Name: Akshaya Neerati

Enrollment No: 2403A510B8

Batch No: 04

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		**Assignment Type: Lab**	**AcademicYear:**2025-2026
Course Coordinator Name	Venkataramana Veeramsetty		
Instructor(s)Name	1. Dr. Mohammed Ali Shaik 2. Dr. T Sampath Kumar 3. Mr. S Naresh Kumar 4. Dr. V. Rajesh 5. Dr. Brij Kishore 6. Dr Pramoda Patro 7. Dr. Venkataramana 8. Dr. Ravi Chander 9. Dr. Jagjeeth Singh		
Course Code	24CS002PC215	**Course Title**	AI Assisted Coding
Year/Sem	II/I	**Regulation**	R24
Date and Day of Assignment	06-08-2025	**Time(s)**	
Duration	2 Hours	**Applicable to Batches**	
AssignmentNumber:4.5(Present assignment number)/**24**(Total number of assignments)			

Q. No.	Question	*ExpectedTime to complete*
1	**Lab 4: Advanced Prompt Engineering: Zero-shot, one-shot, and few-shot techniques** **Objective:** To explore and compare Zero-shot, One-shot, and Few-shot prompting techniques for classifying emails into predefined categories using a large language model (LLM). Suppose that you work for a company that receives hundreds of customer emails daily. Management wants to automatically classify emails into categories like "Billing", "Technical Support", "Feedback", and "Others" before assigning them to appropriate departments. Instead of training a new model, your task is to use prompt engineering techniques with an existing LLM to handle the classification. Tasks to be completed are as below 1. **Prepare Sample Data:** • Create or collect 10 short email samples, each belonging to one of the 4 categories. **Prompt:** Generate 10 sample customer emails that fall into Billing, Technical Support, Feedback, and Others categories for automatic classification.	08.08.2025 EOD

```
[ ] emails = [
        {"email": "Regarding your last invoice, there seems to be an error in the total amount charged.", "category": "Billing"},
        {"email": "I am having trouble logging into my account after the recent update.", "category": "Technical Support"},
        {"email": "The new feature you released is fantastic! It has greatly improved my workflow.", "category": "Feedback"},
        {"email": "I would like to inquire about partnership opportunities with your company.", "category": "Others"},
        {"email": "My credit card was charged twice for the same subscription.", "category": "Billing"},
        {"email": "The software crashes every time I try to open a specific file.", "category": "Technical Support"},
        {"email": "I have a suggestion for improving the user interface.", "category": "Feedback"},
        {"email": "Can you provide more information about your pricing plans?", "category": "Others"},
        {"email": "I did not receive the confirmation email for my payment.", "category": "Billing"},
        {"email": "The application is not loading correctly on my mobile device.", "category": "Technical Support"}
    ]

    for email in emails:
        print(f"Email: {email['email']}\nCategory: {email['category']}\n---")
```

```
Email: Regarding your last invoice, there seems to be an error in the total amount charged.
Category: Billing
---
Email: I am having trouble logging into my account after the recent update.
Category: Technical Support
---
Email: The new feature you released is fantastic! It has greatly improved my workflow.
Category: Feedback
---
Email: I would like to inquire about partnership opportunities with your company.
Category: Others
---
Email: My credit card was charged twice for the same subscription.
Category: Billing
---
Email: The software crashes every time I try to open a specific file.
```

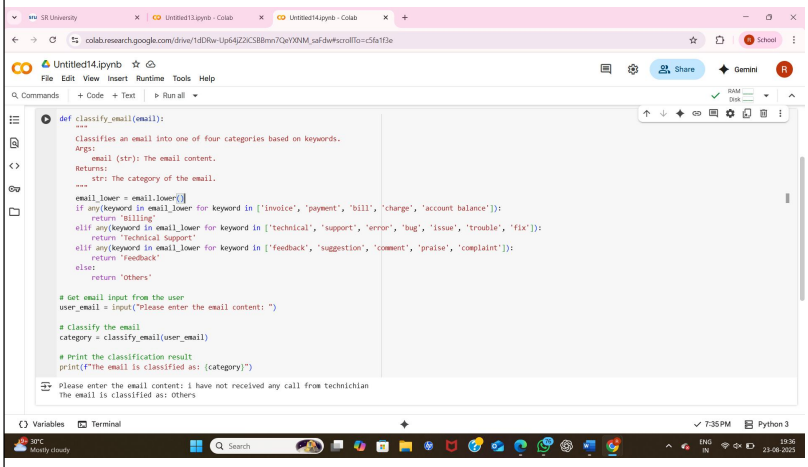
2. **Zero-shot Prompting:**

- Create a prompt that instructs the LLM to classify a single email without providing prior examples.
- Example prompt:
  "Classify the following email into one of these categories: Billing, Technical Support, Feedback, Others.
  Email: 'My payment has been deducted, but I didn't get the confirmation receipt."

