LAB ASSIGNMENT:1.3

NAME: S.NADHIYA

ROLL NO: 2403A510C6

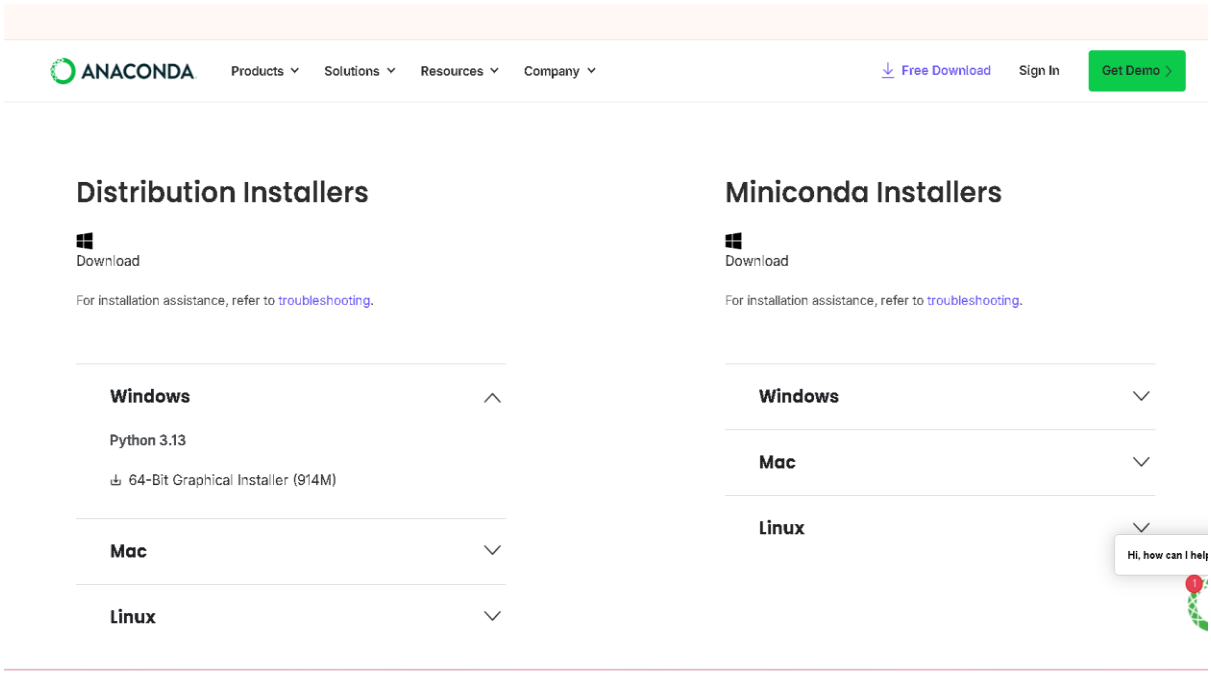
BATCH: 05

BRANCH: CSE

