

Course Title: AI Assisted Coding

Course Code: 23CS002PC304

Faculty Name: Dr. R. Prashant Kumar

Name: Vyshnavi K

HT no: 2303A52273- Batch(36)

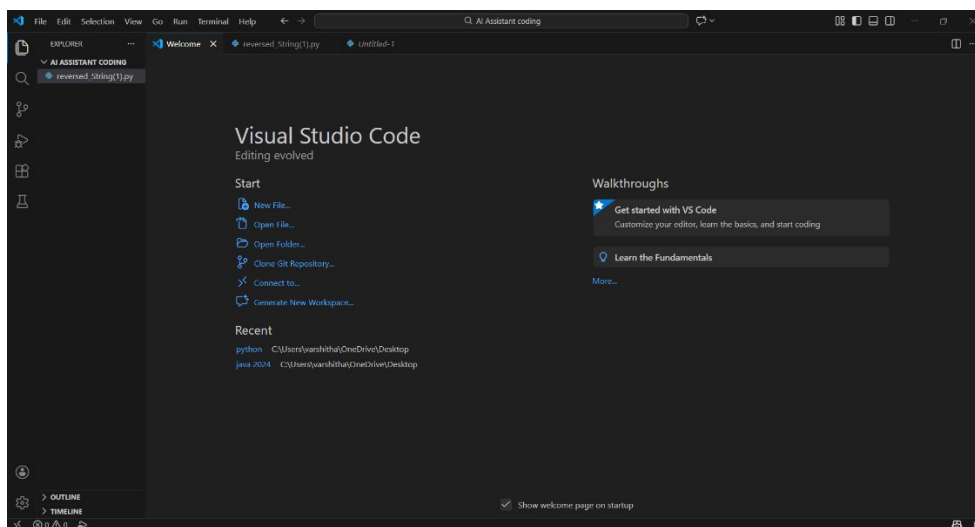
Question:

Lab 1: Environment Setup – GitHub Copilot and VS Code Integration + Understanding AI-assisted Coding Workflow

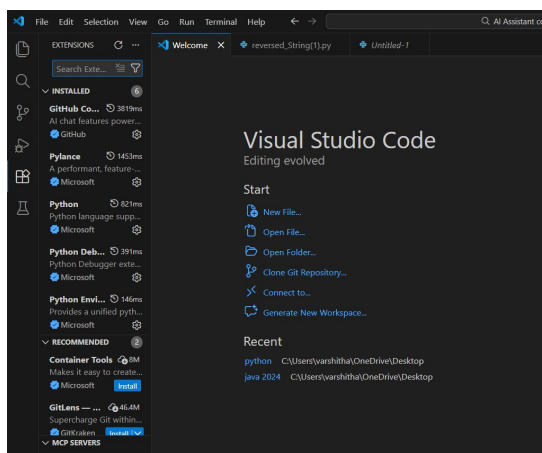
Task 0:

- Install and configure GitHub Copilot in VS Code. Take screenshots of each step.

