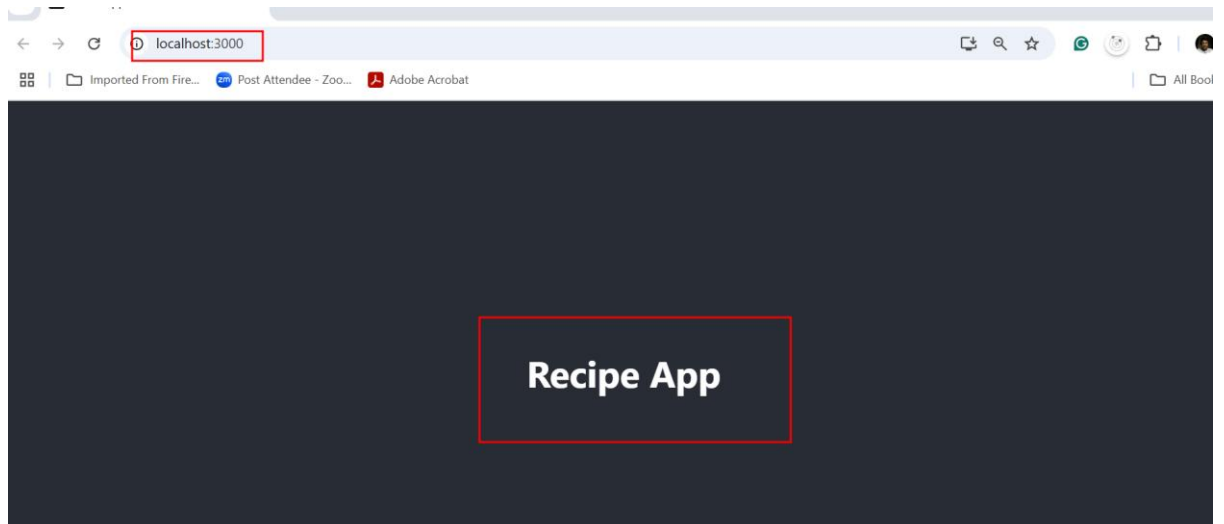
You can now view rec-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.100.103:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

The output will appear as follows:

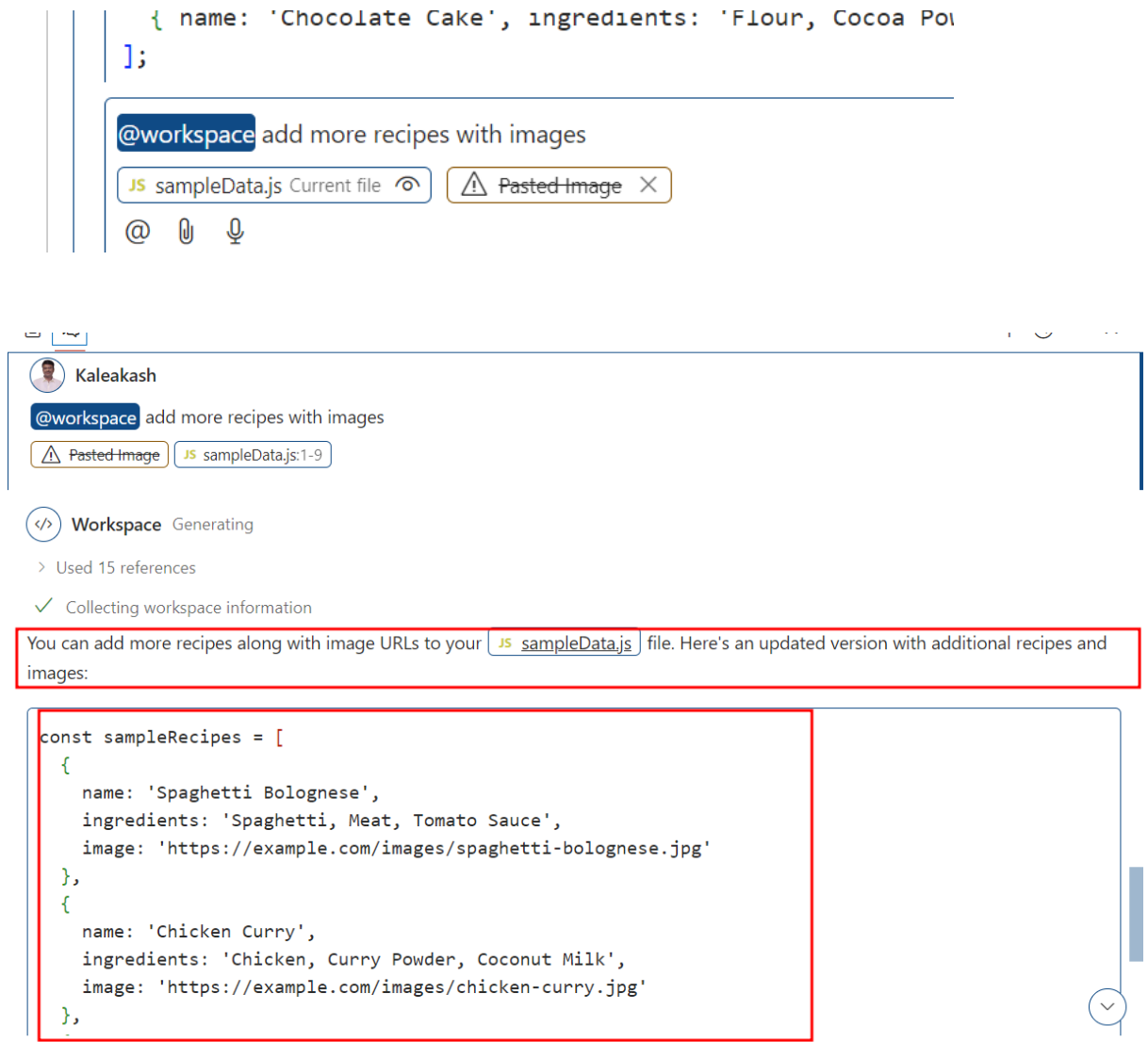


Spaghetti Bolognese Spaghetti, Meat, Tomato Sauce
Chicken Curry Chicken, Curry Powder, Coconut Milk
Beef Stroganoff Beef, Mushrooms, Sour Cream
Vegetable Stir Fry Broccoli, Carrots, Soy Sauce
Fish Tacos Fish, Tortillas, Cabbage, Lime
Add a New Recipe <div><input type="text" value="Recipe Name"/><input type="text" value="Ingredients"/><input type="button" value="Add Recipe"/></div>

5. Generate a React code to add more recipes to the recipe app

5.1 Improve the recipe app by adding more recipes using GitHub Copilot prompts

@workspace add more recipes with images



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

Update the sampleData.js file code

Show new update these recipes in UI with images

image: 'https://example.com/images/chicken-curry.jpg'

},


show new update these recipes in UI with images

JS sampleData.js Current file

@ 📎 🗣️


... 📁 🗨️

ipe

 Kaleakash

show new update these recipes in UI with images

JS sampleData.js:1-25

 GitHub Copilot Generating..

> Used 1 reference

To display the recipes with images in the UI, you need to update the `App.js` file to include the image URLs and render the images. Here's how you can do it:

Update JS App.js

```
import React, { useState } from 'react';
import './App.css';
import sampleRecipes from './sampleData';

function App() {
  const [recipes, setRecipes] = useState(sampleRecipes);
  const [newRecipe, setNewRecipe] = useState({ name: '', ingredients: '', image: '' });

  const handleInputChange = (e) => {
    const { name, value } = e.target;
    setNewRecipe({ ...newRecipe, [name]: value });
  };
}
```

Update the new changes in App.js file as well as App.css file

The final code of all files

Execute the React project using **npm start** after updating the recipe.js file. The output will appear as follows:

6. Create the Recipe Detail Component to display detailed information about each recipe

6.1 Prompt GitHub Copilot to generate a React code for displaying details of each recipe
show it in card format in UI, on click of card it should load the details

