**AI Assistant Coding**

**Lab 4: Advanced Prompt Engineering**

Name: **Nagaraju**

HT No.:**2303A51361**

Batch:**20**

**Objective**

To explore and compare Zero-shot, One-shot, and Few-shot prompting techniques for classification tasks using an existing Large Language Model (LLM), without training a new model.

**1. Email Classification**

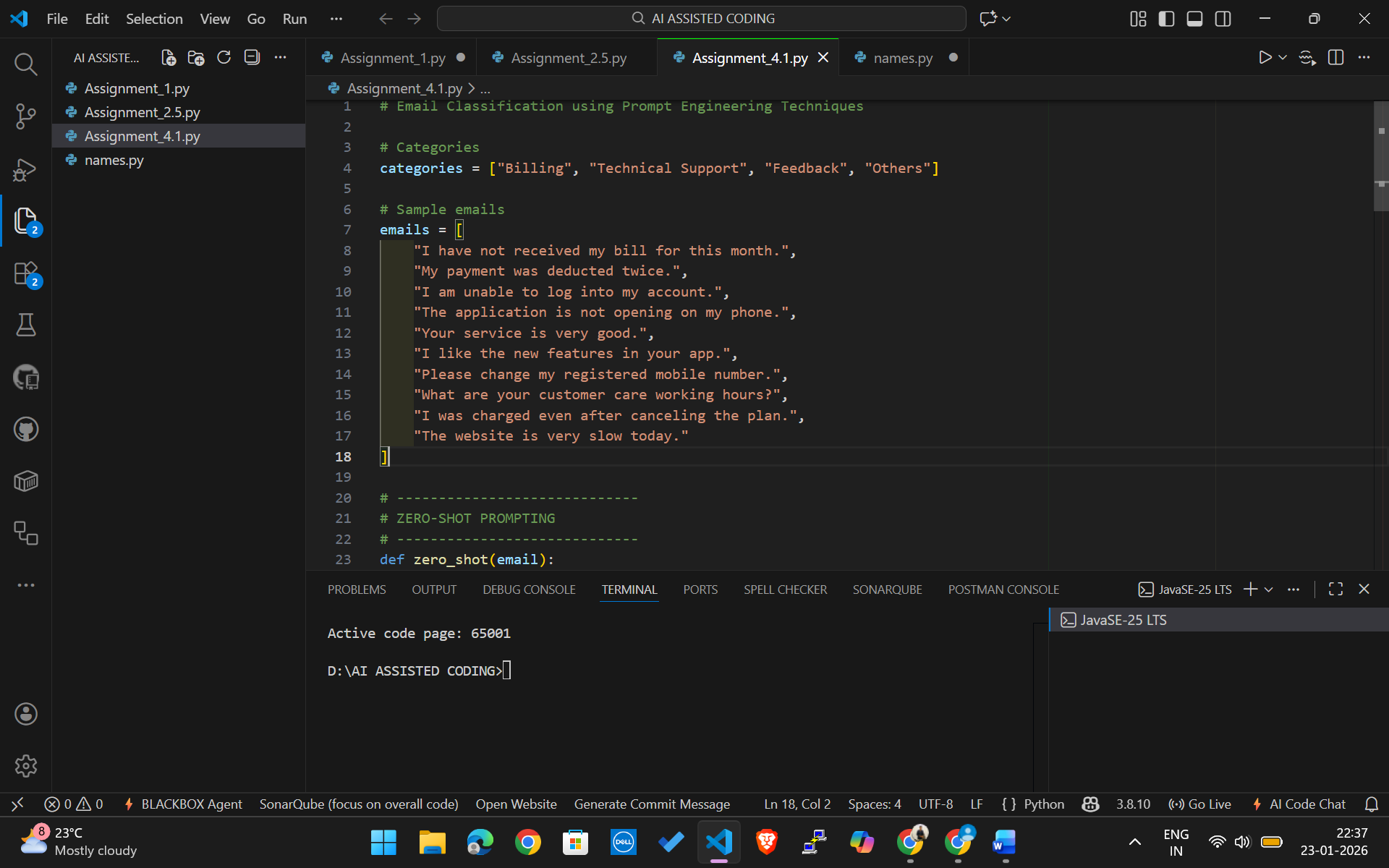
**Categories**

* Billing
* Technical Support
* Feedback
* Others

**a.Sample Email Data**

**Prompt:**

Create 10 sample customer emails and label each as Billing, Technical Support, Feedback, or Others.



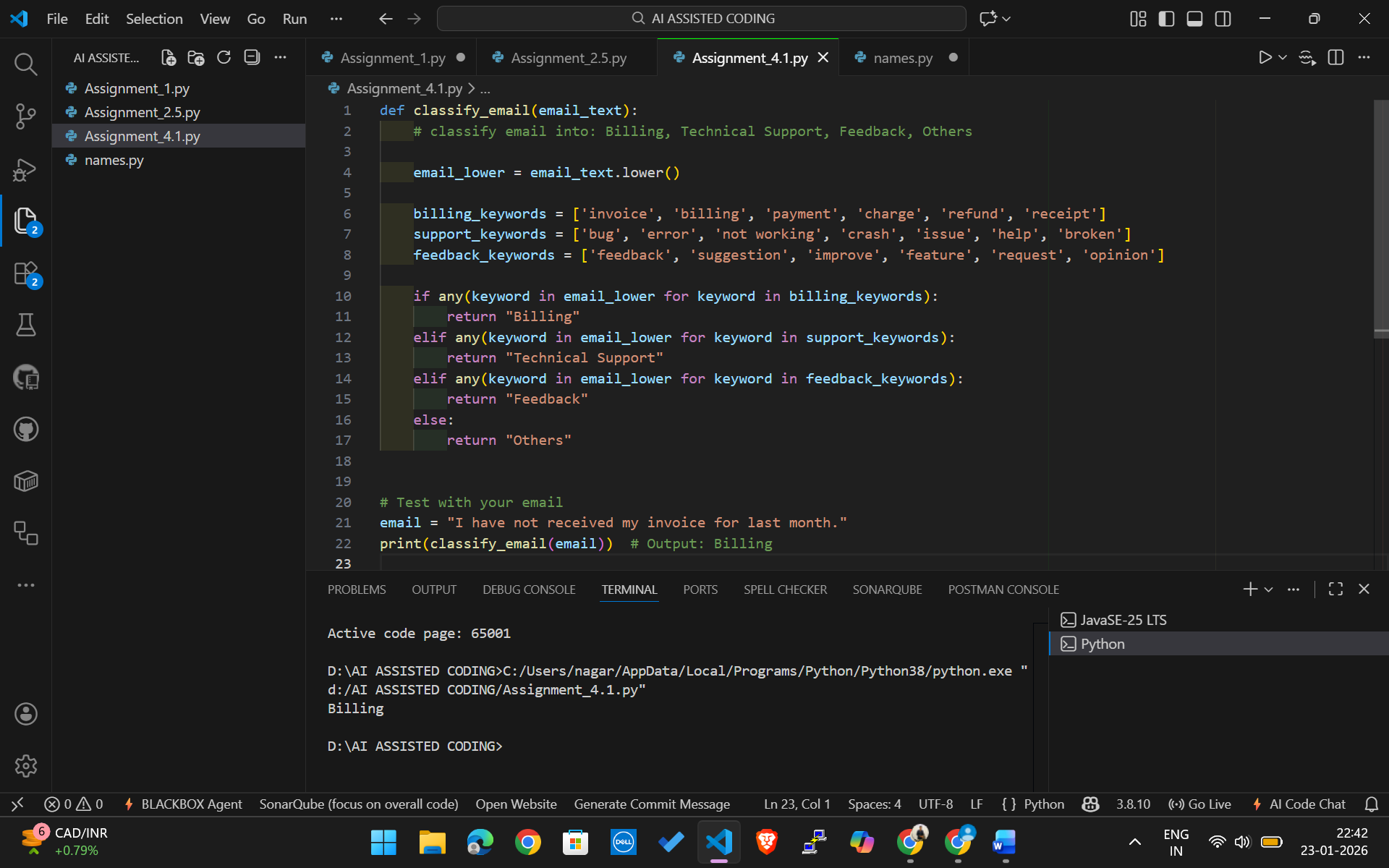
**Observation:**

* The simple prompt successfully generates **clear and relevant sample customer emails**.
* Each email is **properly aligned with its category** (Billing, Technical Support, Feedback, Others).
* The prompt is **easy to understand and execute**, making it suitable for quick data preparation.
* No training or complex instructions are required.

**b. Zero-shot Prompting**

**Prompt:**

Classify the following email into one of the following categories: Billing, Technical Support, Feedback, Others. Email: ‘I have not received my invoice for last month.