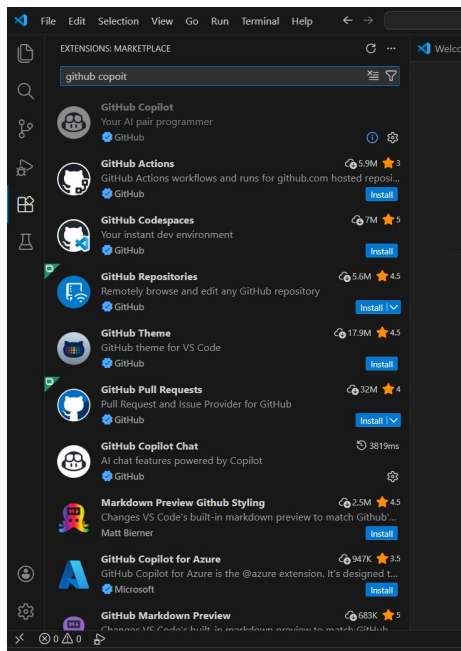
Step 1: Open Visual Studio Code



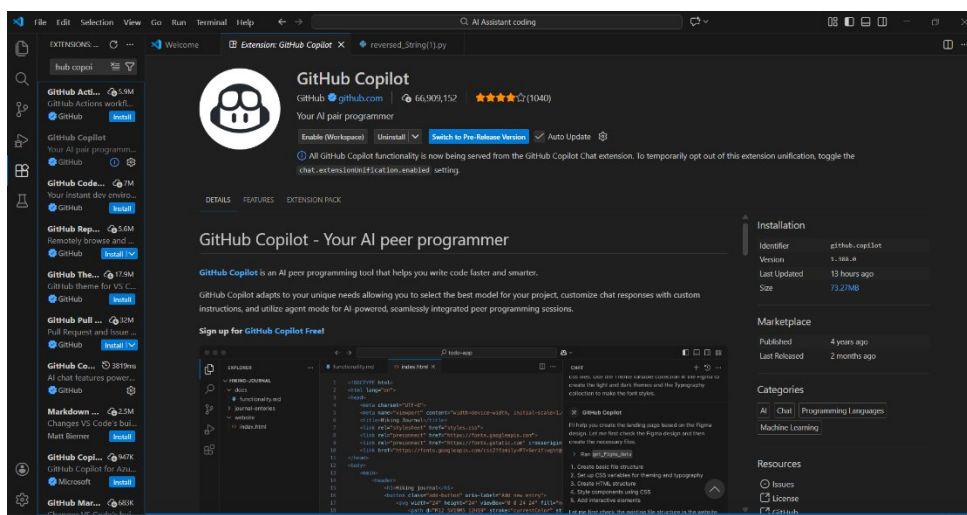
Step 2: Open Extensions Panel



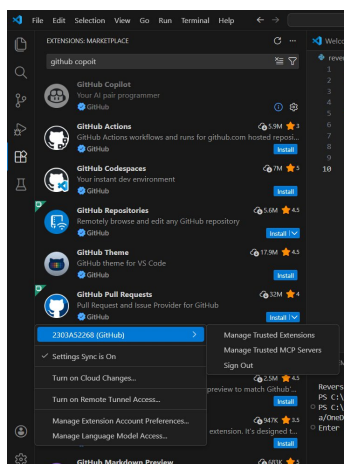
Step 3: Search for GitHub Copilot



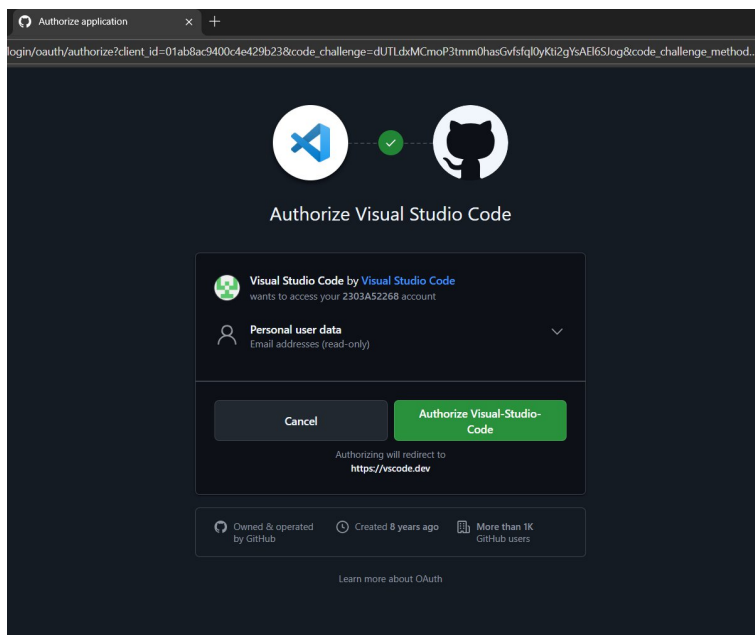
Step 4: Install GitHub Copilot



