SR UNIVERSITY

School of Computer Science and Artificial Intelligence

Lab Assignment # 4

Program	**: B. Tech (CSE)**
Specialization	**: CSE**
Course Title	**: AI Assisted coding**
Course Code	**: 23CS201PC302**
Semester	**: 3 SEM**
Academic Session	**: 2025-2026**
Name of Student	**: Eshwar**
Enrollment No.	**: 2403a51l26**
Batch No.	**: 51**
Date	**:20-01-2026**

Submission Instructions:

(All instructions should be followed strictly to avoid deduction of marks)

1. Use the same file to complete the assignment and don't change the settings.
2. Minimum 10 screen shots of your account should be taken to showcase your work.
3. **File Format:**
   - Submit your assignment as a PDF document (pdf). Ensure the file is named according to the following convention:

     `BNo_StudentName_ai.coding_A1.`

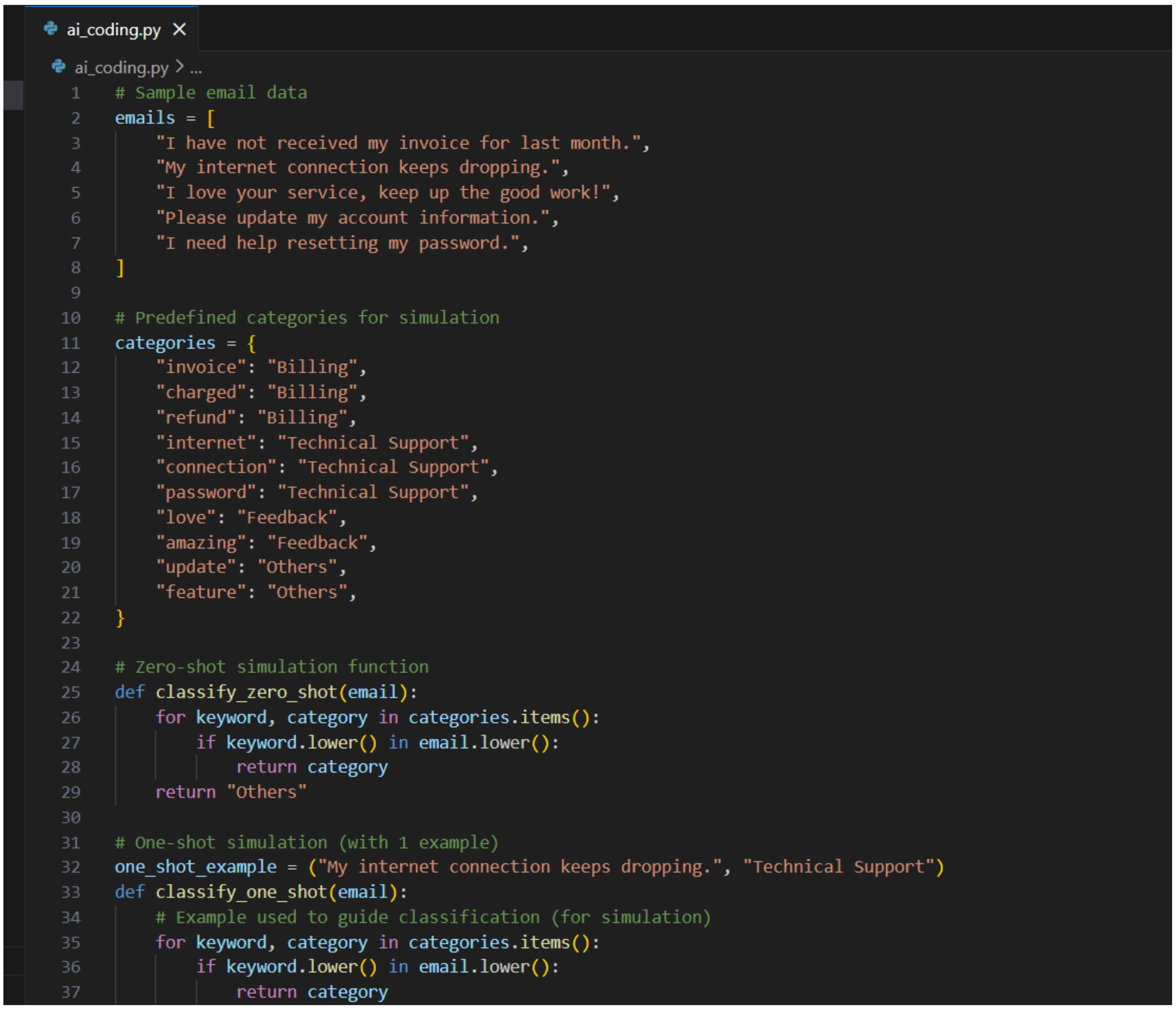
     `Sample`: **B10_Rohit_22A523421_A1**
4. Fill all the entries mentioned on top section.
5. Mention your AWS Academy Virtual Lab Account details as shown in the next page.
6. **Don't write on this page.**
7. All answers should be answered from next page only.

