

AI ASSISTED CODING

LAB EXAM-3

NAME : G.OMKAR

HT.no : 2403A52039

BATCH : 03

+

Q1: Scenario: In the domain of Education, a company is facing a challenge related to code refactoring.

Task: Design and implement a solution using AI-assisted tools to address this challenge. Include code, explanation of AI integration, and test results. Deliverables: Source code, explanation, and output screenshots.

Prompt:

“Design and implement an AI-assisted solution for code refactoring in the Education domain. The AI tool should analyze the existing code, suggest improvements, and refactor the code while keeping the same functionality. Include the Python code, explanation of how AI is integrated, and show test results or output screenshots.”

CODE :

```
.lab6.4.py X |||  
lab6.4.py > ...  
1 # Simple example: AI-assisted code refactoring demo  
2  
3 # Old (unoptimized) function  
4 def calculate_average(scores):  
5     total = 0  
6     for s in scores:  
7         total += s  
8     avg = total / len(scores)  
9     return avg  
10  
11 # Refactored version (AI suggested)  
12 def calculate_average_refactored(scores):  
13     """Return the average of scores using built-in functions."""  
14     return sum(scores) / len(scores)  
15  
16 # Testing both versions  
17 old_result = calculate_average([80, 90, 100])  
18 new_result = calculate_average_refactored([80, 90, 100])  
19  
20 print("Old Result:", old_result)  
21 print("Refactored Result:", new_result)  
22 print("AI Suggestion: Simplified using built-in sum() function.")  
23
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\surya\OneDrive\Desktop\AI assist> & C:/Users/surya/AppData/Local/Microsoft/Windows/OneDrive/Desktop/AI assist/lab6.4.py"  
Old Result: 90.0  
Refactored Result: 90.0  
AI Suggestion: Simplified using built-in sum() function.  
PS C:\Users\surya\OneDrive\Desktop\AI assist>
```

Observation :

The AI-assisted refactoring made the code shorter, easier to read, and more efficient.

Both the old and refactored versions give the same output, proving that the functionality is preserved while improving the code quality.

Q2:

Scenario: In the domain of Transportation, a company is facing a challenge related to backend api development.

Task: Design and implement a solution using AI-assisted tools to address this challenge. Include code, explanation of AI integration, and test results.

Deliverables: Source code, explanation, and output screenshots

Prompt :

In the domain of Transportation, a company is facing a challenge related to backend API development. Design and implement a simple AI-assisted solution that helps generate and manage backend API code for vehicle data (type, capacity, route). Include sample Python code showing CRUD operations, explain how AI assists in generating API templates, and show test results or output.

CODE :

```
lab6.4.py > TransportAPI
1  class TransportAPI:
2      def __init__(self):
3          self.vehicles = {}
4
5      def create_vehicle(self, vid, vehicle_type, capacity, route):
6          self.vehicles[vid] = {"type": vehicle_type, "capacity": capacity, "route": route}
7          return f"Vehicle {vid} created successfully."
8
9      def get_vehicle(self, vid):
10         return self.vehicles.get(vid, "Vehicle not found.")
11
12     def update_vehicle(self, vid, capacity=None, route=None):
13         if vid in self.vehicles:
14             if capacity:
15                 self.vehicles[vid]["capacity"] = capacity
16             if route:
17                 self.vehicles[vid]["route"] = route
18             return f"Vehicle {vid} updated."
19         return "Vehicle not found."
20
21     def delete_vehicle(self, vid):
22         if vid in self.vehicles:
23             del self.vehicles[vid]
24             return f"Vehicle {vid} deleted."
25         return "Vehicle not found."
26
27     # --- AI-assisted suggestion simulation ---
28     def ai_generate_api():
ROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
AI tool suggests CRUD functions for managing transport vehicles.
Vehicle 1 created successfully.
{'type': 'Bus', 'capacity': 50, 'route': 'A-B'}
Vehicle 1 updated.
Vehicle 1 deleted.
Vehicle not found.
PS C:\Users\surya\OneDrive\Desktop\AI assist>
```

Observation:

The AI-assisted function helps suggest or auto-generate backend API templates.

CRUD operations (Create, Read, Update, Delete) work correctly for managing transportation data.

The output confirms the backend logic and AI integration concept