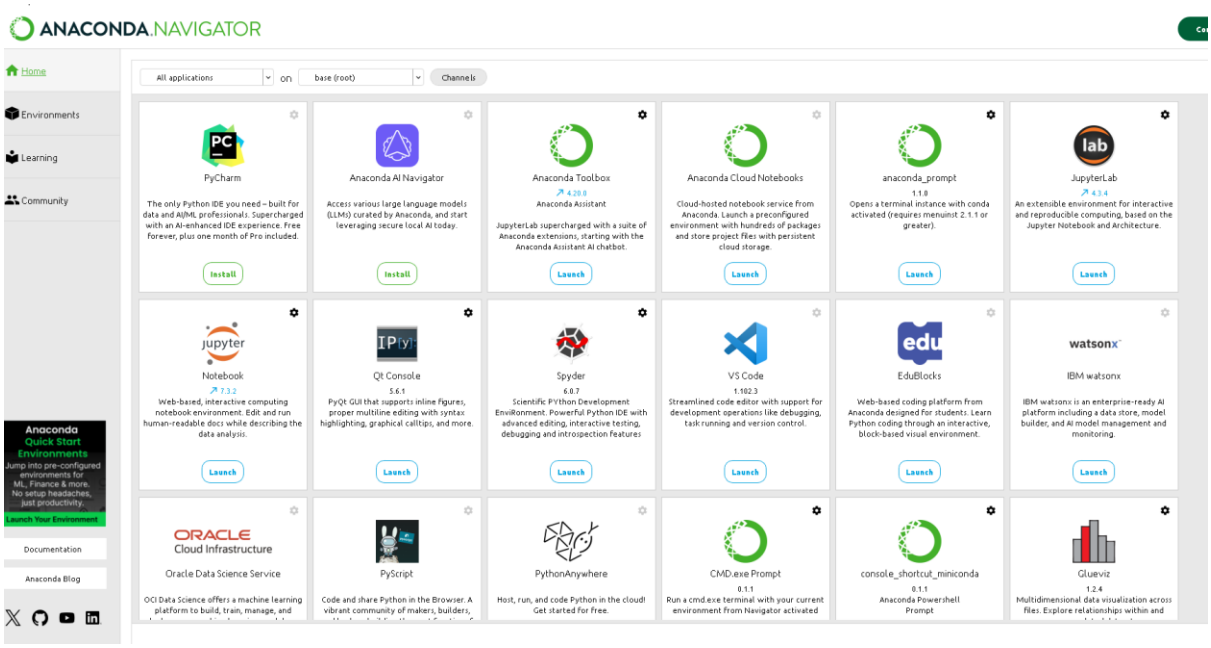
SUB: AI ASSISTED CODING

TASK - 01

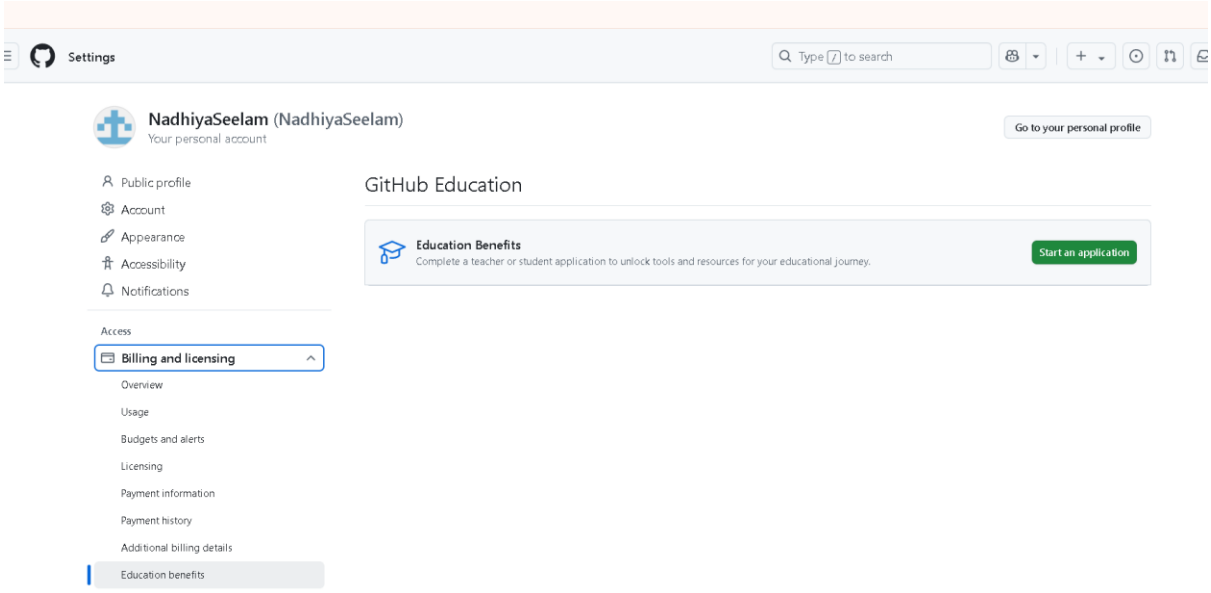
Screen shot-1