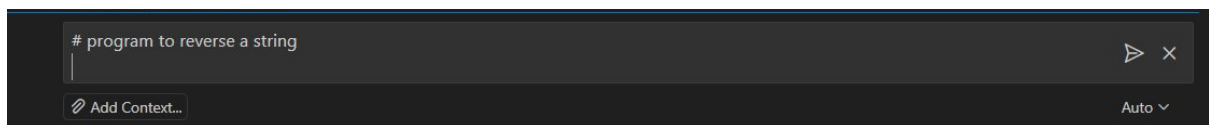
Step 5: Sign in to GitHub Account



Step 6: Authorize GitHub Copilot

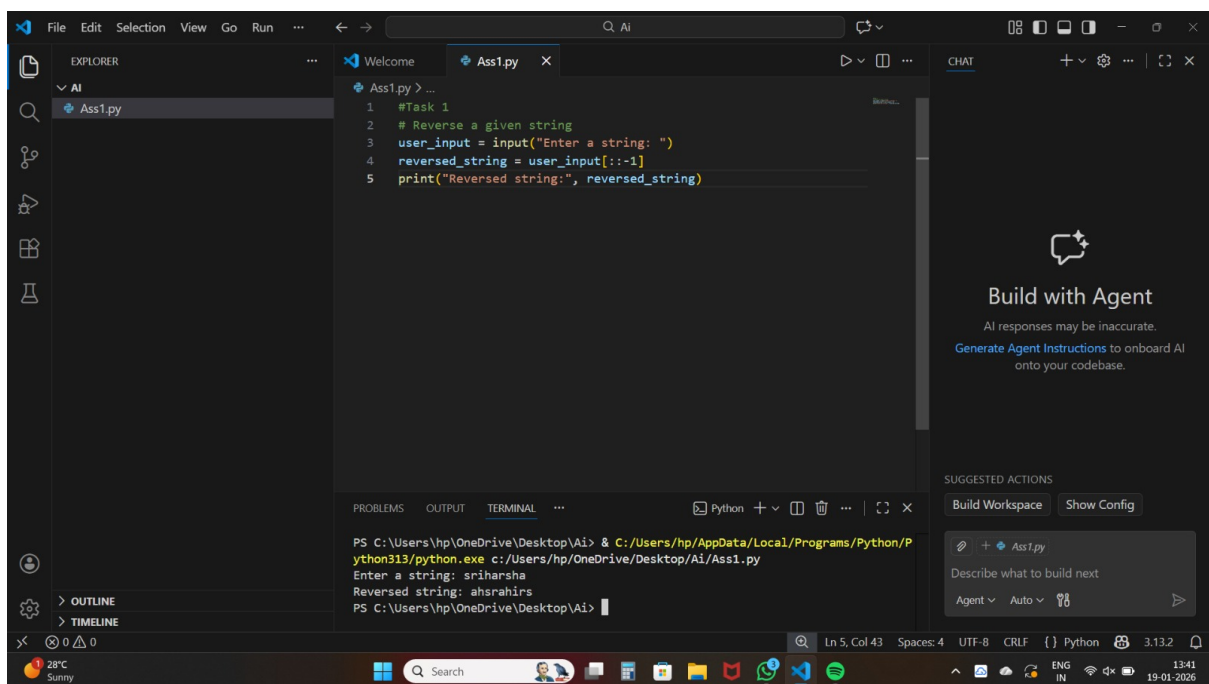


Step 7: Verify Copilot is Enabled



Task 1: AI-Generated Logic Without Modularization (String Reversal Without Functions)

