Lesson 05 Demo 01

Building a To-Do List Application Using GitHub Copilot

Objective: To demonstrate the code generation and enhancement of a To-Do List application using GitHub Copilot

Tools required: Visual Studio Code, Chat GPT and GitHub Copilot

Prerequisites: NA

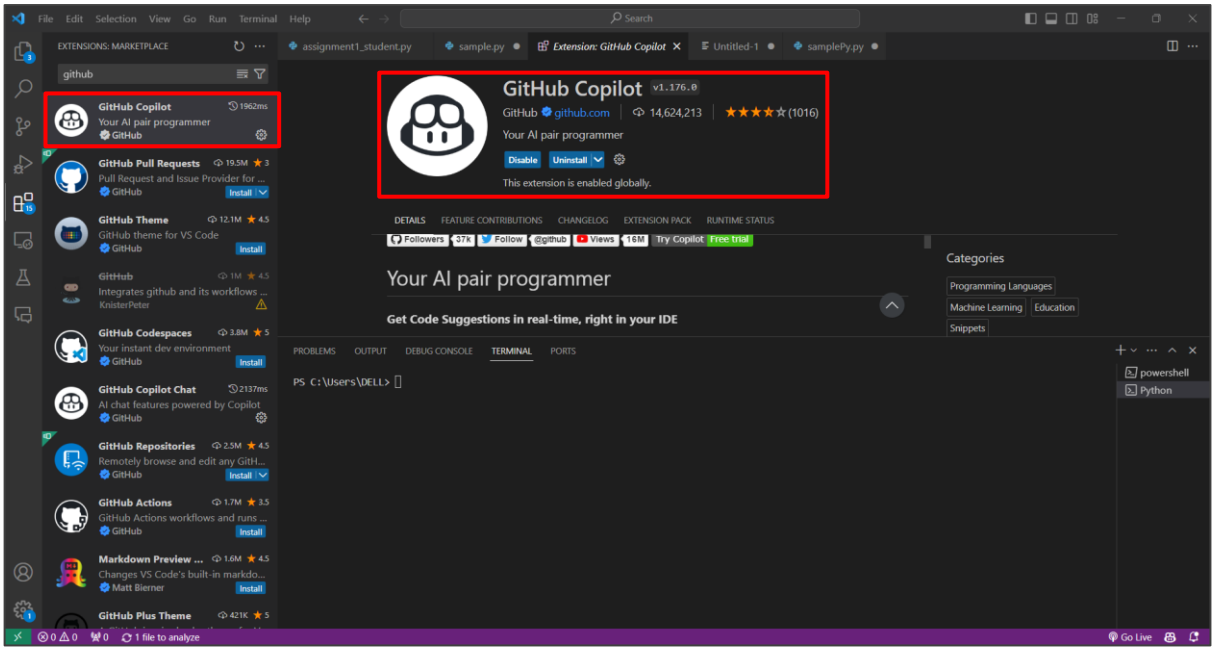
Steps to be followed:

1. Integrate GitHub Copilot with Visual Studio
2. Create React JS Project with help of Chat GPT
3. Create User Defined component with help of GitHub Copilot which help To-Do List application.

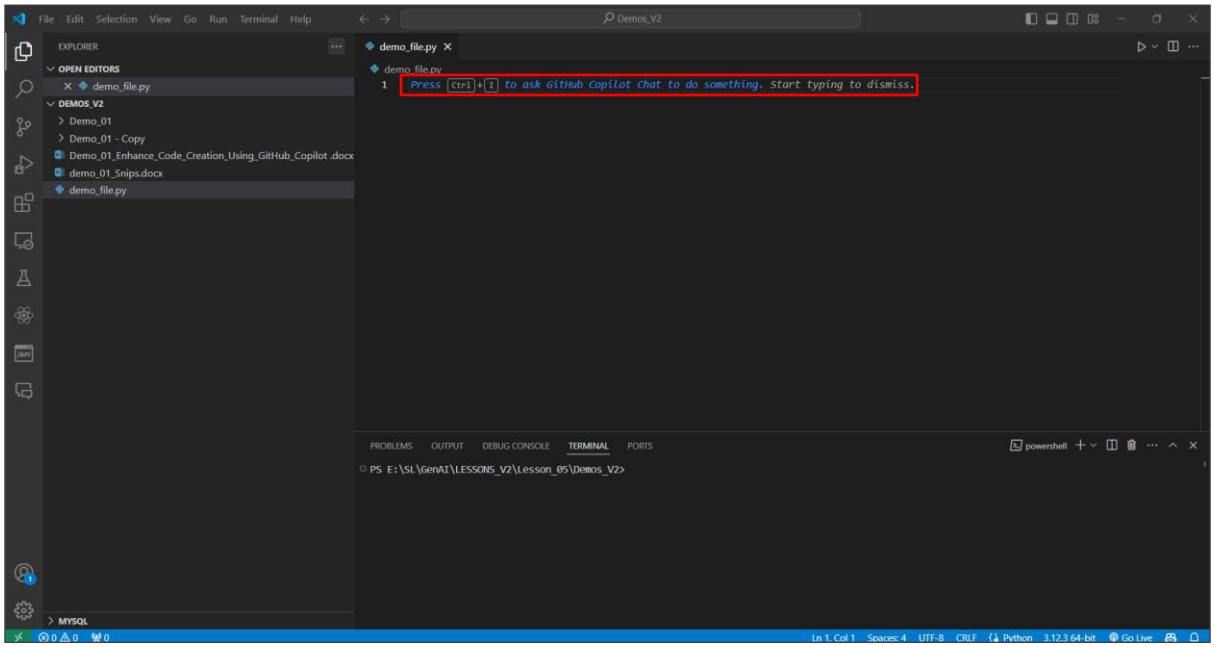
Note: Create an account on https://github.com/features/copilot to access it in Visual Studio. Generative AI tool used in this exercise can produce varied outputs even when presented with similar prompts. Thus, you may get different outputs for the same prompt.

Step 1: Integrate GitHub Copilot with Visual Studio

1.1 Install the **GitHub Copilot** extension in Visual Studio to use generative AI for code generation



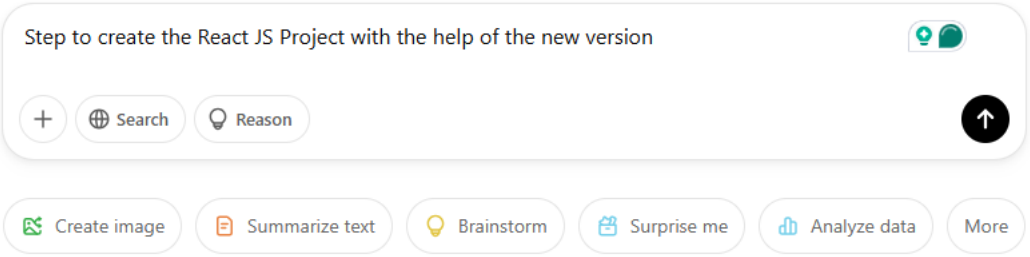
Once the installation and sign-in are completed, a message will appear as shown below:



Step 2: Create React JS Project with help of Chat GPT

2.1 Open the Chat GPT and write the prompt message as

“Step to create the React JS Project with help of new version”

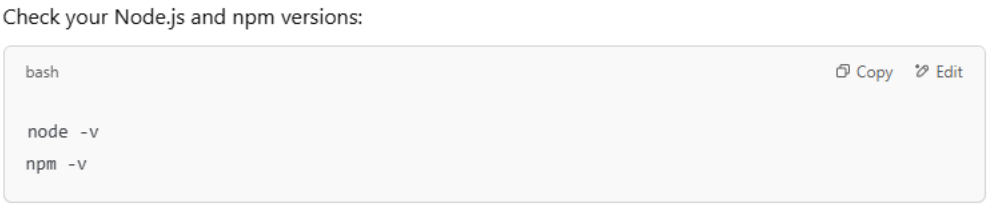


The prompt message generated as

Prerequisites
Ensure the following is installed on your system:

1. **Node.js** (Latest LTS version recommended) – Download Node.js
   If node JS is not installed please installed with below URL as https://nodejs.org/en
2. **npm** (Comes with Node.js) or **pnpm/yarn** (optional package managers)

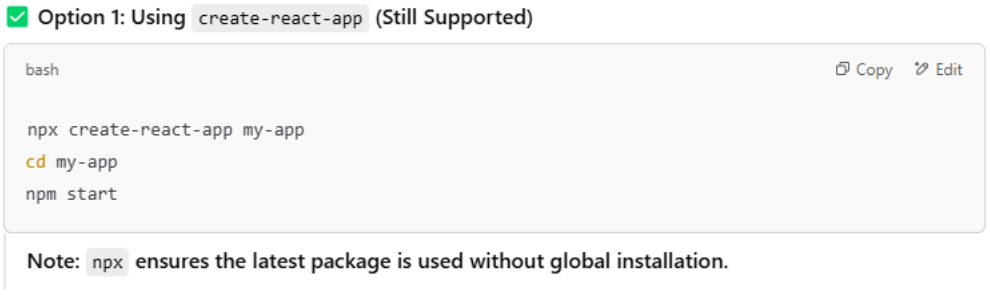
Check your Node.js and npm versions:



🚀 Step 1: Set Up a New React Project

Use the latest **create-react-app alternative**: **Vite** or **Create React with Next.js**.

