

**School of Computer Science and Artificial Intelligence****Lab Assignment # 1**

---

<b>Program</b>	: B. Tech (CSE)
<b>Specialization</b>	:
<b>Course Title</b>	: AI Assisted coding
<b>Course Code</b>	:
<b>Semester</b>	: II
<b>Academic Session</b>	: 2025-2026
<b>Name of Student</b>	: K.Pavan
<b>Enrollment No.</b>	: 2403A51L19
<b>Batch No.</b>	: 51
<b>Date</b>	:16-01-2026

---

**Submission Starts here :****AI LAB – 4**

Advanced Prompt Engineering: Zero-Shot, One-Shot, Few-Shot

**Task 1: Zero-Shot Prompting – Email Classification**

Prompt:

Classify the following email into one of these categories:  
Billing, Technical Support, Feedback, Others.

Email: "I have not received my invoice for last month."

```
print("\nTask 1: Zero-Shot Prompting Email Classification\n")

prompt = """
Classify the following email into one of these categories:
Billing, Technical Support, Feedback, Others.

Email: "I have not received my invoice for last month."
"""

print(prompt)
# Expected Output: Billing
```

Output :

**Zero-Shot Prompt:**

Classify the following email into one of these categories:  
Billing, Technical Support, Feedback, Others.

Email: "I have not received my invoice for last month."

Explanation:

Zero-shot prompting does not provide any examples to the model.

The model relies entirely on its pre-trained knowledge.

This method works well for simple and clear inputs.

## Task 2: One-Shot Prompting – Email Classification

Prompt:

Example:

Email: "The application crashes frequently."

Category: Technical Support

Now classify the following email:

Email: "I was charged twice for my subscription."

```
print("\nTask 2: One-Shot Prompting Email Classification\n")
```

```
prompt = """
```

Example:

Email: "The application crashes frequently."

Category: Technical Support

Now classify the following email:

Email: "I was charged twice for my subscription."

```
"""
```

```
print(prompt)
```

```
# Expected Output: Billing
```

Output :

**One-Shot Prompt:**

Example:

Email: "My internet is not working."

Category: Technical Support

**Now classify:**

Email: "I was charged twice for my subscription."

Explanation:

One-shot prompting provides a single labeled example.

It helps the model understand the expected format.

Accuracy improves compared to zero-shot prompting.

### Task 3: Few-Shot Prompting – Email Classification

Prompt:

Email: "Invoice not generated."

Category: Billing

Email: "App crashes on login."

Category: Technical Support

Email: "Very happy with the service."

Category: Feedback

Now classify the following email:

Email: "Unable to reset my password."

```
print("\nTask 3: Few-Shot Prompting Email Classification\n")

prompt = """
Email: "Invoice not generated."
Category: Billing

Email: "App crashes on login."
Category: Technical Support

Email: "Very happy with the service."
Category: Feedback

Now classify the following email:
Email: "Unable to reset my password."
"""

print(prompt)
# Expected Output: Technical Support
```

Output:

```
Few-Shot Prompt:
Email: "I have not received my bill."
Category: Billing

Email: "The app crashes on startup."
Category: Technical Support

Email: "Great service experience."
Category: Feedback

Now classify:
Email: "I am unable to reset my password."
```

Explanation:

Few-shot prompting provides multiple labeled examples.

This clearly defines the category boundaries.

It gives the highest accuracy among the methods.

### Task 4: Zero-Shot Prompting – Travel Query Classification

Prompt:

Classify the following travel query into one of these categories:

Flight Booking, Hotel Booking, Cancellation, General Travel Info.

```
Query: "Cancel my flight ticket booked for tomorrow."
print("\nTask 4: Zero-Shot Prompting Travel Query Classification\n")

prompt = """
Classify the following travel query into one of these categories:
Flight Booking, Hotel Booking, Cancellation, General Travel Info.

Query: "Cancel my flight ticket booked for tomorrow."
"""

print(prompt)
# Expected Output: Cancellation
```

Output :

## 2. TRAVEL QUERY CLASSIFICATION

zero-Shot Prompt:

classify the following query into:  
Flight Booking, Hotel Booking, Cancellation, General Travel Info.