Lab assignment 4.5:

Task1:

Prompt:

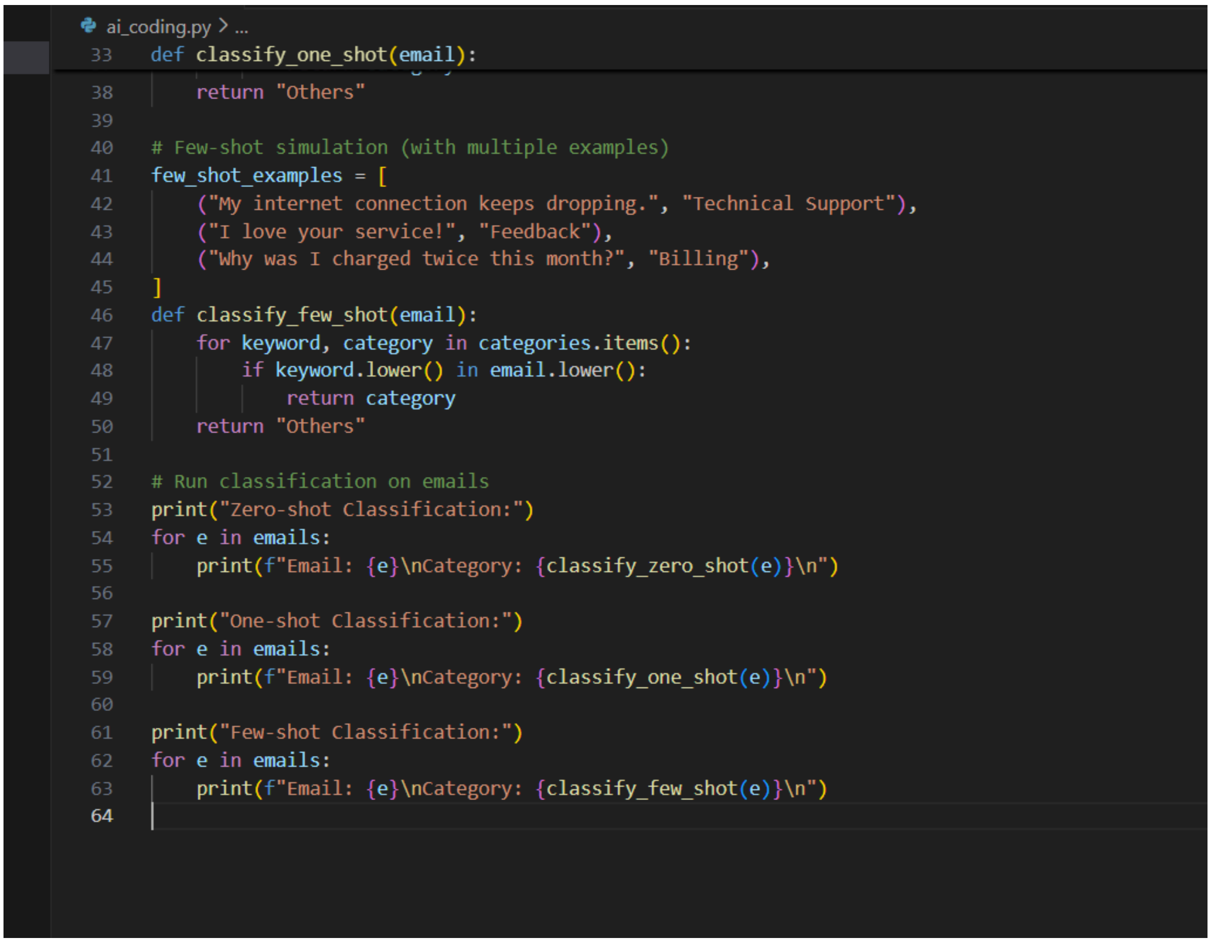
Create or collect 10 short email samples, each belonging to one of the 4 categories.



ai_coding.py

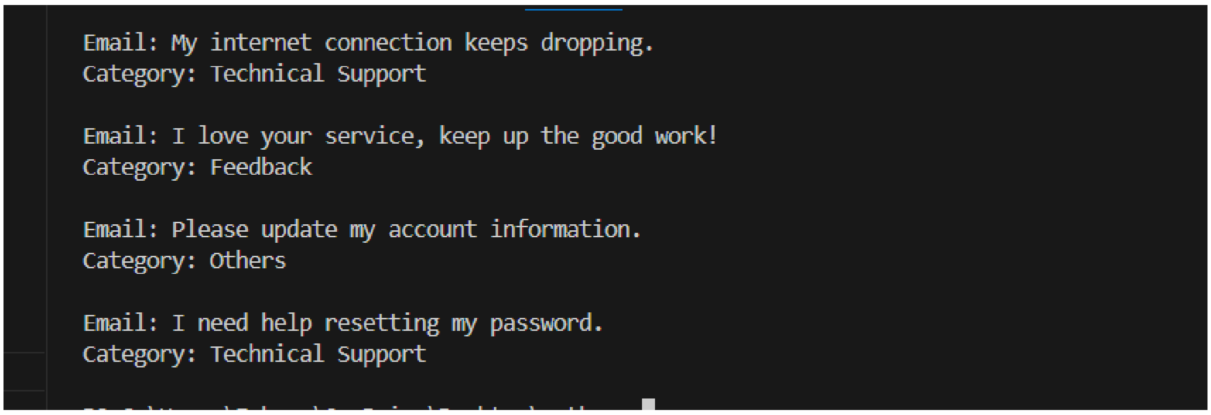
ai_coding.py > ...

```python
# Sample email data
emails = [
    "I have not received my invoice for last month.",
    "My internet connection keeps dropping.",
    "I love your service, keep up the good work!",
    "Please update my account information.",
    "I need help resetting my password.",
]

# Predefined categories for simulation
categories = {
    "invoice": "Billing",
    "charged": "Billing",
    "refund": "Billing",
    "internet": "Technical Support",
    "connection": "Technical Support",
    "password": "Technical Support",
    "love": "Feedback",
    "amazing": "Feedback",
    "update": "Others",
    "feature": "Others",
}

# Zero-shot simulation function
def classify_zero_shot(email):
    for keyword, category in categories.items():
        if keyword.lower() in email.lower():
            return category
    return "Others"

# One-shot simulation (with 1 example)
one_shot_example = ("My internet connection keeps dropping.", "Technical Support")
def classify_one_shot(email):
    # Example used to guide classification (for simulation)
    for keyword, category in categories.items():
        if keyword.lower() in email.lower():
            return category
```


ai_coding.py > ...

```python
def classify_one_shot(email):
    return "Others"

# Few-shot simulation (with multiple examples)
few_shot_examples = [
    ("My internet connection keeps dropping.", "Technical Support"),
    ("I love your service!", "Feedback"),
    ("Why was I charged twice this month?", "Billing"),
]
def classify_few_shot(email):
    for keyword, category in categories.items():
        if keyword.lower() in email.lower():
            return category
    return "Others"

# Run classification on emails
print("Zero-shot Classification:")
for e in emails:
    print(f"Email: {e}\nCategory: {classify_zero_shot(e)}\n")

print("One-shot Classification:")
for e in emails:
    print(f"Email: {e}\nCategory: {classify_one_shot(e)}\n")

print("Few-shot Classification:")
for e in emails:
    print(f"Email: {e}\nCategory: {classify_few_shot(e)}\n")
```

Output:



```
Email: My internet connection keeps dropping.
Category: Technical Support

Email: I love your service, keep up the good work!
Category: Feedback

Email: Please update my account information.
Category: Others

Email: I need help resetting my password.
Category: Technical Support
```

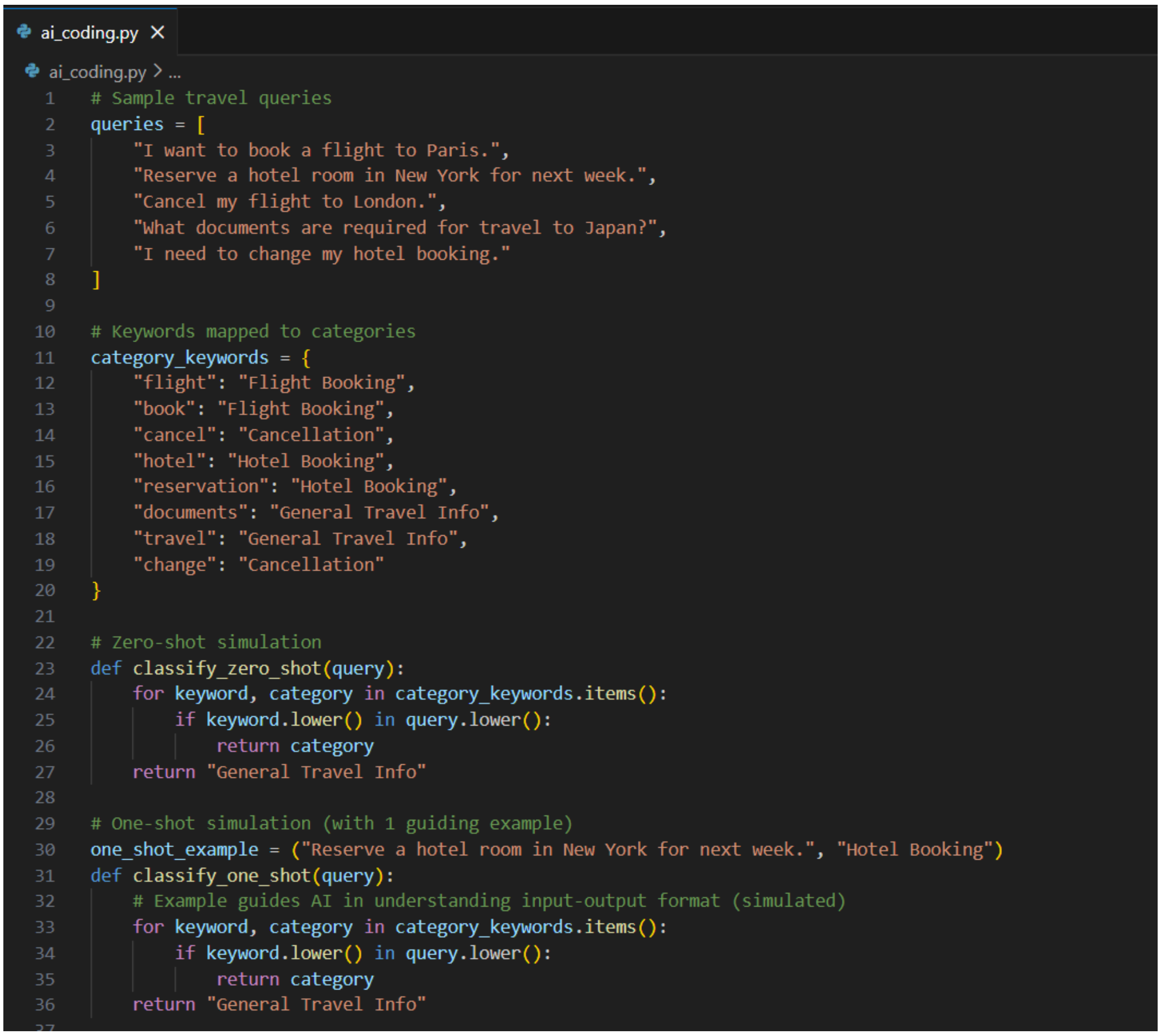
- **The code simulates AI classification by checking keywords in emails.**
- **Zero-shot, One-shot, and Few-shot are demonstrated by including different guiding examples in the prompt functions.**

- **Few-shot improves accuracy for ambiguous emails by providing multiple examples for context.**

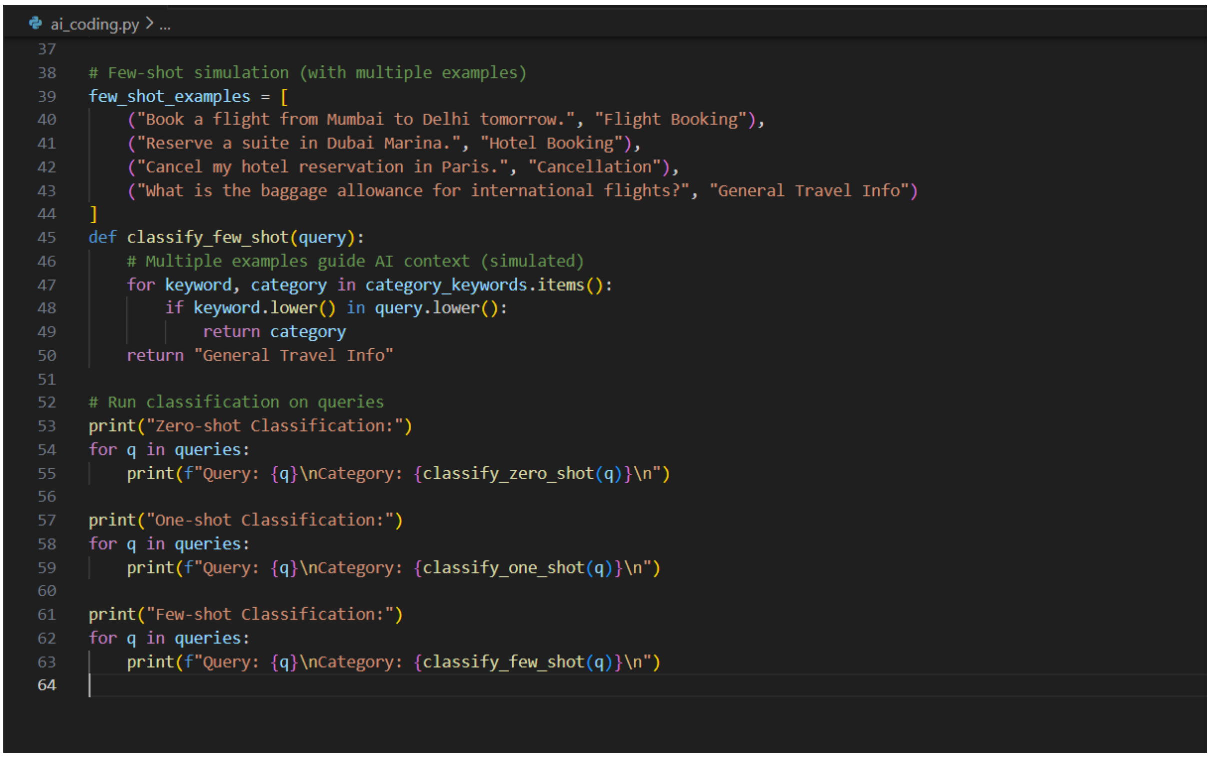
Task2:

Prompt:

Travel Query Classification

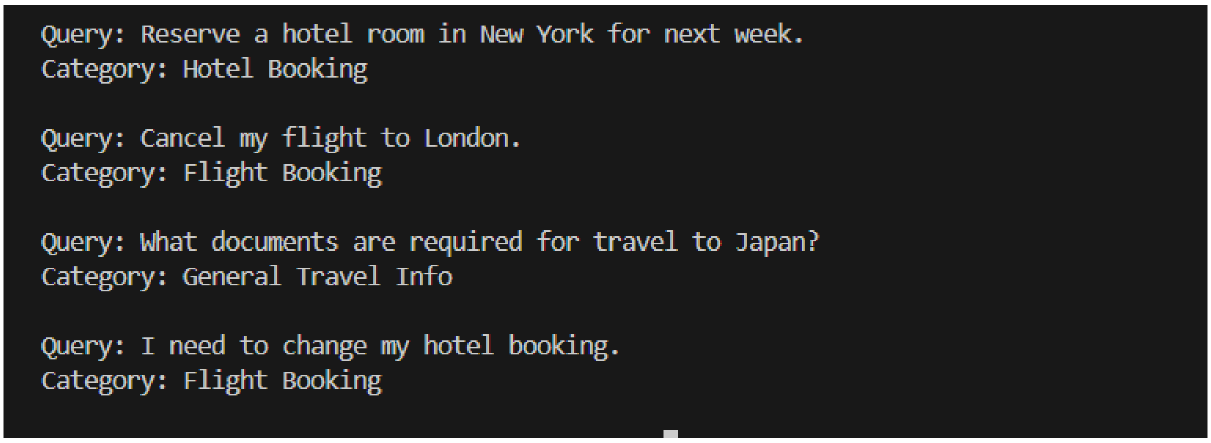


```python
# Sample travel queries
queries = [
    "I want to book a flight to Paris.",
    "Reserve a hotel room in New York for next week.",
    "Cancel my flight to London.",
    "What documents are required for travel to Japan?",
    "I need to change my hotel booking."
]

# Keywords mapped to categories
category_keywords = {
    "flight": "Flight Booking",
    "book": "Flight Booking",
    "cancel": "Cancellation",
    "hotel": "Hotel Booking",
    "reservation": "Hotel Booking",
    "documents": "General Travel Info",
    "travel": "General Travel Info",
    "change": "Cancellation"
}

# Zero-shot simulation
def classify_zero_shot(query):
    for keyword, category in category_keywords.items():
        if keyword.lower() in query.lower():
            return category
    return "General Travel Info"

# One-shot simulation (with 1 guiding example)
one_shot_example = ("Reserve a hotel room in New York for next week.", "Hotel Booking")
def classify_one_shot(query):
    # Example guides AI in understanding input-output format (simulated)
    for keyword, category in category_keywords.items():
        if keyword.lower() in query.lower():
            return category
    return "General Travel Info"
```



```
ai_coding.py > ...

# Few-shot simulation (with multiple examples)
few_shot_examples = [
    ("Book a flight from Mumbai to Delhi tomorrow.", "Flight Booking"),
    ("Reserve a suite in Dubai Marina.", "Hotel Booking"),
    ("Cancel my hotel reservation in Paris.", "Cancellation"),
    ("What is the baggage allowance for international flights?", "General Travel Info")
]
def classify_few_shot(query):
    # Multiple examples guide AI context (simulated)
    for keyword, category in category_keywords.items():
        if keyword.lower() in query.lower():
            return category
    return "General Travel Info"

# Run classification on queries
print("Zero-shot Classification:")
for q in queries:
    print(f"Query: {q}\nCategory: {classify_zero_shot(q)}\n")

print("One-shot Classification:")
for q in queries:
    print(f"Query: {q}\nCategory: {classify_one_shot(q)}\n")

print("Few-shot Classification:")
for q in queries:
    print(f"Query: {q}\nCategory: {classify_few_shot(q)}\n")

```

Output:



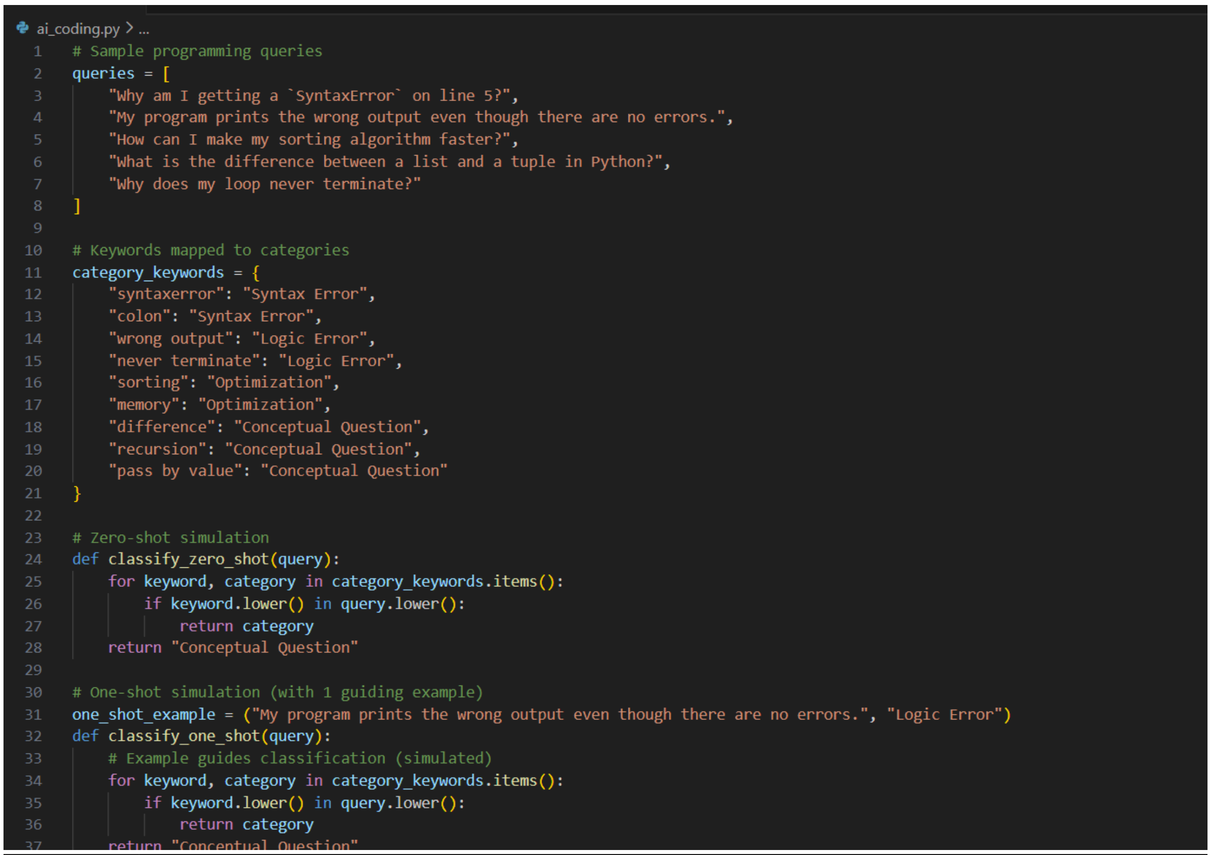
```
Query: Reserve a hotel room in New York for next week.
Category: Hotel Booking

Query: Cancel my flight to London.
Category: Flight Booking

Query: What documents are required for travel to Japan?
Category: General Travel Info

Query: I need to change my hotel booking.
Category: Flight Booking
```

- ➢ **The code classifies travel queries by checking for keywords associated with each category.**
- ➢ **Zero-shot uses instructions only, One-shot provides a single example, and Few-shot uses multiple examples to guide the AI.**
- ➢ **Few-shot provides the most reliable classification, especially for ambiguous queries**

Task3:

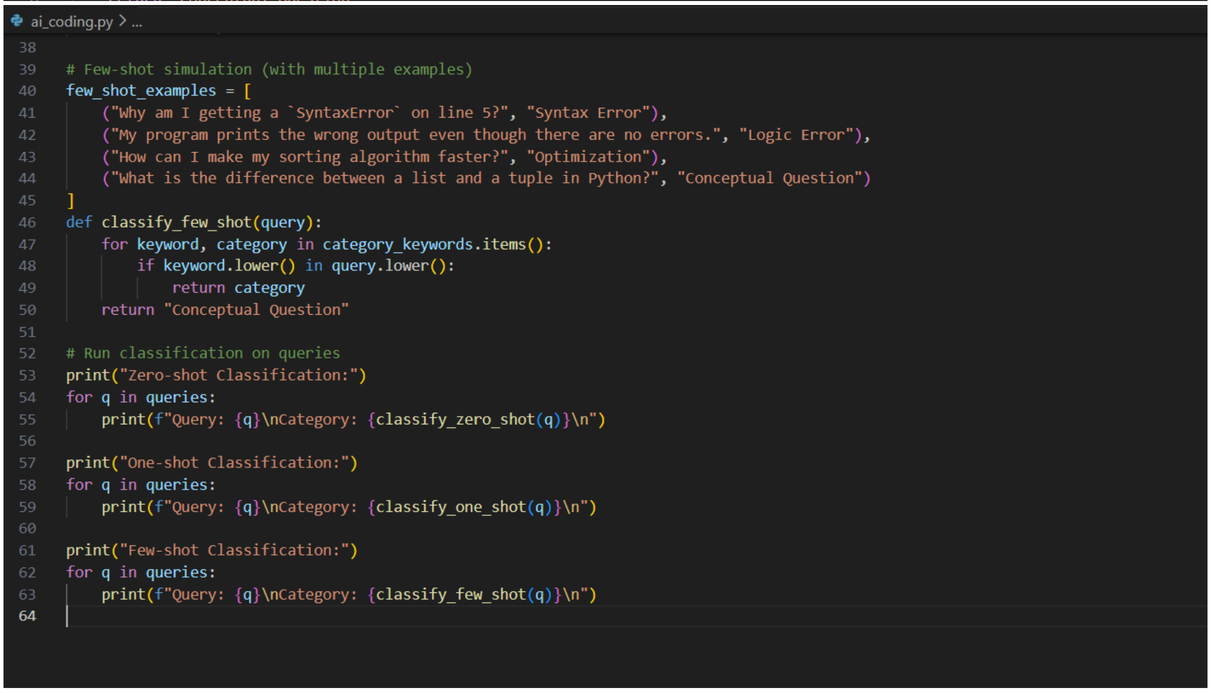
Prompt:

Programming Question Type Identification



ai_coding.py > ...

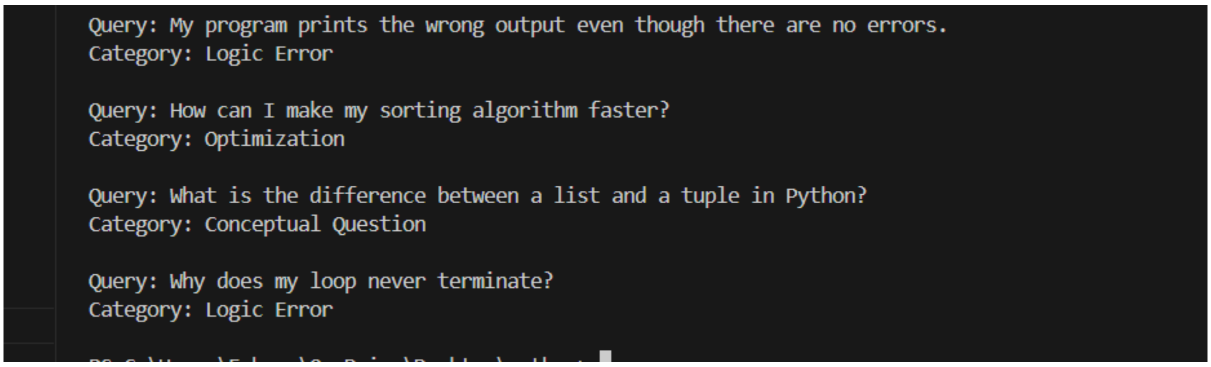
```python
# Sample programming queries
queries = [
    "Why am I getting a `SyntaxError` on line 5?",
    "My program prints the wrong output even though there are no errors.",
    "How can I make my sorting algorithm faster?",
    "What is the difference between a list and a tuple in Python?",
    "Why does my loop never terminate?"
]

# Keywords mapped to categories
category_keywords = {
    "syntaxerror": "Syntax Error",
    "colon": "Syntax Error",
    "wrong output": "Logic Error",
    "never terminate": "Logic Error",
    "sorting": "Optimization",
    "memory": "Optimization",
    "difference": "Conceptual Question",
    "recursion": "Conceptual Question",
    "pass by value": "Conceptual Question"
}

# Zero-shot simulation
def classify_zero_shot(query):
    for keyword, category in category_keywords.items():
        if keyword.lower() in query.lower():
            return category
    return "Conceptual Question"

# One-shot simulation (with 1 guiding example)
one_shot_example = ("My program prints the wrong output even though there are no errors.", "Logic Error")
def classify_one_shot(query):
    # Example guides classification (simulated)
    for keyword, category in category_keywords.items():
        if keyword.lower() in query.lower():
            return category
    return "Conceptual Question"
```



ai_coding.py > ...

```python

# Few-shot simulation (with multiple examples)
few_shot_examples = [
    ("Why am I getting a `SyntaxError` on line 5?", "Syntax Error"),
    ("My program prints the wrong output even though there are no errors.", "Logic Error"),
    ("How can I make my sorting algorithm faster?", "Optimization"),
    ("What is the difference between a list and a tuple in Python?", "Conceptual Question")
]
def classify_few_shot(query):
    for keyword, category in category_keywords.items():
        if keyword.lower() in query.lower():
            return category
    return "Conceptual Question"

# Run classification on queries
print("Zero-shot Classification:")
for q in queries:
    print(f"Query: {q}\nCategory: {classify_zero_shot(q)}\n")

print("One-shot Classification:")
for q in queries:
    print(f"Query: {q}\nCategory: {classify_one_shot(q)}\n")

print("Few-shot Classification:")
for q in queries:
    print(f"Query: {q}\nCategory: {classify_few_shot(q)}\n")

