

Course Title: AI Assisted Coding

Course Code: 23CS002PC304

Faculty Name: Dr. R. Prashant Kumar

Name: K.Keerthi Reddy

HT no: 2303A52263- Batch(36)

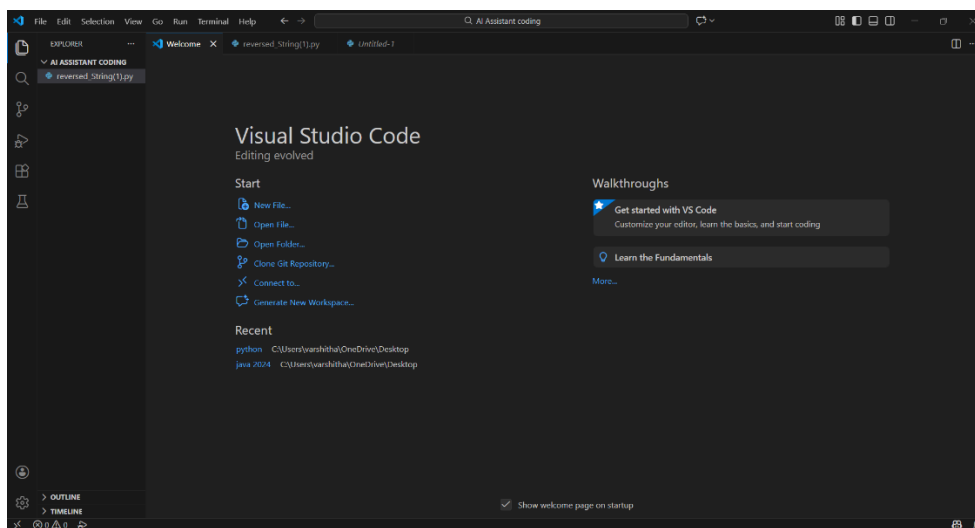
Question:

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

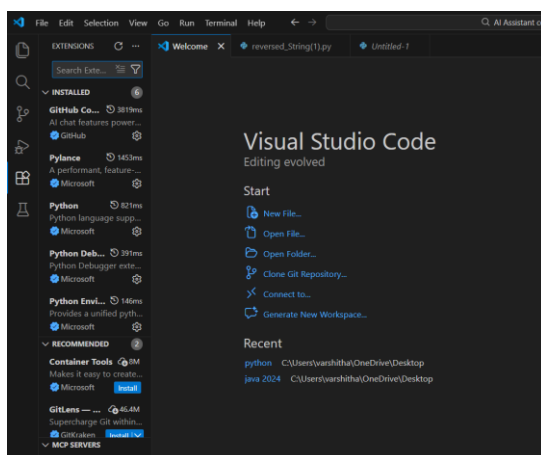
Task 0:

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

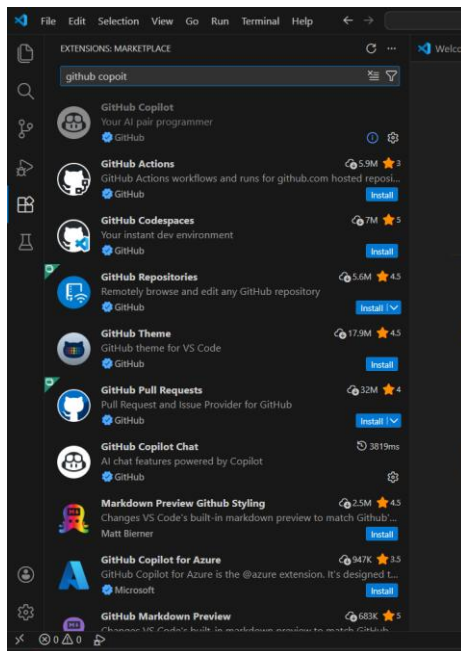
Step 1: Open Visual Studio Code



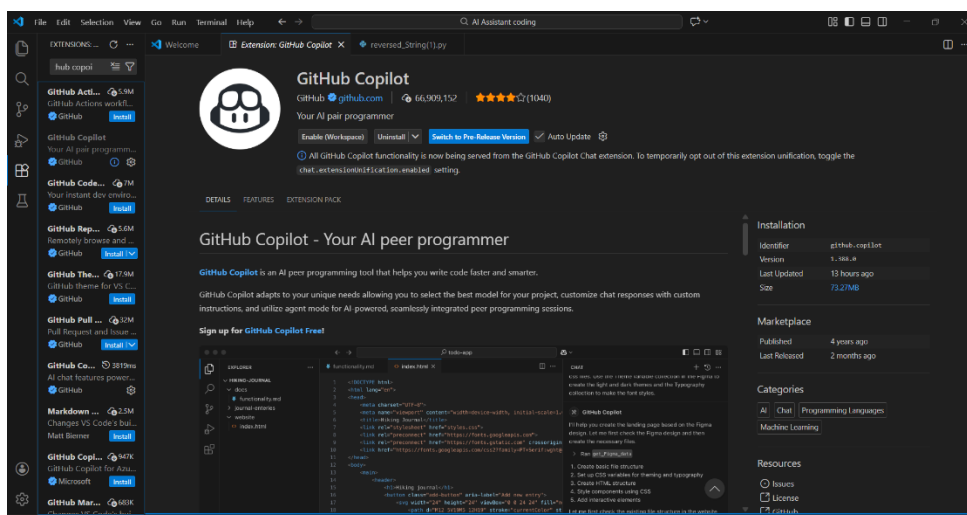
Step 2: Open Extensions Panel



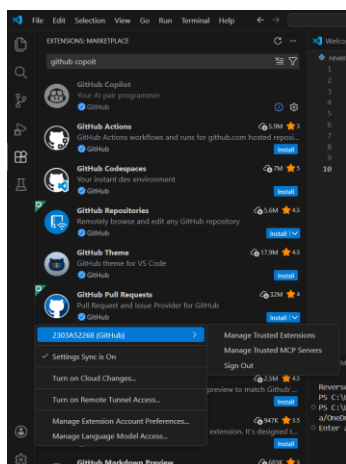
Step 3: Search for GitHub Copilot



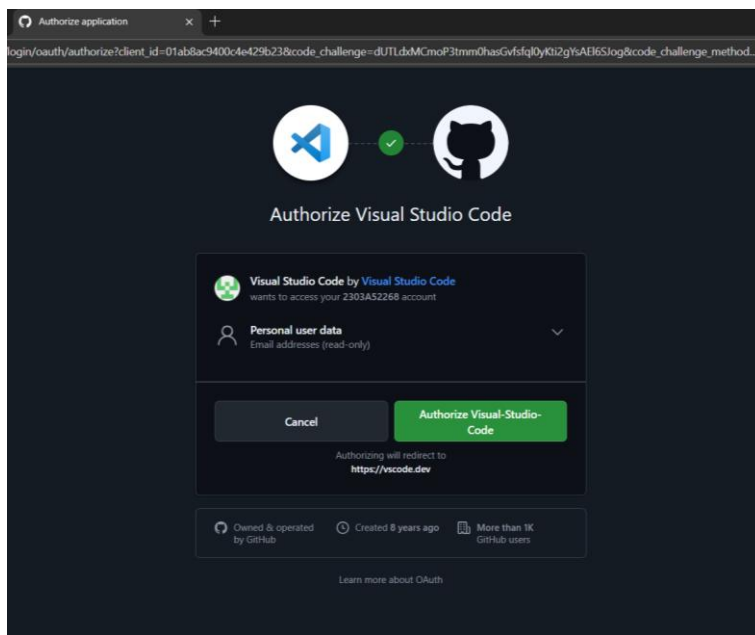
Step 4: Install GitHub Copilot



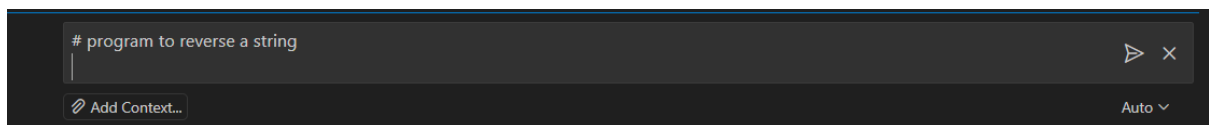
Step 5: Sign in to GitHub Account



Step 6: Authorize GitHub Copilot

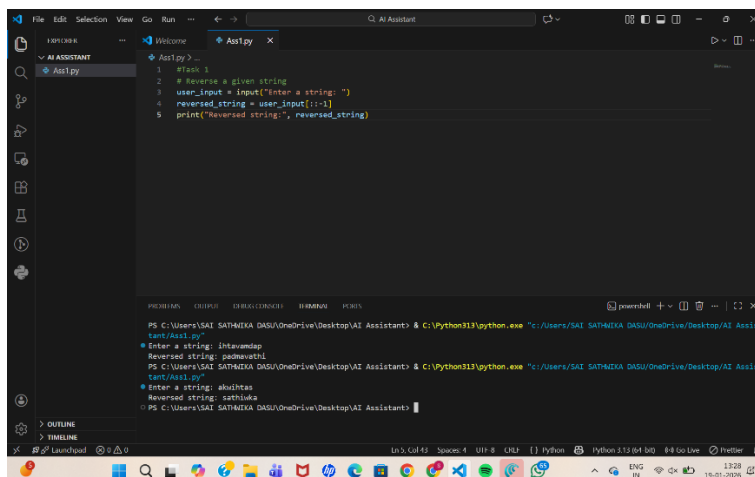


Step 7: Verify Copilot is Enabled



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))

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\SAI SATHNIKA DASU\OneDrive\Desktop\AI Assistant> & C:\Python313\python.exe "c:/Users/SAI SATHNIKA DASU/OneDrive/Desktop/AI Assistant/Ass1.py"

Enter a string: akwihkas
Reversed string: sathiwka

PS C:\Users\SAI SATHNIKA DASU\OneDrive\Desktop\AI Assistant> ^C

PS C:\Users\SAI SATHNIKA DASU\OneDrive\Desktop\AI Assistant> & C:\Python313\python.exe "c:/Users/SAI SATHNIKA DASU/OneDrive/Desktop/AI Assistant/Ass1.py"

Enter a string: sr university
Reversed string: ytrisrevinu rs

PS C:\Users\SAI SATHNIKA DASU\OneDrive\Desktop\AI Assistant>

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

Write a Python function to reverse a string

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

The screenshot shows a VS Code editor with a file named `Ass1.py`. The code defines two functions to reverse a string: `reverse_string(s)` using slicing and `reverse_string_loop(s)` using a loop. The `reverse_string_loop` function iterates through each character of the string and builds a reversed string by prepending each character. The terminal output shows the program being executed twice: first with the input "akwhtas" resulting in "sathiwka", and then with the input "hello world" resulting in "dlrow olleh".

```

18 # Write a python program using a function to reverse a string
19 #Add meaningful comments
20 def reverse_string(s):
21     return s[::-1] #Slicing the string to reverse it
22
23 # Get user input
24 user_input = input("Enter a string: ")
25 # Call the function and display the reversed
26 reverse_str = reverse_string(user_input)
27 print(f"Reversed string: {reverse_str}")
28
29 #Generate a loop based string reversal program in python
30 def reverse_string_loop(s):
31     return s[::-1]
32
33 # Example usage
34 original_string = "Hello, World!"
35 reversed_string = reverse_string_loop(original_string)
36 print(reversed_string)
37

```

```

Enter a string: akwhtas
Reversed string: sathiwka
I Assistant> & C:\Python313\python.exe "c:/Users/SAI SATHMIKA DASU/OneDrive/Desktop/AI Assistant/Ass1.py"
Enter a string: hello world
Reversed string: dlrow olleh
PS C:\Users\SAI SATHMIKA DASU\OneDrive\Desktop\AI Assistant>

```

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.

#Generate a slicing based string reversal program in Python

The screenshot shows a Windows 11 desktop environment. At the bottom, a Windows 10 taskbar is visible with icons for various applications including File Explorer, Edge, and several instances of Visual Studio Code. The main focus is the Visual Studio Code editor window. The Explorer sidebar on the left displays the project structure, with 'Ass1.py' selected under the 'AI ASSISTANT' folder. The main editor area shows a Python script for reversing a string. The script includes comments and two functions: 'reverse_string' and 'reverse_string_loop'. The output window at the bottom shows the execution results, including the reversed string 'sathiwka' and the loop-based reversal output 'dlrow ,olleH'.

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