☑ Option 1: Using create-react-app (Still Supported)



✅ Option 2: Using **Vite** (Faster and More Modern)

bash Copy Edit

```
npm create vite@latest my-app --template react
cd my-app
npm install
npm run dev
```

✅ Option 3: Using **Next.js** (React Framework – Recommended for Fullstack)

bash Copy Edit

```
npx create-next-app@latest my-app
cd my-app
npm run dev
```

🛠️ Step 2: Project Structure

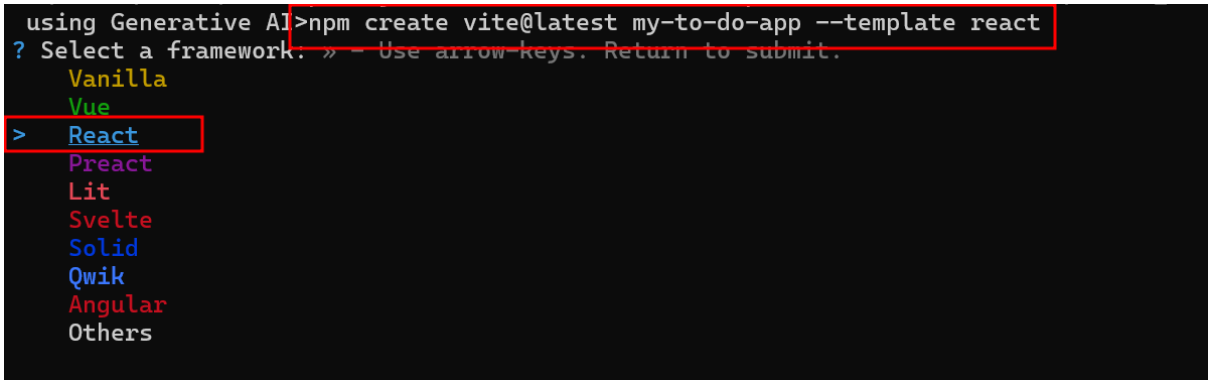
Your project will be structured like this:

pgsql Copy Edit

```
my-app/
├── node_modules/
├── public/
├── src/
│   ├── App.jsx (Main Component)
│   ├── index.jsx (Entry Point)
│   └── styles.css (Global Styles)
├── .gitignore
├── package.json
└── vite.config.js (for Vite projects)
```

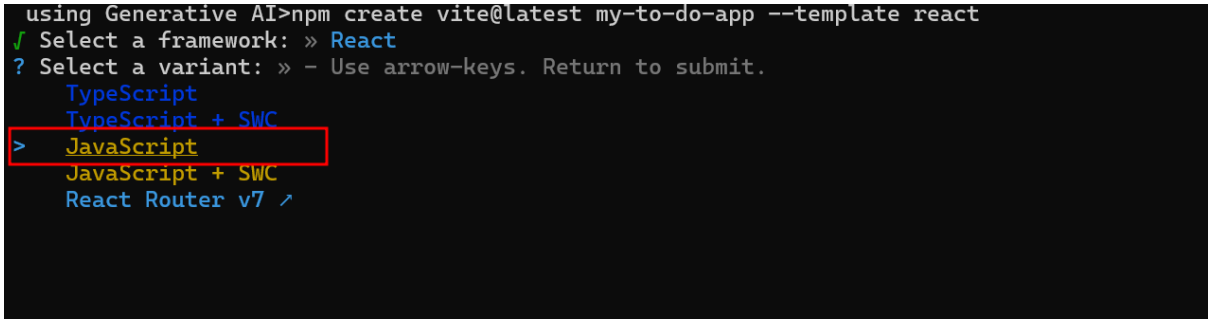
Here in this demo, we use 2nd option.

npm create vite@latest my-to-do-app --template react

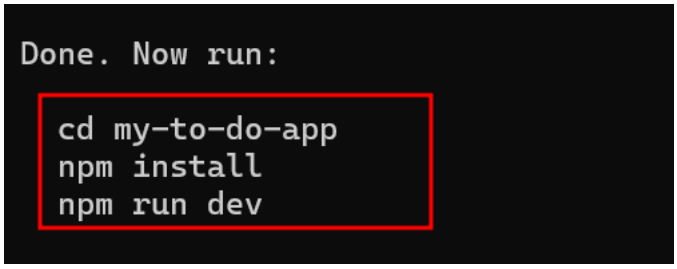


Here select, the option as React

Here select as JavaScript



Now move inside a project folder using cd project-name



Then installed required dependencies. Run the command as **npm install**

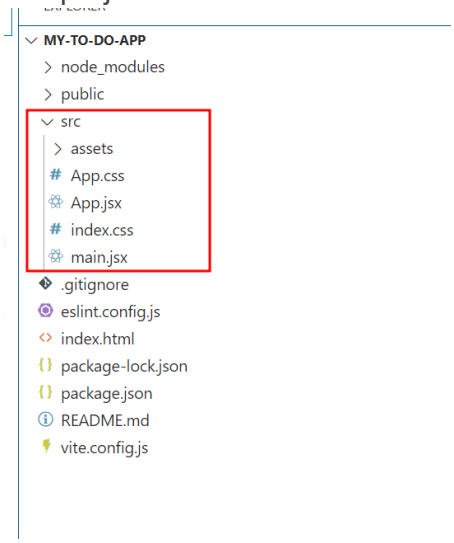
```
Generative AI\my-to-do-app>npm install

258 packages, and audited 259 packages in 14s
```

To run the project we need to use the below command as **npm run dev**

```
using Generative AI\my-to-do-app>npm run dev
```

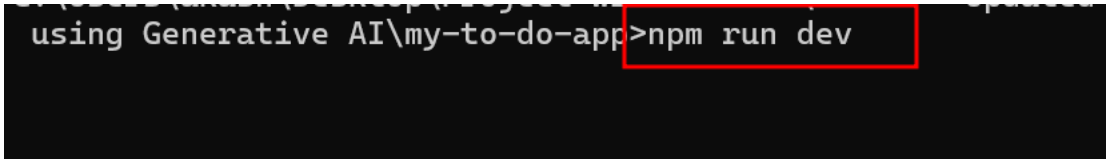
2.3 Now open this project in VS code.

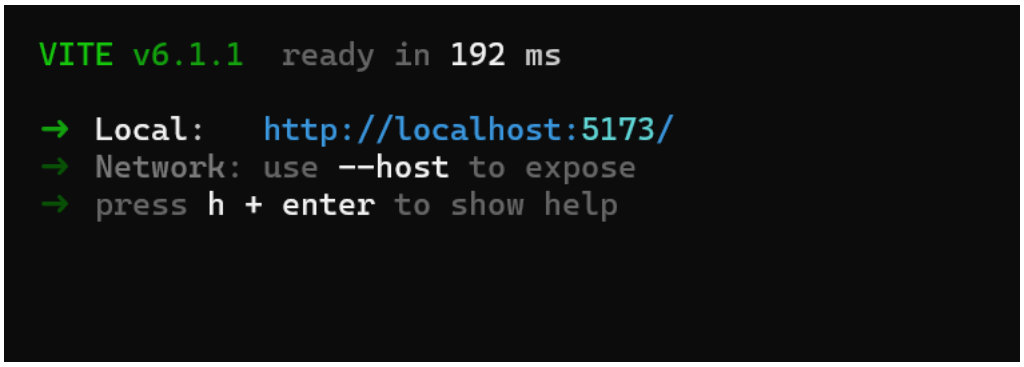
The project structure is



Here we create the project with configuration as vite. **Vite** (pronounced "veet", meaning "fast" in French) is a **next-generation frontend build tool** that provides an extremely fast development experience for **React, Vue, Svelte, and other modern frameworks**. It was created by **Evan You** (the creator of Vue.js).
So in Reat with vite configuration extension of file is .jsx. But if you need to .js also not problem. So file can be .jsx or .js both are same.

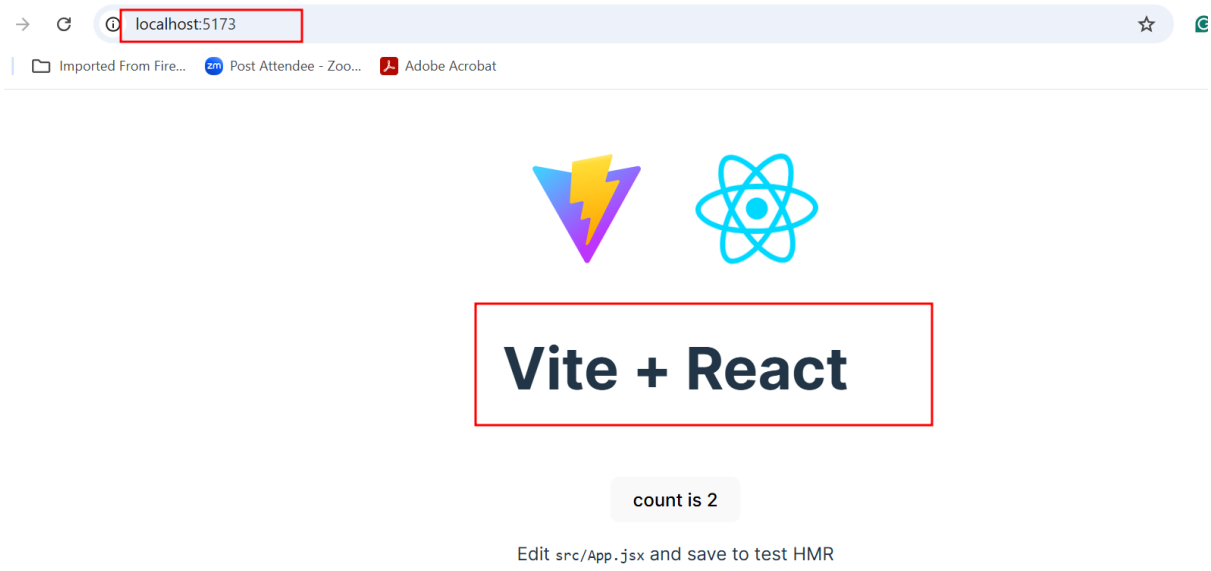
2.4 Now run the project using the **npm run dev** command.





Open the project in below URL.

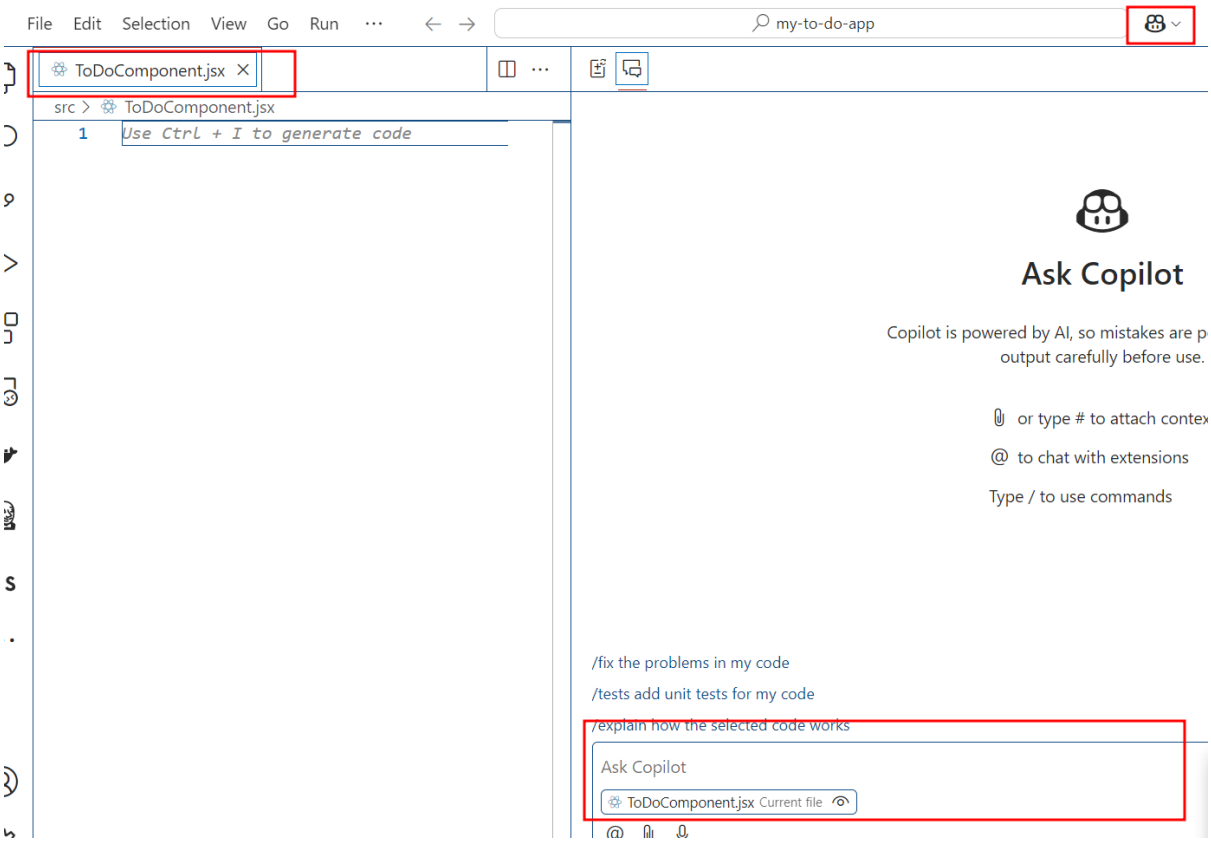
http://localhost:5173/



Step 3 : Create User Defined component with help of GitHub Copilot

3.1 Now we will create the new component with help of GitHub Copilot

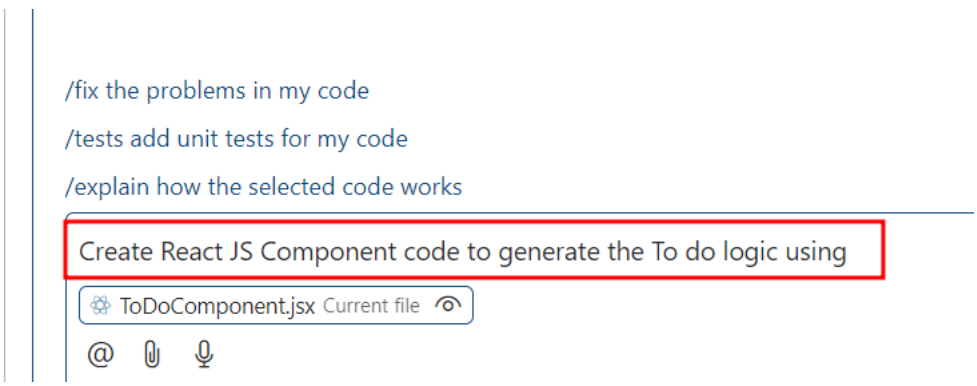
Create a **ToDoComponent.jsx** file inside a src folder.



