**Lesson 05 Demo 04**

**Generating Component Architecture and Building a Structured React Blog Using GitHub Copilot**

**Objective:** To generate component architecture and build a structured React blog application using GitHub Copilot for efficient development and code management

**Tools required:** Visual Studio Code and GitHub Copilot

**Prerequisites:** None

Steps to be followed:

1. Generate component architecture and code using GitHub Copilot

**Note:** Please be advised that GitHub Copilot, as an artificial intelligence tool, can produce varied outputs even when presented with similar prompts.

**Step 1:** **Generate component architecture and code using GitHub Copilot**

* 1. Click on the **Chat** icon, as shown in the screenshot below:  
       
     A screenshot of a computer

     Description automatically generated

* 1. Generate the architecture using the following prompt, as shown in the screenshots below:  
     **Define the architecture for your my-react-blog React application by organizing components and styles within the existing folder setup. Place the index.html as the entry point in the public folder, and use index.js in the src folder for initializing the React DOM. Structure the components folder to include App.js as the root component, alongside BlogList.js for displaying a list of blog post summaries, and BlogPost.js for detailed post views. Utilize App.css in the styles folder to manage the overall styling. Outline component interactions and data flow, ensuring efficient content management before code generation.**

A screenshot of a computer

Description automatically generated  
  
A screenshot of a computer program

Description automatically generated  
  
A screenshot of a computer

Description automatically generated

* 1. Click on the **Copy** icon and save the pseudocode representation of the architecture, as shown in the screenshot below:  
       
     A screenshot of a computer program

     Description automatically generated
  2. Use the following prompt along with the pseudocode representation of the architecture, as shown in the screenshot below:  
     **Generate the React component code for the my-react-blog application. Start with App.js to create a basic layout that incorporates the BlogList and BlogPost components. Implement BlogList.js to fetch and display blog post summaries dynamically, possibly using dummy data or API calls if backend integration is planned. Develop BlogPost.js to render full blog post details, fetched based on post ID passed via props or routing. Style the components using CSS rules in App.css to ensure a consistent and visually appealing design across the blog. Ensure all components handle data effectively and interact seamlessly within the application structure.  
     Here's a pseudocode representation of the architecture:**  
     **public/**

**index.html**

**src/**

**index.js**

**components/**

**App.js**

**BlogList.js**

**BlogPost.js**

**styles/**

**App.css**

A screenshot of a computer program

Description automatically generated

* 1. Click on the **Create Workspace** button, as shown in the screenshot below:  
       
     A screenshot of a computer program

     Description automatically generated  
       
     You will see the following interface:  
       
     A screenshot of a computer

     Description automatically generated
  2. Click on the **package.json** file, as shown in the screenshot below:  
       
     A screenshot of a computer program

     Description automatically generated

|  |
| --- |
| **Note**: View all the files for your reference and make the changes per the requirements |

* 1. Open a new terminal and execute the following command to download and install all project dependencies listed in the **package.json** file into the **node\_modules**:   
     **npm i**  
       
     A screenshot of a computer program

     Description automatically generated
  2. Execute the following to launch the application, as shown in the screenshot below:  
     **npm start**  
       
     A screenshot of a computer program

     Description automatically generated  
       
     You will see the following interface:  
       
     A screenshot of a computer

     Description automatically generated

By following these steps, you have successfully generated a component architecture and built a structured React blog application using GitHub Copilot for efficient development and code management.