AI ASSISTED CODING

LAB-10.4

NAME:K.Manikanta

ENROLL.NO:2403A52040

BATCH:03

TASK-01:

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

```
# buggy_code_task1.py
def add_numbers(a, b)
result = a + b
return reslt
print(add_numbers(10 20))
```

Optimize inefficient logic while keeping the result correct.

```
# buggy_code_task2.py
```

PROMPT:

TASK-02:

Optimize inefficient logic while keeping the result correct.

```
def find_duplicates(nums):
duplicates = []
for i in range(len(nums)):
for j in range(len(nums)):
if i != j and nums[i] == nums[j] and nums[i] not in duplicates:
duplicates.append(nums[i])
return duplicates
numbers = [1,2,3,2,4,5,1,6,1,2]
print(find_duplicates(numbers))
```

PROMPT:

TASK-03:

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

```
# buggy_code_task3.py

def c(n):
x=1
for i in range(1,n+1):
x=x*i
return x
print(c(5))
```

PROMPT:

TASK-04:

Add security practices and exception handling to the code.

```
# buggy_code_task4.py
import sqlite3
def get_user_data(user_id):
conn = sqlite3.connect("users.db")
cursor = conn.cursor()
query = f"SELECT * FROM users WHERE id = {user_id};" #
Potential SQL injection risk
cursor.execute(query)
result = cursor.fetchall()
conn.close()
return result
user_input = input("Enter user ID: ")
print(get_user_data(user_input))
```

PROMPT:

TASK-05:

Generate a review report for this messy code.

```
# buggy_code_task5.py

def calc(x,y,z):
if z=="add":
return x+y
elif z=="sub": return x-y
```

```
elif z=="mul":
return x*y
elif z=="div":
return x/y
else: print("wrong")
print(calc(10,5,"add"))
print(calc(10,0,"div"))
```

PROMPT: