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:

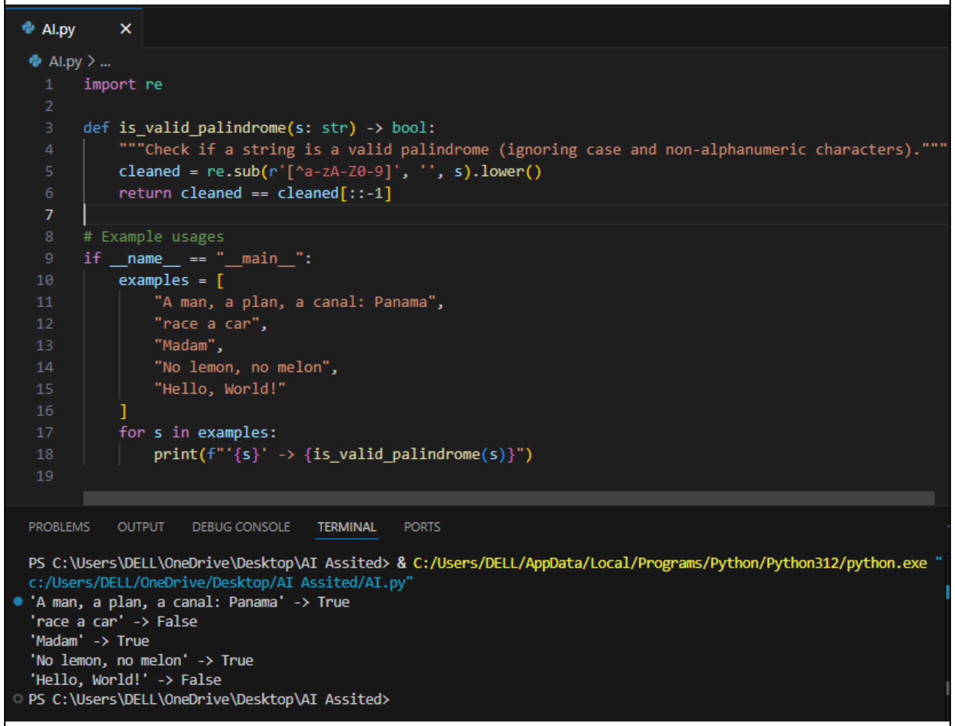
- Set up GitHub Copilot in VS Code successfully.
- Use inline comments and context to generate code with Copilot.
- Evaluate AI-generated code for correctness and readability.
- Compare code suggestions based on different prompts and programming styles.

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.
- **Prompt : write a python code for a function to check if a string is a valid palindrome**

Expected Output#1

- "



```python
import re

def is_valid_palindrome(s: str) -> bool:
    """Check if a string is a valid palindrome (ignoring case and non-alphanumeric characters)."""
    cleaned = re.sub(r'[^a-zA-Z0-9]', '', s).lower()
    return cleaned == cleaned[::-1]

# Example usages
if __name__ == "__main__":
    examples = [
        "A man, a plan, a canal: Panama",
        "race a car",
        "Madam",
        "No lemon, no melon",
        "Hello, World!"
    ]
    for s in examples:
        print(f"'{s}' -> {is_valid_palindrome(s)}")
```

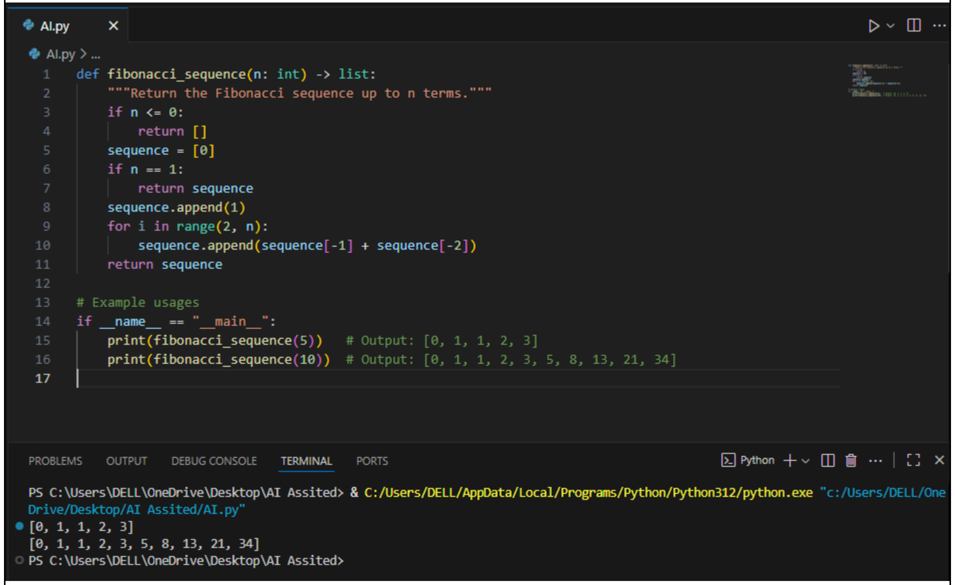
```
PS C:\Users\DELL\OneDrive\Desktop\AI Assited> & C:/Users/DELL/AppData/Local/Programs/Python/Python312/python.exe "
c:/Users/DELL/OneDrive/Desktop/AI Assited/AI.py"
'A man, a plan, a canal: Panama' -> True
'race a car' -> False
'Madam' -> True
'No lemon, no melon' -> True
'Hello, World!' -> False
PS C:\Users\DELL\OneDrive\Desktop\AI Assited>
```

