

Sarika Palle

2403A51L33

B-52

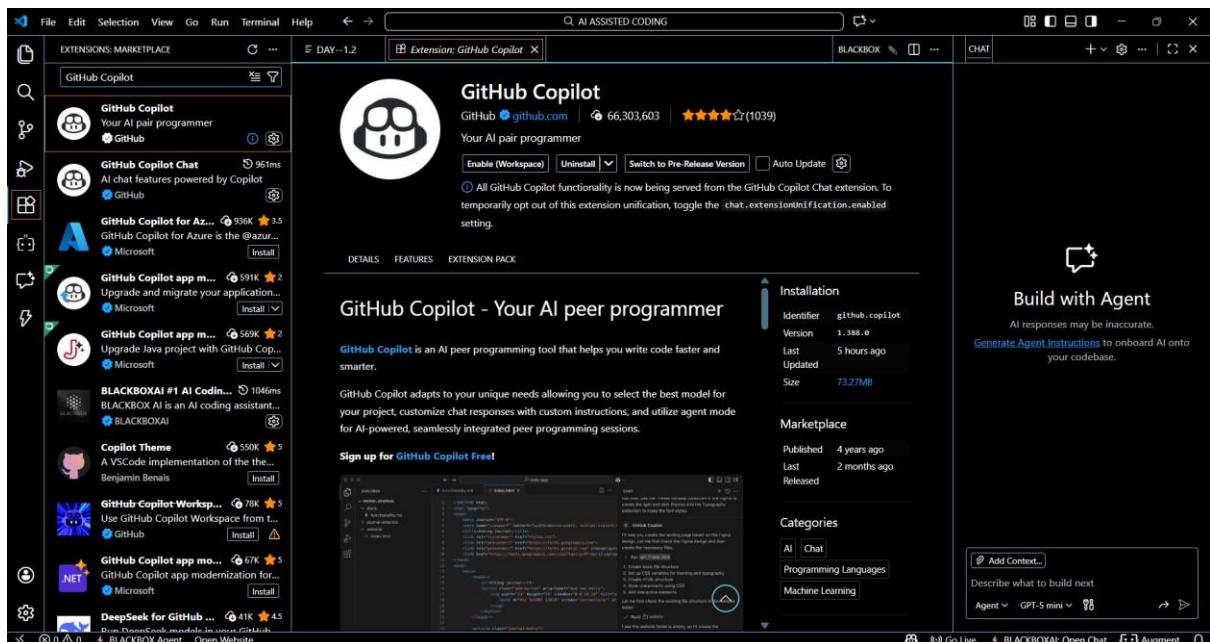
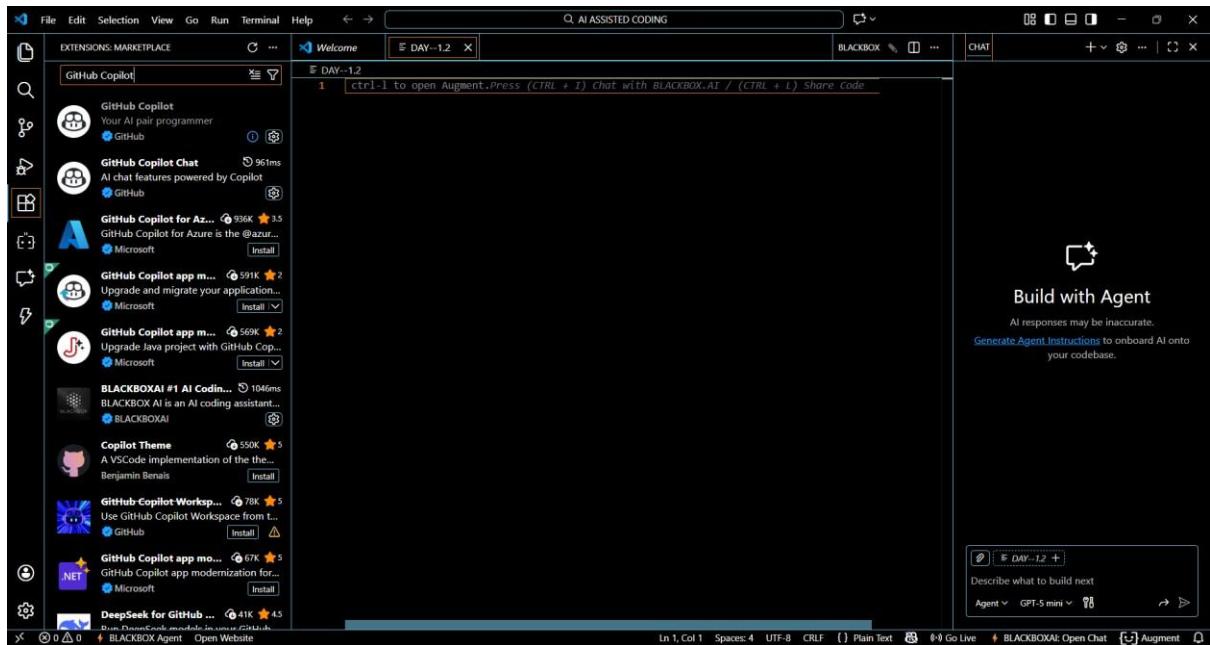
Lab 1.5: AI-Assisted Coding using GitHub Copilot