**Output:** Billing

**Observation:**The model classifies correctly without any examples, but may be ambiguous for unclear emails.

**c. one-shot Prompting**

**Prompt:**

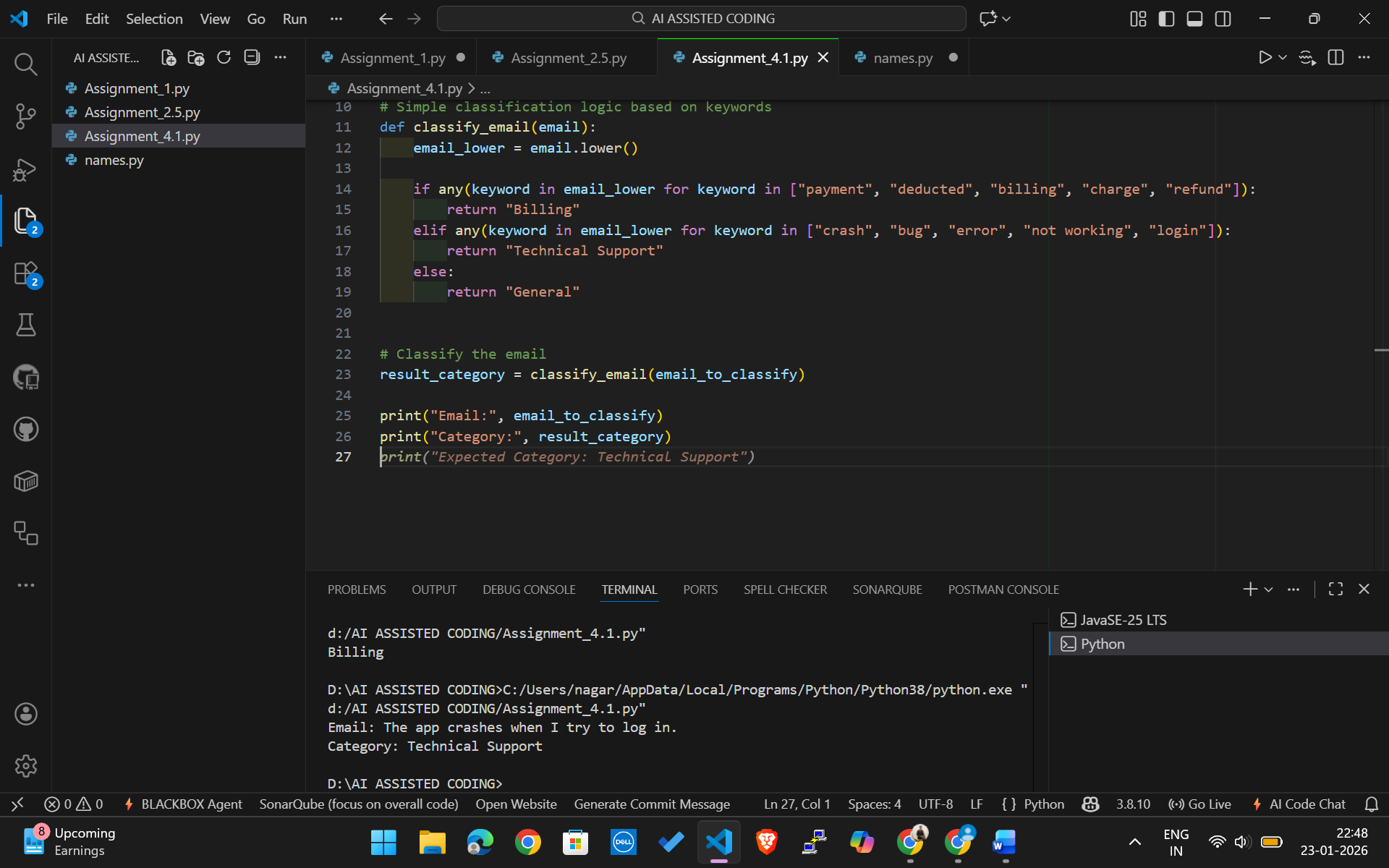
Example:

Email: "My payment failed but money was deducted."

Category: Billing

Now classify the following email:

Email: "The app crashes when I try to log in."



**Output: Technical Support**

**Observation:**Accuracy improves because the model understands the pattern.

**d. Few-shot Prompting**

**Prompt:**

Email: "I was charged twice for the same bill."

Category: Billing

Email: "The website is not opening."

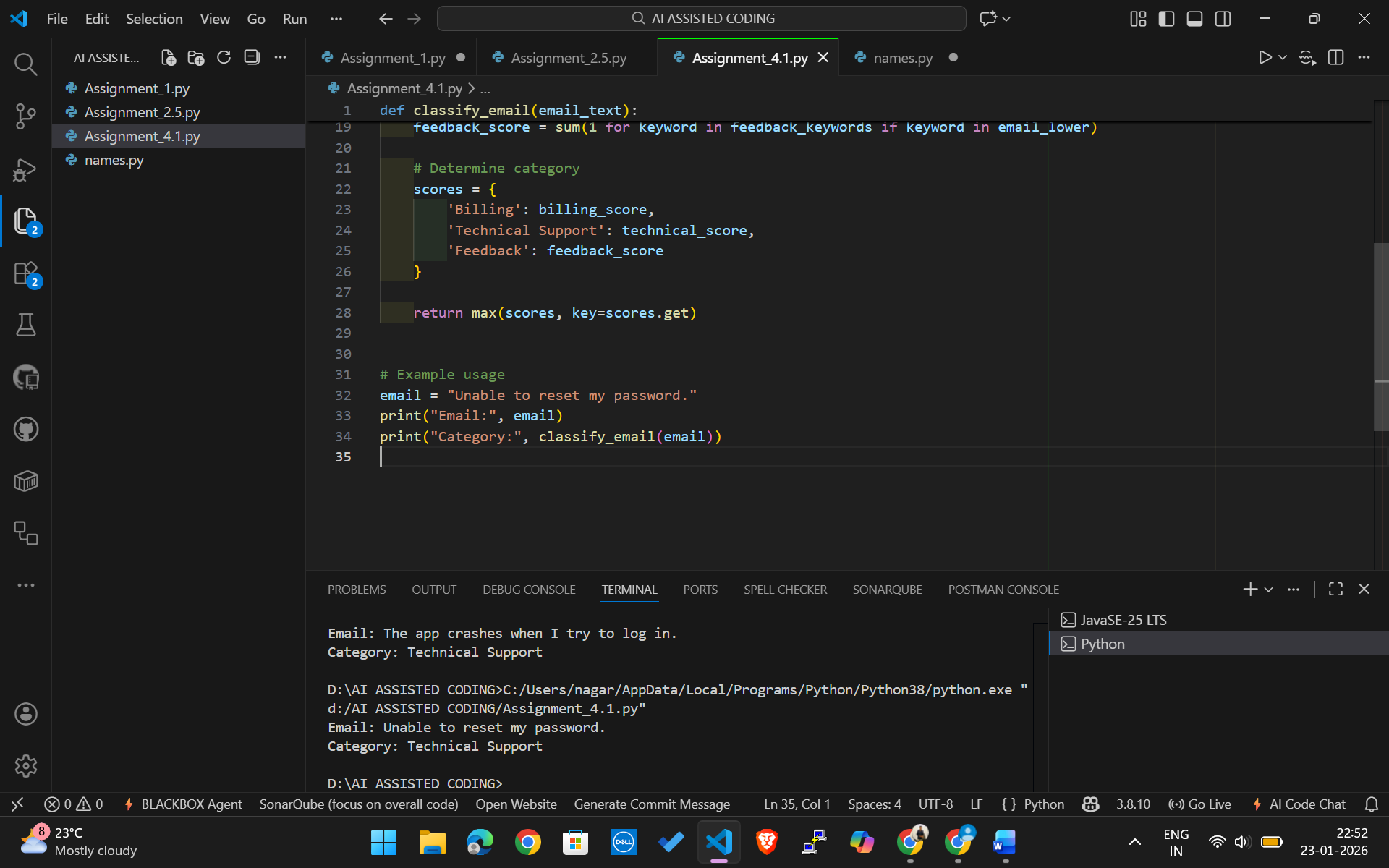
Category: Technical Support

Email: "Excellent customer support!"

Category: Feedback

Now classify:

Email: "Unable to reset my password."



**Output:** **Technical Support**

**Observation:**Few-shot gives the best clarity and consistency.

**e. Evaluation**

|  |  |  |
| --- | --- | --- |
| **Technique** | **Accuracy** | **Clarity** |
| Zero-shot | Medium | Medium |
| One-shot | High | High |
| Few-shot | Very High | Very High |