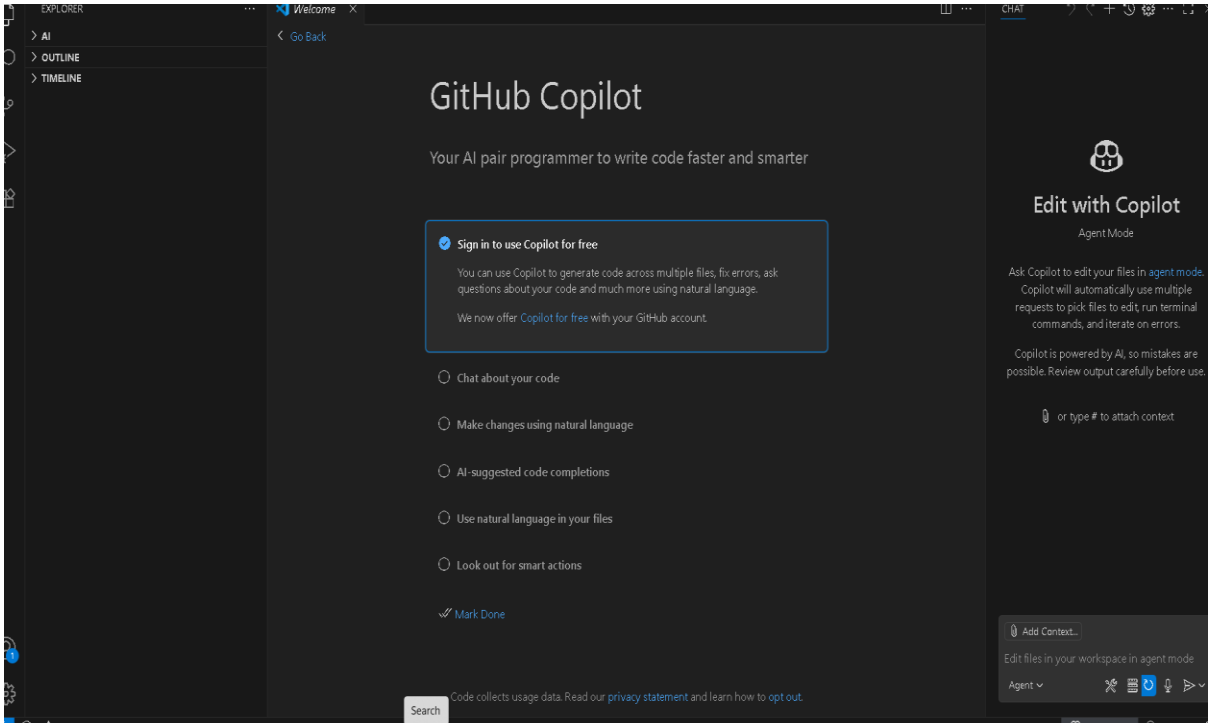
Screen shot-2



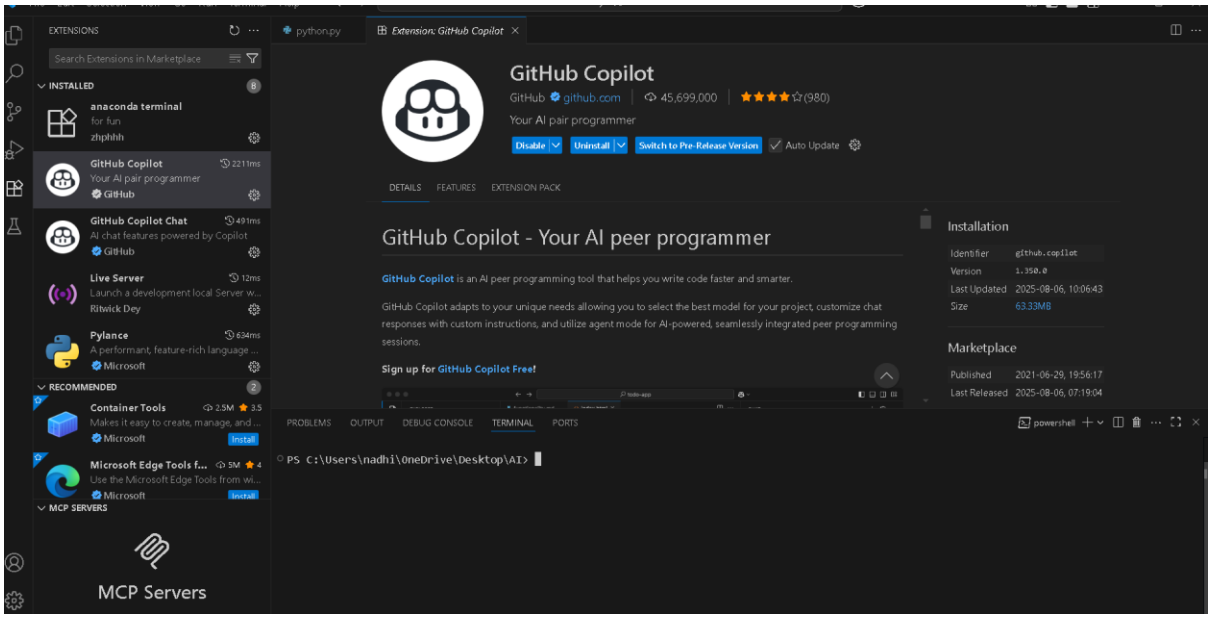
Screen shot-3



Screen shot-4

