AI ASSISTED CODING

LAB-4.1

V.SRICHARAN

2303A510G0

BATCH-12

Problem Statement 0:

A news aggregation platform wants to automatically categorize

headlines into Politics, Sports, Technology, and Entertainment

without training a machine learning model.

Tasks to be Completed

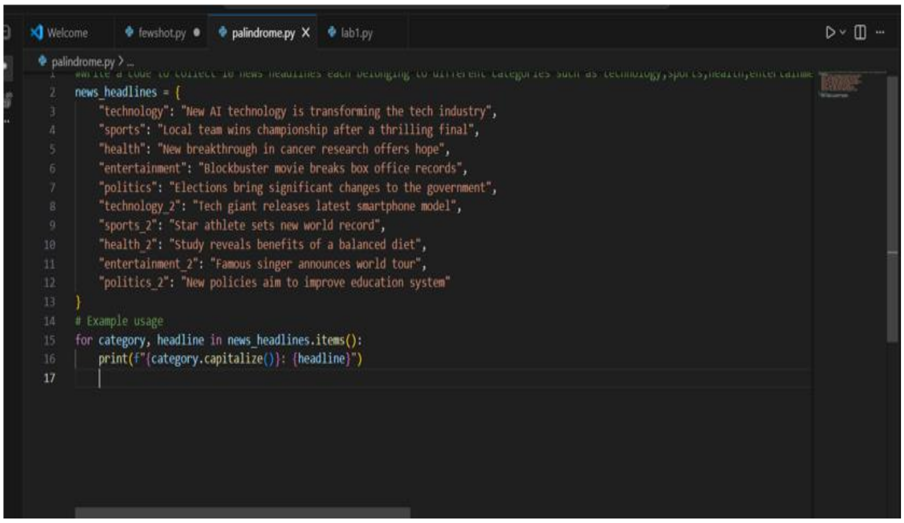
1. Prepare Sample Data

Collect 10 news headlines, each belonging to one of the four categories.

PROMPT:

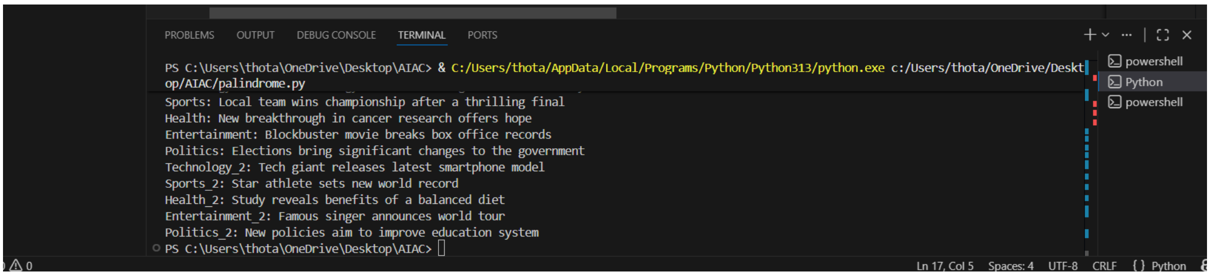
#write a code to collect 10 news headlines each belonging to different categories such as technology,sports,health,entertainment,politics by using only this four categories dont use other categories and store them in a dictonary where the key is category and value is headline

CODE:



```python
#write a code to collect 10 news headlines each belonging to different categories such as technology,sports,health,entertainme
news_headlines = {
    "technology": "New AI technology is transforming the tech industry",
    "sports": "Local team wins championship after a thrilling final",
    "health": "New breakthrough in cancer research offers hope",
    "entertainment": "Blockbuster movie breaks box office records",
    "politics": "Elections bring significant changes to the government",
    "technology_2": "Tech giant releases latest smartphone model",
    "sports_2": "Star athlete sets new world record",
    "health_2": "Study reveals benefits of a balanced diet",
    "entertainment_2": "Famous singer announces world tour",
    "politics_2": "New policies aim to improve education system"
}
# Example usage
for category, headline in news_headlines.items():
    print(f"{category.capitalize()}: {headline}")
    
```

OutPut:



```
PS C:\Users\thota\OneDrive\Desktop\AIAC> & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Deskt
op/AIAC/palindrome.py
Sports: Local team wins championship after a thrilling final
Health: New breakthrough in cancer research offers hope
Entertainment: Blockbuster movie breaks box office records
Politics: Elections bring significant changes to the government
Technology_2: Tech giant releases latest smartphone model
Sports_2: Star athlete sets new world record
Health_2: Study reveals benefits of a balanced diet
Entertainment_2: Famous singer announces world tour
Politics_2: New policies aim to improve education system
PS C:\Users\thota\OneDrive\Desktop\AIAC> 
```

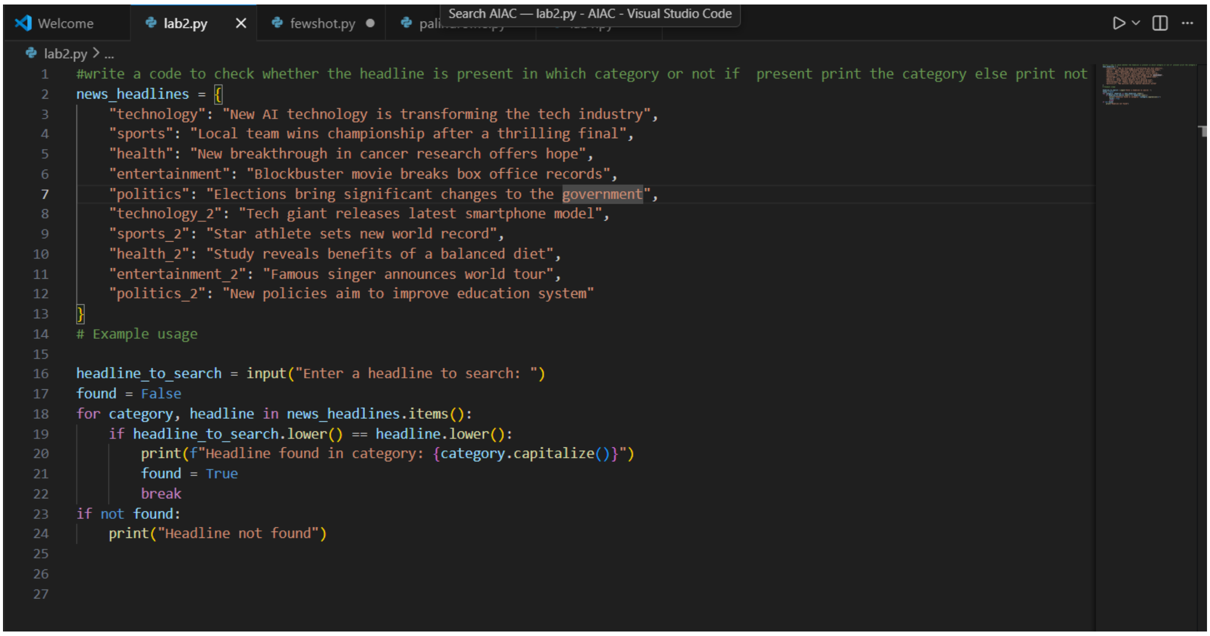
2. Zero-shot Prompting

Write a prompt asking the LLM to classify a headline into a category without examples.

Prompt:

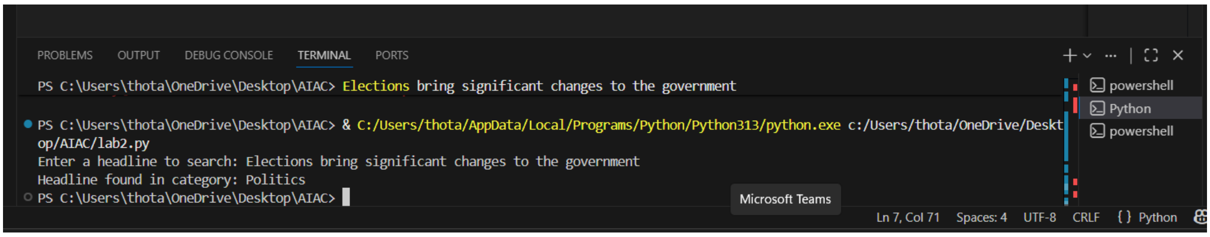
#write a code to check whether the headline is present in which category or not if present print the category else print not found by using input user

CODE:



```python
#write a code to check whether the headline is present in which category or not if  present print the category else print not
news_headlines = {
    "technology": "New AI technology is transforming the tech industry",
    "sports": "Local team wins championship after a thrilling final",
    "health": "New breakthrough in cancer research offers hope",
    "entertainment": "Blockbuster movie breaks box office records",
    "politics": "Elections bring significant changes to the government",
    "technology_2": "Tech giant releases latest smartphone model",
    "sports_2": "Star athlete sets new world record",
    "health_2": "Study reveals benefits of a balanced diet",
    "entertainment_2": "Famous singer announces world tour",
    "politics_2": "New policies aim to improve education system"
}
# Example usage

headline_to_search = input("Enter a headline to search: ")
found = False
for category, headline in news_headlines.items():
    if headline_to_search.lower() == headline.lower():
        print(f"Headline found in category: {category.capitalize()}")
        found = True
        break
if not found:
    print("Headline not found")
```