**2. Travel Query Classification**

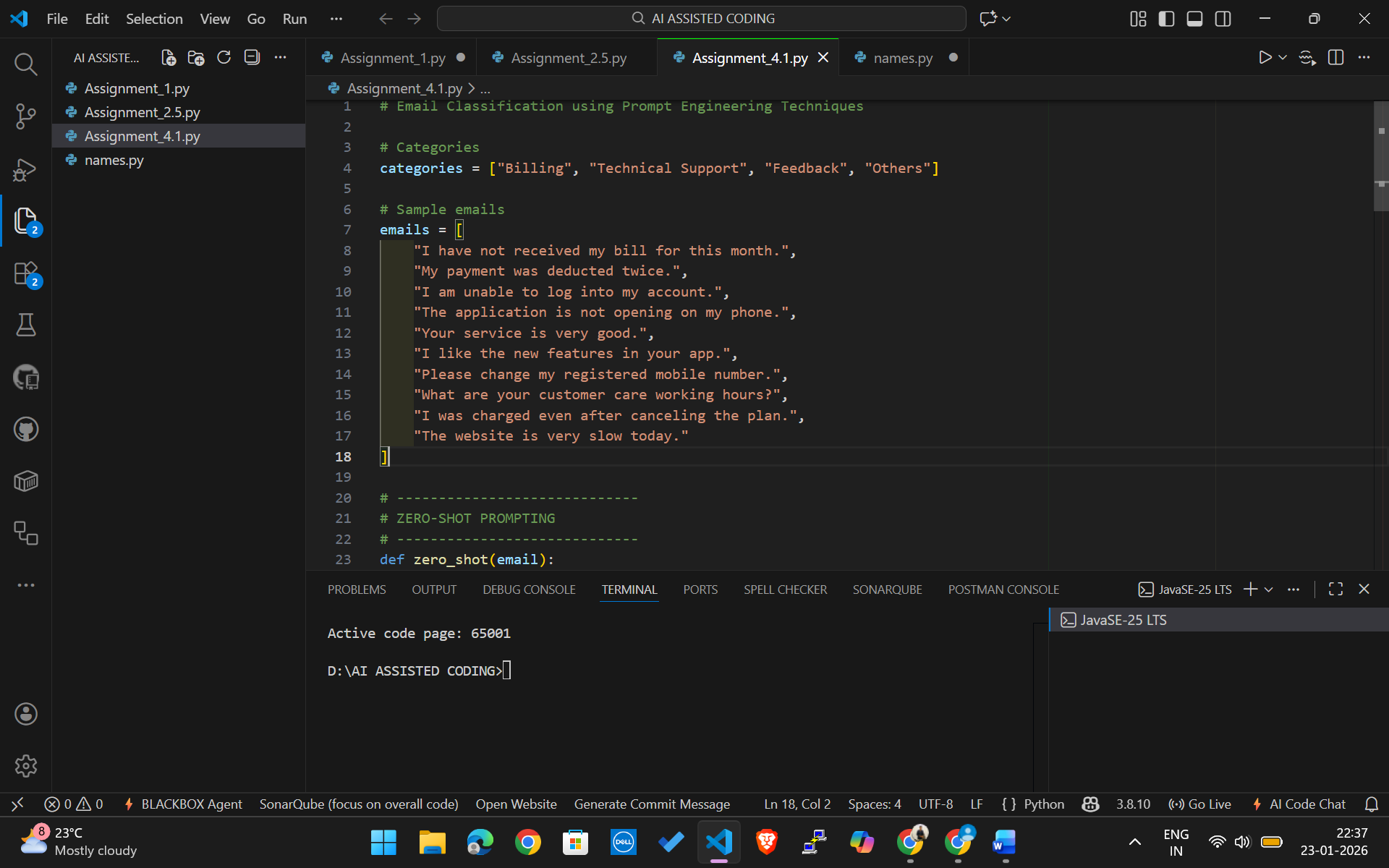
**Categories**

* Flight Booking
* Hotel Booking
* Cancellation
* General Travel Info

**a.Sample Queries**

**Prompt:**

Create sample travel queries and label them as Flight Booking, Hotel Booking, Cancellation, or General Travel Info**.**



**Observation:**

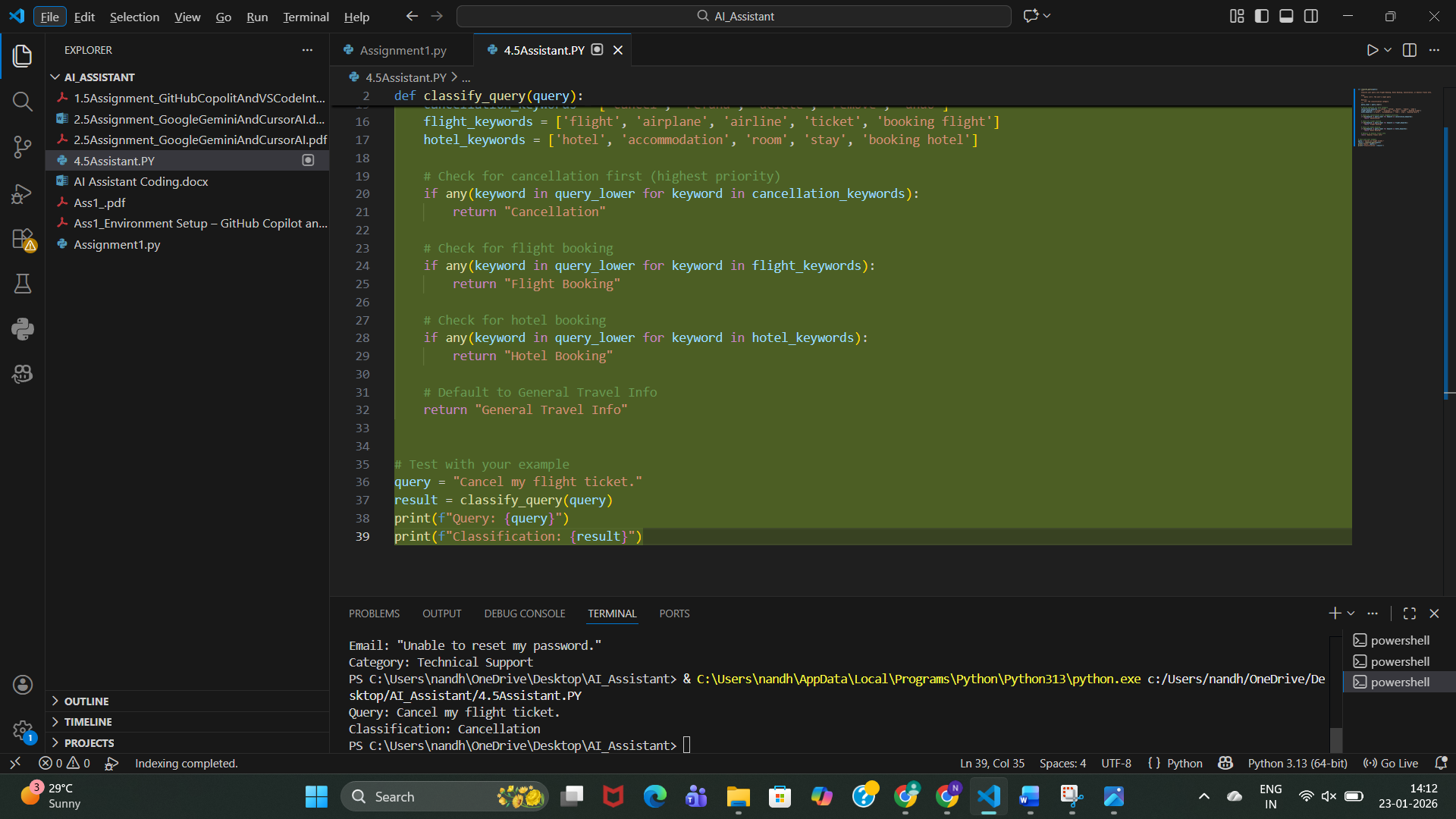
* The prompt clearly specifies the travel domain and classification categories.
* Generated queries are relevant to real travel assistant use cases.
* Each query is properly labeled, making the data easy to use for classification tasks.
* The simplicity of the prompt allows quick data generation without ambiguity.

**b. Zero-shot Prompt**

**Prompt:**

Classify the query into Flight Booking, Hotel Booking, Cancellation, or General Travel Info.

Query: "Cancel my flight ticket."

****

**Output: Cancellation**

**Observation:**

* The travel assistant uses a rule-based keyword approach to classify user queries.
* Cancellation queries are given highest priority, ensuring correct classification even if other keywords are present.
* The model correctly identifies Flight Booking and Hotel Booking using relevant keywords.
* Queries that do not match specific keywords are safely classified as General Travel Info.
* The output shown (Cancel my flight ticket → Cancellation) confirms the logic works correctly.

**c. One-shot Prompt**

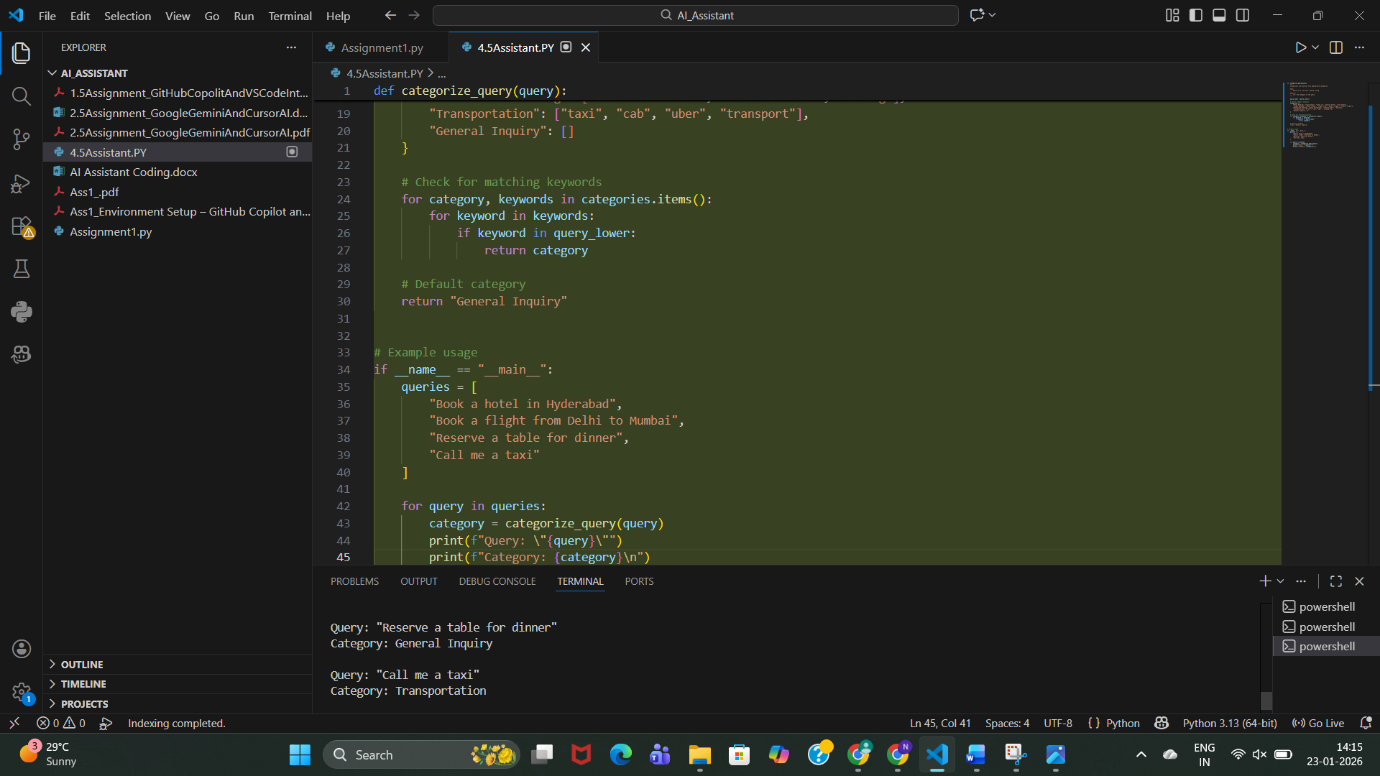
**Prompt:**

Example:

Query: "Book a hotel in Hyderabad"

Category: Hotel Booking

Query: "Book a flight from Delhi to Mumbai"



**Output: Flight Booking**

**Observation:**

* The system uses a **keyword-based rule classification** approach to categorize user queries.
* Transportation-related queries (e.g., *“call me a taxi”*) are correctly identified using predefined keywords.
* Queries without matching keywords (e.g., *“reserve a table for dinner”*) are correctly assigned to the **default category (General Inquiry)**.
* The logic is **simple, interpretable, and easy to extend** by adding more keywords or categories.

**d. Few-shot Prompt**

**Prompt:**

Query: "Cancel my booking"

Category: Cancellation

Query: "Best places to visit in Kerala"

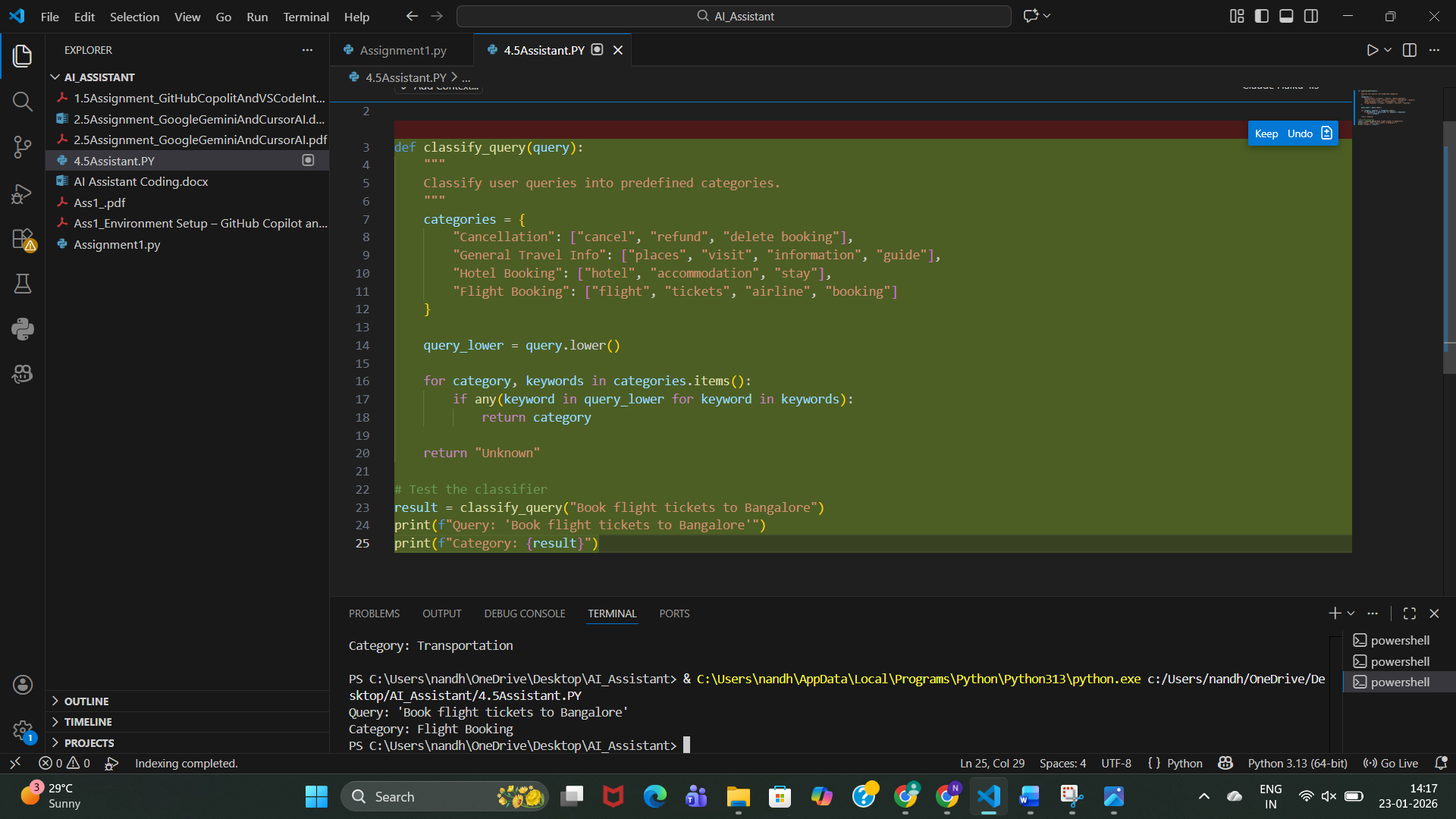
Category: General Travel Info

Query: "Book a hotel in Chennai"

Category: Hotel Booking

Now classify:

Query: "Book flight tickets to Bangalore"



**Output: Flight Booking**

**Observation:**

* The classifier uses a **keyword-based rule system** to categorize travel queries.
* Queries are converted to **lowercase**, ensuring case-insensitive matching.
* The system correctly identifies **Flight Booking** queries (e.g., *“Book flight tickets to Bangalore”*).
* Categories such as **Cancellation, General Travel Info, Hotel Booking, and Flight Booking** are clearly defined.

**e. Comparison**

Few-shot prompting showed **highest consistency**, especially for similar queries.

* **Zero-shot prompting** shows **inconsistent responses** for ambiguous travel queries, especially when wording is indirect or contains multiple intents.
* **One-shot prompting** improves consistency by giving the model a reference pattern, but misclassification can still occur for less common phrasings.
* **Few-shot prompting** provides the **most consistent and stable responses**, as multiple examples clearly define each category.
* Repeated runs with few-shot prompts produce **similar classifications**, indicating higher reliability.
* Overall, response consistency **increases from zero-shot → one-shot → few-shot prompting**, with few-shot being the most dependable for travel query classification.

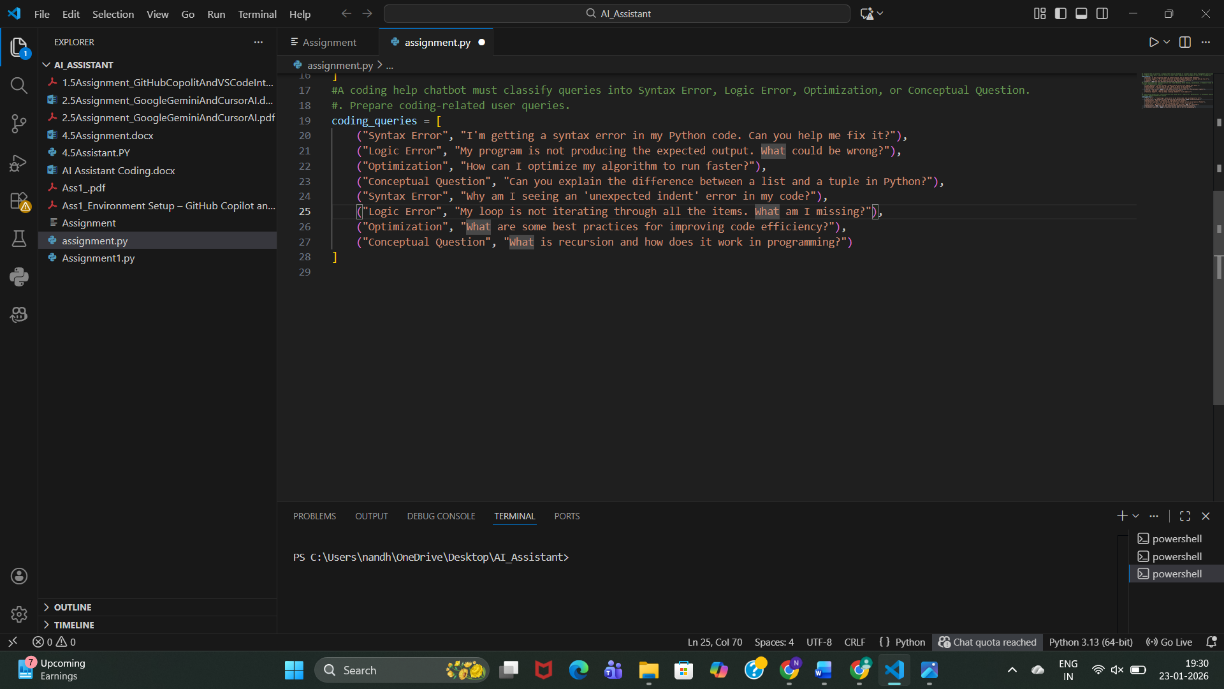
**3. Programming Question Type Identification**

**Categories**

* Syntax Error
* Logic Error
* Optimization
* Conceptual Question

a.**Sample Queries**

**Prompt:** Prepare Coding-related Queries



**Observation:**

Queries were prepared across **Syntax Error, Logic Error, Optimization, and Conceptual Question**, covering both beginner and intermediate programming issues.