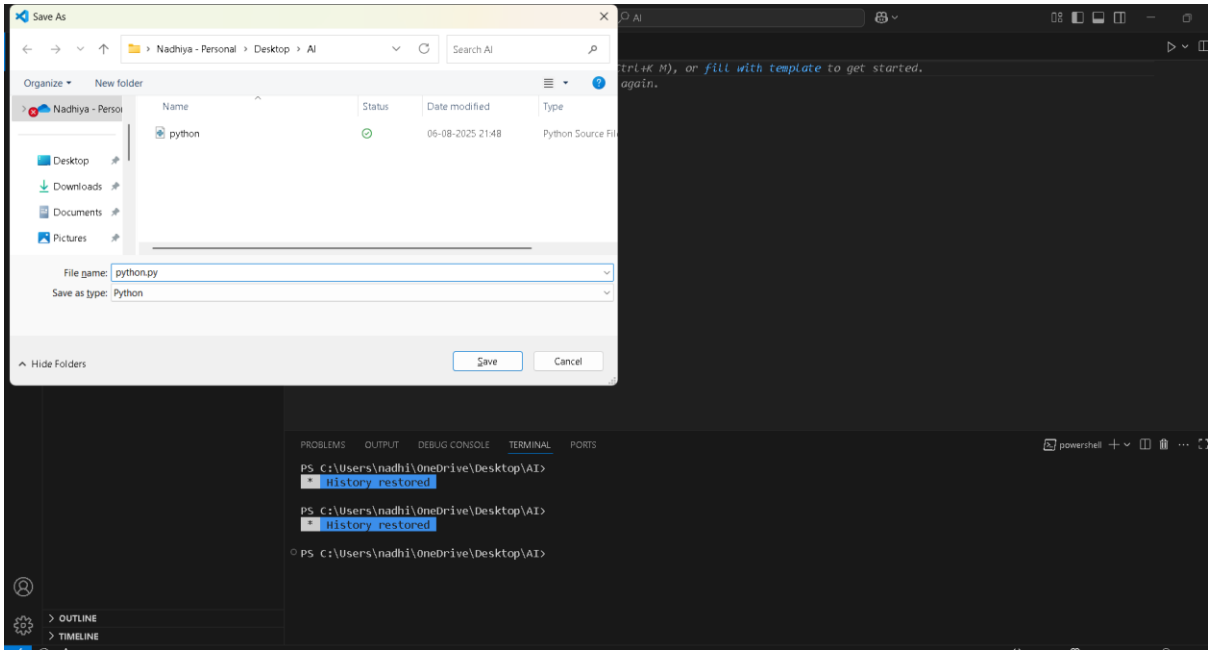


Screen shot-5

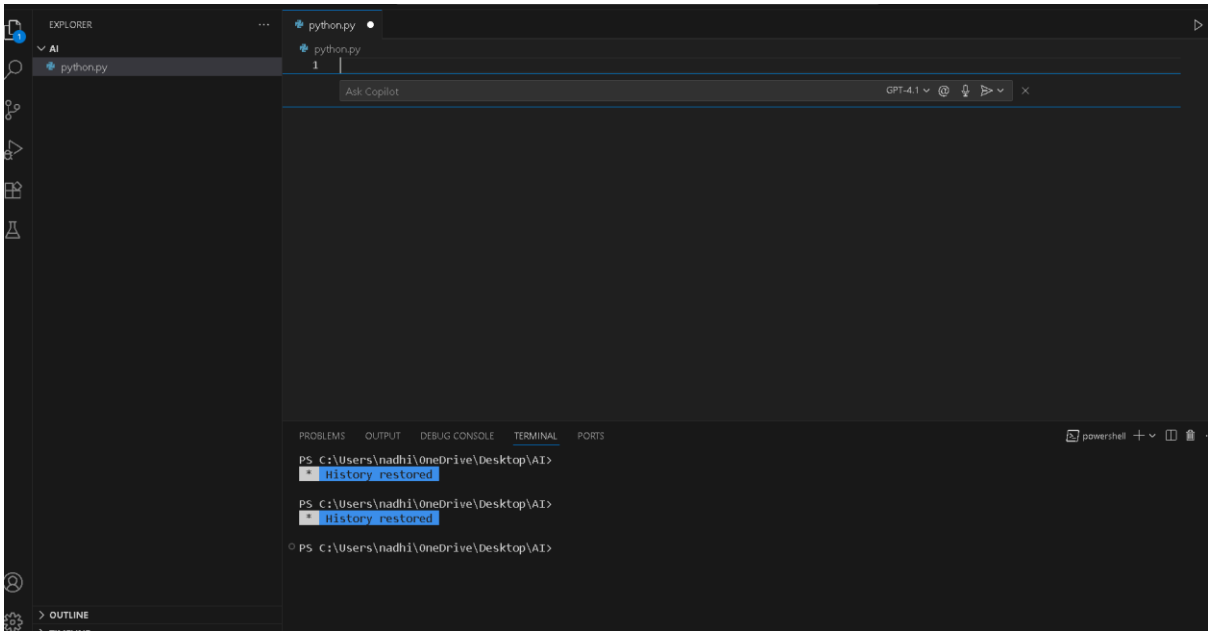


Screen shot-6