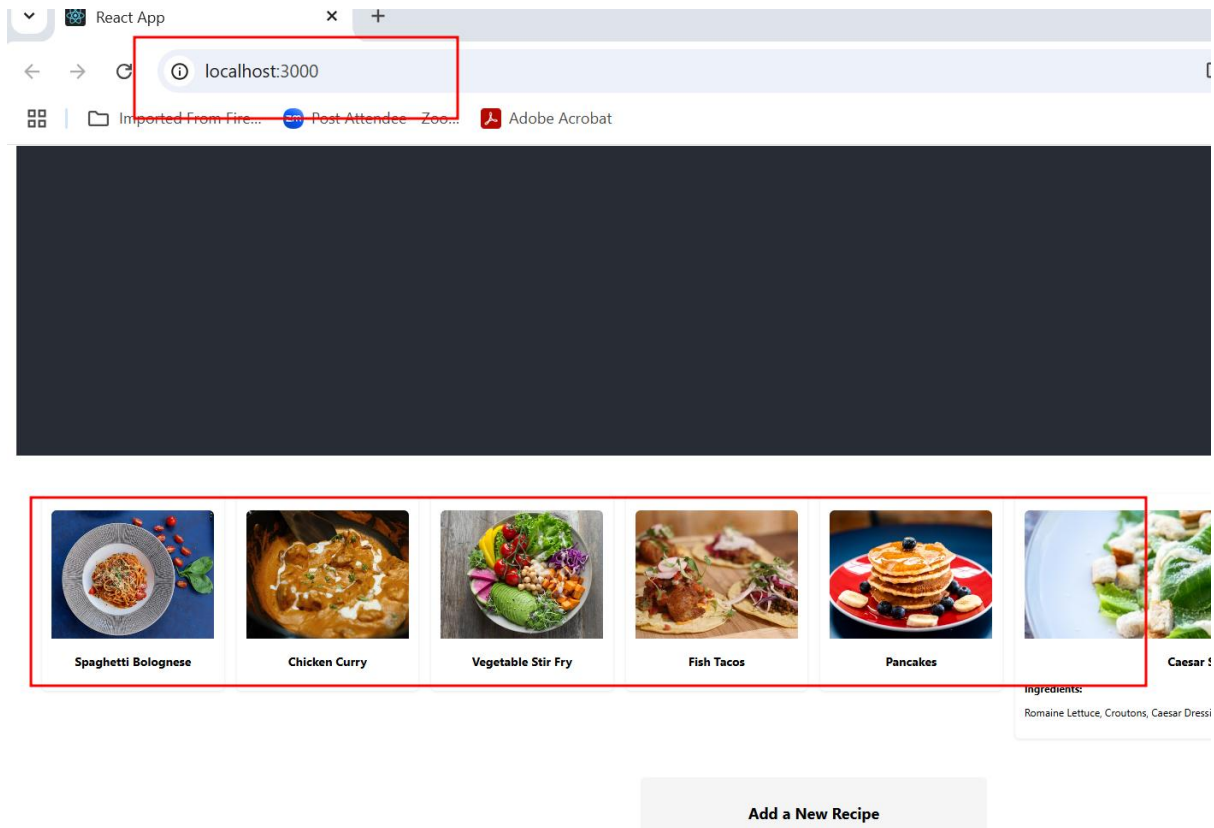
```
|   const handleAddRecipe = () => {  
|  
|   show it in card format in UI, on click of card it should load the details  
|  
|   JS App.js:1-69 Current file
```

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

After updating App.js file, execute the React project using **npm start**. The output will appear as follows:

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

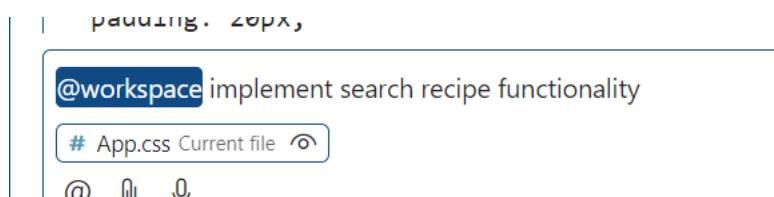
Now check updated output



When you click on particular image you get that image details

7. 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.js and App.css file

The output will appear as follows:

