

# AI ASSISTED CODING

## LAB EXAM-1

NAME : SD.Nayaz HT.no : 2403A52076 BATCH : 04

+

### Q1. Zero-shot Prompting in Healthcare

- **Task 1:** Write a zero-shot prompt that classifies the severity of symptoms without giving any examples

**Prompt:**

**"You are a medical triage assistant. Given a patient's described symptoms, classify the severity of the condition into one of three categories: Mild, Moderate, or Severe.**

**Provide only the classification without explanation".**

**CODE :**

The screenshot shows a code editor window with a Python script named `AI lab exam.py`. The code defines a function `triage(symptoms)` that takes a string of symptoms and returns a severity level ('Severe', 'Moderate', or 'Mild') based on a list of critical symptoms. An example run is shown in the terminal below, where the user enters 'fever' and the program outputs 'Severity: Mild'. The terminal also shows the file path and some status indicators.

```
AI lab exam.py > ...
1 def triage(symptoms):
2     s = symptoms.lower()
3     if any(k in s for k in ["chest pain", "shortness of breath", "unconscious", "severe bleeding"]):
4         return "Severe"
5     elif any(k in s for k in ["high fever", "dizziness", "vomiting", "injury"]):
6         return "Moderate"
7     else:
8         return "Mild"
9
10 # Example run
11 symptoms = input("Enter symptoms: ")
12 print("Severity:", triage(symptoms))
13
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ... | [ ] x

```
PS C:\Users\surya\OneDrive\Desktop\AI assist> & C:/Users/surya/AppData/Local/Microsoft/Windows/Apps/python3.12.exe "c:/Users/surya/OneDrive/Desktop/AI assist/AI lab exam.py"
Enter symptoms: fever
Severity: Mild
PS C:\Users\surya\OneDrive\Desktop\AI assist>
```

Python  
Python  
Python

Ln 13, Col 1 Spaces: 4 UTF-8 CRLF {} Python Chat quota reached



**Observation:**

The code gives a quick diet plan. Without context, it suggests a general healthy diet, and with context (e.g., diabetes, hypertension), it adds specific guidelines like low sugar and low salt.

## Q2. One-shot vs Few-shot for Customer Support

**•Task 1: Write:**

*oA one-shot prompt with 1 example of classification.*

*oA few-shot prompt with 3–4 examples.*

**Prompt : One-shot Prompt**

*Classify emails as Refund, Order Status, or Technical Issue.*

**Example:**

*Email: 'I want my money back for these shoes.' → Refund*

*Email: 'My package is delayed, any update?'*

**Prompt : Few-shot**

*Classify emails as Refund, Order Status, or Technical Issue.*

*Email: 'I want my money back.' → Refund*

*Email: 'Where is my order?' → Order Status*

*Email: 'The website checkout isn't working.' → Technical Issue*

*Email: 'I got the wrong size, need a refund.' → Refund*

*Email: 'I can't track my order with the link.'*

**CODE :**

```

lab6.4.py | AI lab exam.py x |
AI lab exam.py > ...
1 def classify_email(email):
2     e = email.lower()
3     if any(k in e for k in ["refund", "return", "money back", "send it back", "wrong size"]):
4         return "Refund"
5     elif any(k in e for k in ["order", "package", "tracking", "shipped", "arrive"]):
6         return "Order Status"
7     elif any(k in e for k in ["error", "issue", "problem", "not working", "bug", "checkout"]):
8         return "Technical Issue"
9     else:
10        return "Uncategorized"
11
12 # Examples
13 print(classify_email("I want my money back for these shoes"))
14 print(classify_email("My package hasn't arrived yet, any update?"))
15 print(classify_email("The payment page keeps showing an error"))
16

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

Order Status
Technical Issue
PS C:\Users\surya\OneDrive\Desktop\AI assist> & C:/Users/surya/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/surya/OneDrive/Desktop/AI assist/AI lab exam.py"
Refund
Order Status
Technical Issue
PS C:\Users\surya\OneDrive\Desktop\AI assist>

```

In 16, Col 1 Spaces: 4 UTF-8 CRLF {} Python Chat quota reached 3.12.10 (Micro

### **Observation:**

**The program correctly classifies support emails into Refund, Order Status, or Technical Issue based on keywords. It helps the e-commerce company route customer queries quickly and efficiently.**

- Task 2 : Use the same incoming email text for both prompts. Compare how the outputs differ and explain why.**

#### **Prompt : One-shot**

**"Classify emails as Refund, Order Status, or Technical Issue.**

**Email: 'I want my money back for these shoes.' → Refund**

**Email: 'I can't track my order, the link isn't working.'"**

**Likely Output: Order Status**

#### **Prompt : Few-shot**

**"Classify emails as Refund, Order Status, or Technical Issue.**

**Email: 'I want my money back.' → Refund**

**Email: 'Where is my order?' → Order Status**

**Email: 'The website checkout isn't working.' → Technical Issue**

**Email: 'I got the wrong size, need a refund.' → Refund**

**Email: 'I can't track my order, the link isn't working.'"**

**Likely Output: Technical Issue**

## CODE:

The screenshot shows a code editor interface with two tabs: "lab6.4.py" and "AI lab exam.py". The "AI lab exam.py" tab is active, displaying the following Python code:

```
1 email = "I can't track my order, the link isn't working."
2
3 # One-shot (less context)
4 if "refund" in email.lower():
5     one_shot = "Refund"
6 else:
7     one_shot = "Order status"
8
9 # Few-shot (more context)
10 if any(w in email.lower() for w in ["refund", "money back", "return"]):
11     few_shot = "Refund"
12 elif any(w in email.lower() for w in ["error", "not working", "problem"]):
13     few_shot = "Technical Issue"
14 elif "order" in email.lower():
15     few_shot = "Order Status"
16 else:
17     few_shot = "Uncategorized"
18
19 print("One-shot Output:", one_shot)
20 print("Few-shot Output:", few_shot)
21
```

Below the code editor is a terminal window showing the execution of the script:

```
/Desktop/AI assist/AI lab exam.py
One-shot Output: Order Status
Few-shot Output: Order Status
PS C:\Users\surya\OneDrive\Desktop\AI assist> & C:/Users/surya/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/surya/OneDrive/Desktop/AI assist/AI lab exam.py"
/Desktop/AI assist/AI lab exam.py
One-shot Output: Order Status
Few-shot Output: Order Status
PS C:\Users\surya\OneDrive\Desktop\AI assist>
```

The terminal also displays some status information at the bottom:

Ln 21, Col 1 Spaces: 4 UTF-8 CRLF {} Python Chat quota reached 3.12.10 (Micro)

### Observation:

**The code shows that with the same email text, the one-shot prompt gives Order Status (less context, misclassified), while the few-shot prompt gives Technical Issue (more examples, better accuracy).**