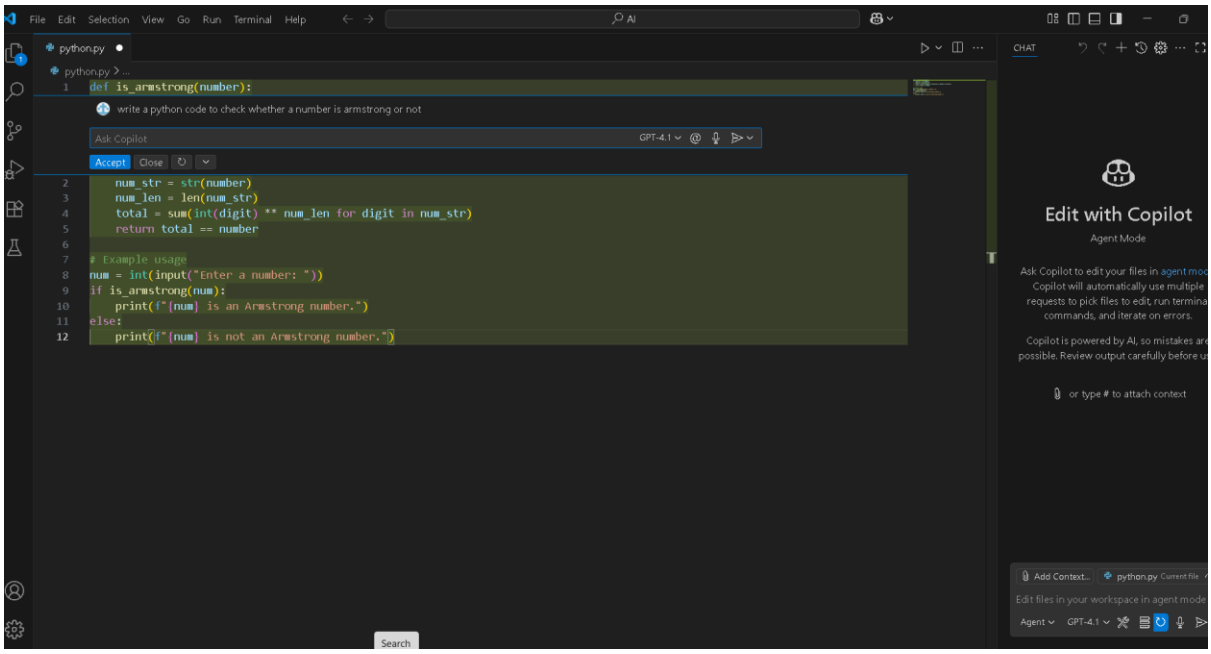
Screen shot-7



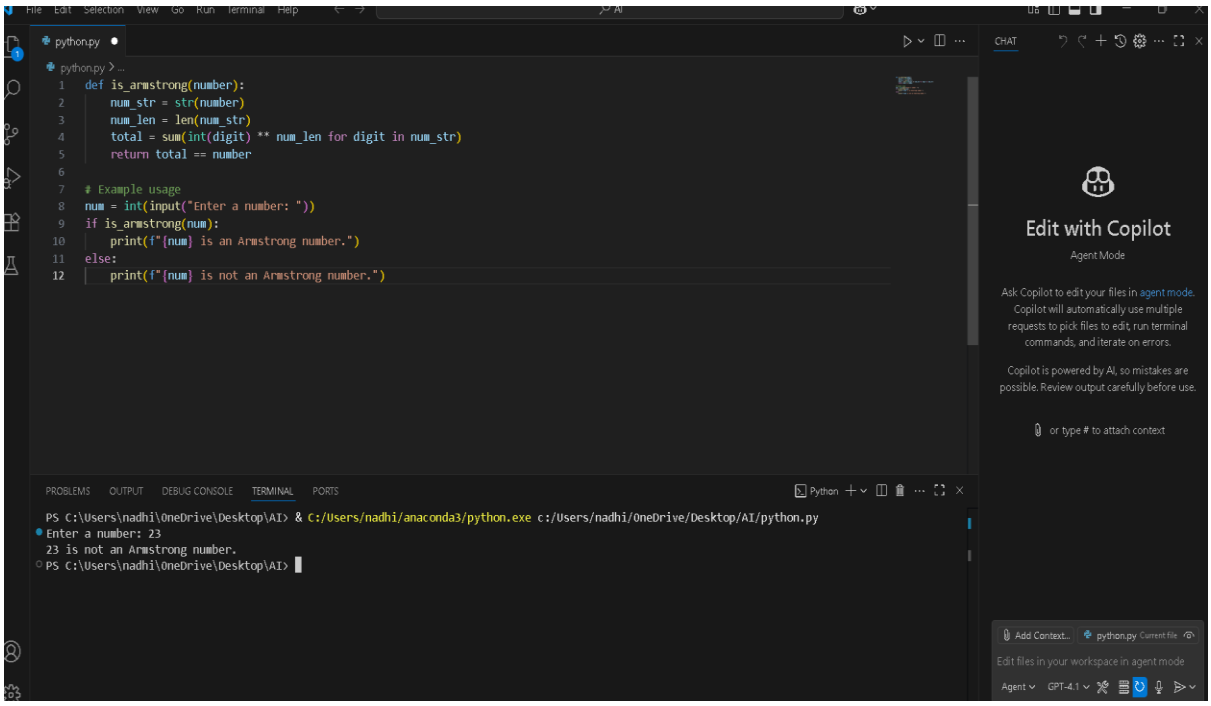
Screen shot-8



Screen shot-9