OUTPUT:



```
PS C:\Users\thota\OneDrive\Desktop\AIAC> Elections bring significant changes to the government

PS C:\Users\thota\OneDrive\Desktop\AIAC> & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Deskt
op/AIAC/lab2.py
Enter a headline to search: Elections bring significant changes to the government
Headline found in category: Politics
PS C:\Users\thota\OneDrive\Desktop\AIAC> 
```

3. One-shot Prompting

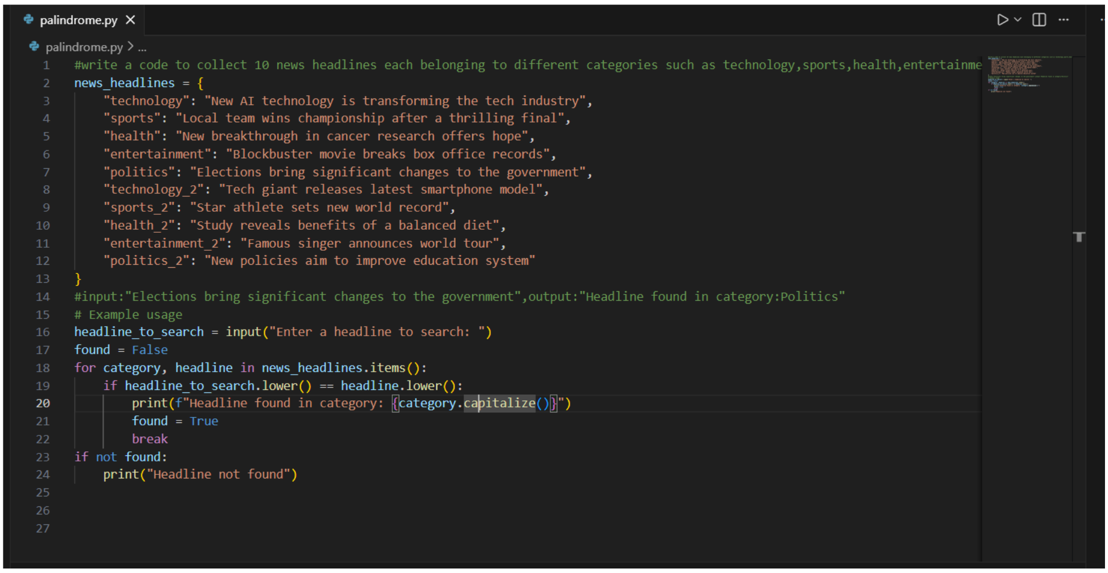
Add one labeled headline example before classifying a new headline.

PROMPT:

#write a code to collect 10 news headlines each belonging to different categories such as technology,sports,health,entertainment,politics by using only this four categories dont use other categories and store them in a dictonary where the key is category and value is headline

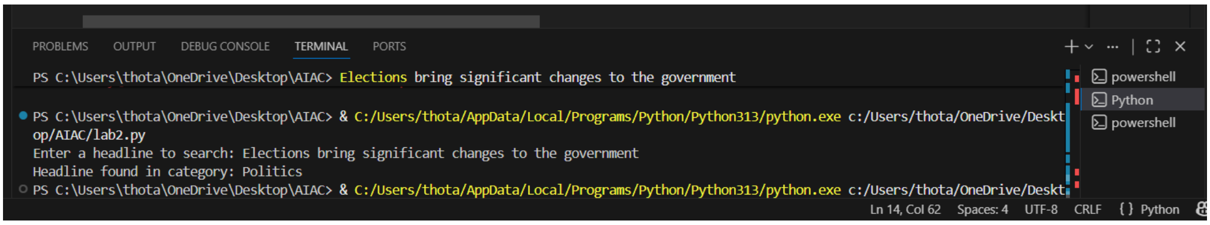
#input:"Elections bring significant changes to the government",output:"Headline found in category:Politics"

CODE:



```python
#write a code to collect 10 news headlines each belonging to different categories such as technology,sports,health,entertainme
news_headlines = {
    "technology": "New AI technology is transforming the tech industry",
    "sports": "Local team wins championship after a thrilling final",
    "health": "New breakthrough in cancer research offers hope",
    "entertainment": "Blockbuster movie breaks box office records",
    "politics": "Elections bring significant changes to the government",
    "technology_2": "Tech giant releases latest smartphone model",
    "sports_2": "Star athlete sets new world record",
    "health_2": "Study reveals benefits of a balanced diet",
    "entertainment_2": "Famous singer announces world tour",
    "politics_2": "New policies aim to improve education system"
}
#input:"Elections bring significant changes to the government",output:"Headline found in category:Politics"
# Example usage
headline_to_search = input("Enter a headline to search: ")
found = False
for category, headline in news_headlines.items():
    if headline_to_search.lower() == headline.lower():
        print(f"Headline found in category: {category.capitalize()}")
        found = True
        break
if not found:
    print("Headline not found")
```