Task 0: Environment Setup

Steps:

1. Install **Visual Studio Code**
2. Open VS Code → Extensions
3. Search **GitHub Copilot**
4. Click **Install**
5. Sign in with GitHub account
6. Enable Copilot suggestions





Explanation: GitHub Copilot was installed and configured in Visual Studio Code by signing in with a GitHub account. This enables AI-based code suggestions directly inside the editor, helping developers write code faster and more efficiently.

Task 1: String Reversal Without Functions

Prompt:

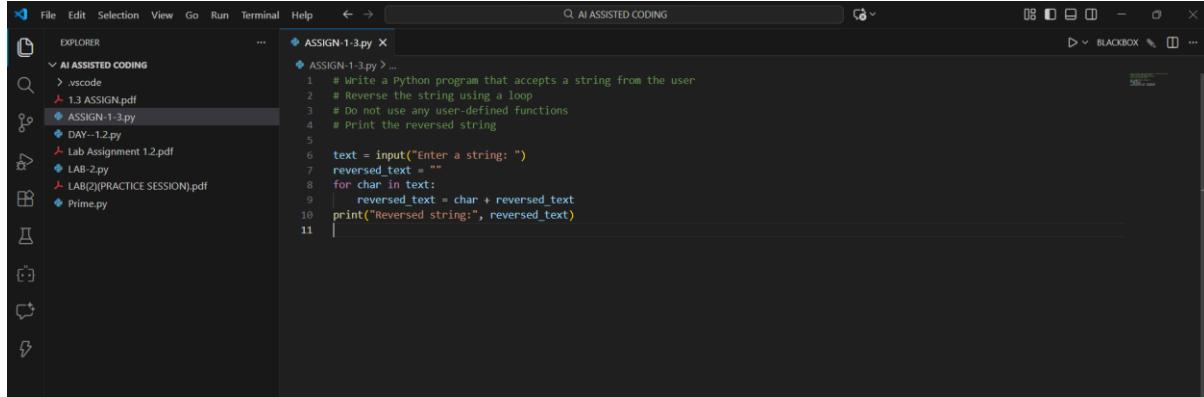
Write a Python program that accepts a string from the user

Reverse the string using a loop

```
# Do not use any user-defined functions
```

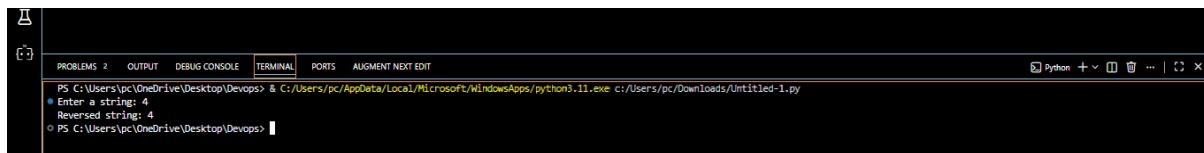
```
# Print the reversed string
```

###CODE:



```
ASSIGN-1-3.py > ...
1  # Write a Python program that accepts a string from the user
2  # Reverse the string using a loop
3  # Do not use any user-defined functions
4  # Print the reversed string
5
6  text = input("Enter a string: ")
7  reversed_text = ""
8  for char in text:
9      reversed_text = char + reversed_text
10 print("Reversed string:", reversed_text)
11 |
```

OUTPUT:



```
PS C:\Users\pc\OneDrive\Desktop\Devops> & C:/Users/pc/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/pc/Downloads/Untitled-1.py
● Enter a string: 4
Reversed string: 4
PS C:\Users\pc\OneDrive\Desktop\Devops>
```

Explanation: In this task, GitHub Copilot generated Python code to reverse a string using a loop without defining any functions. The logic was written directly in the main program, demonstrating basic procedural programming.

Task 2: Code Optimization & Readability

Prompt:

```
# Simplify this string reversal code
```

```
# Remove unnecessary variables
```

```
# Improve readability and efficiency
```

###CODE:

```
13 # Simplify this string reversal code
14 # Remove unnecessary variables
15 # Improve readability and efficiency
16 text = input("Enter a string: ")
17 print("Reversed string:", text[::-1])
```

OUTPUT:

```
PS C:\Users\pc\OneDrive\Desktop\Devops> & C:/Users/pc/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/pc/Downloads/Untitled-1.py
● Enter a string: 2
Reversed string: 2
○ PS C:\Users\pc\OneDrive\Desktop\Devops>
```

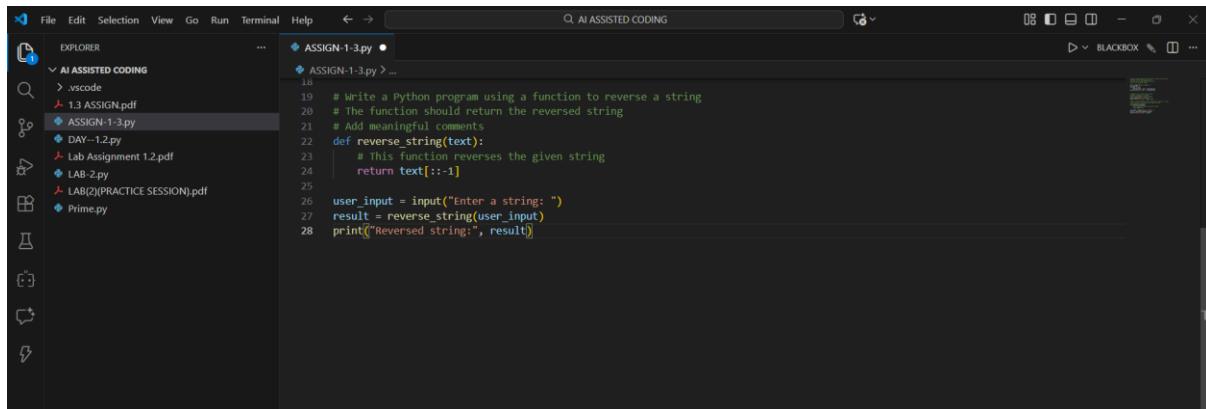
Explanation: The Copilot-generated code was optimized by simplifying the logic and removing unnecessary variables. The improved version produces the same output with better readability and reduced code complexity.

Task 3: String Reversal Using Functions

Prompt:

```
# Write a Python program using a function to reverse a string
# The function should return the reversed string
# Add meaningful comments
```

###CODE:



```
ASSIGN-1-3.py
18 # Write a Python program using a function to reverse a string
19 # The Function should return the reversed string
20 # Add meaningful comments
21 def reverse_string(text):
22     # This function reverses the given string
23     return text[::-1]
24
25
26 user_input = input("Enter a string: ")
27 result = reverse_string(user_input)
28 print("Reversed string:", result)
```

OUTPUT:



```
PS C:\Users\pc\OneDrive\Desktop\Devops> & C:/Users/pc/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/pc/Downloads/Untitled-1.py
● Enter a string: 6
Reversed string: 6
○ PS C:\Users\pc\OneDrive\Desktop\Devops>
```

Explanation: GitHub Copilot was used to create a modular program using a user-defined function to reverse a string. This approach improves reusability, clarity, and makes the code easier to maintain.

Task 4: Procedural vs Modular Comparison

Step 1: Prompt:

```
# Compare string reversal programs with and without functions
```

```
# Discuss clarity, reusability, debugging, and scalability
```

Step 2: Answer (Analysis Output)

Feature	Without Function	With Function
Code clarity	Medium	High
Reusability	Low	High
Debugging	Difficult	Easy
Large applications	Not suitable	Suitable

Explanation: A comparison was made between function-based and non-function-based programs. The analysis shows that modular code is more reusable, easier to debug, and better suited for large-scale applications.