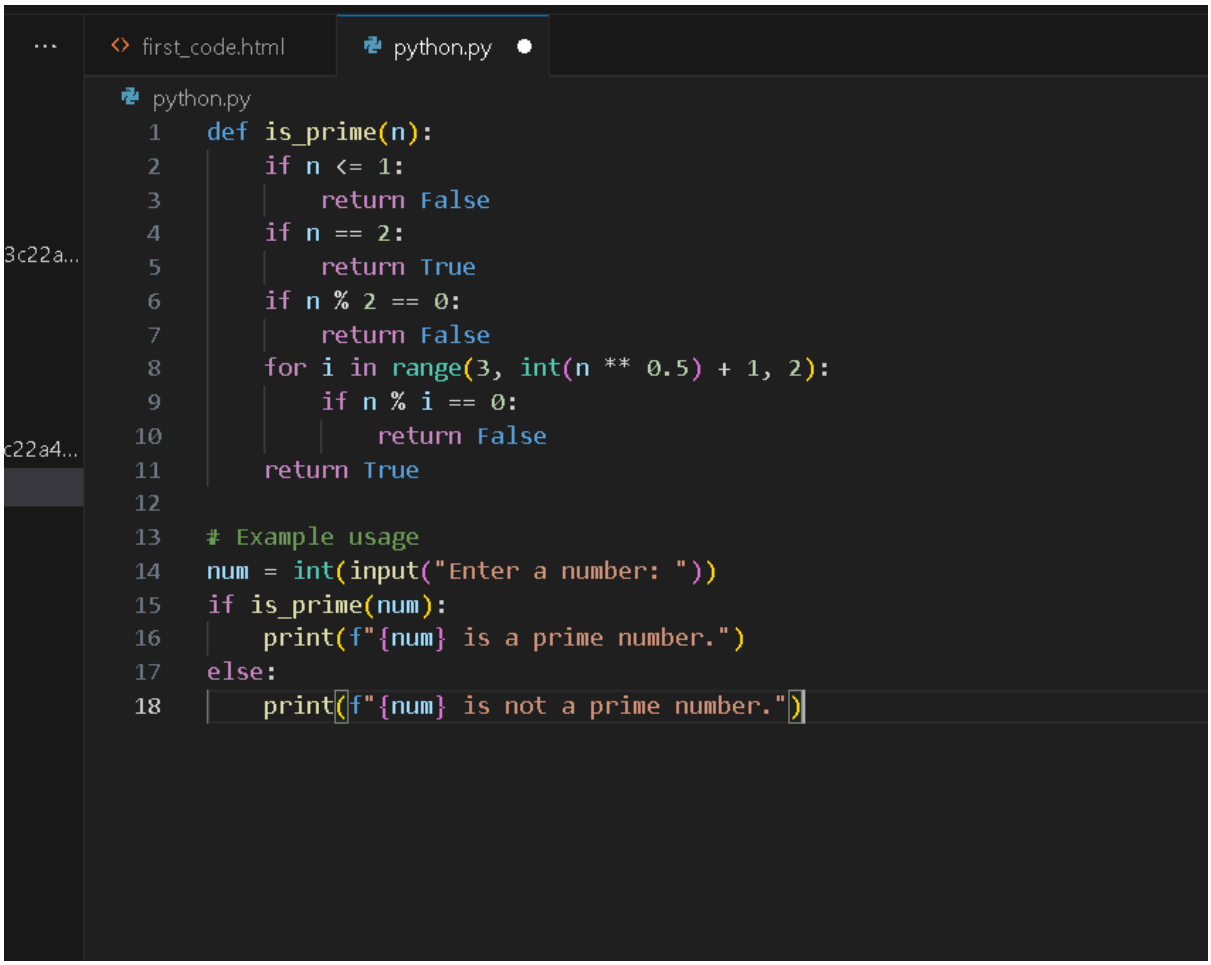


Screen shot-9



TASK – 02

Prompt : Write a python code to check whether a number is prime or not.