OUTPUT:



```
PS C:\Users\thota\OneDrive\Desktop\AIAC> Elections bring significant changes to the government

PS C:\Users\thota\OneDrive\Desktop\AIAC> & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Deskt
op/AIAC/lab2.py
Enter a headline to search: Elections bring significant changes to the government
Headline found in category: Politics
PS C:\Users\thota\OneDrive\Desktop\AIAC> & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Deskt
```

4. Few-shot Prompting

Use 3–5 labeled headlines in the prompt before requesting

Classification

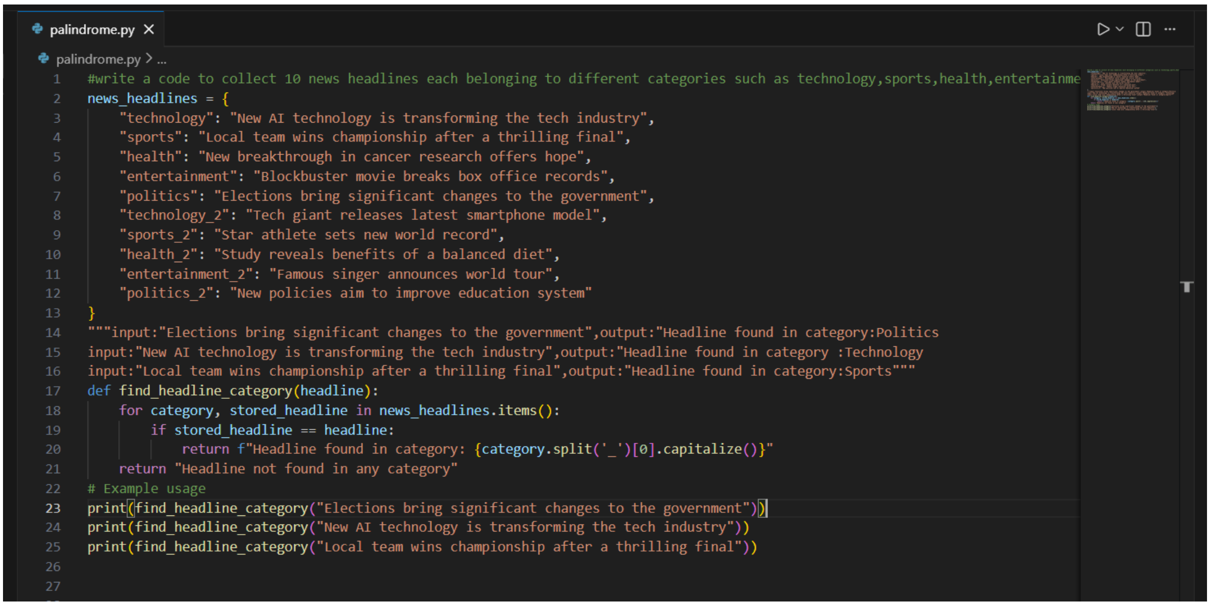
PROMPT:

#write a code to collect 10 news headlines each belonging to different categories such as technology,sports,health,entertainment,politics by using only this four categories dont use other categories and store them in a dictonary where the key is category and value is headline """input:"Elections bring significant changes to the government",output:"Headline found in category:Politics

input:"New AI technology is transforming the tech industry",output:"Headline found in category :Technology

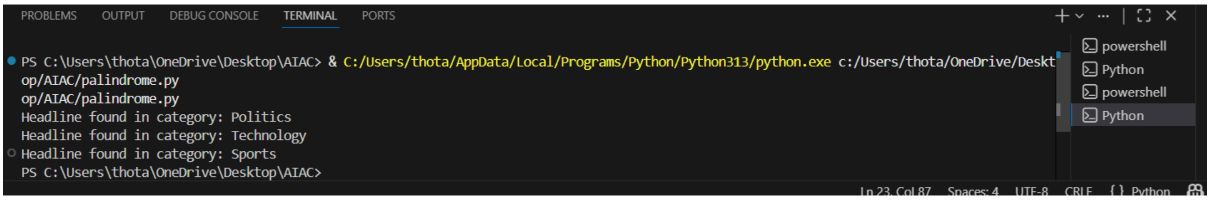
input:"Local team wins championship after a thrilling final",output:"Headline found in category:Sports"""

