AIG02

Solving Your Business Problems Using Prompts and LLMs in SAP Generative AI Hub

EXERCISES AND SOLUTIONS

Course Version: 11
Exercise Duration: 2 Hours

SAP Copyrights, Trademarks and Disclaimers

Typographic Conventions

American English is the standard used in this handbook.

The following typographic conventions are also used.

This information is displayed in the instructor's presentation	
Demonstration	
Procedure	
Warning or Caution	
Hint	
Related or Additional Information	
Facilitated Discussion	
User interface control	*Example text*
Window title	*Example text*

Contents

Unit 1
Exercise 1

Develop a Prompt in SAP AI Launchpad

Business Example

Facility solutions company receives thousands of mails pertaining to customers' requests, complaints, and other messages. The company maintains internal applications to address customer requests and complaints and prioritize them to address these requests in a timely and correct manner.

However, categorizing these mails include manual process to transfer data from mails to internal applications, categorizing, and then prioritizing these tasks.

They turn towards generative AI hub to address this problem.

Developing a prompt to solve a business problem is a stepwise ideation process.

In this exercise, we will develop a prompt to address this problem.

Prerequisites

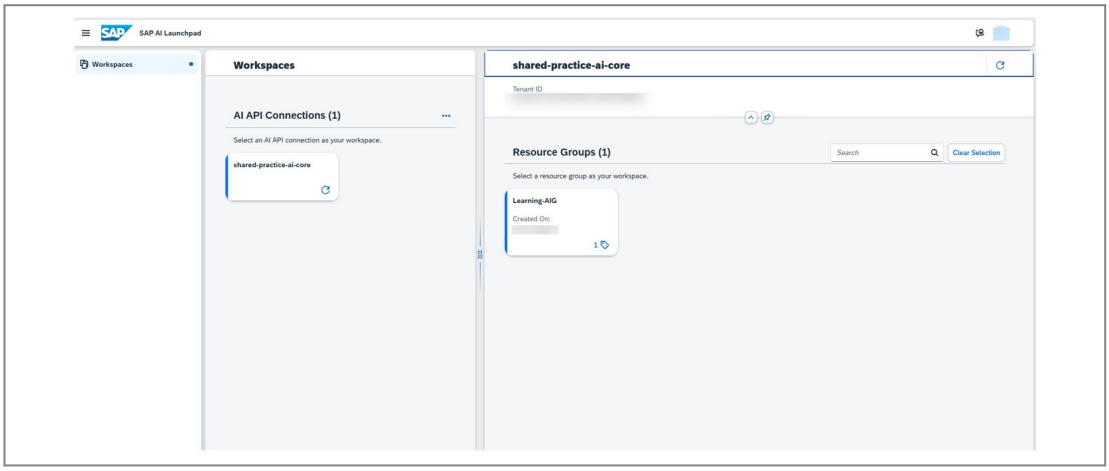
Login credentials for SAP AI Launchpad. Refer to the SSG.

> Caution:
> This SAP's generative AI hub use is limited to the set exercises in the handbook provided by SAP practice system; all other activities are not permitted. User activity may be checked at any time for unauthorized activities. Please read SAP AI Terms and Conditions here .

