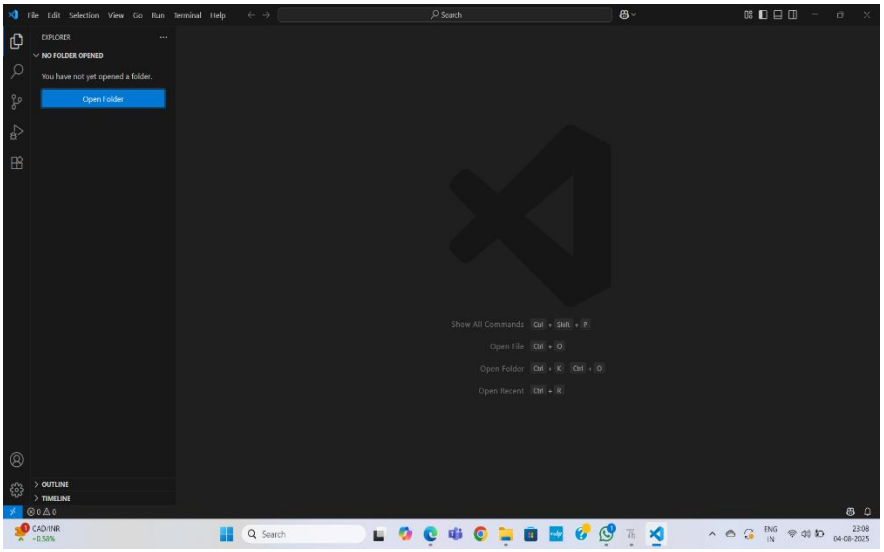
SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
ProgramName:B. Tech		**Assignment Type: Lab**	**AcademicYear:**2025-2026
CourseCoordinatorName	Venkataramana Veeramsetty		
Instructor(s)Name	Dr. V. Venkataramana (Co-ordinator) Dr. T. Sampath Kumar Dr. Pramoda Patro Dr. Brij Kishor Tiwari Dr.J.Ravichander Dr. Mohammand Ali Shaik Dr. Anirodh Kumar Mr. S.Naresh Kumar Dr. RAJESH VELPULA Mr. Kundhan Kumar Ms. Ch.Rajitha Mr. M Prakash Mr. B.Raju Intern 1 (Dharma teja) Intern 2 (Sai Prasad) Intern 3 (Sowmya) NS_2 (Mounika)		
CourseCode	24CS002PC215	**CourseTitle**	AI Assisted Coding
Year/Sem	II/I	**Regulation**	R24
Date and Day of Assignment	Week1 - Tuesday	**Time(s)**	
Duration	2 Hours	**Applicableto Batches**	24CSBTB01 To 24CSBTB39
AssignmentNumber:1.2(Present assignment number)/**24**(Total number of assignments)			

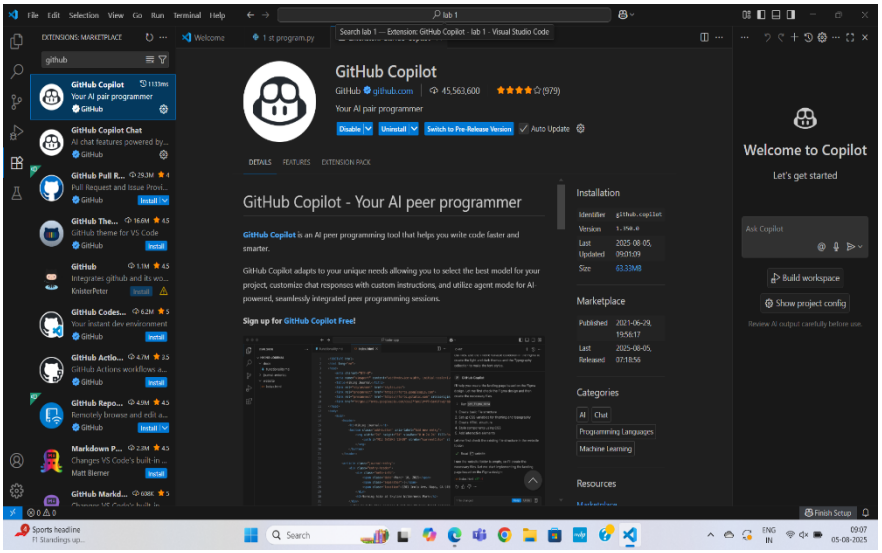
Q.No.	Question	*ExpectedTime to complete*
1	Lab 1: Environment Setup – GitHub Copilot and VS Code Integration **Lab Objectives:** ● To install and configure GitHub Copilot in Visual Studio Code. ● To explore AI-assisted code generation using GitHub Copilot.	Week1 - wednesday

- To analyze the accuracy and effectiveness of Copilot's code suggestions.
- To understand prompt-based programming using comments and code context

Lab Outcomes (LOs):
After completing this lab, students will be able to:

- Set up GitHub Copilot in VS Code successfully.
- Use inline comments and context to generate code with Copilot.
- Evaluate AI-generated code for correctness and readability.
- Compare code suggestions based on different prompts and programming styles.





Task Description#1

- Write a comment: # Function to check if a string is a valid palindrome (ignoring spaces and case) and allow Copilot to complete it.

Expected Output#1

- A function that correctly returns True for phrases like "A man a plan a canal Panama"

```python
# function to check if a sring is a palindraome
def is_palindrome(s: str) -> bool:
    cleaned = ''.join(c.lower() for c in s if c.isalnum())
    return cleaned == cleaned[::-1]

    # Example usage
if __name__ == "__main__":
    test_str = "A man a plan a canal Panama"
    print(is_palindrome(test_str))  # Output: True

```

```python
# alternative implementation
def is_palindrome_alt(s: str) -> bool:
    left, right = 0, len(s) - 1
    while left < right:
        while left < right and not s[left].isalnum():
            left += 1
        while left < right and not s[right].isalnum():
            right -= 1
        if s[left].lower() != s[right].lower():
            return False
        left += 1
        right -= 1
    return True

    # Example usage
test_str2 = "A man a plan a canal Panama"
print(is_palindrome_alt(test_str2))  # Output: True
```

Output :

```
PS C:\Users\Ansika\OneDrive\Desktop\AI assisted> & C:/Users/Ansika/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Ansika/OneDrive/Desktop/AI assisted/lab 2/palindrome_checker.py"
True
```

Task Description#2

- Generate a Python function that returns the Fibonacci sequence up to n terms. Prompt with only a function header and docstring

Expected Output#2

- AI completes the function logic using loop or recursion with accurate output

```python
#fibonacci sequance up to n terms
def fibonacci_sequence(n):
    sequence = []
    a, b = 0, 1
    for _ in range(n):
        sequence.append(a)
        a, b = b, a + b
    return sequence

    # Example usage:
if __name__ == "__main__":
    n = int(input("Enter number of terms: "))
    print("loops: ",fibonacci_sequence(n))

```

```python
# alternative implementation using recursion
def fibonacci_recursive(n, a=0, b=1, sequence=None):
    if sequence is None:
        sequence = []
    if n == 0:
        return sequence
    sequence.append(a)
    return fibonacci_recursive(n - 1, b, a + b, sequence)
n = int(input("Enter number of terms: "))
print("recursion: ",fibonacci_recursive(n))
```

Output:

```
Enter number of terms: 6
loops:  [0, 1, 1, 2, 3, 5]
```

Task Description#3

- Write a comment like # Function to reverse a string and use Copilot to generate the function.

Expected Output#3

- Auto-completed reverse function

```python
# Function to reverse a string
def reverse_string(s):
    """
    This function takes a string 's' as input and returns the reversed string.
    """
    # Use slicing to reverse the string
    return s[::-1]

# Example usage
if __name__ == "__main__":
    input_str = "Hello, World!"
    reversed_str = reverse_string(input_str)
    print("Original string:", input_str)
    print("Reversed string:", reversed_str)

```

```python
# Alternative implementation using recursion
def reverse_string_recursive(s):
    """
    This function takes a string 's' as input and returns the reversed string using recursion.
    """
    if len(s) == 0:
        return s
    else:
        return s[-1] + reverse_string_recursive(s[:-1])

# Example usage
if __name__ == "__main__":
    input_str = "Hello, World!"
    reversed_str = reverse_string(input_str)
    print("Original string:", input_str)
    print("Reversed string:", reversed_str)
```

Output:

```
Original string: Hello, World!
Reversed string: !dlroW ,olleH
```

Task Description#4

- Generate a program that simulates a basic calculator (add, subtract, multiply, divide). Write the comment: # Simple calculator with 4 operations and let AI complete it.

Expected Output#4

- Fully working calculator with input/output and operator selection logic

```python
def add(x, y):
    return x + y

def subtract(x, y):
    return x - y

def multiply(x, y):
    return x * y

def divide(x, y):
    if y == 0:
        return "Error: Division by zero"
    return x / y

def main():
    print("Simple Calculator")
    print("Select operation:")
    print("1. Add")
    print("2. Subtract")
    print("3. Multiply")
    print("4. Divide")

    choice = input("Enter choice (1/2/3/4): ")

    if choice in ('1', '2', '3', '4'):
        try:
            num1 = float(input("Enter first number: "))
            num2 = float(input("Enter second number: "))
        except ValueError:
            print("Invalid input. Please enter numbers.")
            return

        if choice == '1':
            print("Result:", add(num1, num2))
        elif choice == '2':
            print("Result:", subtract(num1, num2))
        elif choice == '3':
            print("Result:", multiply(num1, num2))
        elif choice == '4':
            print("Result:", divide(num1, num2))
    else:
        print("Invalid choice.")

if __name__ == "__main__":
    main()
```

Output:

```
Simple Calculator
Select operation:
1. Add
2. Subtract
3. Multiply
4. Divide
Enter choice (1/2/3/4): 3
Enter first number: 12
Enter second number: 3
Result: 36.0
```

Task Description#5

- Use a comment to instruct AI to write a function that reads a file and returns the number of lines..

Expected Output#5

- Functional implementation using open() or with open() and readlines()

```python
#write a code use function that reads a file and returns the number of the lines in the file
def read_file(file_path):
    try:
        with open('Why story matters.txt', 'r') as file:
            lines = file.readlines()
            return len(lines)
    except FileNotFoundError:
        print(f"The file at {file_path} was not found.")
        return 0
    except Exception as e:
        print(f"An error occurred: {e}")
        return 0
print(read_file('Why story matters.txt'))  # Example usage, replace 'who.txt' with your file path
```

Output:

```
PS C:\Users\Ansika\OneDrive\Desktop\AI assisted> & C:/Users/Ansika/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Ansika/OneDrive/Desktop/AI assisted/lab 2/file reader.py"
8
```

Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots

Evaluation Criteria:

Criteria	Max Marks
Task #1	0.5
Task #2	0.5
Task #3	0.5
Task #4	0.5
Task #5	0.5
Total	**2.5 Marks**