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.	

- 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.

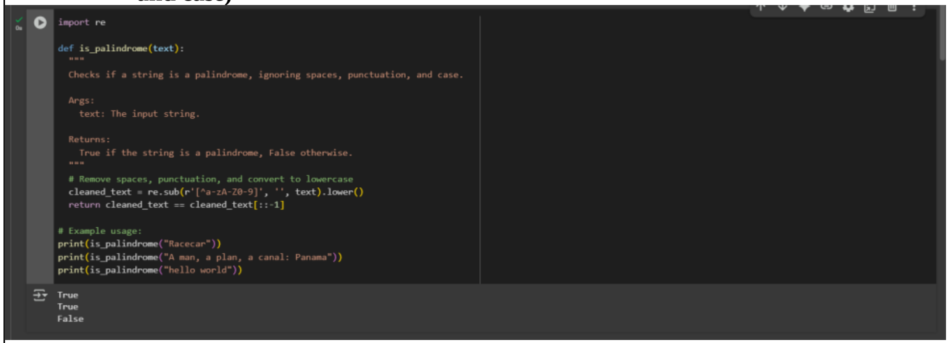
Task Description#1

- 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"

Prompt : Function to check if a string is a valid palindrome (ignoring spaces,punctuation, and case)



```
import re

def is_palindrome(text):
  """
  Checks if a string is a palindrome, ignoring spaces, punctuation, and case.

  Args:
    text: The input string.

  Returns:
    True if the string is a palindrome, False otherwise.
  """
  # Remove spaces, punctuation, and convert to lowercase
  cleaned_text = re.sub(r'[^a-zA-Z0-9]', '', text).lower()
  return cleaned_text == cleaned_text[::-1]

# Example usage:
print(is_palindrome("Racecar"))
print(is_palindrome("A man, a plan, a canal: Panama"))
print(is_palindrome("hello world"))
```

```
True
True
False
```

OBSERVATIONS:

1. **Co-Pilot will auto generate a function named is_valid_palindrome**
2. **It will remove spaces from a string and convert it to lower cases**
3. **It wil compare the string in reverse to check if it's a palindrome or no**

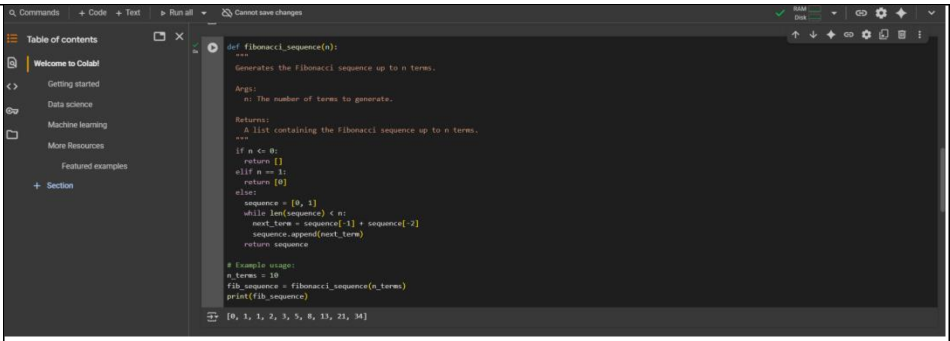
Task Description#2

- 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

PROMPT:Write a python function that returns the fibonacci sequence up to n terms



```python
def fibonacci_sequence(n):
  """
  Generates the Fibonacci sequence up to n terms.

  Args:
    n: The number of terms to generate.

  Returns:
    A list containing the Fibonacci sequence up to n terms.
  """
  if n <= 0:
    return []
  elif n == 1:
    return [0]
  else:
    sequence = [0, 1]
    while len(sequence) < n:
      next_term = sequence[-1] + sequence[-2]
      sequence.append(next_term)
    return sequence

# Example usage:
n_terms = 10
fib_sequence = fibonacci_sequence(n_terms)
print(fib_sequence)
```

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

OBSERVATIONS:

1) **It implements the Fibonacci sequence iteratively starting with a=0 and b=1**
2) **It might also add a docstring**

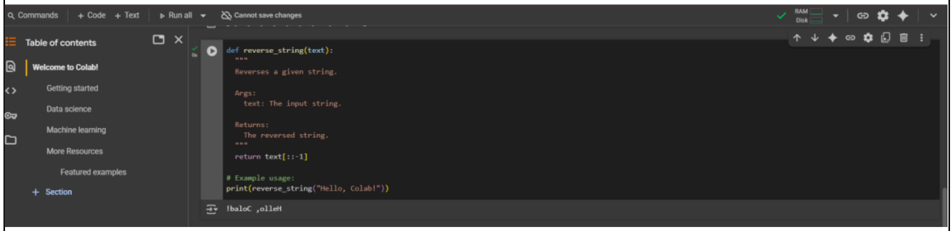
Task Description#3

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

Expected Output#3

- Auto-completed reverse function

PROMPT:Write a python function to reverse a string



```python
def reverse_string(text):
  """
  Reverses a given string.

  Args:
    text: The input string.

  Returns:
    The reversed string.
  """
  return text[::-1]

# Example usage:
print(reverse_string("Hello, Colab!"))
```

```
!baloC ,olleH
```