Task Description#2

- Generate a Python function that returns the Fibonacci sequence up to n terms. Prompt with only a function header and docstring
- **Prompt : write a python code for a function that returns the Fibonacci sequence up to n terms**

Expected Output#2

●



```
def fibonacci_sequence(n: int) -> list:
    """Return the Fibonacci sequence up to n terms."""
    if n <= 0:
        return []
    sequence = [0]
    if n == 1:
        return sequence
    sequence.append(1)
    for i in range(2, n):
        sequence.append(sequence[-1] + sequence[-2])
    return sequence

# Example usages
if __name__ == "__main__":
    print(fibonacci_sequence(5))   # Output: [0, 1, 1, 2, 3]
    print(fibonacci_sequence(10))  # Output: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

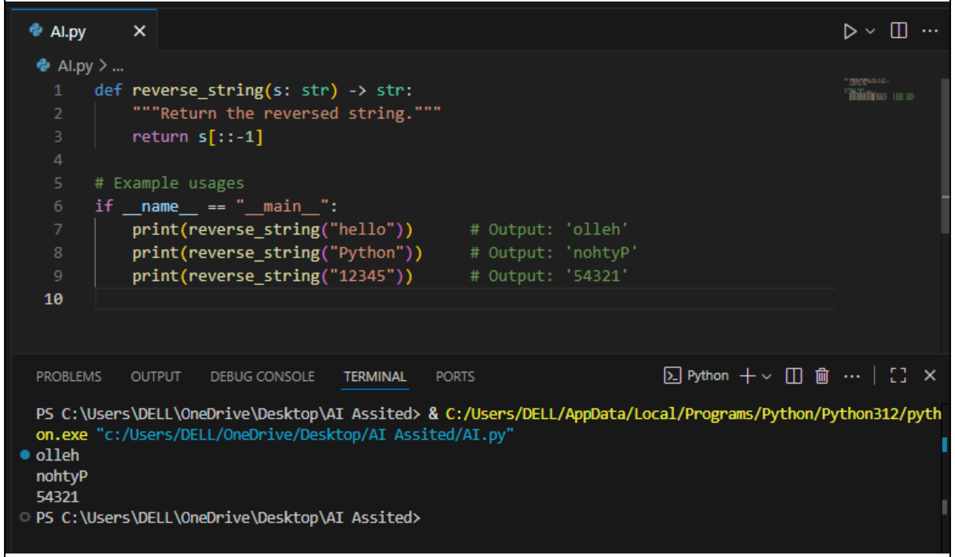
```
PS C:\Users\DELL\OneDrive\Desktop\AI Assited> & C:/Users/DELL/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/DELL/OneDrive/Desktop/AI Assited/AI.py"
[0, 1, 1, 2, 3]
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
PS C:\Users\DELL\OneDrive\Desktop\AI Assited>
```

Task Description#3

- Write a comment like # Function to reverse a string and use Copilot to generate the function.
- **Prompt : write a python code for Function to reverse a string**

Expected Output#3

●



```
def reverse_string(s: str) -> str:
    """Return the reversed string."""
    return s[::-1]

