

Hepsiba Devara

2403A51L25

B-51

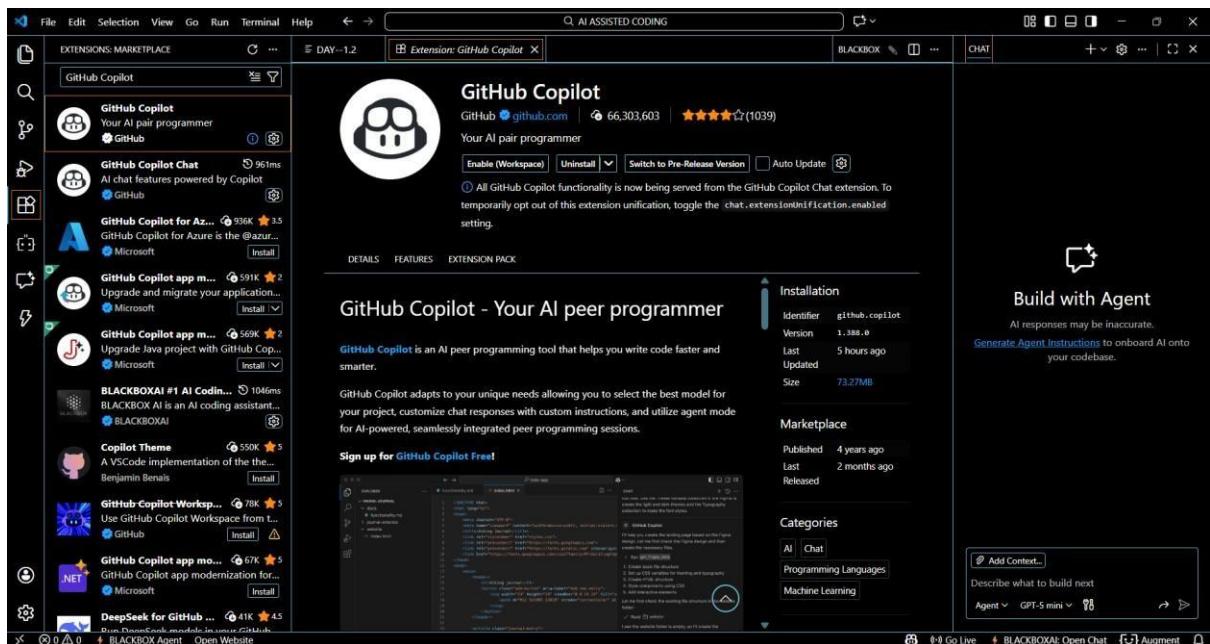
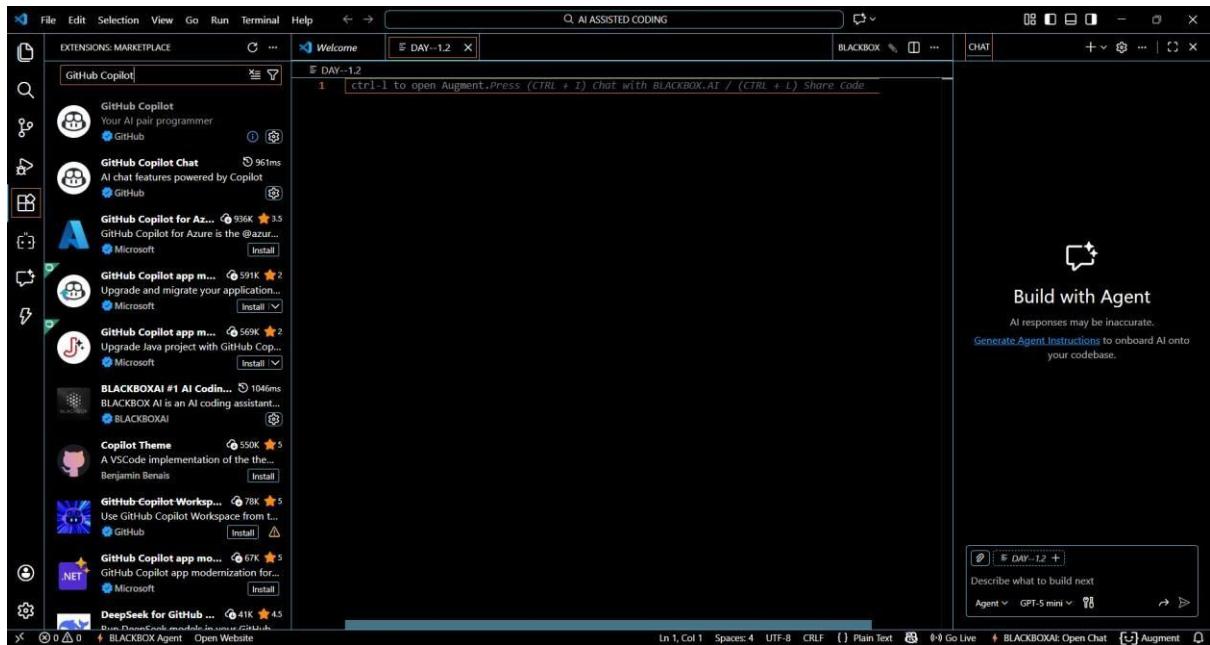
Lab 1.5: AI-Assisted Coding using GitHub Copilot