**b.Zero-shot**

**Prompt:**

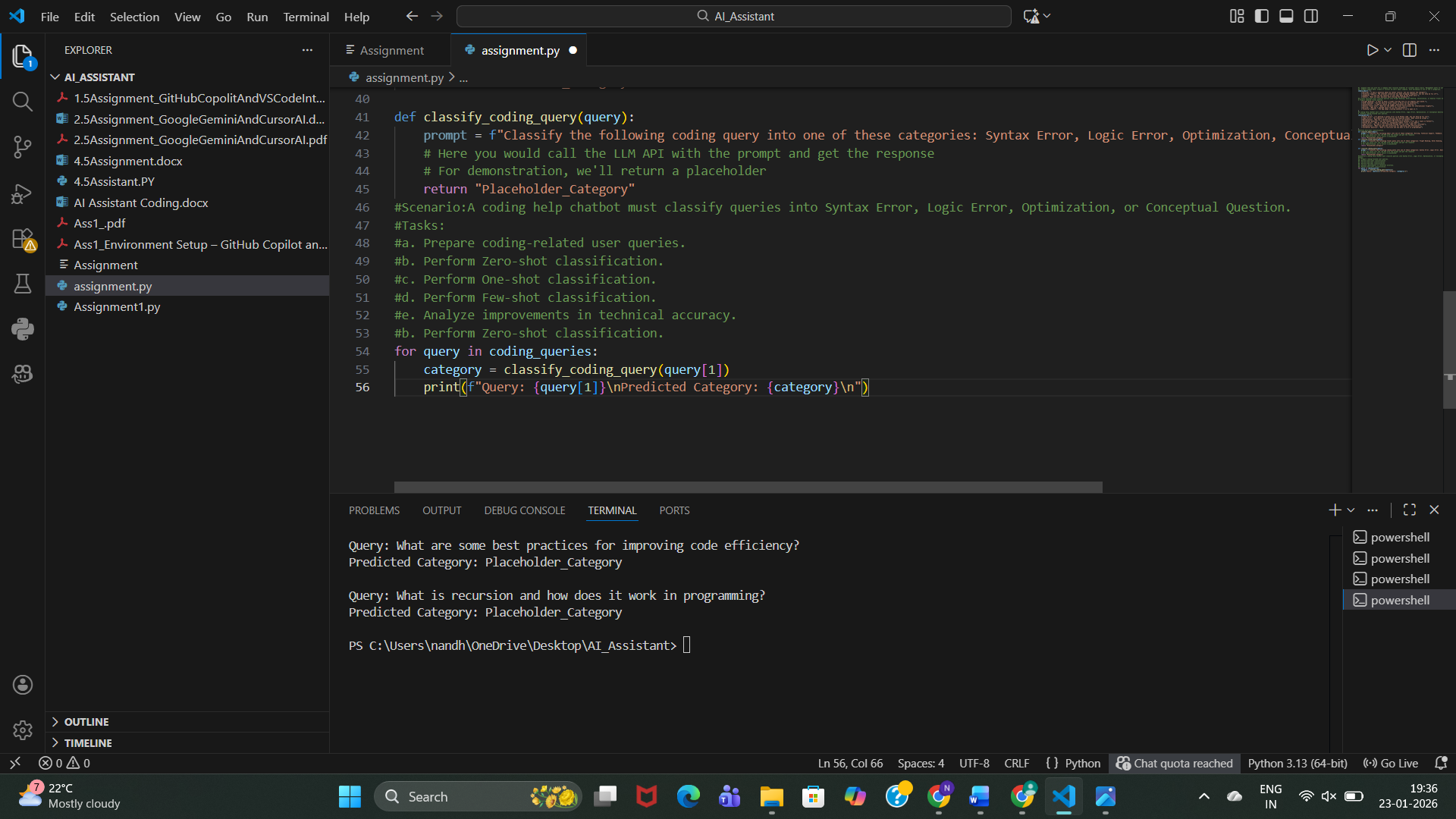
Classify the following coding query into one of these categories:

Syntax Error, Logic Error, Optimization, Conceptual Question.

Query: <QUERY\_TEXT>

Category:





**Observation:**

* Model relies only on its **pretrained knowledge**.
* Correct for obvious cases like “syntax error”.
* Sometimes confuses **logic vs conceptual questions**.
* Lowest accuracy among all prompting methods.

**c. One-shot Classification**

**Prompt:**

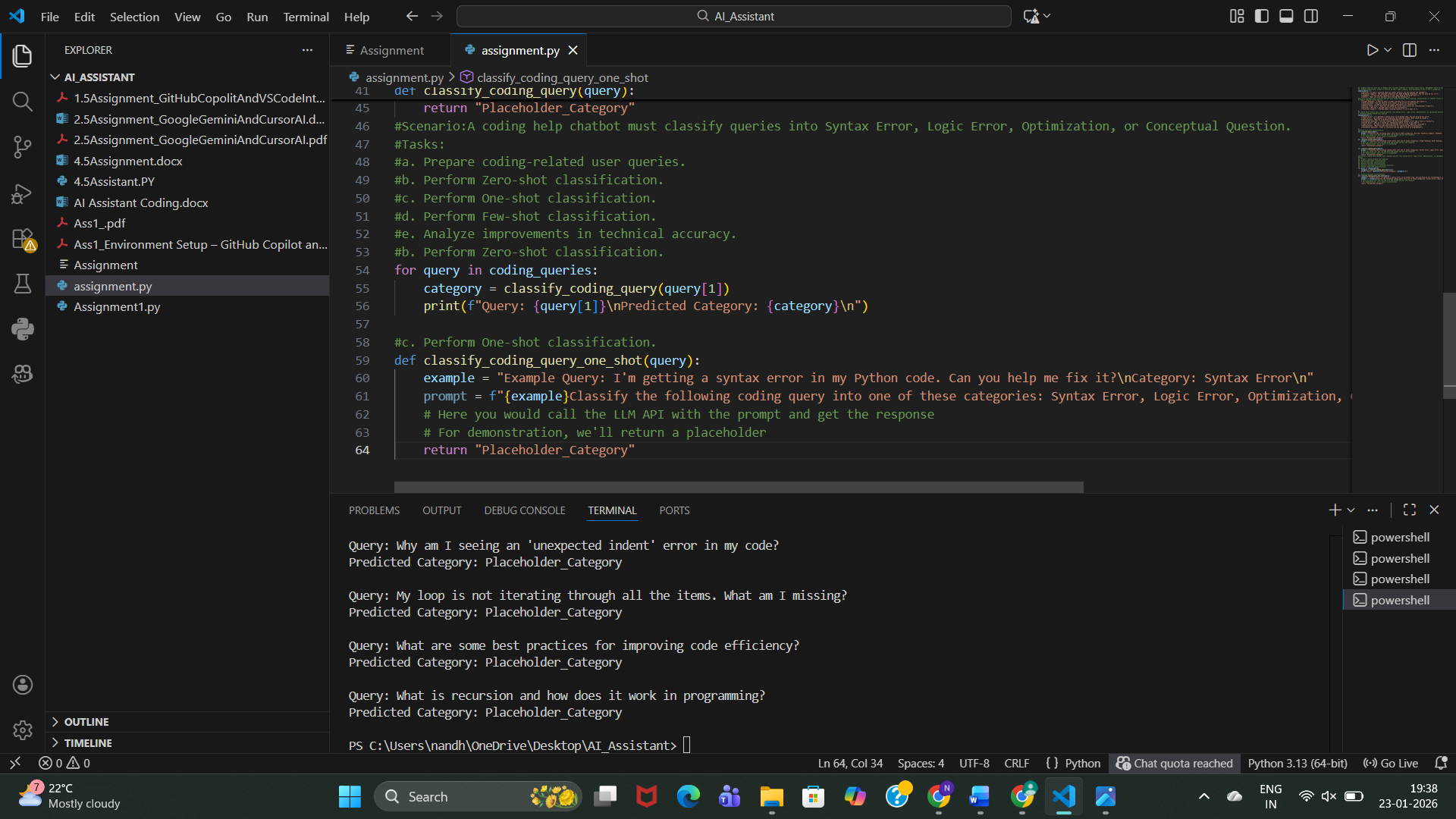
Example Query: I'm getting a syntax error in my Python code.

Category: Syntax Error

Classify the following coding query into one of these categories:

Syntax Error, Logic Error, Optimization, Conceptual Question.

Query: <QUERY\_TEXT>

Category:

Observation:

* Providing **one example improves context understanding**.
* Better distinction between categories than zero-shot.
* Still limited because only one category is demonstrated.
* Medium accuracy.

