Assignment-4.3

Name: Y. Poojitha

Hall ticket No:2303A51499

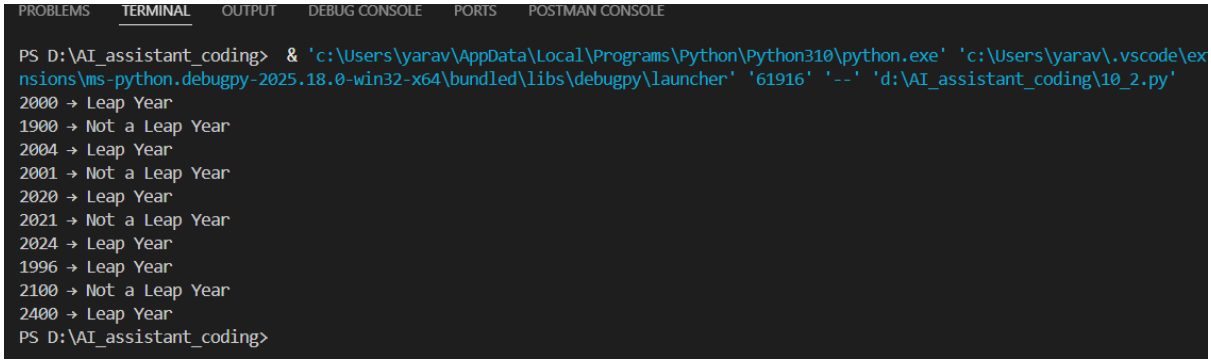
Batch:08

Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques

Task 1: Zero-Shot Prompting – Leap Year Check

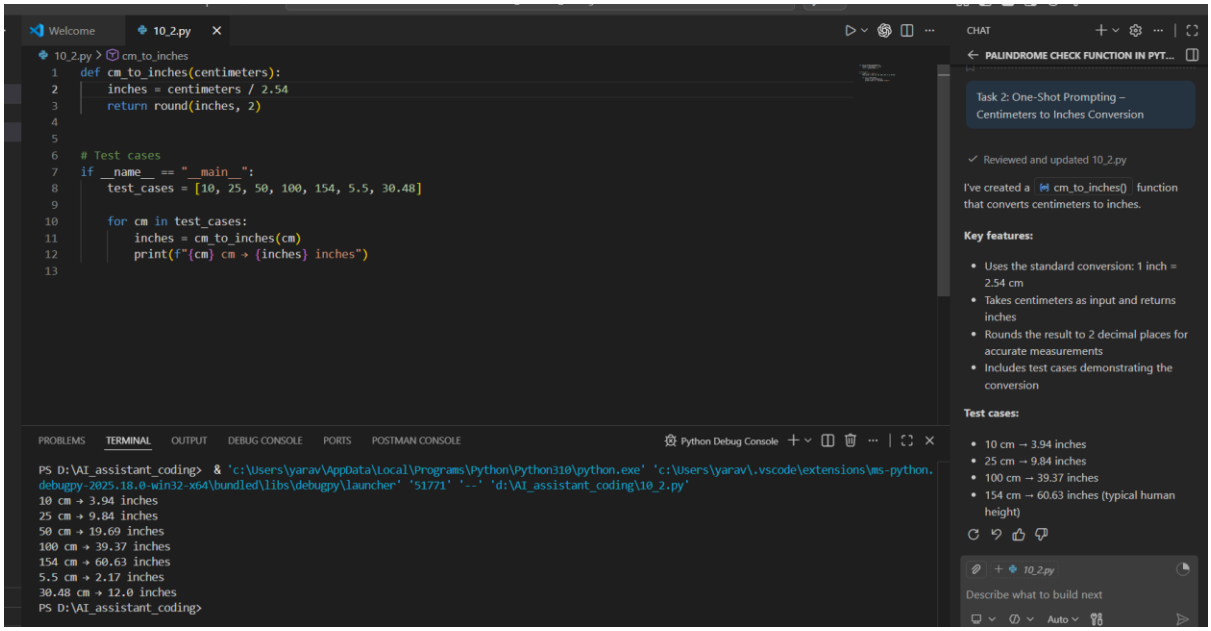

```python
def is_leap_year(year):
    if year % 400 == 0:
        return True

    if year % 100 == 0:
        return False

    if year % 4 == 0:
        return True

    return False

# Test cases
if __name__ == "__main__":
    test_cases = [2000, 1900, 2004, 2001, 2020, 2021, 2024, 1996, 2100, 2400]

    for year in test_cases:
        result = is_leap_year(year)
        status = "Leap Year" if result else "Not a Leap Year"
        print(f"{year} → {status}")
```

Output:

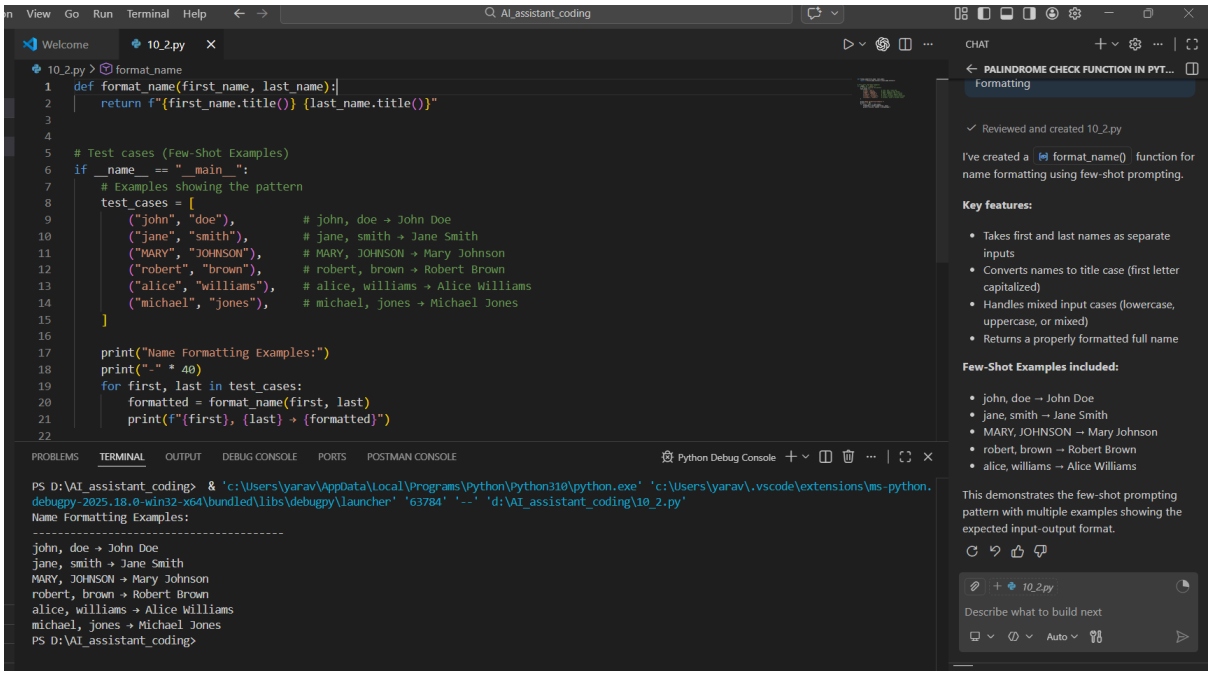


Task 2: One-Shot Prompting

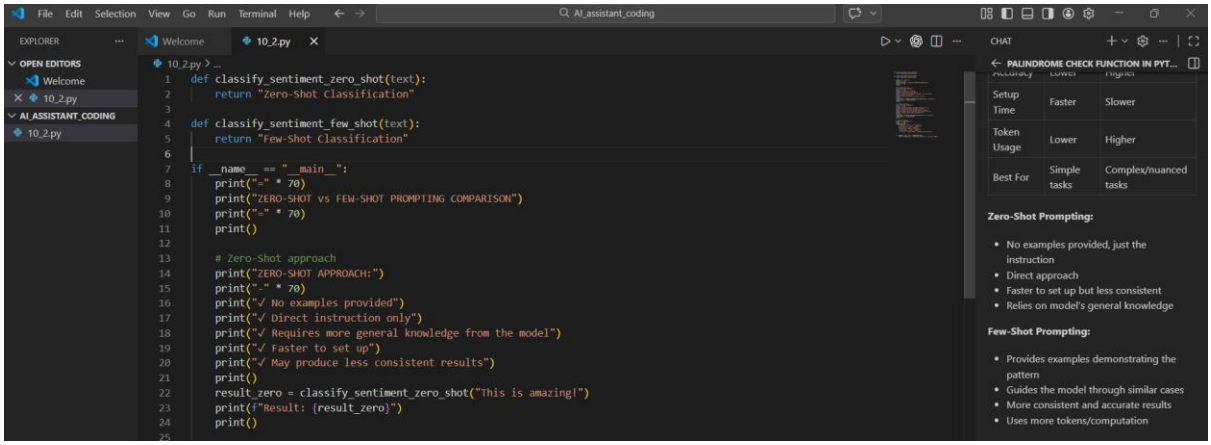
Centimeters to Inches Conversion



Task 3: Few-Shot Prompting – Name Formatting



Task 4: Comparative Analysis – Zero-Shot vs Few-Shot


File Edit Selection View Go Run Terminal Help
AI_assistant_coding
EXPLORER
OPEN EDITORS
Welcome
10_2.py
AI_ASSISTANT_CODING
10_2.py
Welcome
10_2.py
10_2.py > ...
def classify_sentiment_zero_shot(text):
return "Zero-Shot Classification"
def classify_sentiment_few_shot(text):
return "Few-Shot Classification"
if __name__ == "__main__":
print("=" * 70)
print("ZERO-SHOT vs FEW-SHOT PROMPTING COMPARISON")
print("=" * 70)
print()
# Zero-Shot approach
print("ZERO-SHOT APPROACH:")
print("-" * 70)
print("✓ No examples provided")
print("✓ Direct instruction only")
print("✓ Requires more general knowledge from the model")
print("✓ Faster to set up")
print("✓ May produce less consistent results")
print()
result_zero = classify_sentiment_zero_shot("This is amazing!")
print(f"Result: {result_zero}")
print()
CHAT
PALINDROME CHECK FUNCTION IN PYT...
Setup Time Faster Slower
Token Usage Lower Higher
Best For Simple tasks Complex/nuanced tasks
Zero-Shot Prompting:
No examples provided, just the instruction
Direct approach
Faster to set up but less consistent
Relies on model's general knowledge
Few-Shot Prompting:
Provides examples demonstrating the pattern
Guides the model through similar cases
More consistent and accurate results
Uses more tokens/computation


# Few-Shot approach
print("FEW-SHOT APPROACH:")
print("-" * 70)
print("✓ Multiple examples provided (typically 2-5)")
print("✓ Demonstrates the expected pattern")
print("✓ Guides the model through similar examples")
print("✓ Produces more consistent and accurate results")
print("✓ Requires more tokens/computation")
print()
result_few = classify_sentiment_few_shot("This is amazing!")
print(f"Result: {result_few}")
print()
# Key Differences
print("KEY DIFFERENCES:")
print("-" * 70)
comparison_data = [
("Examples", "None", "2-5+ examples"),
("Consistency", "Lower", "Higher"),
("Accuracy", "Lower", "Higher"),
("Setup Time", "Faster", "Slower"),
("Token Usage", "Lower", "Higher"),
("Use Case", "Simple tasks", "Complex/nuanced tasks"),
]
for feature, zero, few in comparison_data:
print(f"{feature:15} | Zero-Shot: {zero:20} | Few-Shot: {few}")
Usage
Best For
Zero-Shot
No exa
instruct
Direct a
Faster t
Relies o
Few-Shot P
Provide
pattern
Guides
More c
Uses m
The file incl
implementa
would struc
Describe wh