# Example usages
if __name__ == "__main__":
    print(reverse_string("hello"))      # Output: 'olleh'
    print(reverse_string("Python"))     # Output: 'nohtyP'
    print(reverse_string("12345"))      # Output: '54321'
```

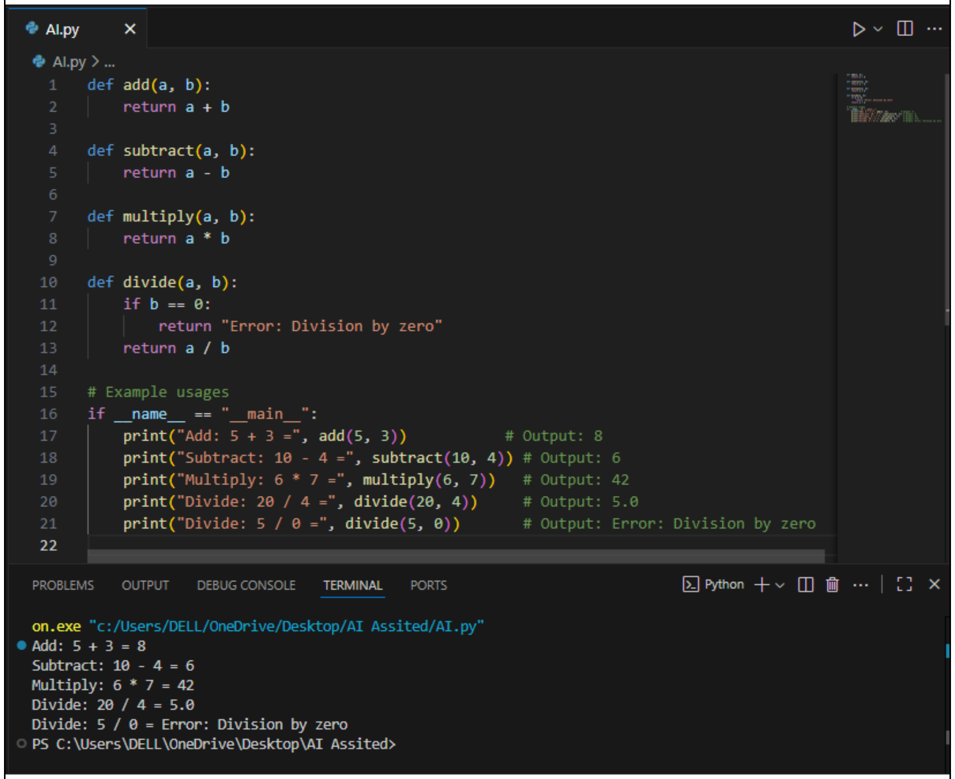
```
PS C:\Users\DELL\OneDrive\Desktop\AI Assited> & C:/Users/DELL/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/DELL/OneDrive/Desktop/AI Assited/AI.py"
olleh
nohtyP
54321
PS C:\Users\DELL\OneDrive\Desktop\AI Assited>
```

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.
- **Prompt : python code only for a program that simulates a basic calculator (add, subtract, multiply, divide)**

Expected Output#4

●



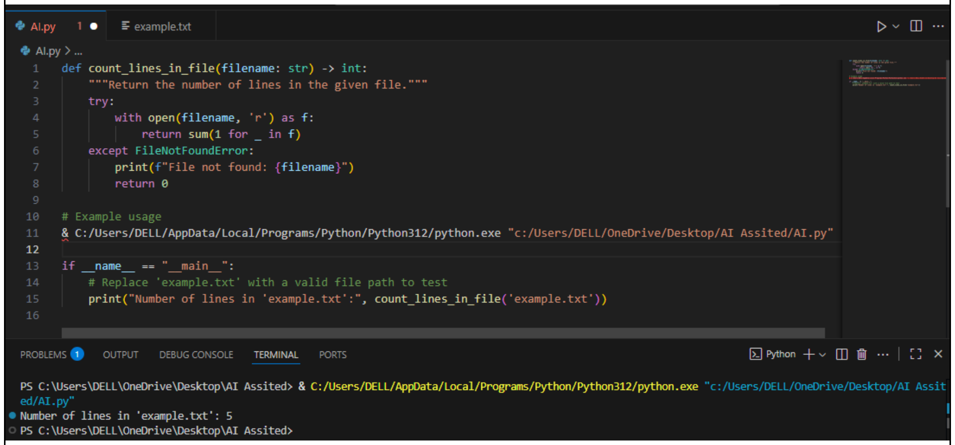
```python
def add(a, b):
    return a + b

def subtract(a, b):
    return a - b

def multiply(a, b):
    return a * b

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

# Example usages
if __name__ == "__main__":
    print("Add: 5 + 3 =", add(5, 3))          # Output: 8
    print("Subtract: 10 - 4 =", subtract(10, 4)) # Output: 6
    print("Multiply: 6 * 7 =", multiply(6, 7))   # Output: 42
    print("Divide: 20 / 4 =", divide(20, 4))     # Output: 5.0
    print("Divide: 5 / 0 =", divide(5, 0))       # Output: Error: Division by zero
```

```
on.exe "c:/Users/DELL/OneDrive/Desktop/AI Assited/AI.py"
Add: 5 + 3 = 8
Subtract: 10 - 4 = 6
Multiply: 6 * 7 = 42
Divide: 20 / 4 = 5.0
Divide: 5 / 0 = Error: Division by zero
PS C:\Users\DELL\OneDrive\Desktop\AI Assited>
```

Task Description#5

- Use a comment to instruct AI to write a function that reads a file and returns the number of lines..
- Prompt : python code only for a function that reads a file and returns the number of lines..

Expected Output#5



```python
def count_lines_in_file(filename: str) -> int:
    """Return the number of lines in the given file."""
    try:
        with open(filename, 'r') as f:
            return sum(1 for _ in f)
    except FileNotFoundError:
        print(f"File not found: {filename}")
        return 0

# Example usage
& C:/Users/DELL/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/DELL/OneDrive/Desktop/AI Assited/AI.py"

if __name__ == "__main__":
    # Replace 'example.txt' with a valid file path to test
    print("Number of lines in 'example.txt':", count_lines_in_file('example.txt'))
```

```
PS C:\Users\DELL\OneDrive\Desktop\AI Assited> & C:/Users/DELL/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/DELL/OneDrive/Desktop/AI Assit
ed/AI.py"
Number of lines in 'example.txt': 5
PS C:\Users\DELL\OneDrive\Desktop\AI Assited>
```

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**		