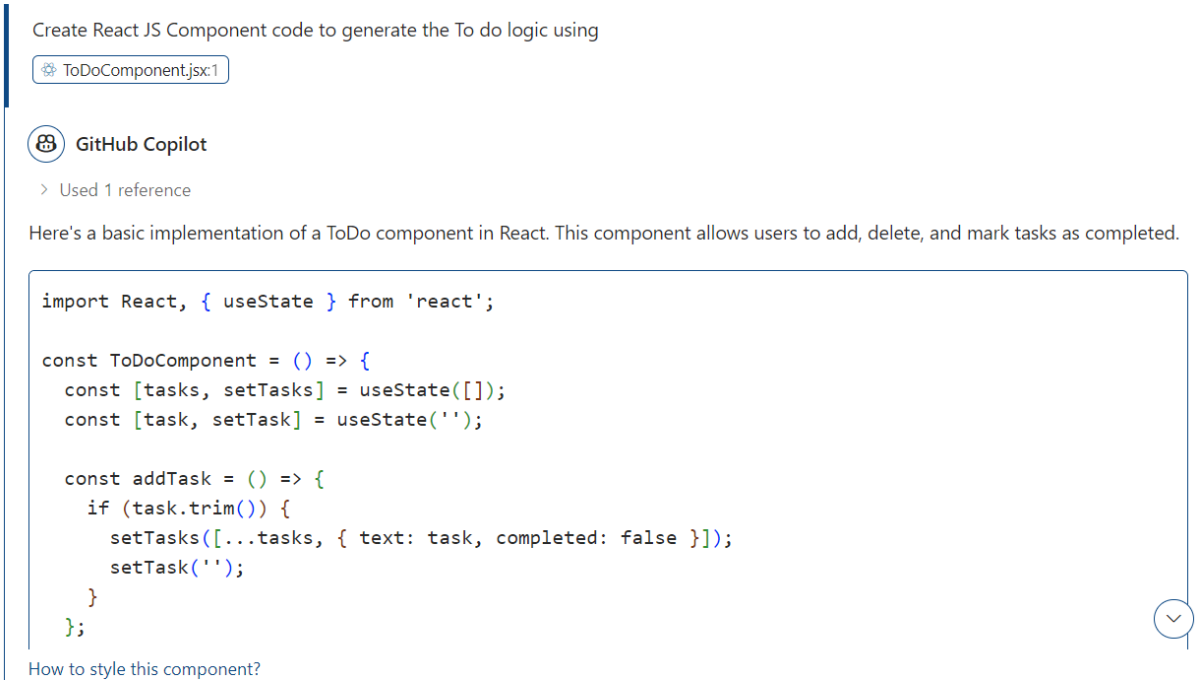
Open the GitHub copilot chat option and write the prompt message to generate the React JS component code.

Then Open the Github copilot terminal and write the prompt message as

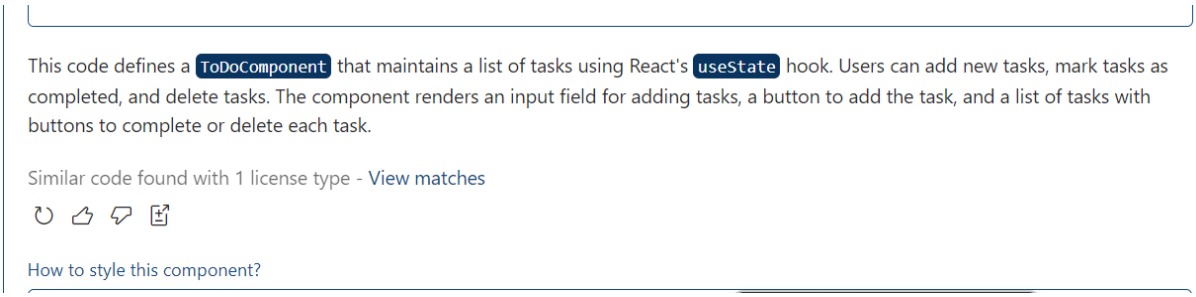
“Create React JS Component code to generate the To do logic using”



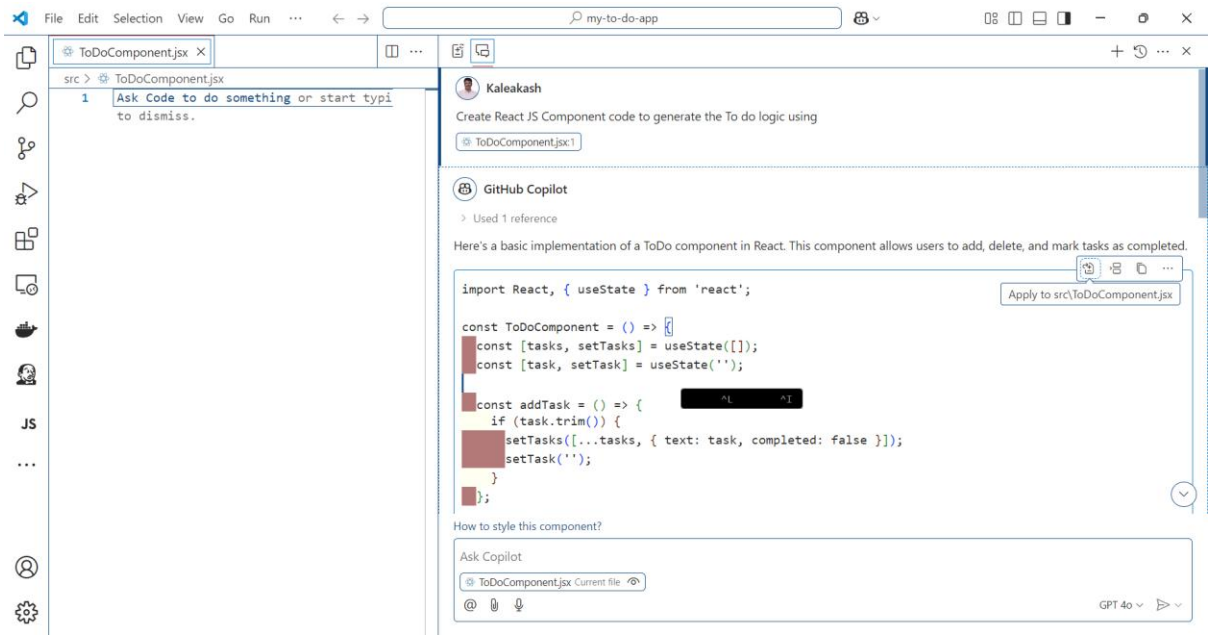
After prompt it generate the code.



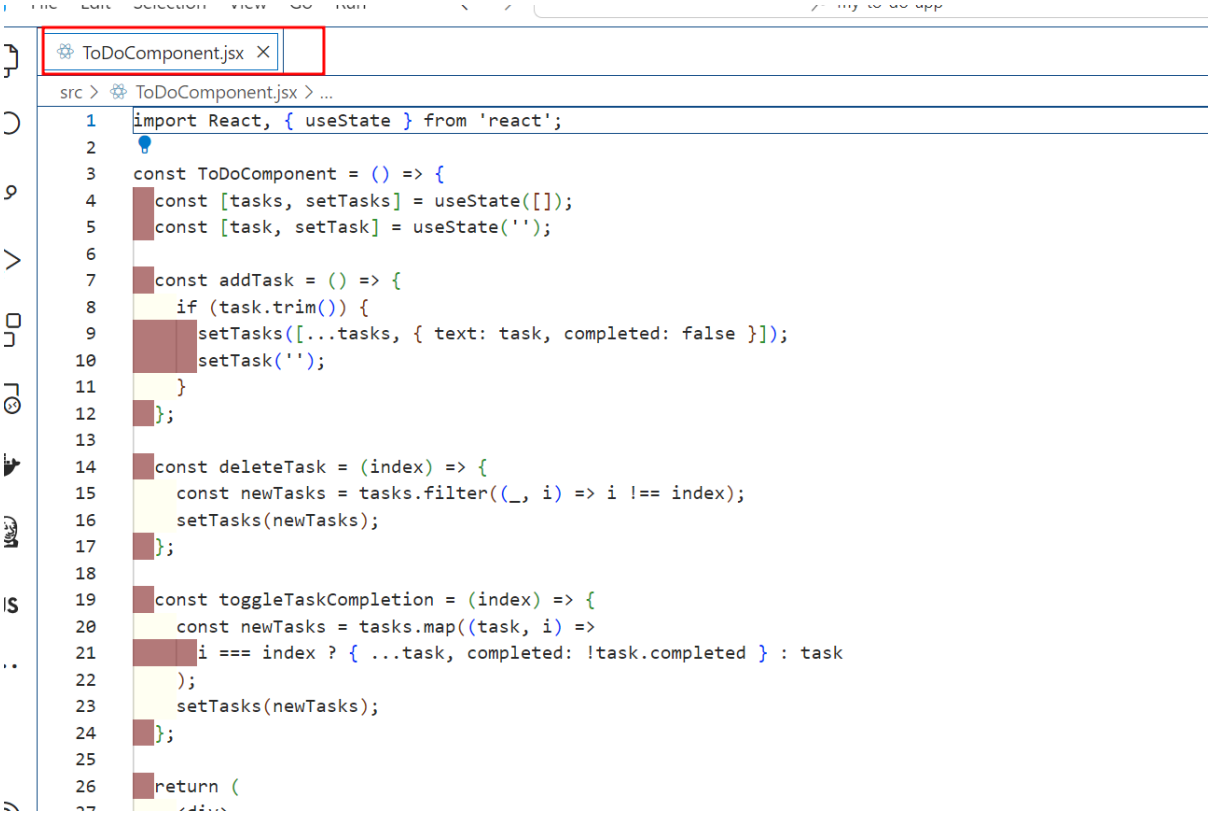
Even GitHub copilot provide the explanation about the generated code.