OBSERVATIONS:

1) Writes a very short, direct solution because string reversal in Python is extremely common and has a one-liner approach
2) The copilot stores the reverse srting in variable
3) It reades the input from string() so the program feels complete and interactive

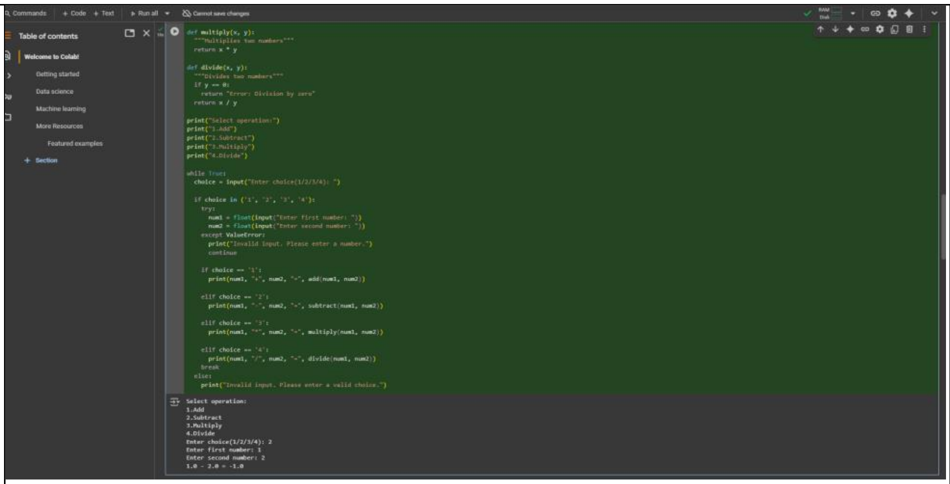
Task Description#4

- 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

PROMPT:Write a python program to stimulate a basic calculator which could Add, subtract, multiply, divide



```python
def multiply(x, y):
  """Multiplies two numbers"""
  return x * y

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

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

while True:
  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 a number.")
      continue

    if choice == '1':
      print(num1, "+", num2, "=", add(num1, num2))

    elif choice == '2':
      print(num1, "-", num2, "=", subtract(num1, num2))

    elif choice == '3':
      print(num1, "*", num2, "=", multiply(num1, num2))

    elif choice == '4':
      print(num1, "/", num2, "=", divide(num1, num2))
    break
  else:
    print("Invalid input. Please enter a valid choice.")
```

```
Select operation:
1.Add
2.Subtract
3.Multiply
4.Divide
Enter choice(1/2/3/4): 2
Enter first number: 1
Enter second number: 2
1.0 - 2.0 = -1.0
```

OBSERVATIONS:

1) **It makes a menu for ADD, Subtract, Multiply, Divide**
2) **Takes user input for choice and two numbers**
3) **Uses if/elif/else todecide the operation**

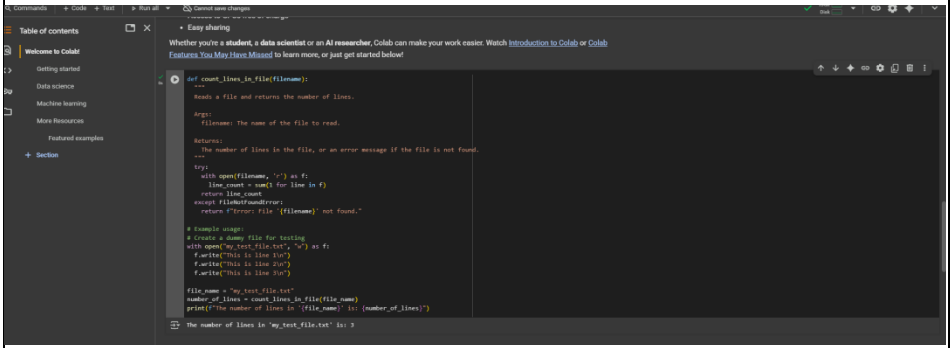
Task Description#5

- 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()

PROMPT:Create a python function that reads a file and returns the number of lines



- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch Introduction to Colab or Colab Features You May Have Missed to learn more, or just get started below!

```python
def count_lines_in_file(filename):
  """
  Reads a file and returns the number of lines.

  Args:
    filename: The name of the file to read.

  Returns:
    The number of lines in the file, or an error message if the file is not found.
  """
  try:
    with open(filename, 'r') as f:
      line_count = sum(1 for line in f)
    return line_count
  except FileNotFoundError:
    return f"Error: File '{filename}' not found."

# Example usage:
# Create a dummy file for testing
with open("my_test_file.txt", "w") as f:
  f.write("This is line 1\n")
  f.write("This is line 2\n")
  f.write("This is line 3\n")

file_name = "my_test_file.txt"
number_of_lines = count_lines_in_file(file_name)
print(f"The number of lines in '{file_name}' is: {number_of_lines}")
```

```
The number of lines in 'my_test_file.txt' is: 3
```

OBSERVATIONS: The function is designed to **read a file** and **return the number of lines** it contains.The logic is wrapped in a function, making it reusable and testable.

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**		