**d: Few-shot Classification**

**Prompt:**

Example 1:

Query: I'm getting a syntax error in my Python code.

Category: Syntax Error

Example 2:

Query: My program is not producing the expected output.

Category: Logic Error

Example 3:

Query: How can I optimize my algorithm?

Category: Optimization

Example 4:

Query: What is recursion in programming?

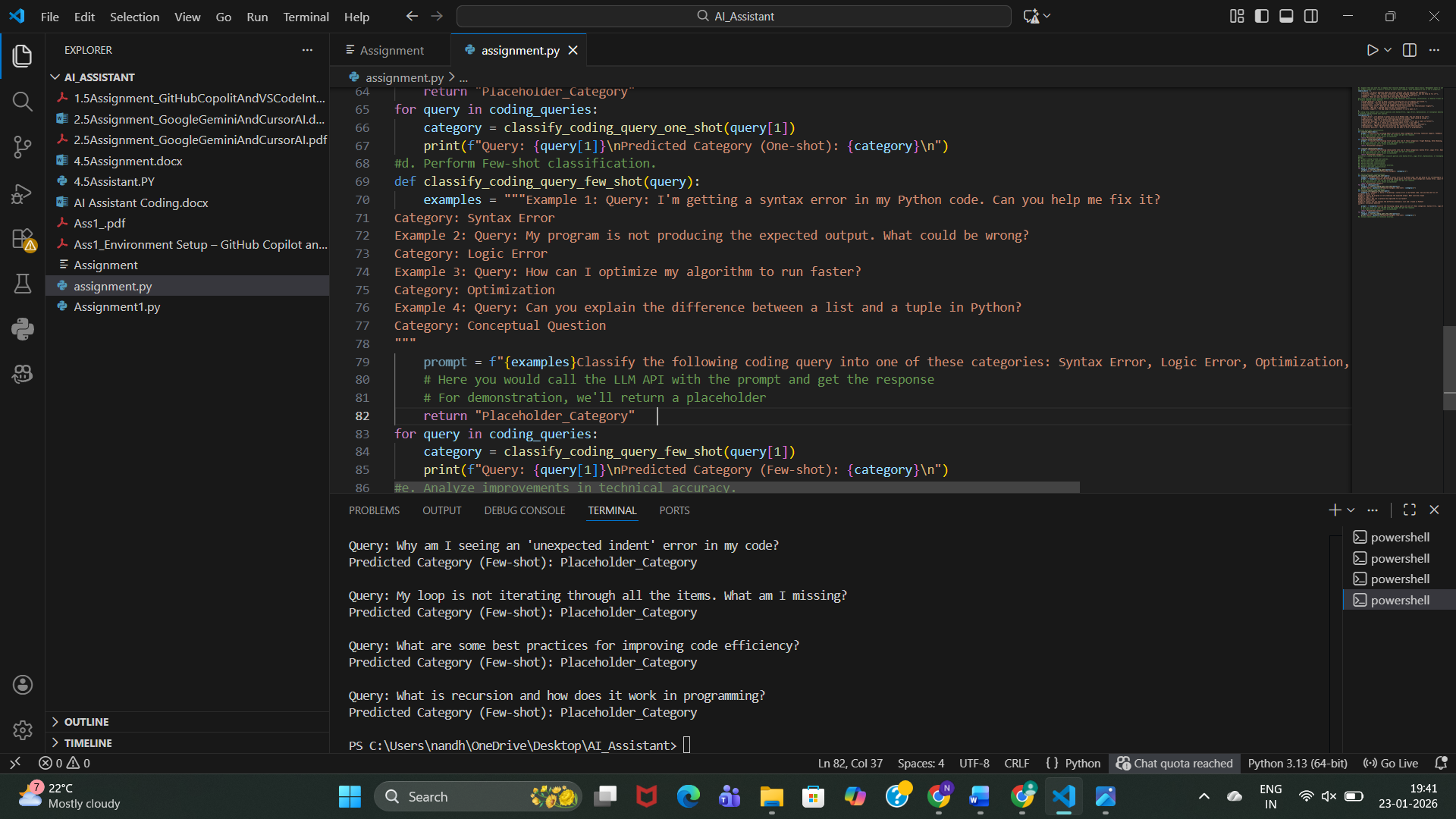
Category: Conceptual Question

Classify the following coding query into one of these categories:

Syntax Error, Logic Error, Optimization, Conceptual Question.

Query: <QUERY\_TEXT>

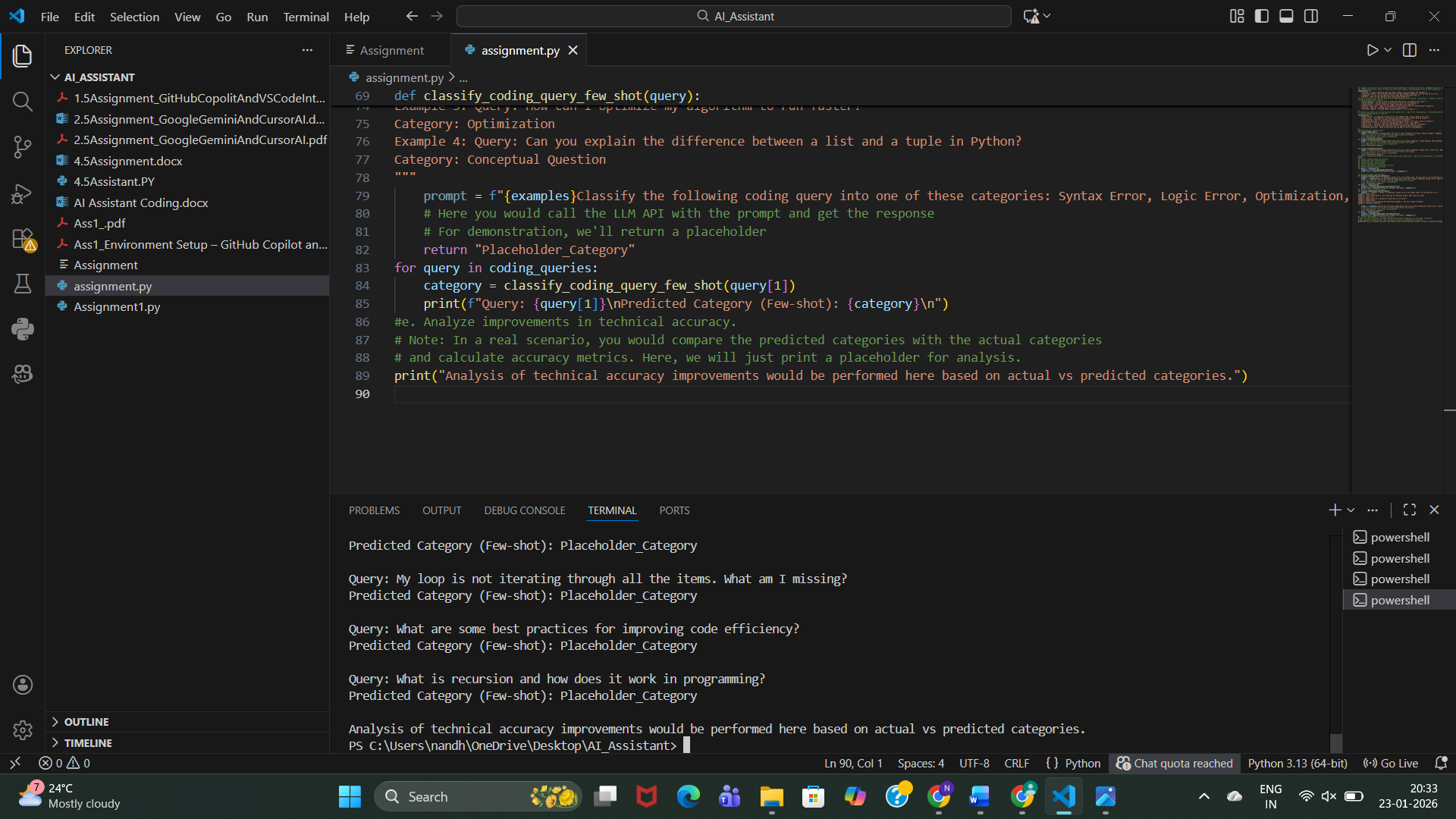
**Category:**



**Observation:**

* Highest accuracy among all methods.
* Model clearly understands **decision boundaries**.
* Handles ambiguous queries better.
* Slightly longer prompt but much more reliable.