program to reverse a string

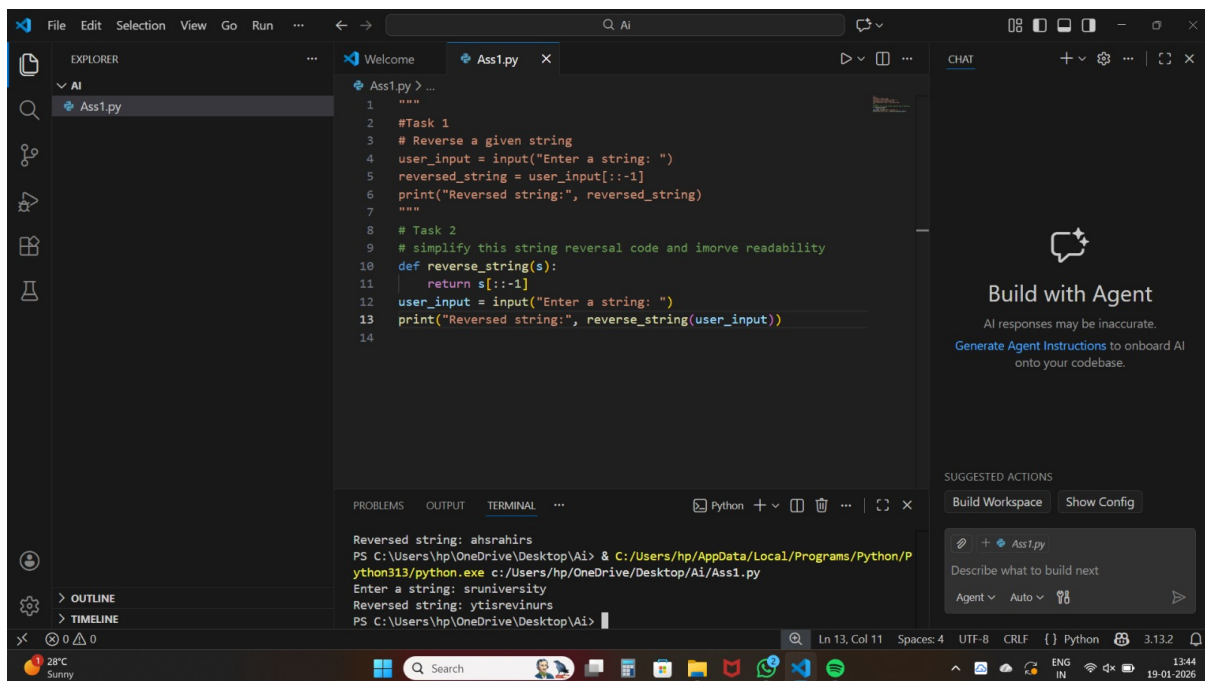


Explanation

- The input() function takes a string from the user.
- An empty string rev is created to store the reversed result.
- The for loop iterates through the string from the last character to the first.
- Each character is appended to rev.
- The final reversed string is printed.
- The logic is written directly in the main code without using functions

Task 2: Efficiency & Logic Optimization (Readability Improvement)

Simplified String Reversal Code



```

1  """
2  #Task 1
3  # Reverse a given string
4  user_input = input("Enter a string: ")
5  reversed_string = user_input[::-1]
6  print("Reversed string:", reversed_string)
7  """
8  # Task 2
9  # simplify this string reversal code and improve readability
10 def reverse_string(s):
11     return s[::-1]
12 user_input = input("Enter a string: ")
13 print("Reversed string:", reverse_string(user_input))
14

```

Reversed string: ahsrahirs
PS C:\Users\hp\OneDrive\Desktop\Ai> & C:/Users/hp/AppData/Local/Programs/Python/Python313/python.exe c:/Users/hp/OneDrive/Desktop/Ai/Ass1.py
Enter a string: sruniversi
Reversed string: ytisrevinurs
PS C:\Users\hp\OneDrive\Desktop\Ai>

Explanation of Optimization

- The loop and extra variable were removed
- Python slicing reverses the string in a single step
- Code is shorter, cleaner, and easier to understand

Time Complexity Explanation

- Original code: **O(n)** (manual loop)
- Optimized code: **O(n)** (built-in slicing)

- Although complexity remains the same, slicing is **faster in practice** due to internal optimization

Task 3: Modular Design Using AI Assistance (String Reversal Using Functions)

Write a Python function to reverse a string

```

9 # Task 2
10 # simplify this string reversal code and improve readability
11 def reverse_string(s):
12     return s[::-1]
13 user_input = input("Enter a string: ")
14 print("Reversed string:", reverse_string(user_input))
15 """
16 # Task 3
17 # Write a python program using a function to reverse a string
18 #Add meaningful comments
19 def reverse_string(s):
20     return s[::-1] #Slicing the string to reverse it
21 # Get user input
22 user_input = input("Enter a string: ")
23 # Call the function and display the reversed
24 reverse_str = reverse_string(user_input)
25 print(f"Reversed string: {reverse_str}")

```

Reversed string: ytisrevinurs
PS C:\Users\hp\OneDrive\Desktop\Ai> & C:/Users/hp/AppData/Local/Programs/Python/Python313/python.exe c:/Users/hp/OneDrive/Desktop/Ai/Ass1.py
Enter a string: warangal
Reversed string: lagnaraw
PS C:\Users\hp\OneDrive\Desktop\Ai>

Explanation

- A function `reverse_string()` is defined to reverse a string.
- The function takes one parameter `text`.
- The slicing method `[::-1]` is used to reverse the string.
- The reversed string is returned to the caller.
- User input is passed to the function.
- The result is printed.
- This modular approach improves reusability and readability.

Task 4: Comparative Analysis – Procedural vs Modular Approach (With vs Without Functions)

Aspect	Without Function (Procedural)	With Function (Modular)
Code Clarity	Moderate	High
Reusability	Not reusable	Highly reusable
Debugging	Difficult	Easier
Maintainability	Low	High
Large-scale Suitability	Poor	Good

Task 5: AI-Generated Iterative vs Recursive Fibonacci Approaches (Different Algorithmic Approaches to String Reversal)