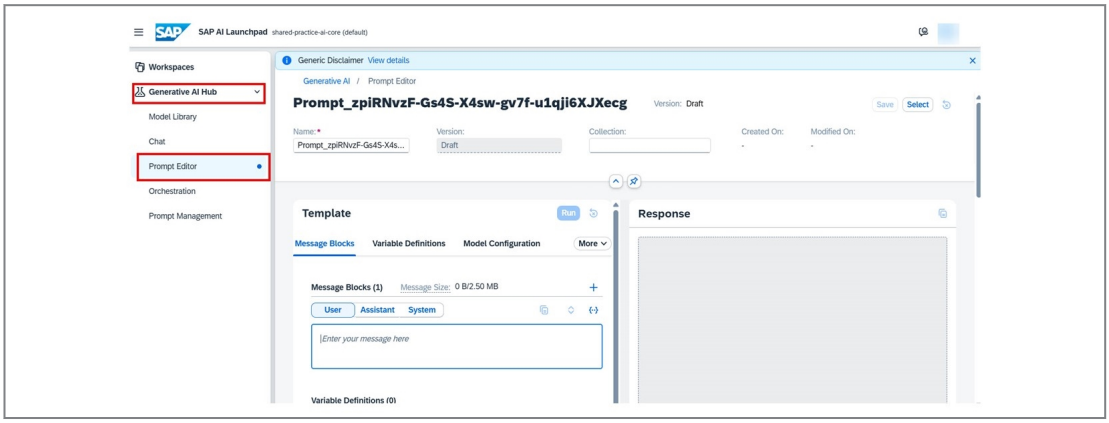
Task 1: Develop a Basic Prompt

We will now find the urgency and sentiment in an incoming mail.

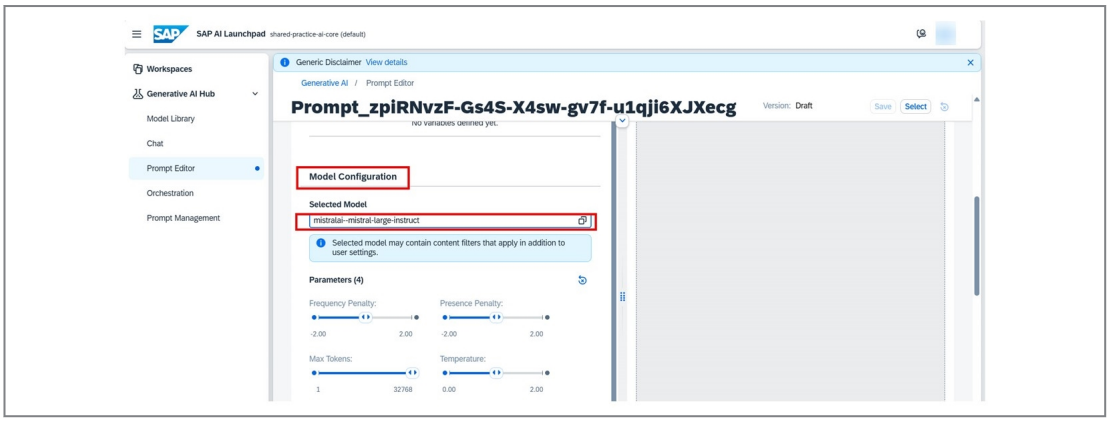
1. Log into Generative AI Hub , you will see the SAP AI Launchpad. Ensure that the **Learning-AIG** resource group is selected. After a few moments, Generative AI Hub option is displayed in the left pane.



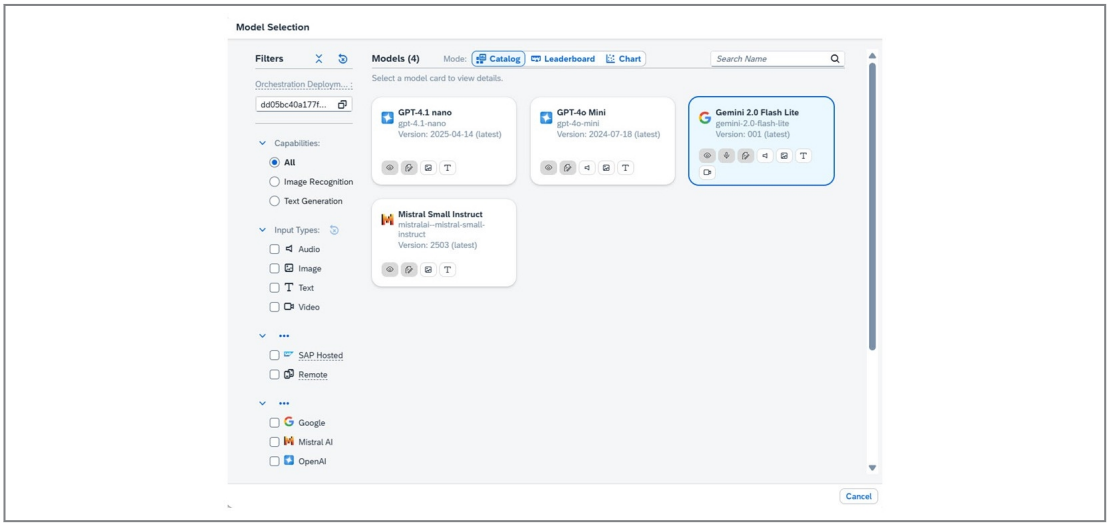
2. Expand the Generative AI Hub and then select *Prompt Editor* in the left pane.