CODE:



```python
#write a code to collect 10 news headlines each belonging to different categories such as technology,sports,health,entertainme
news_headlines = {
    "technology": "New AI technology is transforming the tech industry",
    "sports": "Local team wins championship after a thrilling final",
    "health": "New breakthrough in cancer research offers hope",
    "entertainment": "Blockbuster movie breaks box office records",
    "politics": "Elections bring significant changes to the government",
    "technology_2": "Tech giant releases latest smartphone model",
    "sports_2": "Star athlete sets new world record",
    "health_2": "Study reveals benefits of a balanced diet",
    "entertainment_2": "Famous singer announces world tour",
    "politics_2": "New policies aim to improve education system"
}
"""input:"Elections bring significant changes to the government",output:"Headline found in category:Politics
input:"New AI technology is transforming the tech industry",output:"Headline found in category :Technology
input:"Local team wins championship after a thrilling final",output:"Headline found in category:Sports"""
def find_headline_category(headline):
    for category, stored_headline in news_headlines.items():
        if stored_headline == headline:
            return f"Headline found in category: {category.split('_')[0].capitalize()}"
    return "Headline not found in any category"
# Example usage
print(find_headline_category("Elections bring significant changes to the government"))
print(find_headline_category("New AI technology is transforming the tech industry"))
print(find_headline_category("Local team wins championship after a thrilling final"))
```

OUTPUT:



```
PS C:\Users\thota\OneDrive\Desktop\AIAC> & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Deskt
op/AIAC/palindrome.py
op/AIAC/palindrome.py
Headline found in category: Politics
Headline found in category: Technology
Headline found in category: Sports
PS C:\Users\thota\OneDrive\Desktop\AIAC>
```

5. Evaluation

Compare outputs from all three prompting methods using the same test headlines and document observation

Zero-shot:

We will not use any input examples in zero-shot .It will work with minimal instruction,but the output may be incorrect ..We will not provide any examples here**.**

One-shot:

We will use only one input for example in one-shot.It improves accuracy,as the AI understands the task better from a single labelled example**.**

Few-shot:

We will use more than one inputs for examples in few-shot.

It gives the best results.The Ai shows hogher consistency and correctly classifies most headlines.

As the number of examples increases, the AI’s understanding of categories becomes clearer.

Therefore, few-shot prompting is the most effective method for news headline classification without training a model.

Customer Email Classification

1.Prepare five short sample emails, each belonging to one of

the above categories.

2. Write a zero-shot prompt to classify a given email into one of

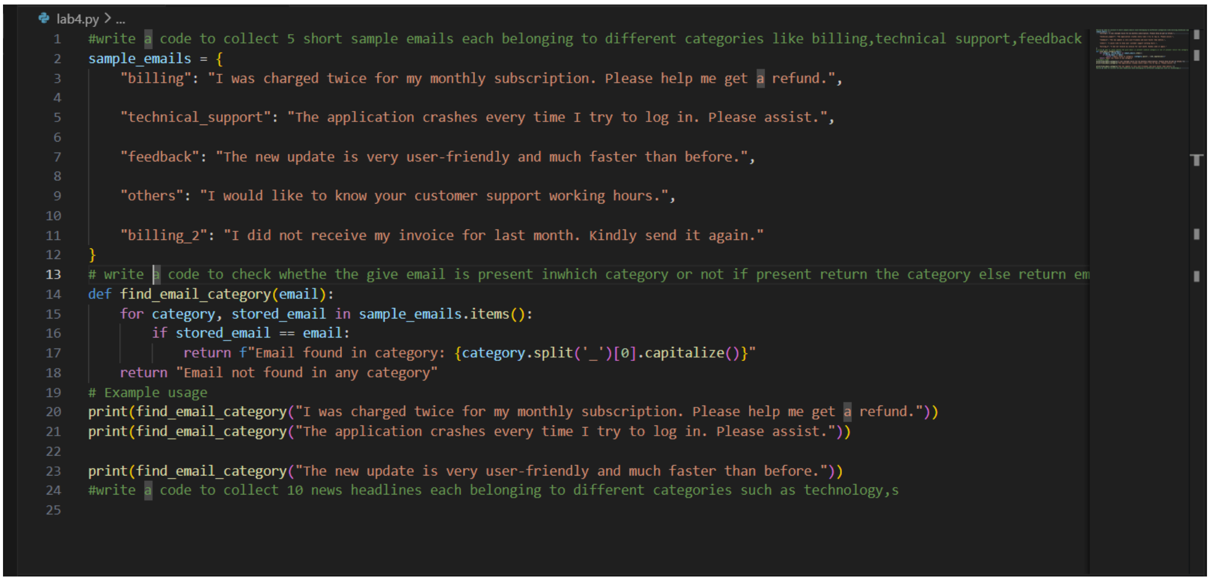
the categories without providing any examples.