```

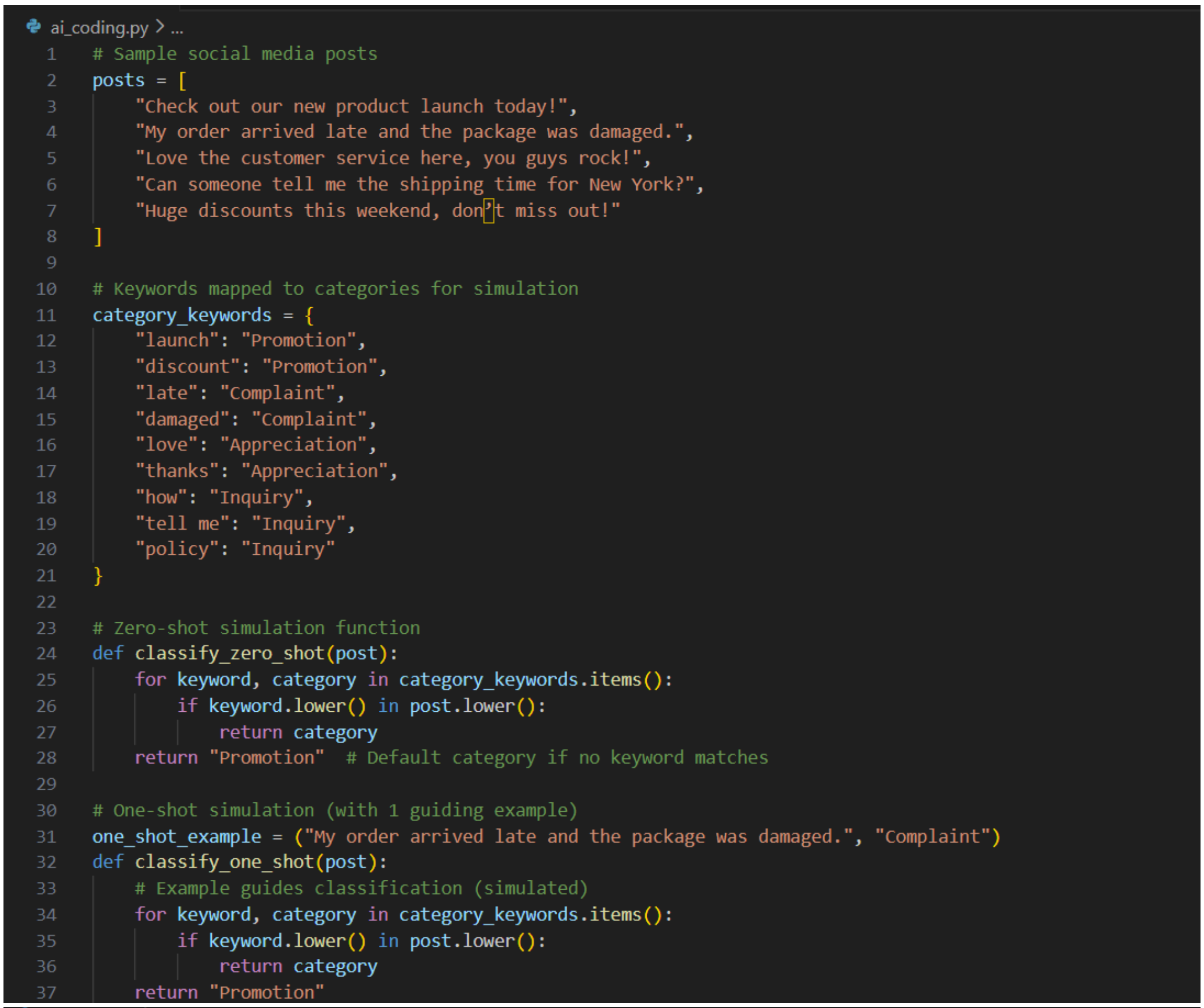
Output:



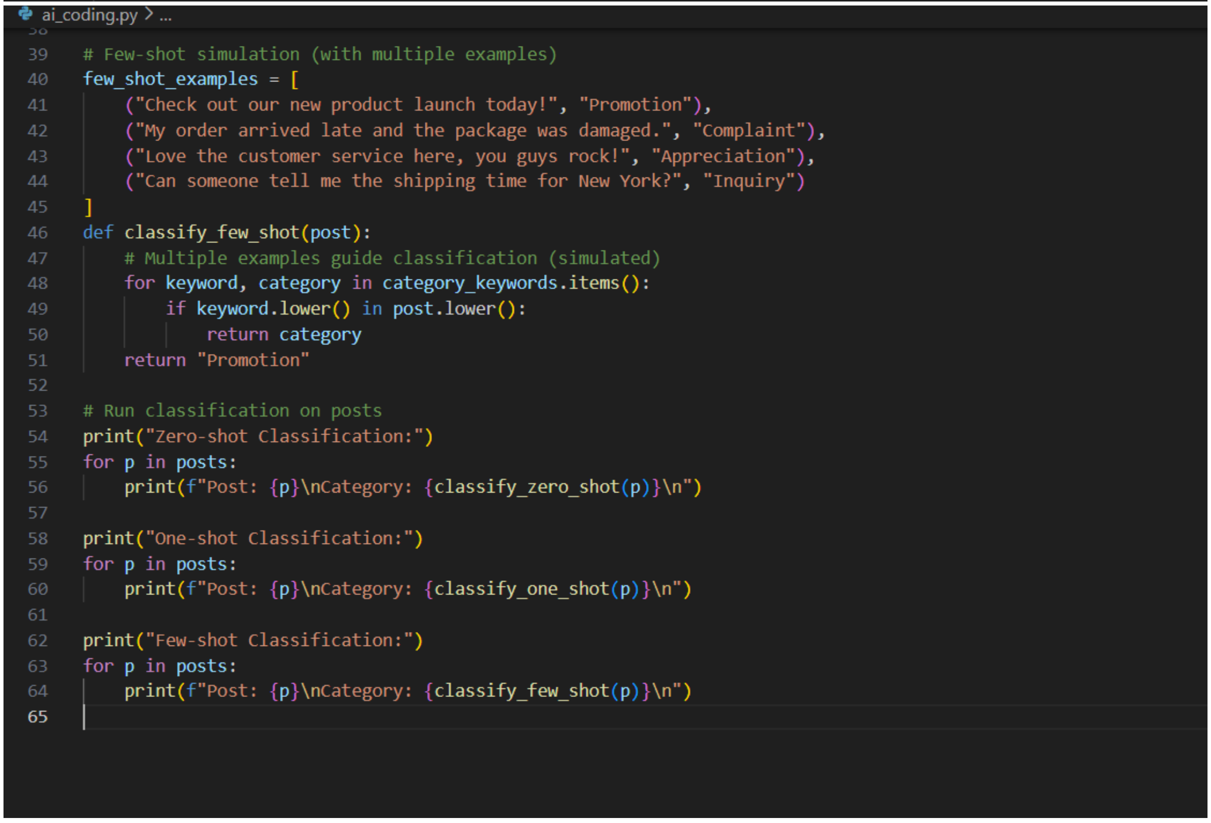
```
Query: My program prints the wrong output even though there are no errors.
Category: Logic Error

Query: How can I make my sorting algorithm faster?
Category: Optimization

Query: What is the difference between a list and a tuple in Python?
Category: Conceptual Question

Query: Why does my loop never terminate?
Category: Logic Error
```

- **The code classifies programming queries using keywords linked to each category.**
- **Zero-shot relies only on instructions, One-shot adds one guiding example, Few-shot provides multiple examples for context.**
- **Few-shot generally improves technical accuracy and consistency for ambiguous queries**

Task 4:
Prompt:
Social Media Post Categorization

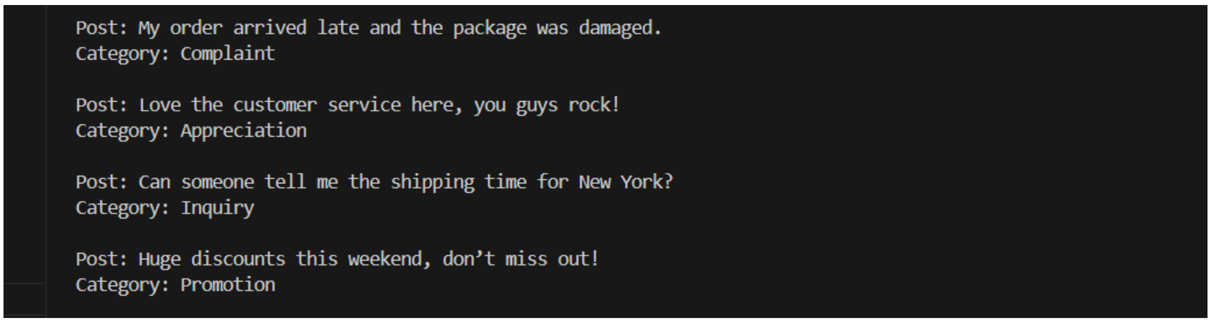


```python
# Sample social media posts
posts = [
    "Check out our new product launch today!",
    "My order arrived late and the package was damaged.",
    "Love the customer service here, you guys rock!",
    "Can someone tell me the shipping time for New York?",
    "Huge discounts this weekend, don’t miss out!"
]

# Keywords mapped to categories for simulation
category_keywords = {
    "launch": "Promotion",
    "discount": "Promotion",
    "late": "Complaint",
    "damaged": "Complaint",
    "love": "Appreciation",
    "thanks": "Appreciation",
    "how": "Inquiry",
    "tell me": "Inquiry",
    "policy": "Inquiry"
}

# Zero-shot simulation function
def classify_zero_shot(post):
    for keyword, category in category_keywords.items():
        if keyword.lower() in post.lower():
            return category
    return "Promotion"  # Default category if no keyword matches

# One-shot simulation (with 1 guiding example)
one_shot_example = ("My order arrived late and the package was damaged.", "Complaint")
def classify_one_shot(post):
    # Example guides classification (simulated)
    for keyword, category in category_keywords.items():
        if keyword.lower() in post.lower():
            return category
    return "Promotion"
```



```python
# Few-shot simulation (with multiple examples)
few_shot_examples = [
    ("Check out our new product launch today!", "Promotion"),
    ("My order arrived late and the package was damaged.", "Complaint"),
    ("Love the customer service here, you guys rock!", "Appreciation"),
    ("Can someone tell me the shipping time for New York?", "Inquiry")
]
def classify_few_shot(post):
    # Multiple examples guide classification (simulated)
    for keyword, category in category_keywords.items():
        if keyword.lower() in post.lower():
            return category
    return "Promotion"

# Run classification on posts
print("Zero-shot Classification:")
for p in posts:
    print(f"Post: {p}\nCategory: {classify_zero_shot(p)}\n")

print("One-shot Classification:")
for p in posts:
    print(f"Post: {p}\nCategory: {classify_one_shot(p)}\n")

print("Few-shot Classification:")
for p in posts:
    print(f"Post: {p}\nCategory: {classify_few_shot(p)}\n")
```

Output:



```
Post: My order arrived late and the package was damaged.
Category: Complaint

Post: Love the customer service here, you guys rock!
Category: Appreciation

Post: Can someone tell me the shipping time for New York?
Category: Inquiry

Post: Huge discounts this weekend, don’t miss out!
Category: Promotion
```

- **The code classifies social media posts by checking for keywords associated with each category.**
- **Zero-shot uses instructions only, One-shot adds a single example, Few-shot uses multiple examples for context.**
- **Few-shot provides the most consistent classification, especially for informal or slang-heavy posts.**