3. Navigate downward and select *Model Configuration*. Choose the appropriate model.



4. The *Model Selection* dialog box is displayed. You can also use the search bar to find the correct model.

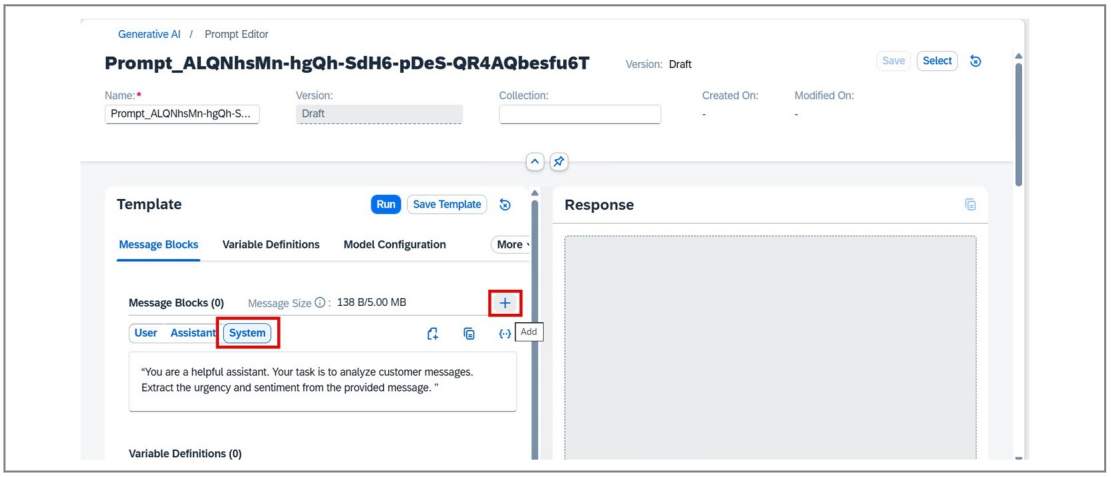


5. For this exercise, we start with using a Google model. Select the latest Google model.

6. In **Message Blocks,** select the **System** role. Copy the following prompt and paste it in the System text box in Message Blocks.

```
Prompt:
"You are a helpful assistant. Your task is to analyze customer
messages. Extract the urgency and sentiment from the provided message.
"
```

7. Select **Add Role**, to add the user role.



8. Ensure that the User role is selected. Copy the following prompt and paste it in the User text box in Message Blocks.

```
text box in Message Blocks.
"
Analyze the following message:
---
Subject: Urgent HVAC System Repair Needed

Dear Support Team,

I hope this message finds you well. My name is [Sender], and I am
reaching out to you from [Residential Complex Name], where I have been
residing for the past few years. I have always appreciated the
meticulous care and attention your team provides in maintaining our
facilities.

However, I am currently facing a pressing issue with the HVAC system
in my apartment. Over the past few days, the system has been
malfunctioning, resulting in inconsistent temperatures and, at times,
complete shutdowns. Given the current weather conditions, this has
become quite unbearable and is affecting my daily routine
significantly.

I have attempted to troubleshoot the problem by resetting the system
and checking the thermostat settings, but these efforts have not
yielded any improvement. The situation seems to be beyond my control
and requires professional intervention.

I kindly request that a repair team be dispatched immediately to
address this urgent issue. The urgency of the matter cannot be
overstated, as it is impacting not only my comfort but also my ability
to carry out daily activities effectively.

Thank you for your prompt attention to this matter. I look forward to
your swift response and resolution.

Best regards,
[Sender]
"
```