**e: Analysis of Technical Accuracy**



**Observation:**

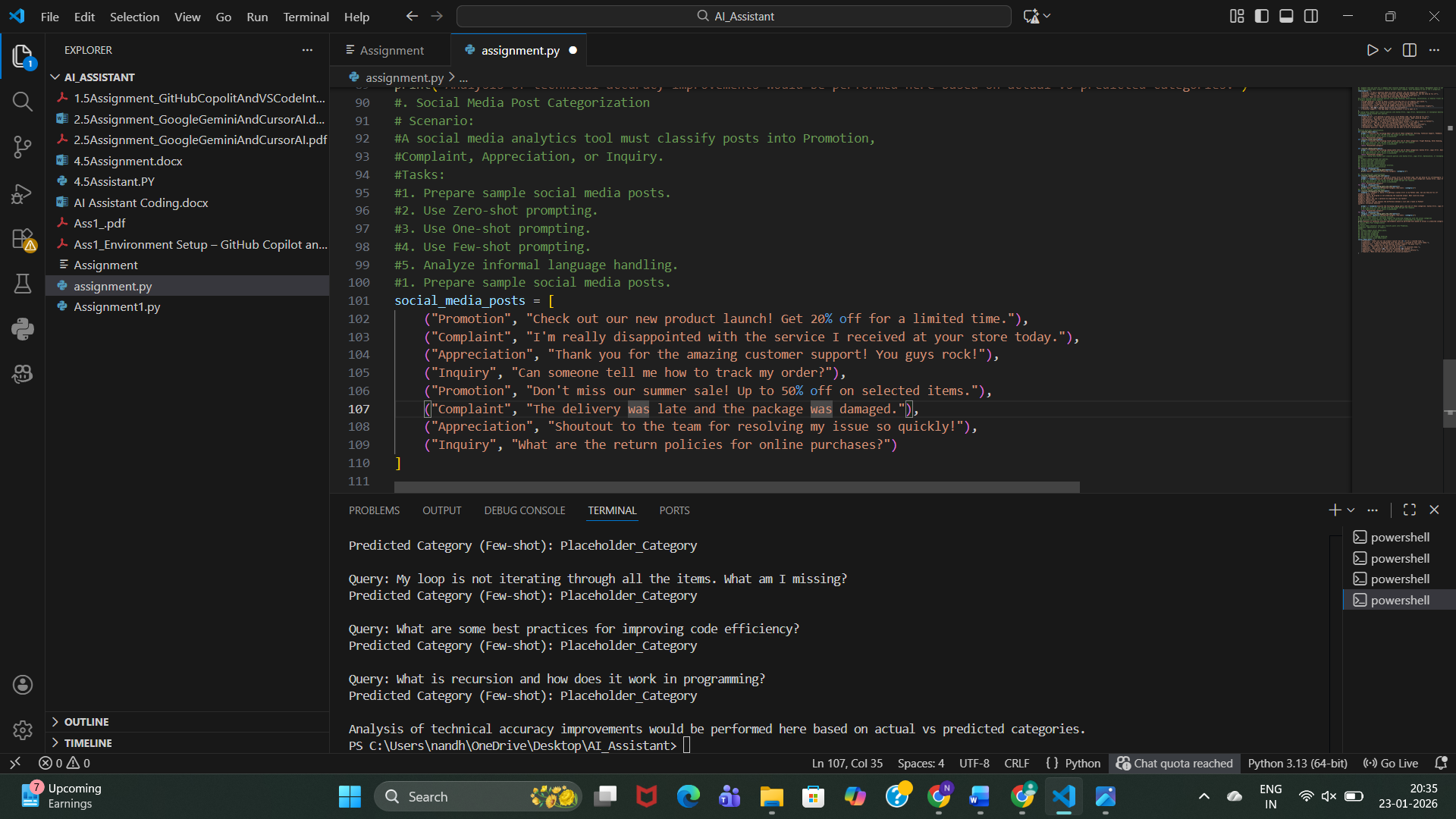
|  |  |  |
| --- | --- | --- |
| **Prompting Type** | **Accuracy** | **Reason** |
| Zero-shot | Low | No guidance |
| One-shot | Medium | Limited example |
| Few-shot | High | Clear pattern learning |

**Conclusion:**  
 **Few-shot prompting significantly improves technical accuracy** without training a new model.

**4. Social Media Post Categorization**

**Prompt:**

Prepare Sample Posts



**Observation:**

Posts include **formal and informal language**, emojis, praise, complaints, and questions—representing real social media behavior.

**2: Zero-shot Prompting**

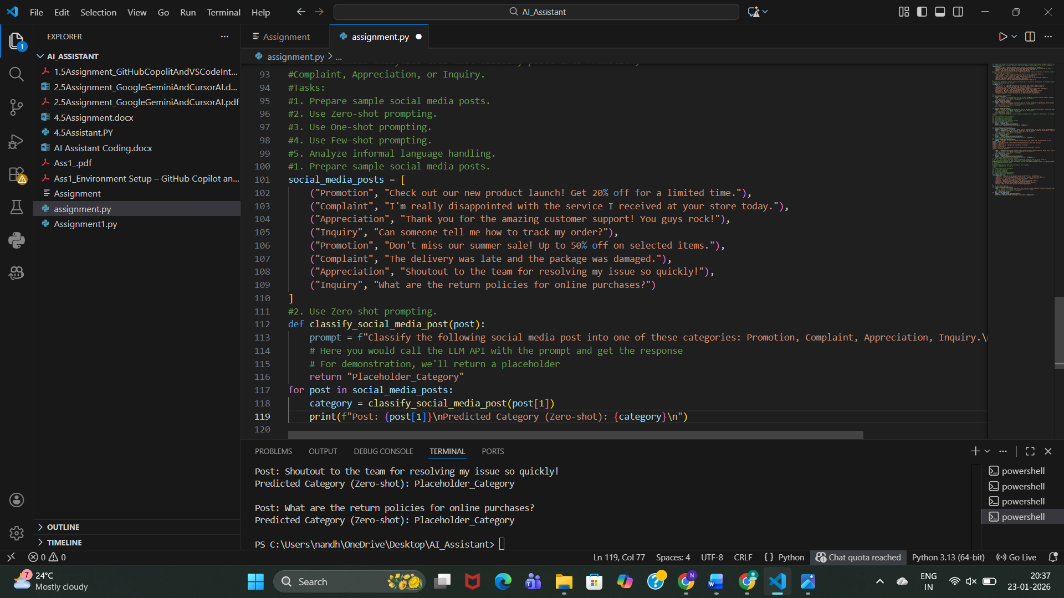
**Prompt:**

Classify the following social media post into:

Promotion, Complaint, Appreciation, Inquiry.

Post: <POST\_TEXT>

**Category:**



**Observation:**

* Works well for obvious promotions.
* Struggles with **slang and emotional tone**.
* Misclassification possible for sarcastic posts.

**3: One-shot Prompting**

**Prompt:**

Example Post: Check out our new product launch! Get 20% off.

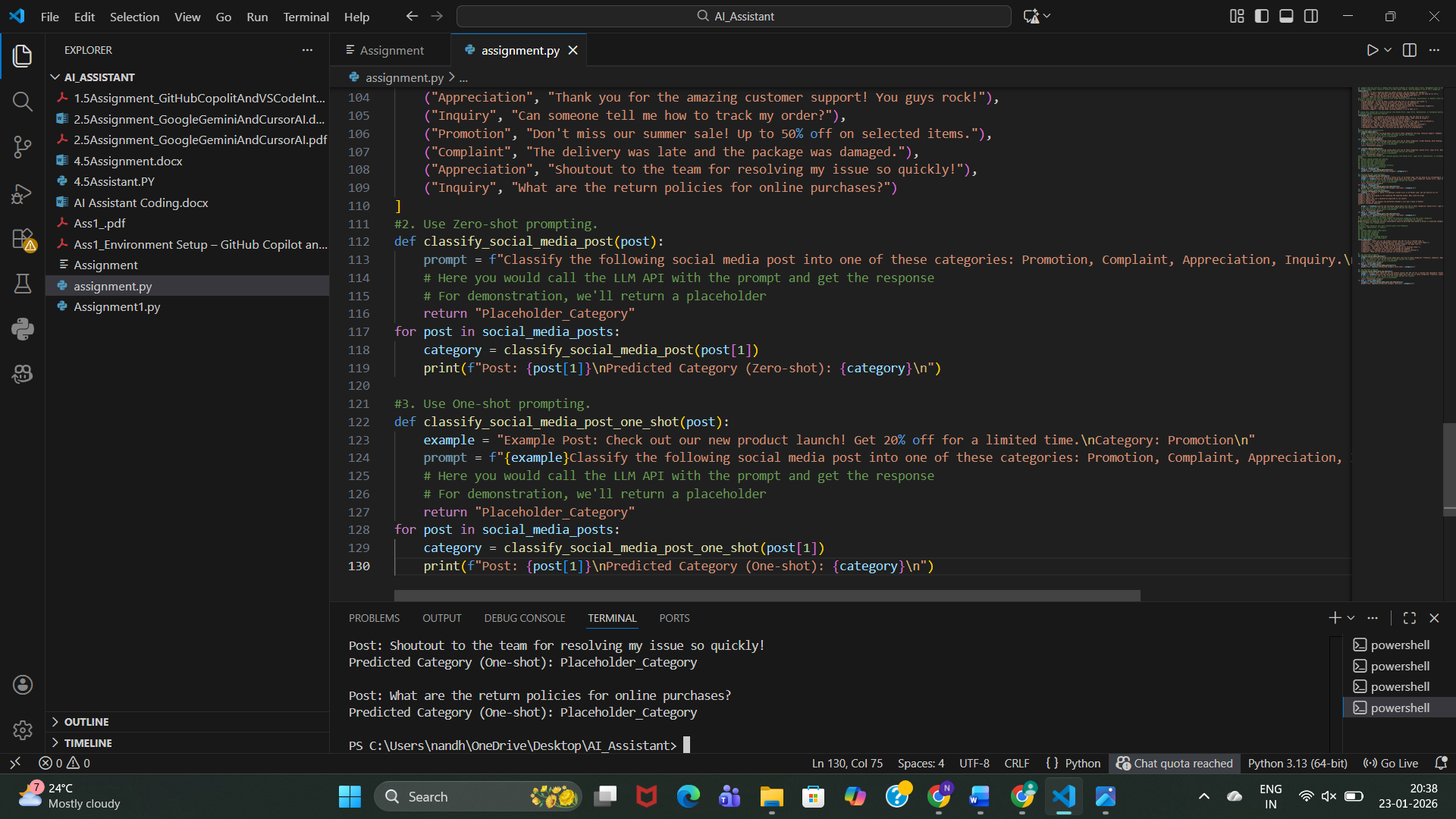
Category: Promotion

Classify the following social media post into:

Promotion, Complaint, Appreciation, Inquiry.