If you like this code you can click on apply this code the component



Once you click on Apply to src/ToDoComponent.jsx. This code copy and paste automatically in component tsx file.



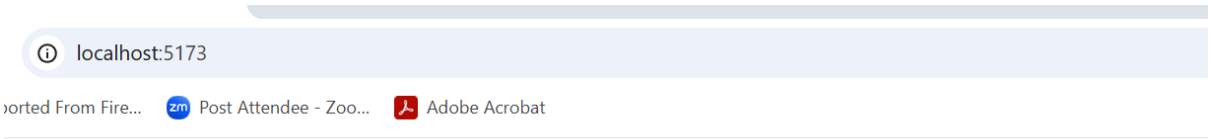
Now Open the App.jsx file and include this component.

App.jsx

src > App.jsx > ...

```jsx
import './App.css'
import ToDoComponent from './ToDoComponent'

function App() {

  return (
    <div>
      <ToDoComponent></ToDoComponent>
    </div>
  )
}

export default App
```

Now check the updated output on browser

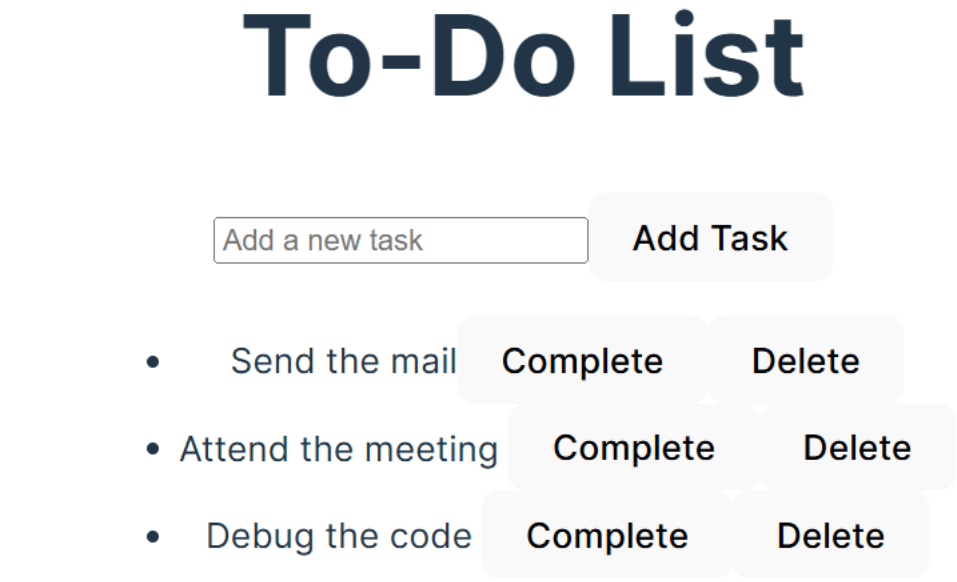


To-Do List

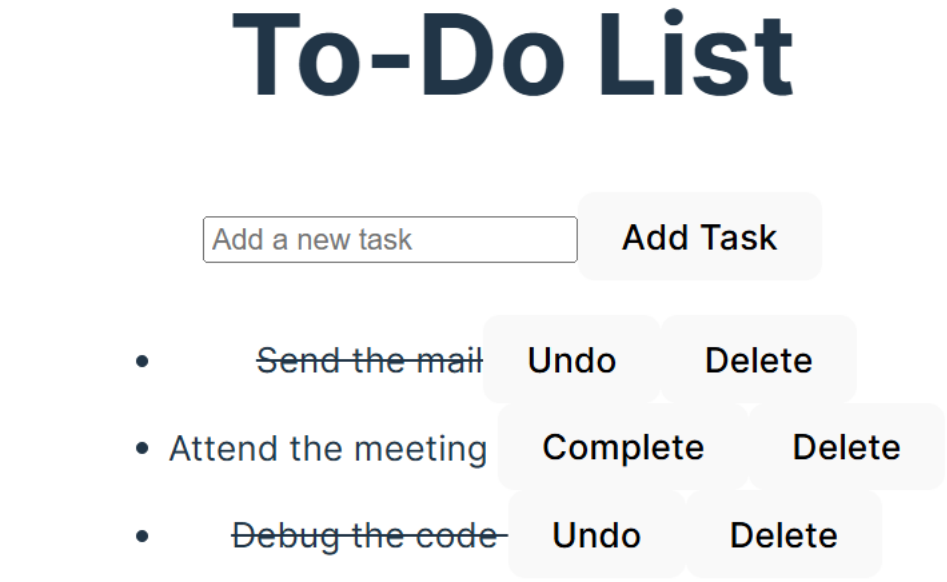
Add a new task

Add Task

Add few more task you want to do.



Make few task complete.



Remove few task.

To-Do List

Add a new task Add Task

- ~~Send the mail~~ Undo Delete
- ~~Debug the code~~ Undo Delete