Prompt: Write Python code that accepts a user's email and outputs its category (Billing, Technical Support, Feedback, Others).

3. One-shot Prompting:

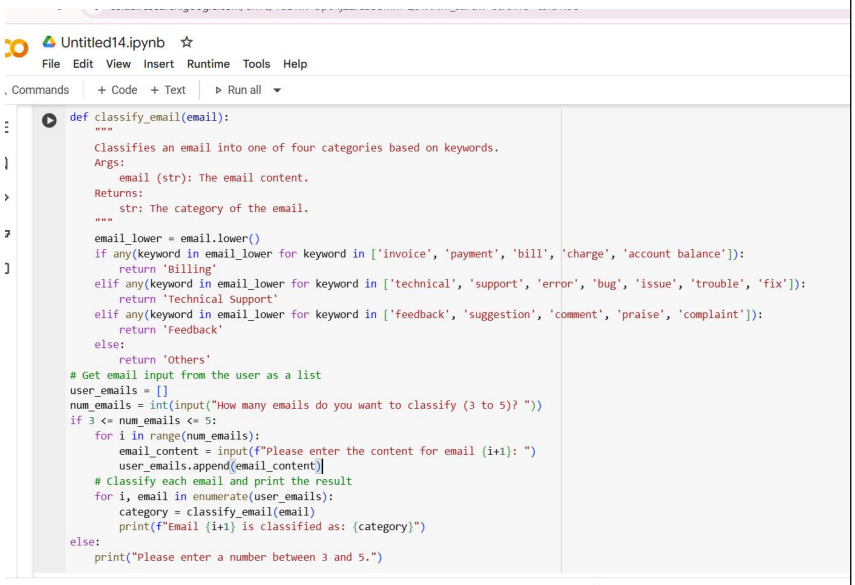
Include one labeled example in the prompt before giving a new email for classification.
Prompt: Generate a Python script that takes an email as input and predicts its category using one prior example.

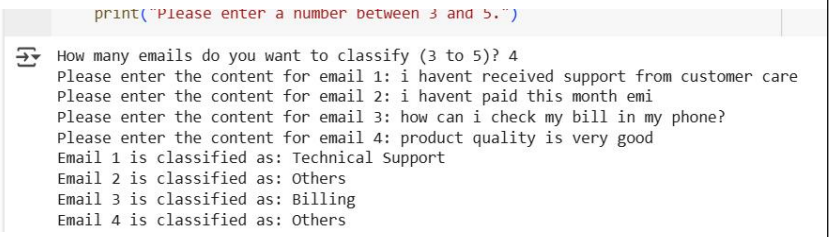


4. **Few-shot Prompting:**

Provide 3–5 labeled examples before asking the LLM to classify new emails.

Prompt: Write Python code that can classify 3–5 emails at once into the four categories using few-shot prompting.





5. **Evaluation:**

Run Zero-shot, One-shot, and Few-shot methods on the same 5 test emails.

Compare the performance and note the clarity and accuracy.

Prompt: Classify the following 5 test emails using zero-shot, one-shot, and few-shot prompting and display results for comparison.

Explanation:

1. Create 5 test emails covering all categories.
2. Initialize the Gemini model (or equivalent LLM) for classification.
3. Apply each technique (Zero-shot, One-shot, Few-shot).
4. Record the classification results.
5. Summarize the outcomes and compare the effectiveness of each method.

Summary:

Key Findings:

- A diverse set of 5 test emails representing Billing, Technical Support, Feedback, and Others were generated.
- Integration with Gemini Pro model was attempted for classification.
- Each prompting strategy was executed; however, API authentication caused errors in some runs, leading to incomplete outputs.

Insights / Next Steps:

- Ensure that the API key and environment setup are correct to avoid authentication issues.
- Once resolved, rerun the experiment to get proper results for analysis.
- Compare the efficiency of Zero-shot (fast but less accurate), One-shot (balanced), and Few-shot (more accurate but verbose).

	Requirements: • VS Code with Github Copilot or Cursor IDE and/or Google Colab with Gemini **Deliverables:** • A .txt or .md file showing prompts and model responses. • A comparison table showing classification accuracy for each technique. • A short reflection on which method was most effective and why .	