NAME: Kolipaka Sanjana

ROLL.NO:2303a52139

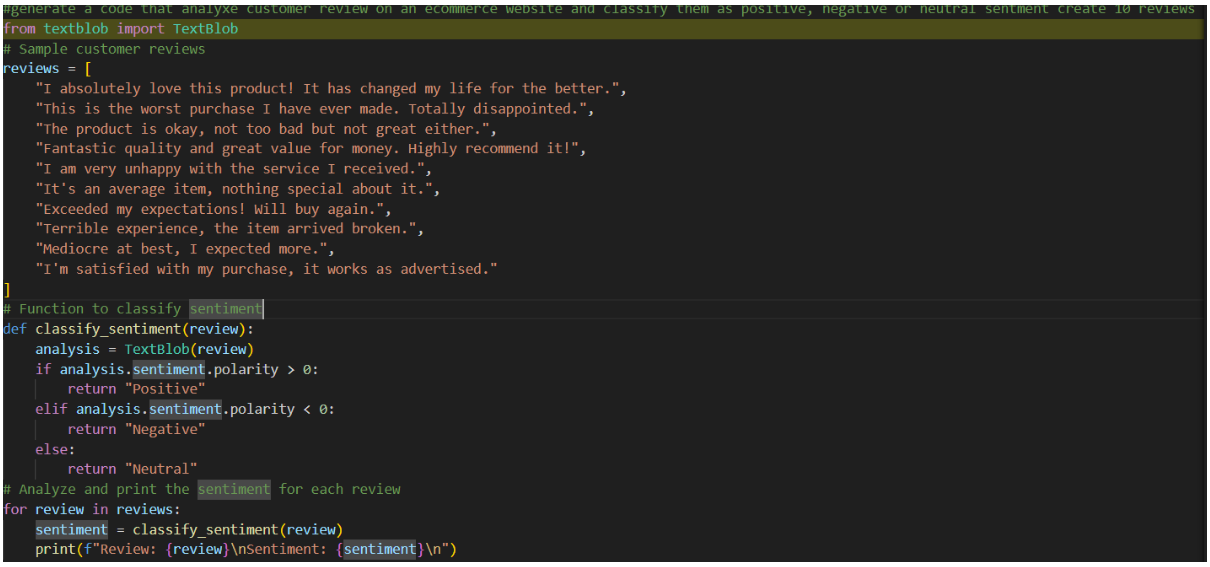
BATCH:41

ASSIGNMENT 4.4

TASK1:

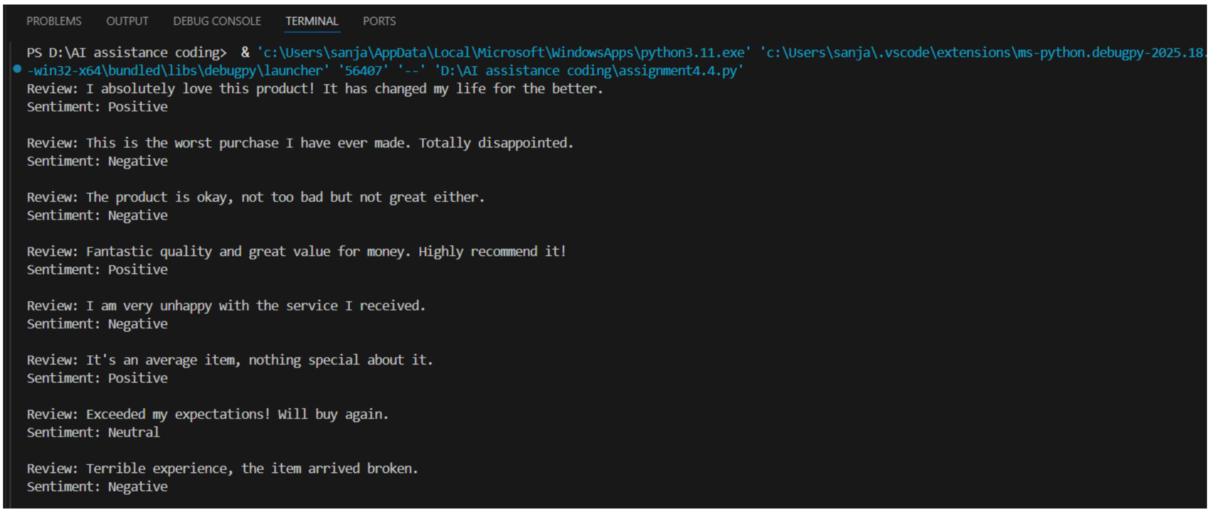
ZERO SHOT PROMPT: generate a code that analyxe customer review on an ecommerce website and classify them as positive, negative or neutral sentment create 10 reviews inside the code itself and print the classification result for each review.

CODE:



```python
#generate a code that analyxe customer review on an ecommerce website and classify them as positive, negative or neutral sentment create 10 reviews
from textblob import TextBlob
# Sample customer reviews
reviews = [
    "I absolutely love this product! It has changed my life for the better.",
    "This is the worst purchase I have ever made. Totally disappointed.",
    "The product is okay, not too bad but not great either.",
    "Fantastic quality and great value for money. Highly recommend it!",
    "I am very unhappy with the service I received.",
    "It's an average item, nothing special about it.",
    "Exceeded my expectations! Will buy again.",
    "Terrible experience, the item arrived broken.",
    "Mediocre at best, I expected more.",
    "I'm satisfied with my purchase, it works as advertised."
]
# Function to classify sentiment
def classify_sentiment(review):
    analysis = TextBlob(review)
    if analysis.sentiment.polarity > 0:
        return "Positive"
    elif analysis.sentiment.polarity < 0:
        return "Negative"
    else:
        return "Neutral"
# Analyze and print the sentiment for each review
for review in reviews:
    sentiment = classify_sentiment(review)
    print(f"Review: {review}\nSentiment: {sentiment}\n")
```

OUTPUT:



```
PS D:\AI assistance coding>  & 'c:\Users\sanja\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\sanja\.vscode\extensions\ms-python.debugpy-2025.18
-win32-x64\bundled\libs\debugpy\launcher' '56407' '--' 'D:\AI assistance coding\assignment4.4.py'
Review: I absolutely love this product! It has changed my life for the better.
Sentiment: Positive

Review: This is the worst purchase I have ever made. Totally disappointed.
Sentiment: Negative

Review: The product is okay, not too bad but not great either.
Sentiment: Negative

Review: Fantastic quality and great value for money. Highly recommend it!
Sentiment: Positive

Review: I am very unhappy with the service I received.
Sentiment: Negative

Review: It's an average item, nothing special about it.
Sentiment: Positive

Review: Exceeded my expectations! Will buy again.
Sentiment: Neutral

Review: Terrible experience, the item arrived broken.
Sentiment: Negative
```

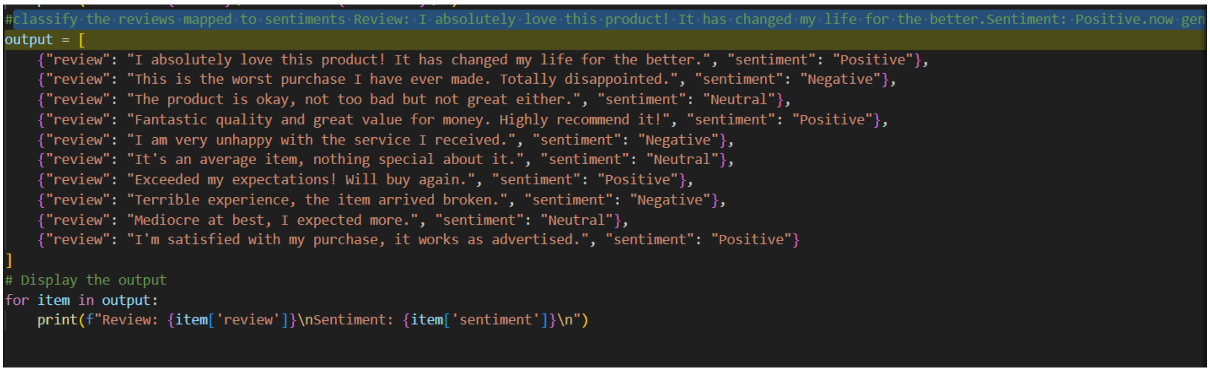
ONE- SHOT: classify the reviews mapped to sentiments

Review: I absolutely love this product! It has changed my life for the better.

Sentiment: Positive.

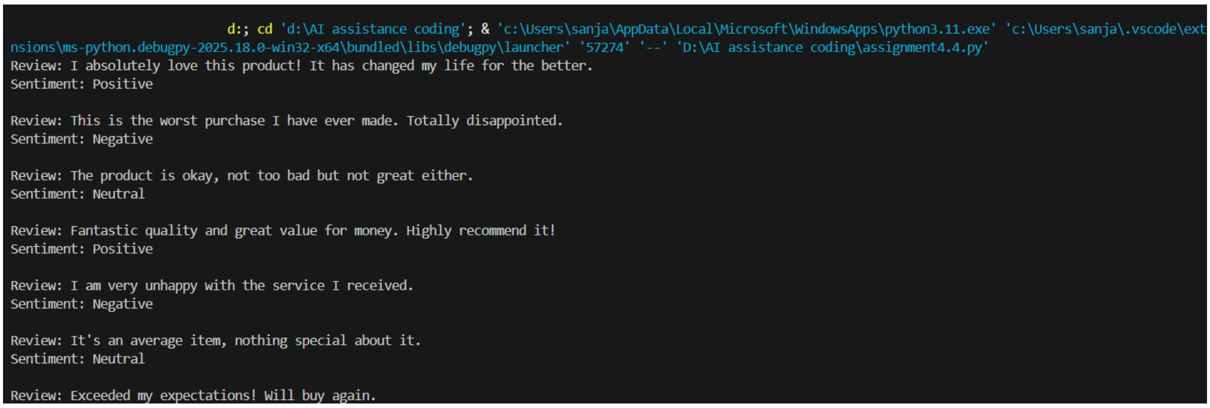
now generate python code to display the output for all reviews in the same format as above.i want data in code itself not in output.

CODE:



```
#classify the reviews mapped to sentiments Review: I absolutely love this product! It has changed my life for the better.Sentiment: Positive.now gen
output = [
    {"review": "I absolutely love this product! It has changed my life for the better.", "sentiment": "Positive"},
    {"review": "This is the worst purchase I have ever made. Totally disappointed.", "sentiment": "Negative"},
    {"review": "The product is okay, not too bad but not great either.", "sentiment": "Neutral"},
    {"review": "Fantastic quality and great value for money. Highly recommend it!", "sentiment": "Positive"},
    {"review": "I am very unhappy with the service I received.", "sentiment": "Negative"},
    {"review": "It's an average item, nothing special about it.", "sentiment": "Neutral"},
    {"review": "Exceeded my expectations! Will buy again.", "sentiment": "Positive"},
    {"review": "Terrible experience, the item arrived broken.", "sentiment": "Negative"},
    {"review": "Mediocre at best, I expected more.", "sentiment": "Neutral"},
    {"review": "I'm satisfied with my purchase, it works as advertised.", "sentiment": "Positive"}
]
# Display the output
for item in output:
    print(f"Review: {item['review']}\nSentiment: {item['sentiment']}\n")
```

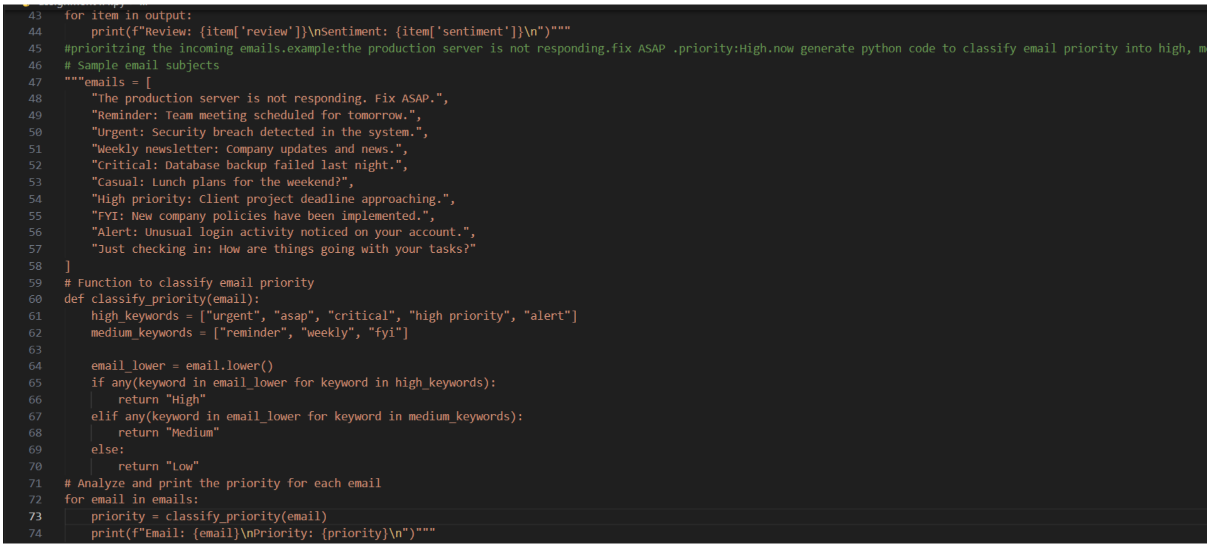
OUTPUT:



```
                              d:; cd 'd:\AI assistance coding'; & 'c:\Users\sanja\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\sanja\.vscode\ext
nsions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '57274' '--' 'D:\AI assistance coding\assignment4.4.py'
Review: I absolutely love this product! It has changed my life for the better.
Sentiment: Positive

Review: This is the worst purchase I have ever made. Totally disappointed.
Sentiment: Negative

Review: The product is okay, not too bad but not great either.
Sentiment: Neutral

Review: Fantastic quality and great value for money. Highly recommend it!
Sentiment: Positive

Review: I am very unhappy with the service I received.
Sentiment: Negative

Review: It's an average item, nothing special about it.
Sentiment: Neutral

Review: Exceeded my expectations! Will buy again.
```

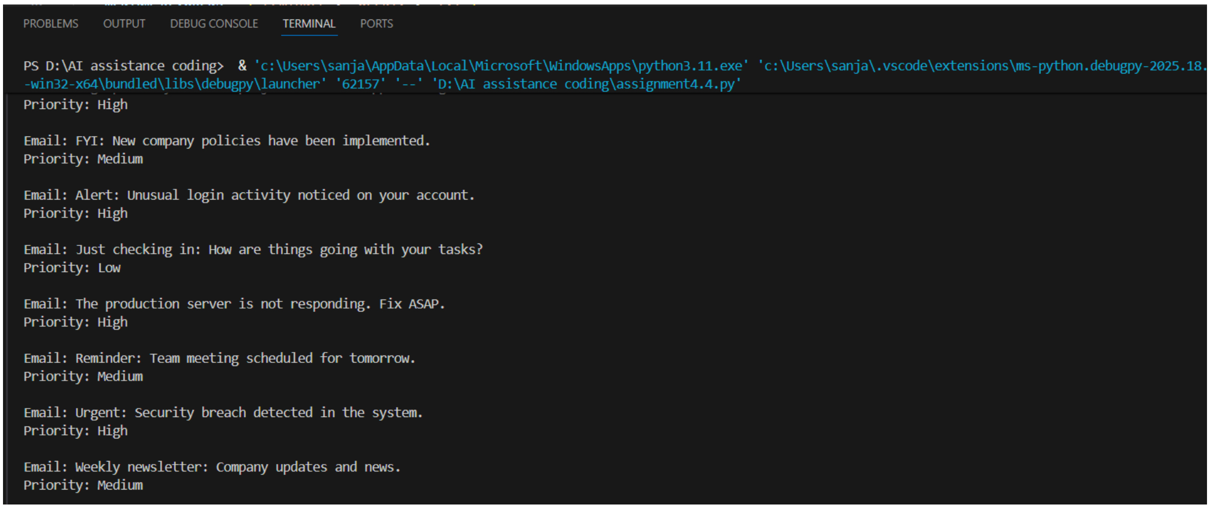
TASK2:

ONE SHOT PROMPT: Prioritzing the incoming emails. Example: The production server is not responding. fix ASAP . Priority: High. Now generate python code to classify email priority into high, medium and low based on certain keywords. Create 10 email samples inside the code itself and print the classification result for each email.



```
for item in output:
    print(f"Review: {item['review']}\nSentiment: {item['sentiment']}\n")"""
#prioritzing the incoming emails.example:the production server is not responding.fix ASAP .priority:High.now generate python code to classify email priority into high, m
# Sample email subjects
"""emails = [
    "The production server is not responding. Fix ASAP.",
    "Reminder: Team meeting scheduled for tomorrow.",
    "Urgent: Security breach detected in the system.",
    "Weekly newsletter: Company updates and news.",
    "Critical: Database backup failed last night.",
    "Casual: Lunch plans for the weekend?",
    "High priority: Client project deadline approaching.",
    "FYI: New company policies have been implemented.",
    "Alert: Unusual login activity noticed on your account.",
    "Just checking in: How are things going with your tasks?"
]
# Function to classify email priority
def classify_priority(email):
    high_keywords = ["urgent", "asap", "critical", "high priority", "alert"]
    medium_keywords = ["reminder", "weekly", "fyi"]

    email_lower = email.lower()
    if any(keyword in email_lower for keyword in high_keywords):
        return "High"
    elif any(keyword in email_lower for keyword in medium_keywords):
        return "Medium"
    else:
        return "Low"
# Analyze and print the priority for each email
for email in emails:
    priority = classify_priority(email)
    print(f"Email: {email}\nPriority: {priority}\n")"""
```

OUTPUT:

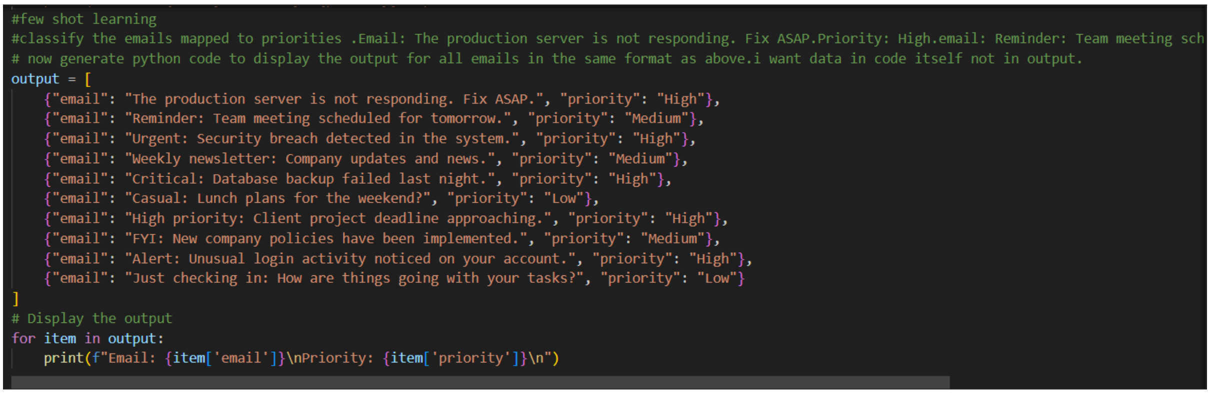


```
PS D:\AI assistance coding>  & 'c:\Users\sanja\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\sanja\.vscode\extensions\ms-python.debugpy-2025.18.
-win32-x64\bundled\libs\debugpy\launcher' '62157' '--' 'D:\AI assistance coding\assignment4.4.py'
Priority: High

Email: FYI: New company policies have been implemented.
Priority: Medium

Email: Alert: Unusual login activity noticed on your account.
Priority: High

Email: Just checking in: How are things going with your tasks?
Priority: Low

Email: The production server is not responding. Fix ASAP.
Priority: High

Email: Reminder: Team meeting scheduled for tomorrow.
Priority: Medium

Email: Urgent: Security breach detected in the system.
Priority: High

Email: Weekly newsletter: Company updates and news.
Priority: Medium
```

FEW SHOT PROMPT:

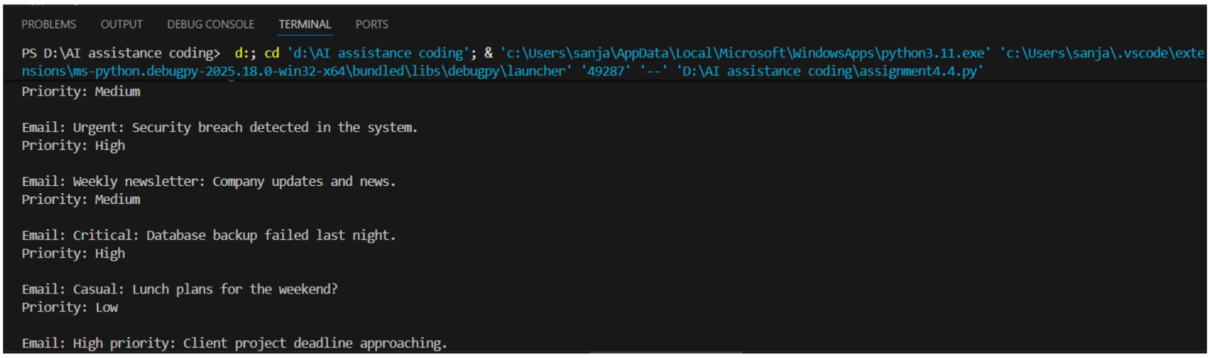
Classify the emails mapped to priorities .Email: The production server is not responding. Fix ASAP. Priority: High. email: Reminder: Team meeting scheduled for tomorrow.Priority: Medium. Now generate python code to display the output for all emails in the same format as above.i want data in code itself not in output.

CODE:



```python
#few shot learning
#classify the emails mapped to priorities .Email: The production server is not responding. Fix ASAP.Priority: High.email: Reminder: Team meeting sch
# now generate python code to display the output for all emails in the same format as above.i want data in code itself not in output.
output = [
    {"email": "The production server is not responding. Fix ASAP.", "priority": "High"},
    {"email": "Reminder: Team meeting scheduled for tomorrow.", "priority": "Medium"},
    {"email": "Urgent: Security breach detected in the system.", "priority": "High"},
    {"email": "Weekly newsletter: Company updates and news.", "priority": "Medium"},
    {"email": "Critical: Database backup failed last night.", "priority": "High"},
    {"email": "Casual: Lunch plans for the weekend?", "priority": "Low"},
    {"email": "High priority: Client project deadline approaching.", "priority": "High"},
    {"email": "FYI: New company policies have been implemented.", "priority": "Medium"},
    {"email": "Alert: Unusual login activity noticed on your account.", "priority": "High"},
    {"email": "Just checking in: How are things going with your tasks?", "priority": "Low"}
]
# Display the output
for item in output:
    print(f"Email: {item['email']}\nPriority: {item['priority']}\n")
```

OUTPUT:

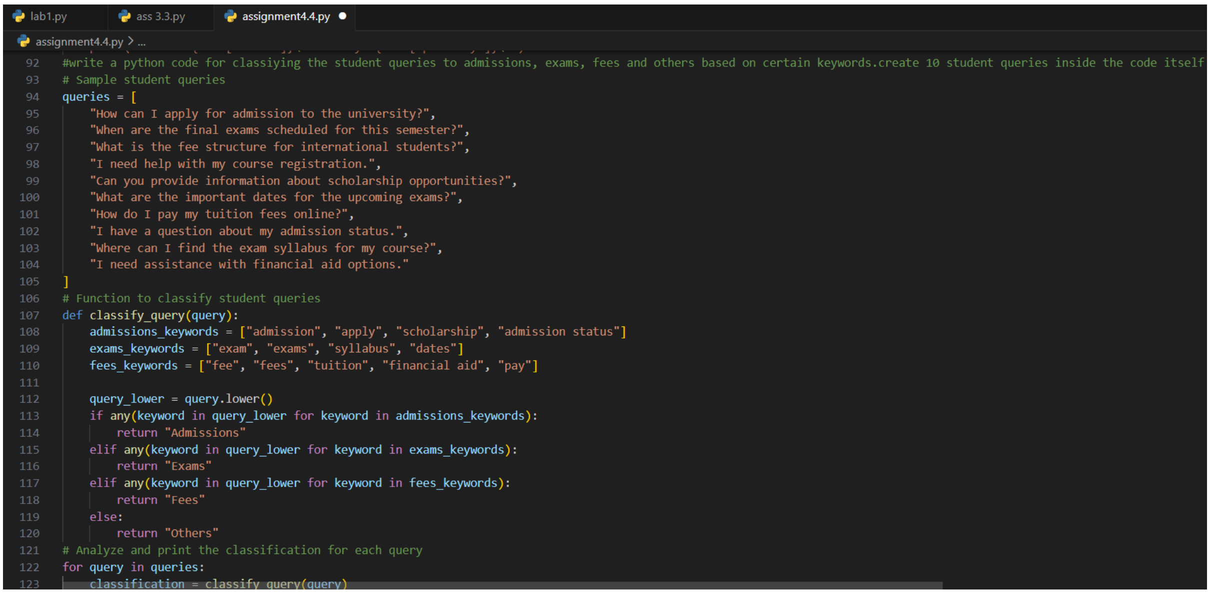


```
PS D:\AI assistance coding>  d:; cd 'd:\AI assistance coding'; & 'c:\Users\sanja\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\sanja\.vscode\exte
nsions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '49287' '--' 'D:\AI assistance coding\assignment4.4.py'
Priority: Medium

Email: Urgent: Security breach detected in the system.
Priority: High

Email: Weekly newsletter: Company updates and news.
Priority: Medium

Email: Critical: Database backup failed last night.
Priority: High

Email: Casual: Lunch plans for the weekend?
Priority: Low

Email: High priority: Client project deadline approaching.
```

TASK3:

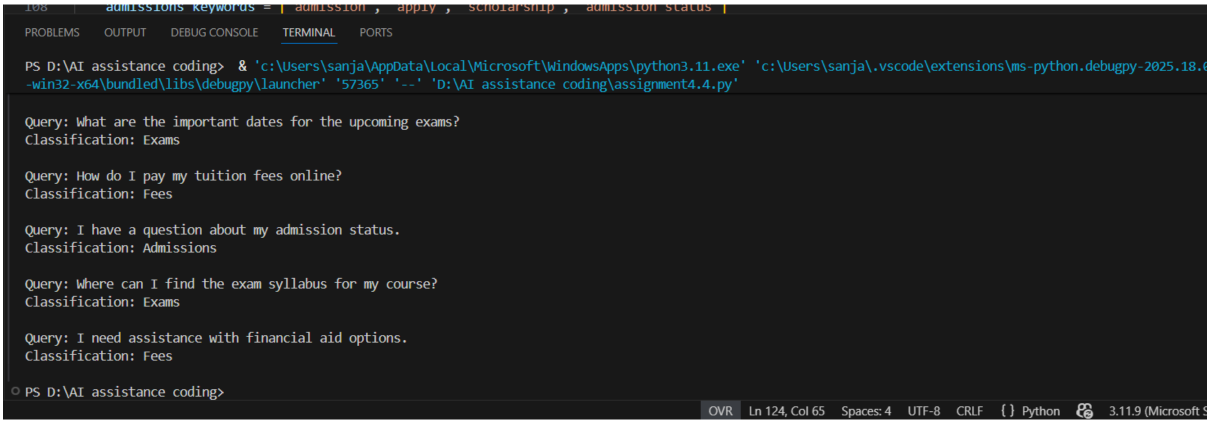
ZERO SHOT PROMPT: Write a python code for classiying the student queries to admissions, exams, fees and others based on certain keywords. Create 10 student queries inside the code itself and print the classification result for each query.

CODE:



```python
#write a python code for classiying the student queries to admissions, exams, fees and others based on certain keywords.create 10 student queries inside the code itself
# Sample student queries
queries = [
    "How can I apply for admission to the university?",
    "When are the final exams scheduled for this semester?",
    "What is the fee structure for international students?",
    "I need help with my course registration.",
    "Can you provide information about scholarship opportunities?",
    "What are the important dates for the upcoming exams?",
    "How do I pay my tuition fees online?",
    "I have a question about my admission status.",
    "Where can I find the exam syllabus for my course?",
    "I need assistance with financial aid options."
]
# Function to classify student queries
def classify_query(query):
    admissions_keywords = ["admission", "apply", "scholarship", "admission status"]
    exams_keywords = ["exam", "exams", "syllabus", "dates"]
    fees_keywords = ["fee", "fees", "tuition", "financial aid", "pay"]

    query_lower = query.lower()
    if any(keyword in query_lower for keyword in admissions_keywords):
        return "Admissions"
    elif any(keyword in query_lower for keyword in exams_keywords):
        return "Exams"
    elif any(keyword in query_lower for keyword in fees_keywords):
        return "Fees"
    else:
        return "Others"
# Analyze and print the classification for each query
for query in queries:
    classification = classify_query(query)
```

OUTPUT:

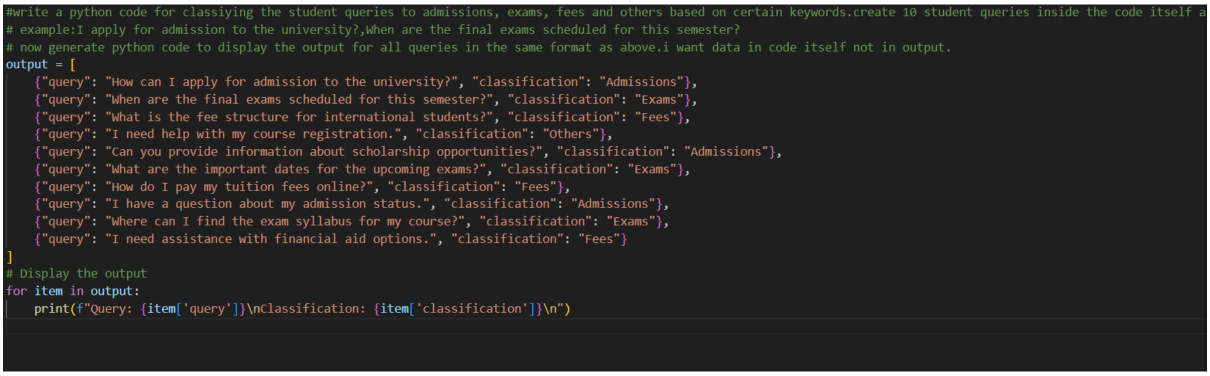


```
PS D:\AI assistance coding>  & 'c:\Users\sanja\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\sanja\.vscode\extensions\ms-python.debugpy-2025.18.0
-win32-x64\bundled\libs\debugpy\launcher' '57365' '--' 'D:\AI assistance coding\assignment4.4.py'

Query: What are the important dates for the upcoming exams?
Classification: Exams

Query: How do I pay my tuition fees online?
Classification: Fees

Query: I have a question about my admission status.
Classification: Admissions

Query: Where can I find the exam syllabus for my course?
Classification: Exams

Query: I need assistance with financial aid options.
Classification: Fees

PS D:\AI assistance coding>
```

ONE SHOT:

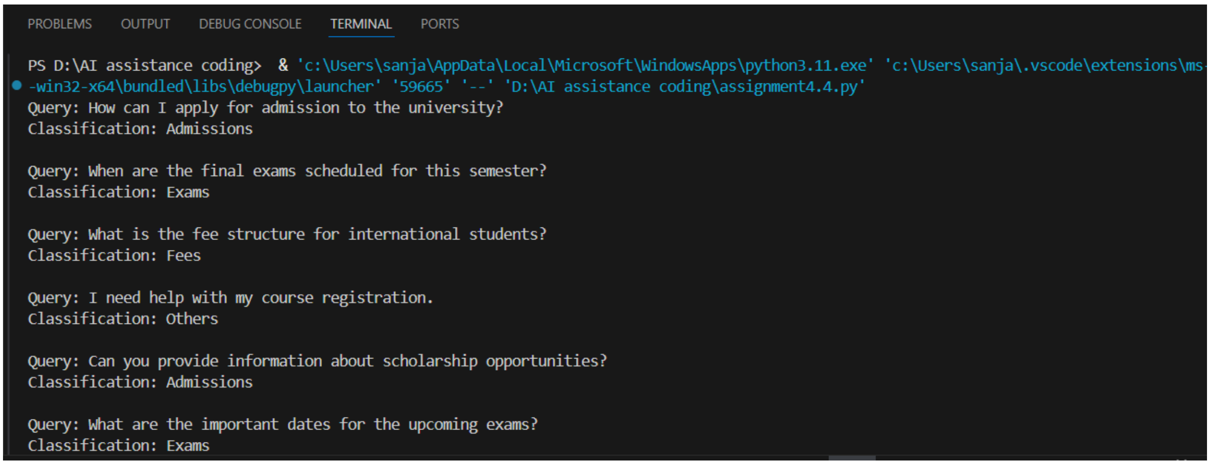
Write a python code for classiying the student queries to admissions, exams, fees and others based on certain keywords.create 10 student queries inside the code itself and print the classification reult for each query.Example: I apply for admission to the university?,When are the final exams scheduled for this semester?Now generate python code to display the output for all queries in the same format as above.i want data in code itself not in output.

CODE:



```
#write a python code for classiying the student queries to admissions, exams, fees and others based on certain keywords.create 10 student queries inside the code itself a
# example:I apply for admission to the university?,When are the final exams scheduled for this semester?
# now generate python code to display the output for all queries in the same format as above.i want data in code itself not in output.
output = [
    {"query": "How can I apply for admission to the university?", "classification": "Admissions"},
    {"query": "When are the final exams scheduled for this semester?", "classification": "Exams"},
    {"query": "What is the fee structure for international students?", "classification": "Fees"},
    {"query": "I need help with my course registration.", "classification": "Others"},
    {"query": "Can you provide information about scholarship opportunities?", "classification": "Admissions"},
    {"query": "What are the important dates for the upcoming exams?", "classification": "Exams"},
    {"query": "How do I pay my tuition fees online?", "classification": "Fees"},
    {"query": "I have a question about my admission status.", "classification": "Admissions"},
    {"query": "Where can I find the exam syllabus for my course?", "classification": "Exams"},
    {"query": "I need assistance with financial aid options.", "classification": "Fees"}
]
# Display the output
for item in output:
    print(f"Query: {item['query']}\nClassification: {item['classification']}\n")
```

OUTPUT:

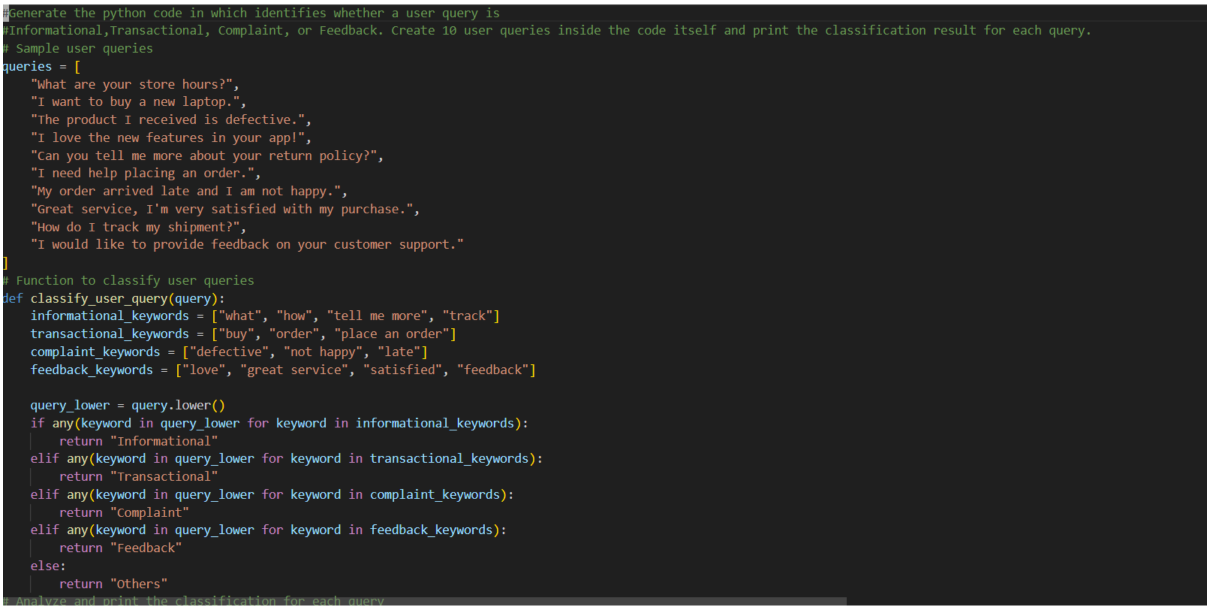


```
PS D:\AI assistance coding>  & 'c:\Users\sanja\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\sanja\.vscode\extensions\ms-
-win32-x64\bundled\libs\debugpy\launcher' '59665' '--' 'D:\AI assistance coding\assignment4.4.py'
Query: How can I apply for admission to the university?
Classification: Admissions

Query: When are the final exams scheduled for this semester?
Classification: Exams

Query: What is the fee structure for international students?
Classification: Fees

Query: I need help with my course registration.
Classification: Others

Query: Can you provide information about scholarship opportunities?
Classification: Admissions

Query: What are the important dates for the upcoming exams?
Classification: Exams
```

TASK4:

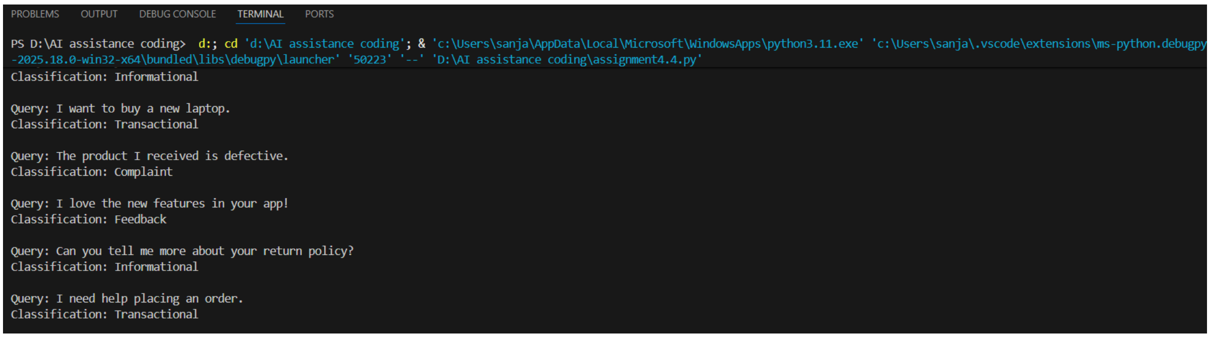
ZERO SHOT: Generate the Python code that identifies whether a user query is Informational, Transactional, Complaint, or Feedback. Create 10 user queries inside the code itself and print the classification result for each query.

CODE:



```
#Generate the python code in which identifies whether a user query is
#Informational,Transactional, Complaint, or Feedback. Create 10 user queries inside the code itself and print the classification result for each query.
# Sample user queries
queries = [
    "What are your store hours?",
    "I want to buy a new laptop.",
    "The product I received is defective.",
    "I love the new features in your app!",
    "Can you tell me more about your return policy?",
    "I need help placing an order.",
    "My order arrived late and I am not happy.",
    "Great service, I'm very satisfied with my purchase.",
    "How do I track my shipment?",
    "I would like to provide feedback on your customer support."
]
# Function to classify user queries
def classify_user_query(query):
    informational_keywords = ["what", "how", "tell me more", "track"]
    transactional_keywords = ["buy", "order", "place an order"]
    complaint_keywords = ["defective", "not happy", "late"]
    feedback_keywords = ["love", "great service", "satisfied", "feedback"]

    query_lower = query.lower()
    if any(keyword in query_lower for keyword in informational_keywords):
        return "Informational"
    elif any(keyword in query_lower for keyword in transactional_keywords):
        return "Transactional"
    elif any(keyword in query_lower for keyword in complaint_keywords):
        return "Complaint"
    elif any(keyword in query_lower for keyword in feedback_keywords):
        return "Feedback"
    else:
        return "Others"
# Analyze and print the classification for each query
```

OUTPUT:

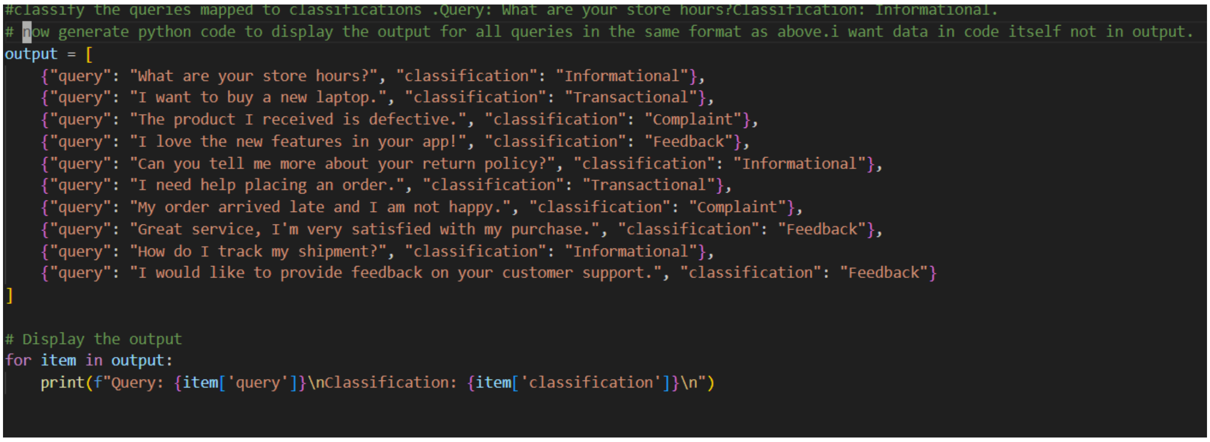


```
PS D:\AI assistance coding>  d:; cd 'd:\AI assistance coding'; & 'c:\Users\sanja\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\sanja\.vscode\extensions\ms-python.debugpy
-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '50223' '--' 'D:\AI assistance coding\assignment4.4.py'
Classification: Informational

Query: I want to buy a new laptop.
Classification: Transactional

Query: The product I received is defective.
Classification: Complaint

Query: I love the new features in your app!
Classification: Feedback

Query: Can you tell me more about your return policy?
Classification: Informational

Query: I need help placing an order.
Classification: Transactional
```

ONE SHOT:

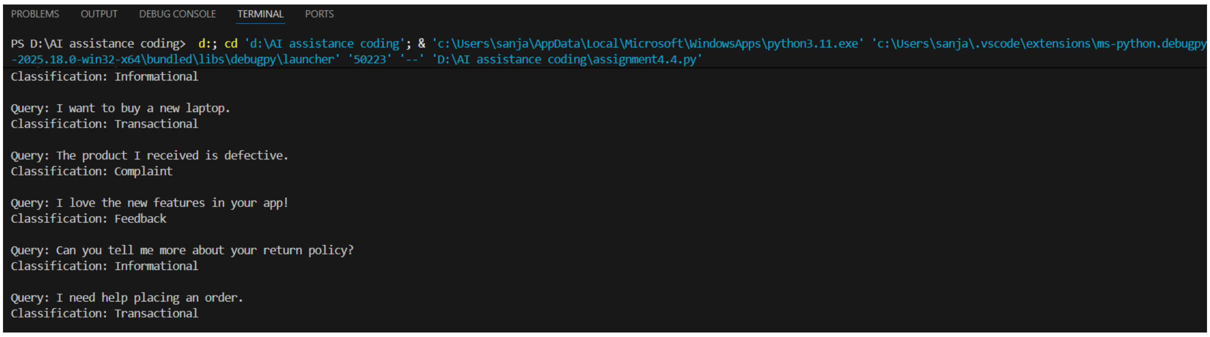
Classify the queries mapped to classifications .Query: What are your store hours? Classification: Informational.Now generate python code to display the output for all queries in the same format as above. i want data in code itself not in output.

CODE:



```python
#classify the queries mapped to classifications .Query: What are your store hours?Classification: Informational.
# now generate python code to display the output for all queries in the same format as above.i want data in code itself not in output.
output = [
    {"query": "What are your store hours?", "classification": "Informational"},
    {"query": "I want to buy a new laptop.", "classification": "Transactional"},
    {"query": "The product I received is defective.", "classification": "Complaint"},
    {"query": "I love the new features in your app!", "classification": "Feedback"},
    {"query": "Can you tell me more about your return policy?", "classification": "Informational"},
    {"query": "I need help placing an order.", "classification": "Transactional"},
    {"query": "My order arrived late and I am not happy.", "classification": "Complaint"},
    {"query": "Great service, I'm very satisfied with my purchase.", "classification": "Feedback"},
    {"query": "How do I track my shipment?", "classification": "Informational"},
    {"query": "I would like to provide feedback on your customer support.", "classification": "Feedback"}
]

# Display the output
for item in output:
    print(f"Query: {item['query']}\nClassification: {item['classification']}\n")
```

OUTPUT:



```
PS D:\AI assistance coding>  d:; cd 'd:\AI assistance coding'; & 'c:\Users\sanja\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\sanja\.vscode\extensions\ms-python.debugpy
-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '50223' '--' 'D:\AI assistance coding\assignment4.4.py'
Classification: Informational

Query: I want to buy a new laptop.
Classification: Transactional

Query: The product I received is defective.
Classification: Complaint

Query: I love the new features in your app!
Classification: Feedback

Query: Can you tell me more about your return policy?
Classification: Informational

Query: I need help placing an order.
Classification: Transactional
```