Query: "Cancel my flight ticket."

Explanation:

The model classifies the query without prior examples.

It uses keywords and intent understanding.

Effective for direct and simple travel queries.

## Task 5: One-Shot Prompting – Travel Query Classification

Prompt:

Example:

Query: "Book a flight from Delhi to London"

Category: Flight Booking

Now classify the following query:

Query: "I need a hotel in Goa for 3 nights."

```
print("\nTask 5: One-Shot Prompting Travel Query Classification\n")

prompt = """
Example:
Query: "Book a flight from Delhi to London"
Category: Flight Booking

Now classify the following query:
Query: "I need a hotel in Goa for 3 nights."
"""

print(prompt)
# Expected Output: Hotel Booking
```

Output :

**One-Shot Prompt:**

**Example:**

**Query:** "Book a flight to London"

**Category:** Flight Booking

**Now classify:**

**Query:** "I need a hotel in Goa."

**Explanation:**

One-shot prompting improves consistency in output.

The example guides the model's understanding.

Better accuracy than zero-shot prompting.

### **Task 6: Few-Shot Prompting – Travel Query Classification**

Prompt:

Query: "Book flight tickets to Dubai"

Category: Flight Booking

Query: "Cancel my hotel reservation"

Category: Cancellation

Query: "Find hotels near airport"

Category: Hotel Booking

Now classify the following query:

Query: "What documents are required for international travel?"

```
print("\nTask 6: Few-Shot Prompting Travel Query Classification\n")

prompt = """
Query: "Book flight tickets to Dubai"
Category: Flight Booking

Query: "Cancel my hotel reservation"
Category: Cancellation

Query: "Find hotels near airport"
Category: Hotel Booking

Now classify the following query:
Query: "What documents are required for international travel?"
"""

print(prompt)
# Expected Output: General Travel Info
```

Output :

```
Few-Shot Prompt:  
Query: "Book flight tickets to Dubai"  
Category: Flight Booking  
  
Query: "Cancel my hotel reservation"  
Category: Cancellation  
  
Query: "Find hotels near airport"  
Category: Hotel Booking  
  
Now classify:  
Query: "What documents are needed for international travel?"
```

Explanation:

Few-shot prompting reduces ambiguity in classification.

The model generalizes better to unseen queries.

Most reliable method for travel assistants.

## Task 7: Zero-Shot Prompting – Programming Question Classification

Prompt:

Classify the following programming question into:

Syntax Error, Logic Error, Optimization, Conceptual Question.

Question: "My program runs but gives incorrect output."

```
print("\nTask 7: Zero-Shot Prompting  Programming Question Classification\n")  
  
prompt = """  
Classify the following programming question into:  
Syntax Error, Logic Error, Optimization, Conceptual Question.  
  
Question: "My program runs but gives incorrect output."  
"""  
  
print(prompt)  
# Expected Output: Logic Error
```

### 3. PROGRAMMING QUESTION CLASSIFICATION

Zero-Shot Prompt:

Classify the following programming question into:

Syntax Error, Logic Error, Optimization, Conceptual Question.

Question: "My program runs but gives wrong output."

Explanation:

The model infers the issue without examples.

Suitable for clearly described programming problems.

May struggle with ambiguous technical questions.

## Task 8: One-Shot Prompting – Programming Question Classification

Prompt:

Example:

Question: "Missing semicolon in Java code"

Category: Syntax Error

Now classify the following question:

Question: "My program is slow for large inputs."

```
print("\nTask 8: One-Shot Prompting  Programming Question Classification\n")

prompt = """
Example:
Question: "Missing semicolon in Java code"
Category: Syntax Error

Now classify the following question:
Question: "My program is slow for large inputs."
"""

print(prompt)
# Expected Output: Optimization
```

**One-Shot Prompt:**

Example:

Question: "Missing semicolon in Java"

Category: Syntax Error

Now classify:

Question: "Program is slow for large inputs."