#Generate a loop based string reversal program in Python

```

19 #Add meaningful comments
20 def reverse_string(s):
21     return s[::-1] #Slicing the string to reverse it
22 # Get user input
23 user_input = input("Enter a string: ")
24 # Call the function and display the reversed
25 reverse_str = reverse_string(user_input)
26 print(f"Reversed string: {reverse_str}")
27 """
28 #Generate a loop based string reversal program in python
29 def reverse_string_loop(s):
30     return s[::-1]
31 # Example usage
32 original_string = "Hello, World!"
33 reversed_string = reverse_string_loop(original_string)
34 print(reversed_string)

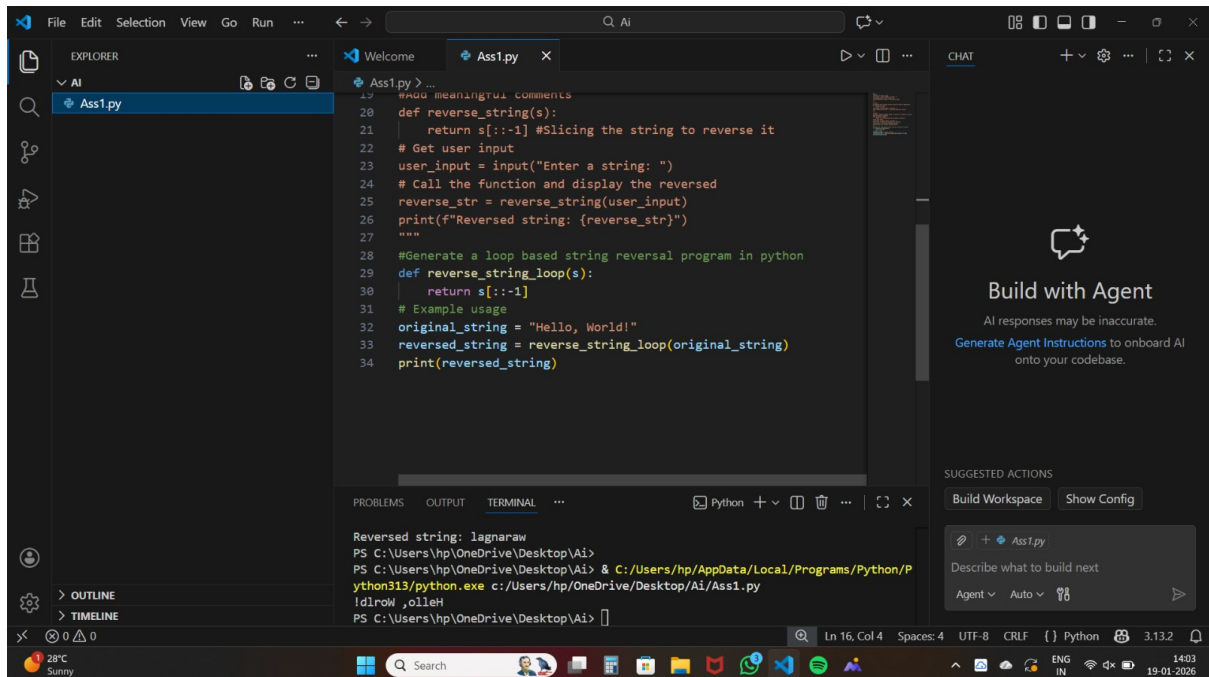
```

Reversed string: dlroW ,olleH

Explanation

- The user inputs a string.
- An empty string rev is created.
- The loop reads each character from left to right.
- Each character is added at the beginning of rev, reversing the order.
- The reversed string is printed.
- This method helps understand string manipulation logic.

#Generate a slicing based string reversal program in Python



```
19 #Add meaningful comments
20 def reverse_string(s):
21     return s[::-1] #Slicing the string to reverse it
22 # Get user input
23 user_input = input("Enter a string: ")
24 # Call the function and display the reversed
25 reverse_str = reverse_string(user_input)
26 print(f"Reversed string: {reverse_str}")
27 """
28 #Generate a loop based string reversal program in python
29 def reverse_string_loop(s):
30     return s[::-1]
31 # Example usage
32 original_string = "Hello, World!"
33 reversed_string = reverse_string_loop(original_string)
34 print(reversed_string)
```

Reversed string: lagnaraw
PS C:\Users\hpn\OneDrive\Desktop\Ai> & C:/Users/hp/AppData/Local/Programs/Python/Python313/python.exe c:/Users/hp/OneDrive/Desktop/Ai/Ass1.py
Idlrow ,olleH
PS C:\Users\hpn\OneDrive\Desktop\Ai>

Explanation

- The string is taken from the user.
- Python slicing reverses the string efficiently.
- The reversed string is printed directly.
- This approach is best for large inputs and real-world applications.

Comparison of Approaches

Aspect	Loop-Based	Slicing-Based
Execution Flow	Step-by-step reversal	Single operation
Time Complexity	O(n)	O(n)
Performance for Large Inputs	Slower	Faster
Readability	Moderate	Very High
Best Usage	Learning logic	Production code