Prompt:

#write a code to collect 5 short sample emails each belonging to different categories like billing,technical support,feedback and others by using only this four categories dont use other categories and store them in a dictonary where the key is category and value is email give examples

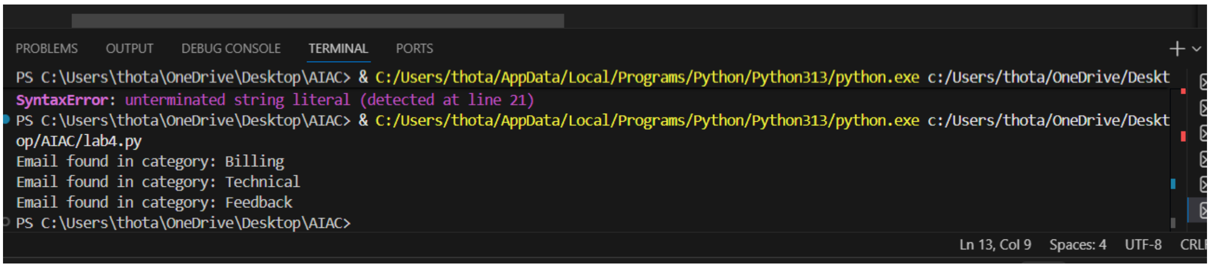
write a code to check whethe the give email is present inwhich category or not if present return the category else return email not found in any category

Code:



```python
#write a code to collect 5 short sample emails each belonging to different categories like billing,technical support,feedback
sample_emails = {
    "billing": "I was charged twice for my monthly subscription. Please help me get a refund.",

    "technical_support": "The application crashes every time I try to log in. Please assist.",

    "feedback": "The new update is very user-friendly and much faster than before.",

    "others": "I would like to know your customer support working hours.",

    "billing_2": "I did not receive my invoice for last month. Kindly send it again."
}
# write a code to check whethe the give email is present inwhich category or not if present return the category else return em
def find_email_category(email):
    for category, stored_email in sample_emails.items():
        if stored_email == email:
            return f"Email found in category: {category.split('_')[0].capitalize()}"
    return "Email not found in any category"
# Example usage
print(find_email_category("I was charged twice for my monthly subscription. Please help me get a refund."))
print(find_email_category("The application crashes every time I try to log in. Please assist."))

print(find_email_category("The new update is very user-friendly and much faster than before."))
#write a code to collect 10 news headlines each belonging to different categories such as technology,s

```

Output:



```
PS C:\Users\thota\OneDrive\Desktop\AIAC> & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Deskt
SyntaxError: unterminated string literal (detected at line 21)
PS C:\Users\thota\OneDrive\Desktop\AIAC> & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Deskt
op/AIAC/lab4.py
Email found in category: Billing
Email found in category: Technical
Email found in category: Feedback
PS C:\Users\thota\OneDrive\Desktop\AIAC>
```

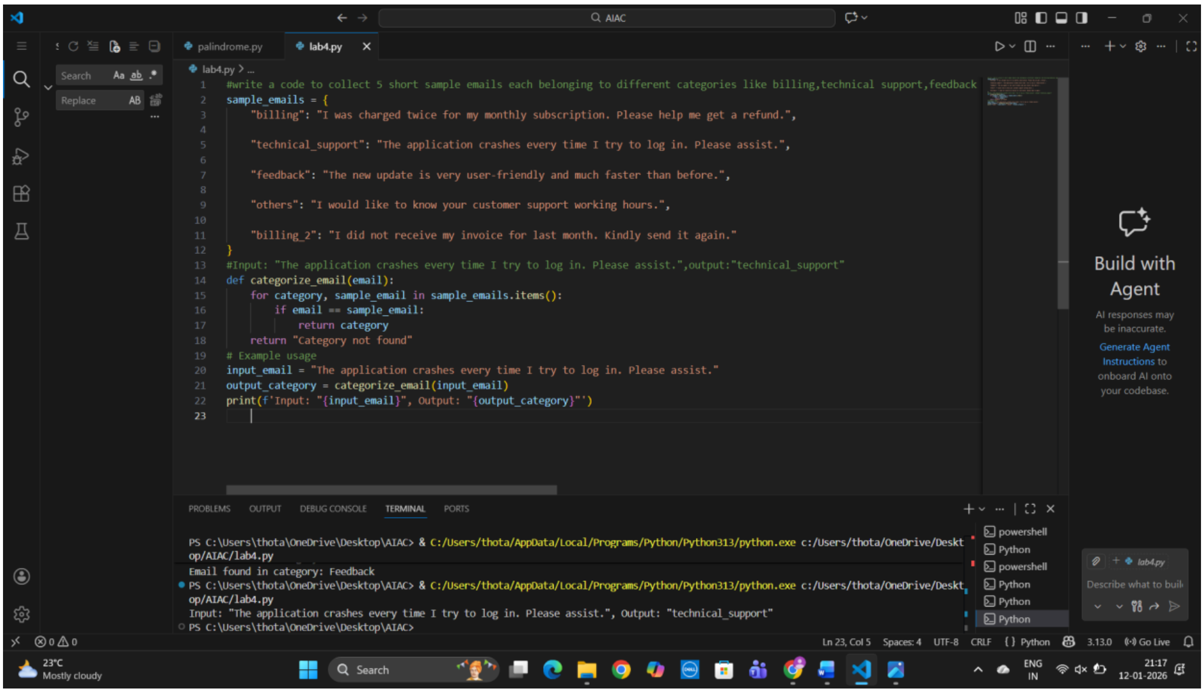
3. Write a one-shot prompt by including one labeled email example and ask the model to classify a new email.