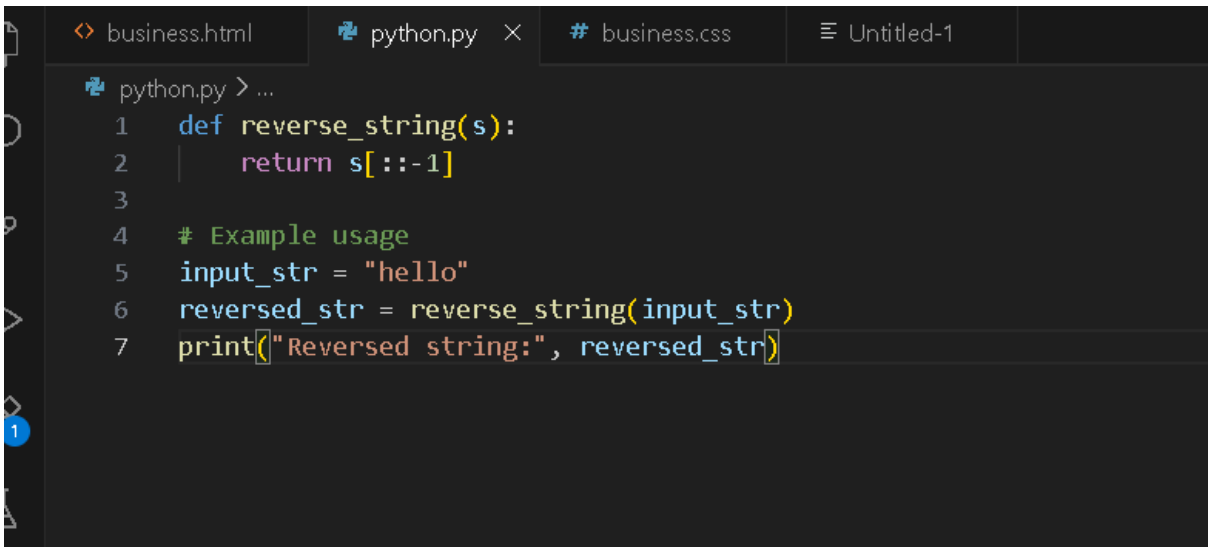


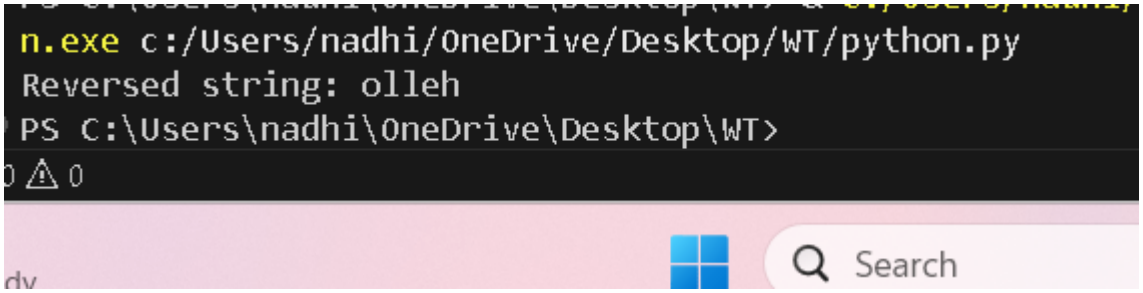
Output: 2 is prime number

Explanation:

- he code checks if a given number is prime.
- A prime number is a number greater than 1 that has no divisors other than 1 and itself.
- The code usually:
  - Returns False if the number is less than or equal to 1.
  - Loops from 2 up to the square root of the number.
  - If the number is divisible by any of these, it returns False.
  - If no divisors are found, it returns True.

Task-3:





Explanation:

- The function reverse_string(s) takes a string s as input and returns its reverse using slicing (s[::-1]).
- The example usage sets input_str to "hello".
- It calls reverse_string(input_str), which returns "olleh", and stores it in reversed_str.
- Finally, it prints Reversed string: olleh to the console

Task-04:

```python
# Recursive version of factorial
def factorial_recursive(n):
    """
    Calculate factorial of n recursively.
    """
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial_recursive(n - 1)

# Iterative version of factorial
def factorial_iterative(n):
    """
    Calculate factorial of n iteratively.
    """
    result = 1
    for i in range(2, n + 1):
        result *= i
    return result

# Example usage
if __name__ == "__main__":
    num = 5
    print("Recursive:", factorial_recursive(num))  # Output: 120
    print("Iterative:", factorial_iterative(num))  # Output: 120
```