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:

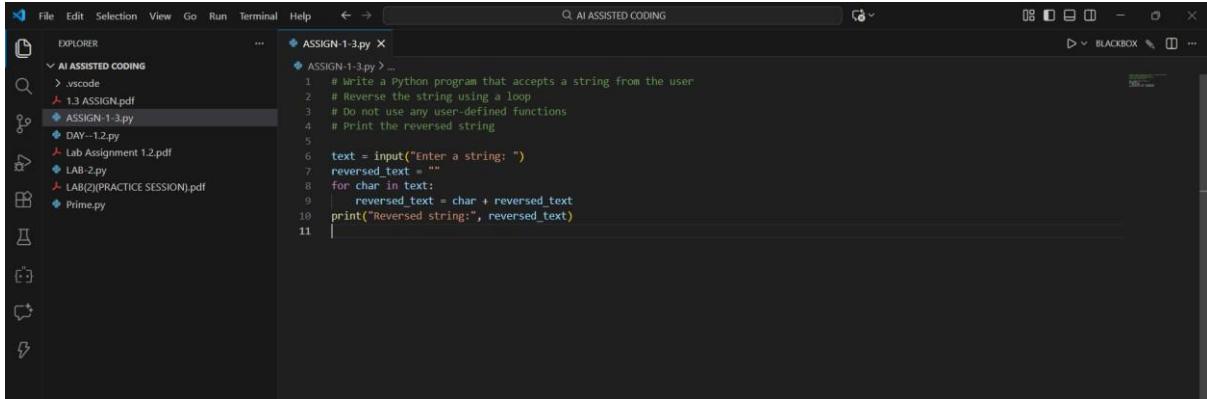
```
# Write a Python program that accepts a string from the user
```

```
# Reverse the string using a loop
```

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

```
# Print the reversed string
```

###CODE:



OUTPUT:



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:

The screenshot shows the VS Code interface with the 'AI ASSISTED CODING' extension active. The Explorer sidebar on the left lists files like 'vscode', '1.3 ASSIGN.pdf', 'ASSIGN-1-3.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 Python code for reversing a string:

```
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:

The screenshot shows the VS Code terminal window. It displays the command 'python Untitled-1.py' being run, followed by the output: 'Enter a string: 2' and 'Reversed string: 2'. The terminal also shows the current working directory as '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:
```

The screenshot shows the VS Code interface with the 'AI ASSISTED CODING' extension active. The Explorer sidebar on the left lists files like 'vscode', '1.3 ASSIGN.pdf', 'ASSIGN-1-3.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 Python code for reversing a string using a function:

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

OUTPUT:



A screenshot of a terminal window titled "Python". The tab bar includes "PROBLEMS 2", "OUTPUT", "DEBUG CONSOLE", "TERMINAL", "PORTS", and "AUGMENT NEXT EDIT". The terminal content shows a Python session:

```
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 nonfunction-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:
```

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

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:**

```
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:



A screenshot of a terminal window within a code editor interface. The terminal tab is active at the top. The command entered is `C:/Users/pc/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/pc/Downloads/Untitled-1.py`. The output shows:

```
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.