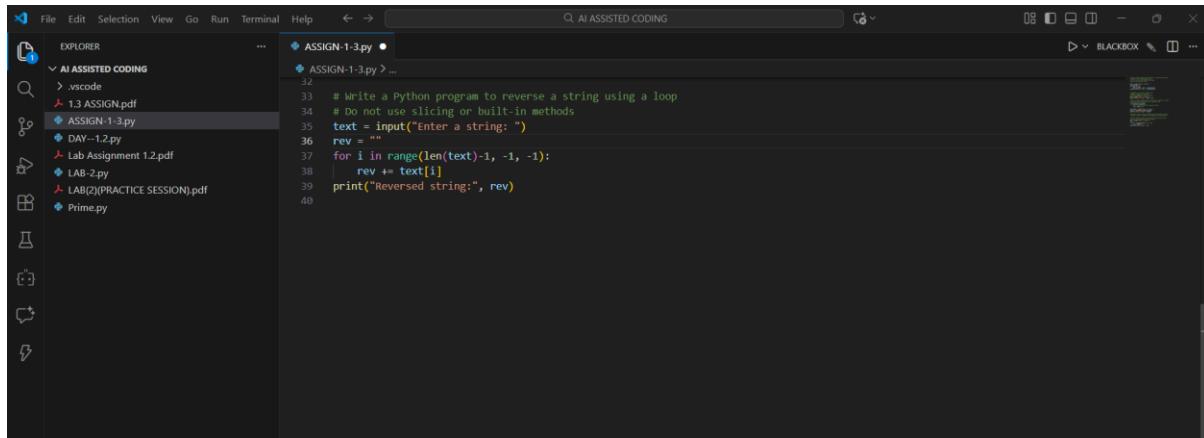
Task 5: Loop vs Built-in Reversal

Step 1: Loop-Based Prompt

```
# Write a Python program to reverse a string using a loop
```

```
# Do not use slicing or built-in methods
```

###CODE:



The screenshot shows the VS Code interface. The left sidebar (Explorer) lists files including 'ASSIGN-1-3.py' (the active file), '1.3 ASSIGN.pdf', 'DAY-1-2.py', 'Lab Assignment 1.2.pdf', 'LAB-2.py', 'LAB(2)(PRACTICE SESSION).pdf', and 'Prime.py'. The main editor area contains the following Python code:

```
32
33 # Write a Python program to reverse a string using a loop
34 # Do not use slicing or built-in methods
35 text = input("Enter a string: ")
36 rev = ""
37 for i in range(len(text)-1, -1, -1):
38     rev += text[i]
39 print("Reversed string:", rev)
40
```

OUTPUT:



The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the following output:

```
PS C:\Users\pc\OneDrive\Desktop\Devops> & C:/Users/pc/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/pc/Downloads/Untitled-1.py
● Enter a string: 6
Reversed string: 6
○ PS C:\Users\pc\OneDrive\Desktop\Devops>
```

Step 2: Built-in Prompt

```
# Write a Python program to reverse a string using slicing
```

###CODE:

The screenshot shows the VS Code interface with the "AI ASSISTED CODING" extension active. The Explorer sidebar on the left lists files including ".vscode", "1.3 ASSIGN.pdf", "ASSIGN-1.2.py", "DAY-1.2.py", "Lab Assignment 1.2.pdf", "LAB-2.py", "LAB(2)(PRACTICE SESSION).pdf", and "Prime.py". The main editor area displays a Python script named "ASSIGN-1.3.py":

```
41 # Write a Python program to reverse a string using slicing
42 text = input("Enter a string: ")
43 print("Reversed string:", text[::-1])
44
```

OUTPUT:

The screenshot shows the VS Code terminal window. The tab bar at the top indicates the current tab is "TERMINAL". The terminal output shows the execution of the Python script and its results:

```
PS C:\Users\pc\OneDrive\Desktop\Devops> & C:/Users/pc/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/pc/Downloads/Untitled-1.py
● Enter a string: 24
Reversed string: 42
○ PS C:\Users\pc\OneDrive\Desktop\Devops>
```

Explanation: Two different string reversal approaches were generated using Copilot: loop-based and built-in slicing. Both methods have the same time complexity, but the built-in approach is more concise and readable.