Explanation:

One example helps identify performance-related issues.

Improves technical understanding of categories.

More accurate than zero-shot prompting.

## Task 9: Few-Shot Prompting – Programming Question Classification

Prompt:

Question: "What is inheritance?"

Category: Conceptual Question

Question: "Extra bracket causes compilation error"

Category: Syntax Error

Question: "Infinite loop issue"

Category: Logic Error

Now classify the following question:

Question: "How can I optimize this algorithm?"

```
print("\nTask 9: Few-Shot Prompting Programming Question Classification\n")

prompt = """
Question: "What is inheritance?"
Category: Conceptual Question

Question: "Extra bracket causes compilation error"
Category: Syntax Error

Question: "Infinite loop issue"
Category: Logic Error

Now classify the following question:
Question: "How can I optimize this algorithm?"
"""

print(prompt)
# Expected Output: Optimization
```

**Few-Shot Prompt:**

Question: "What is inheritance?"

Category: Conceptual Question

Question: "Extra bracket causes error"

Category: Syntax Error

Question: "Infinite loop issue"

Category: Logic Error

**Now classify:**

Question: "How can I optimize this algorithm?"

Explanation:

Multiple examples clarify category distinctions.

The model better understands technical intent.

Provides the highest classification accuracy.

## Task 10: Zero-Shot Prompting – Social Media Post Categorization

Prompt:

Classify the following social media post into:

Promotion, Complaint, Appreciation, Inquiry.

Post: "The delivery was delayed again."

```
print("\nTask 10: Zero-Shot Prompting Social Media Post Categorization\n")

prompt = """
Classify the following social media post into:
Promotion, Complaint, Appreciation, Inquiry.

Post: "The delivery was delayed again."
"""

print(prompt)
# Expected Output: Complaint
```

#### 4. SOCIAL MEDIA POST CATEGORIZATION

**Zero-Shot Prompt:**

Classify the following post into:  
Promotion, Complaint, Appreciation, Inquiry.

Post: "The delivery was delayed again."

Explanation:

The model classifies based on sentiment alone.

No prior examples are provided.

Works well for clear complaints.

#### Task 11: One-Shot Prompting – Social Media Post Categorization

Prompt:

Example:

Post: "Love the new update!"

Category: Appreciation

Now classify the following post:

Post: "Any discounts available?"

```
print("\nTask 11: One-Shot Prompting Social Media Post Categorization\n")

prompt = """
Example:
Post: "Love the new update!"
Category: Appreciation

Now classify the following post:
Post: "Any discounts available?"
"""

print(prompt)
# Expected Output: Inquiry
```

**One-Shot Prompt:**

**Example:**

**Post:** "Love the new update!"

**Category:** Appreciation

**Now classify:**

**Post:** "Any discounts available?"

**Explanation:**

One-shot prompting improves sentiment recognition.

Helps the model follow a consistent format.

Better accuracy than zero-shot.

## **Task 12: Few-Shot Prompting – Social Media Post Categorization**

Prompt:

Post: "Great customer support experience"

Category: Appreciation

Post: "Worst app experience ever"

Category: Complaint

Post: "Limited time offer! Buy now"

Category: Promotion

Now classify the following post:

Post: "When will the next update be released?"

```
print("\nTask 12: Few-Shot Prompting Social Media Post Categorization\n")
```

```
prompt = """
```

Post: "Great customer support experience"

Category: Appreciation

Post: "Worst app experience ever"

Category: Complaint

Post: "Limited time offer! Buy now"

Category: Promotion

Now classify the following post:

Post: "When will the next update be released?"

```
"""
```

```
print(prompt)
```

```
# Expected Output: Inquiry
```

**Few-Shot Prompt:**

Post: "Great customer support experience"

Category: Appreciation

Post: "Worst app experience ever"

Category: Complaint

Post: "Limited time offer! Buy now"

Category: Promotion

**Now classify:**

Post: "When will the next update be released?"

Explanation:

Few-shot prompting handles informal language better.

Reduces confusion between similar categories.

Most effective method for social media data.