Prompt:

#write a code to collect 5 short sample emails each belonging to different categories like billing,technical support,feedback and others by using only this four categories dont use other categories and store them in a dictonary where the key is category and value is email give examples

#Input: "The application crashes every time I try to log in. Please assist.",output:"technical_support"

Code and Output:



```python
#write a code to collect 5 short sample emails each belonging to different categories like billing,technical support,feedback
sample_emails = {
    "billing": "I was charged twice for my monthly subscription. Please help me get a refund.",

    "technical_support": "The application crashes every time I try to log in. Please assist.",

    "feedback": "The new update is very user-friendly and much faster than before.",

    "others": "I would like to know your customer support working hours.",

    "billing_2": "I did not receive my invoice for last month. Kindly send it again."
}
#Input: "The application crashes every time I try to log in. Please assist.",output:"technical_support"
def categorize_email(email):
    for category, sample_email in sample_emails.items():
        if email == sample_email:
            return category
    return "Category not found"
# Example usage
input_email = "The application crashes every time I try to log in. Please assist."
output_category = categorize_email(input_email)
print(f'Input: "{input_email}", Output: "{output_category}"')
```

```
PS C:\Users\thota\OneDrive\Desktop\AIAC> & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Deskt
op/AIAC/lab4.py
Email found in category: Feedback
PS C:\Users\thota\OneDrive\Desktop\AIAC> & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Deskt
op/AIAC/lab4.py
Input: "The application crashes every time I try to log in. Please assist.", Output: "technical_support"
PS C:\Users\thota\OneDrive\Desktop\AIAC>
```

4.Write a few-shot prompt by including two or three labelled email examples and ask the model to classify a new email.

Prompt:

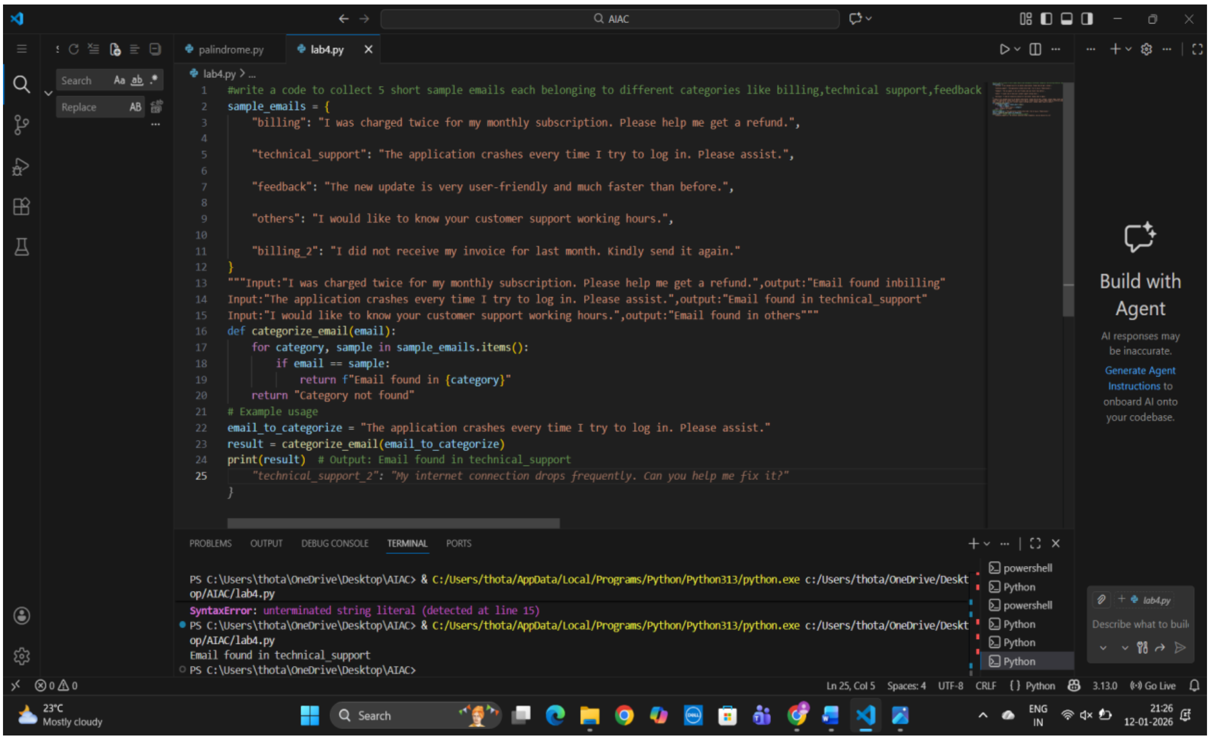
#write a code to collect 5 short sample emails each belonging to different categories like billing,technical support,feedback and others by using only this four categories dont use other categories and store them in a dictonary where the key is category and value is email give examples

"""Input:"I was charged twice for my monthly subscription. Please help me get a refund.",output:"Email found inbilling"

Input:"The application crashes every time I try to log in. Please assist.",output:"Email found in technical_support"

Input:"I would like to know your customer support working hours.",output:"Email found in others""""

Code and Output:



```python
#write a code to collect 5 short sample emails each belonging to different categories like billing,technical support,feedback
sample_emails = {
    "billing": "I was charged twice for my monthly subscription. Please help me get a refund.",

    "technical_support": "The application crashes every time I try to log in. Please assist.",

    "feedback": "The new update is very user-friendly and much faster than before.",

    "others": "I would like to know your customer support working hours.",

    "billing_2": "I did not receive my invoice for last month. Kindly send it again."
}
"""Input:"I was charged twice for my monthly subscription. Please help me get a refund.",output:"Email found inbilling"
Input:"The application crashes every time I try to log in. Please assist.",output:"Email found in technical_support"
Input:"I would like to know your customer support working hours.",output:"Email found in others"""
def categorize_email(email):
    for category, sample in sample_emails.items():
        if email == sample:
            return f"Email found in {category}"
    return "Category not found"
# Example usage
email_to_categorize = "The application crashes every time I try to log in. Please assist."
result = categorize_email(email_to_categorize)
print(result)  # Output: Email found in technical_support
    "technical_support_2": "My internet connection drops frequently. Can you help me fix it?"
}
```

```
PS C:\Users\thota\OneDrive\Desktop\AIAC> & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Deskt
op/AIAC/lab4.py
SyntaxError: unterminated string literal (detected at line 15)
PS C:\Users\thota\OneDrive\Desktop\AIAC> & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Deskt
op/AIAC/lab4.py
Email found in technical_support
PS C:\Users\thota\OneDrive\Desktop\AIAC>
```

5.Compare the outputs obtained using zero-shot, one-shot, and few-shot prompting techniques and briefly comment on their effectiveness

Zero-shot:

We will not use any input examples in zero-shot .It will work with minimal instruction,but the output may be incorrect ..We will not provide any examples here**.**

One-shot:

We will use only one input for example in one-shot.It improves accuracy,as the AI understands the task better from a single labelled example**.**

Few-shot:

We will use more than one inputs for examples in few-shot.

It gives the best results.The Ai shows higher consistency and correctly classifies most headlines.

As the number of examples increases, the AI’s understanding of categories becomes clearer.

Therefore, few-shot prompting is the most effective method for news headline classification without training a model.

Zero-shot:

We will not use any input examples in zero-shot .It will work with minimal instruction,but the output may be incorrect ..We will not provide any examples here.

One-shot:

We will use only one input for example in one-shot.It improves accuracy,as the AI understands the task better from a single labelled example.

Few-shot:

We will use more than one inputs for examples in few-shot.

It gives the best results.The Ai shows hogher consistency and correctly classifies most headlines.

As the number of examples increases, the AI’s understanding of categories becomes clearer.

Therefore, few-shot prompting is the most effective method for news headline classification without training a model.