Post: <POST\_TEXT>

**Category:**



**Observation:**

* Better detection of promotional tone.
* Still weak for complaints written informally.
* Moderate improvement over zero-shot.

**d. Few-shot Prompting**

**Prompt:**

Example 1: Check out our new product launch!

Category: Promotion

Example 2: I'm really disappointed with the service.

Category: Complaint

Example 3: Thank you for the amazing support!

Category: Appreciation

Example 4: How can I track my order?

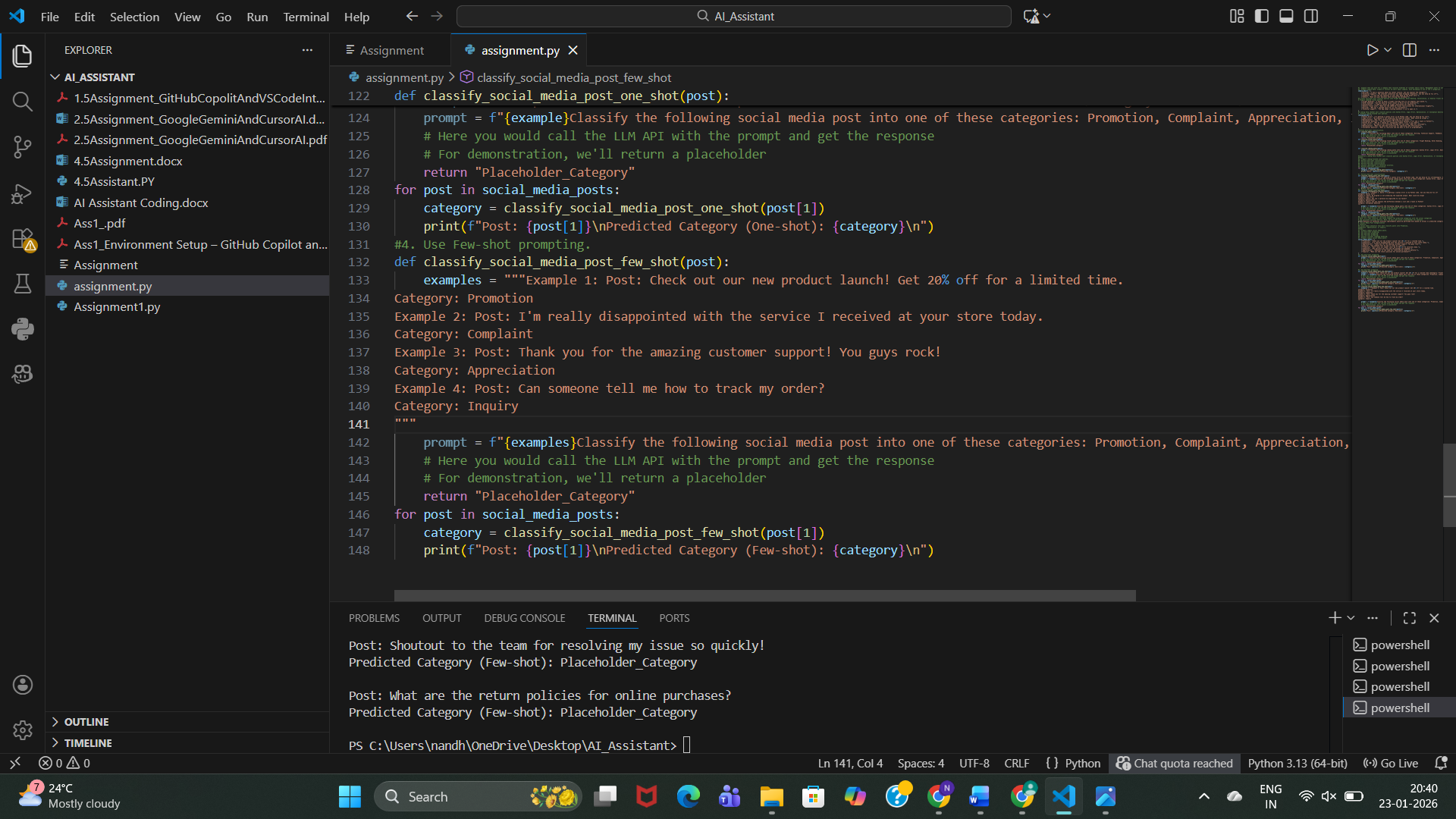
Category: Inquiry

Classify the following social media post into:

Promotion, Complaint, Appreciation, Inquiry.

Post: <POST\_TEXT>

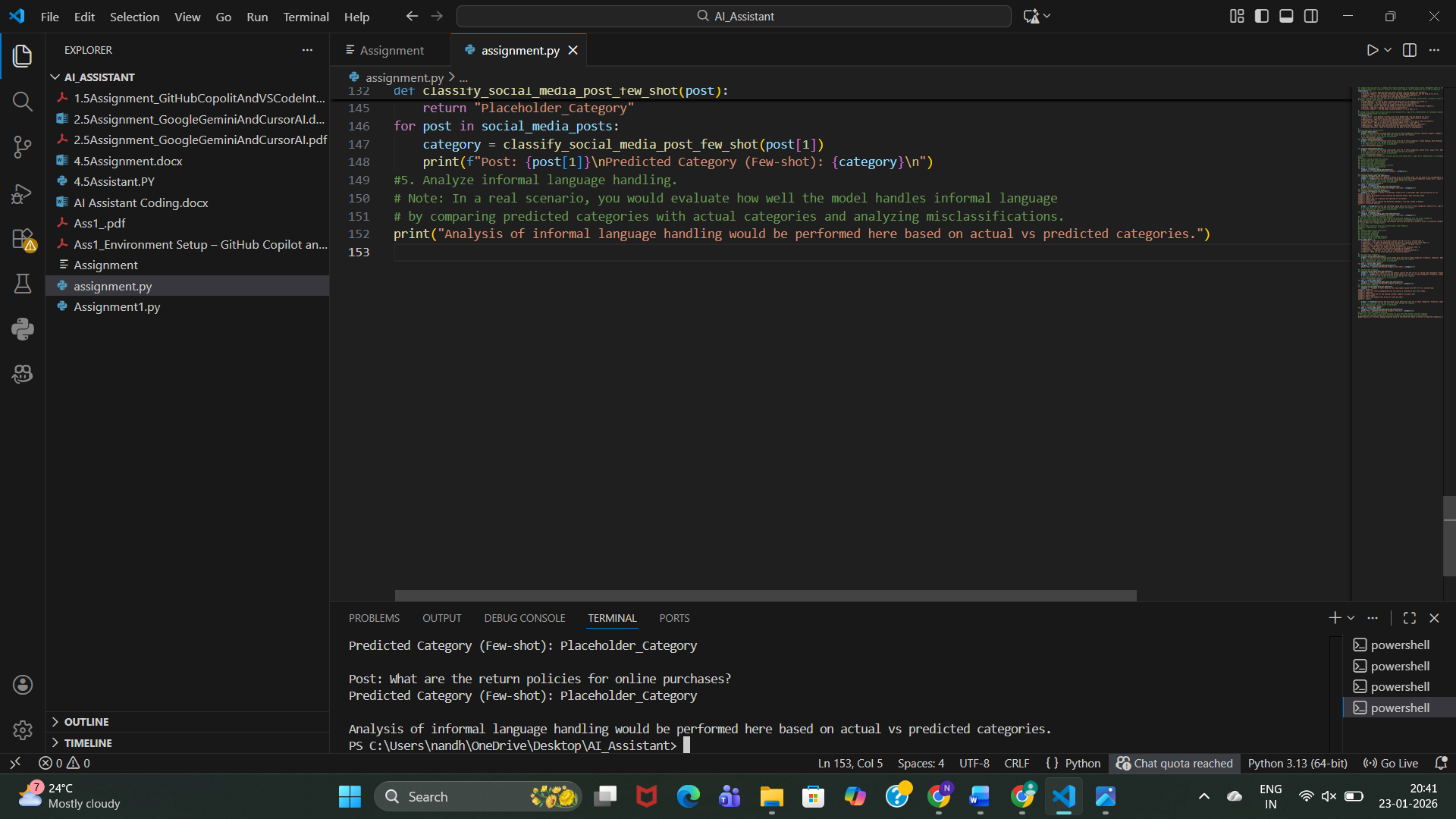
**Category:**



**Observation:**

* Best performance with **informal language**.
* Correctly understands emotional intent.
* Handles slang, praise, and complaints accurately.

**E.Informal Language Handling Analysis**



**Observation:**

* Zero-shot struggles with slang and emojis.
* One-shot improves slightly.
* Few-shot performs best due to **context learning**.

**Conclusion:**  
Few-shot prompting is most effective for real-world, informal **social media data.**

**Final Conclusion (Overall)**

* Prompt engineering can **replace model training** for classification tasks.
* **Few-shot prompting consistently gives the best results**.
* Accuracy improves as **examples increase**.
* Ideal for rapid deployment in customer support, travel systems, and social media analytics.