AI ASSISTED LAB-10.3

NAME:B.VISHNU VARDHAN

ROLL NO:2403A510F2

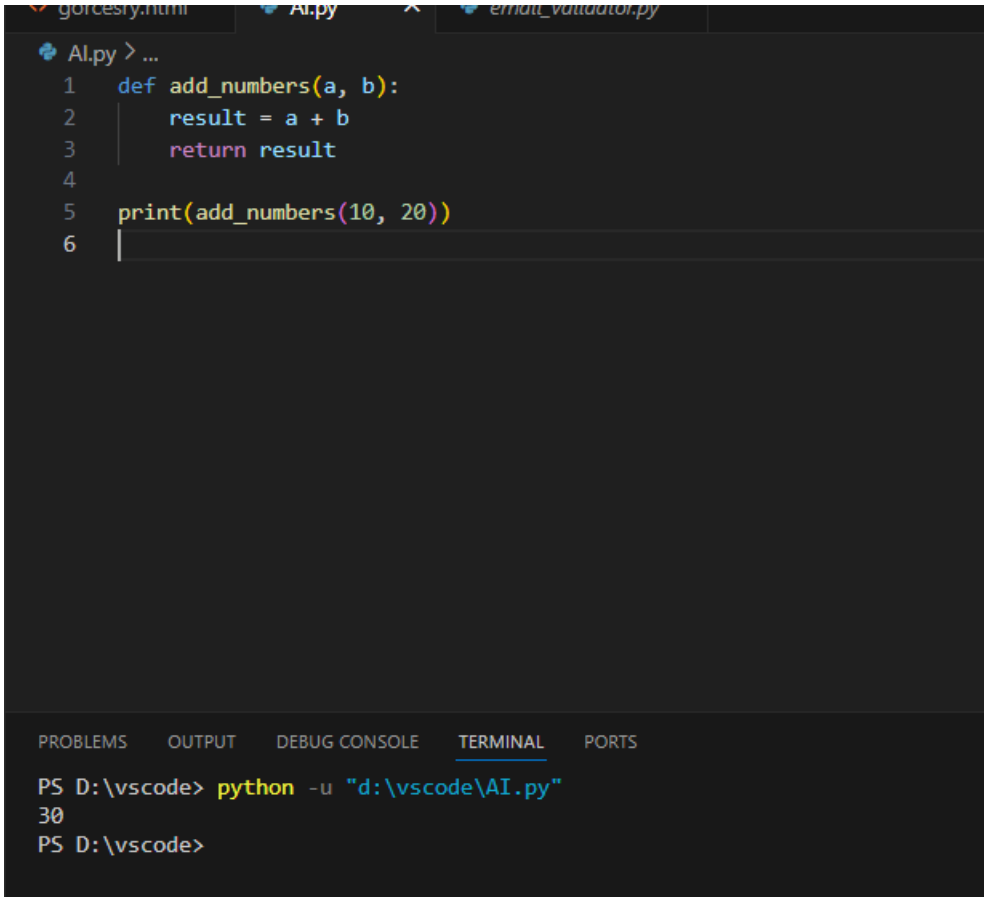
BATCH:06

TASK 1: SYNTAX AND ERROR DETECTION

PROMPT:

Identify and fix syntax, indentation, and variable errors in the given script.

CODE:



OUTPUT:

30

OBSERVATION:

Fixed missing colon after function, corrected variable name from 'reslt' to 'result', and added a comma between arguments.

AI ASSISTED LAB-10.3

TASK 2: LOGICAL AND PERFORMANCE ISSUE REVIEW

PROMPT:

Optimize inefficient logic while keeping the result correct.

CODE:

gorcesry.html | AI.py | email_validator.py

AI.py > find_duplicates

```python
def find_duplicates(nums):
    seen = set()
    duplicates = set()
    for num in nums:
        if num in seen:
            duplicates.add(num)
        else:
            seen.add(num)
    return list(duplicates)

numbers = [1, 2, 3, 2, 4, 5, 1, 6, 1, 2]
print(find_duplicates(numbers))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\vscode> python -u "d:\vscode\AI.py"
[1, 2]
PS D:\vscode>
```

OUTPUT:

[1, 2]

OBSERVATION:

Replaced nested loops with set-based logic for O(n) efficiency.

TASK 3: CODE REFACTORING FOR READABILITY

PROMPT:

Refactor messy code into clean, PEP 8–compliant, well-structured code.

CODE:

AI ASSISTED LAB-10.3

gorcesry.html | AI.py | email_validator.py

AI.py > ...

```python
def calculate_factorial(n):
    """Calculate the factorial of a given non-negative integer n."""
    factorial = 1
    for i in range(1, n + 1):
        factorial *= i
    return factorial

print(calculate_factorial(5))
```

OUTPUT:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\vscode> python -u "d:\vscode\AI.py"
120
PS D:\vscode>
```

OBSERVATION:

Renamed function and variables, added docstring, and improved readability and formatting.

TASK 4: SECURITY AND ERROR HANDLING ENHANCEMENT

PROMPT:

Add security practices and exception handling to the code.

AI ASSISTED LAB-10.3

CODE:

gorcesry.html | AI.py | email_validator.py

AI.py > get_user_data

```python
import sqlite3

def get_user_data(user_id):
    try:
        conn = sqlite3.connect("users.db")
        cursor = conn.cursor()
        query = "SELECT * FROM users WHERE id = ?"
        cursor.execute(query, (user_id,))
        result = cursor.fetchall()
    except sqlite3.Error as e:
        print(f"Database error: {e}")
        result = []
    finally:
        conn.close()
    return result

user_input = input("Enter user ID (numeric only): ")
if user_input.isdigit():
    print(get_user_data(int(user_input)))
else:
    print("Invalid input. Please enter a numeric user ID.")
```

OUTPUT:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\vscode> python -u "d:\vscode\AI.py"
Enter user ID (numeric only): sai
Invalid input. Please enter a numeric user ID.
PS D:\vscode>
```

OBSERVATION:

Used parameterized query to prevent SQL injection, added try-except for database errors, and input validation.

TASK 5: AUTOMATED CODE REVIEW REPORT GENERATION

PROMPT:

Generate a review report and fix messy code.

CODE:

AI ASSISTED LAB-10.3

gorcesry.html | AI.py | email_validator.py

AI.py > ...

```python
def calculate(x, y, operation):
    """Perform an arithmetic operation on two numbers."""
    try:
        if operation == "add":
            return x + y
        elif operation == "sub":
            return x - y
        elif operation == "mul":
            return x * y
        elif operation == "div":
            return x / y
        else:
            return "Invalid operation"
    except ZeroDivisionError:
        return "Error: Division by zero"

print(calculate(10, 5, "add"))
print(calculate(10, 0, "div"))
```

OUTPUT:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\vscode> python -u "d:\vscode\AI.py"
15
Error: Division by zero
PS D:\vscode>
```

OBSERVATION:

Added docstring, proper indentation, descriptive function name, and handled division by zero error.