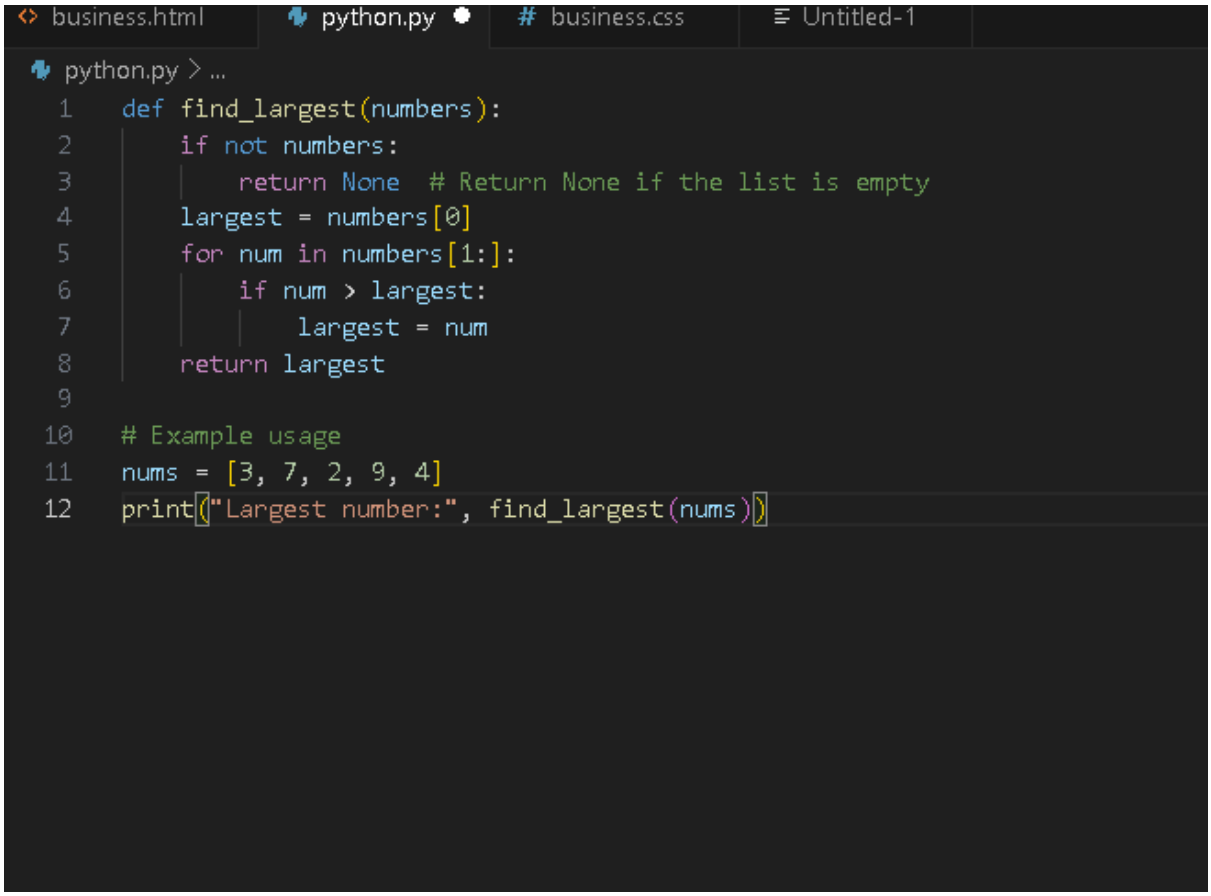
```
PS C:\Users\nadhi\OneDrive\Desktop\WT> & C:/Users/nadhi/AppData/Local/Programs/Python/Python313/python.exe c:/Users/nadhi/OneDrive/Desktop
/WT/python.py
Recursive: 120
Iterative: 120
```

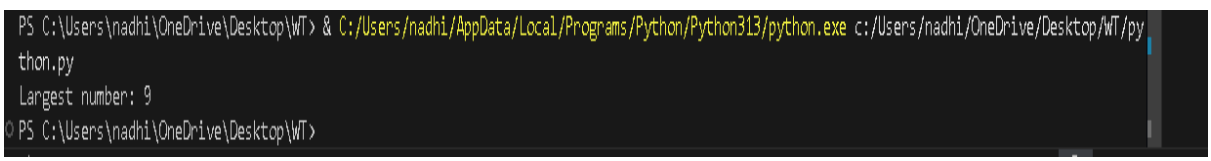
Explanation:

- This function calculates the factorial of n using recursion.
    - If n is 0 or 1, it returns 1 (base case).
    - Otherwise, it returns n * factorial_recursive(n - 1).

- **factorial_iterative(n):**
  This function calculates the factorial of n using a loop.
  - It initializes result to 1.
  - Then multiplies result by each number from 2 up to n.
- **Example usage:**
  - If the script is run directly, it sets num = 5.
  - It prints the factorial of 5 using both the recursive and iterative functions.
  - Both methods output 120.

Task-05:





Explanation:

- **find_largest(numbers):**
  This function takes a list of numbers and returns the largest value.
    - If the list is empty, it returns None.
    - It starts by assuming the first number is the largest.
    - It then loops through the rest of the list, updating largest if it finds a bigger number.
    - Finally, it returns the largest number found.
- **Example usage:**
    - A list nums = [3, 7, 2, 9, 4] is defined.
    - The function is called with this list, and the result is printed.
    - Output: Largest number: 9