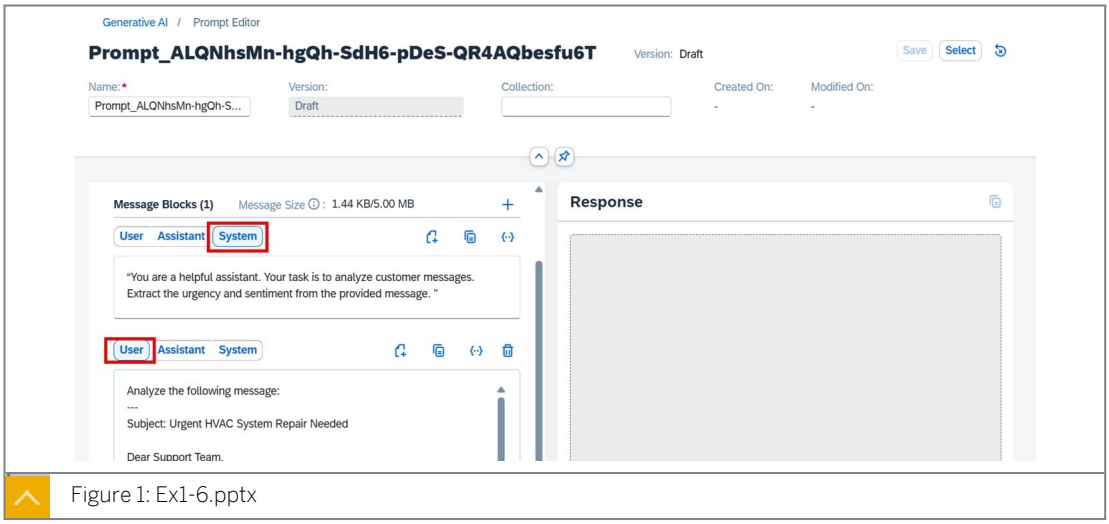
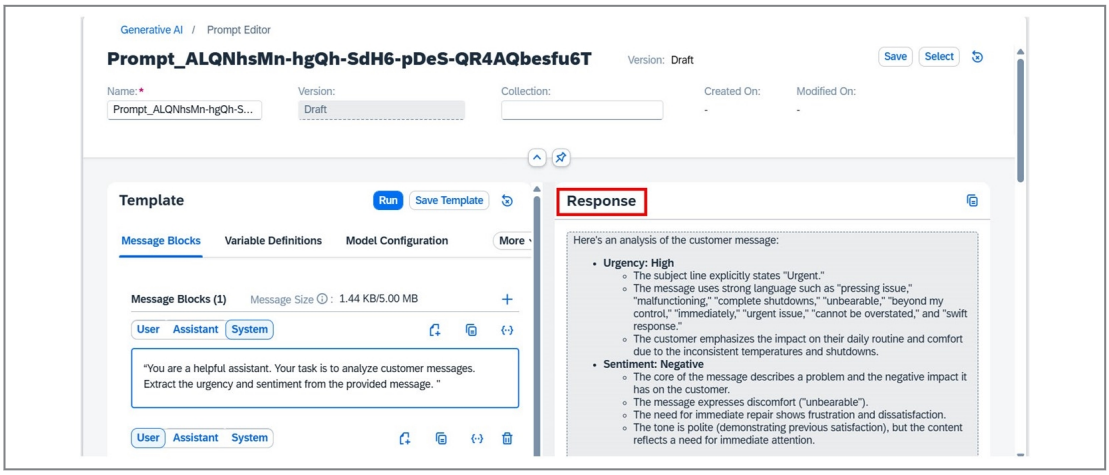
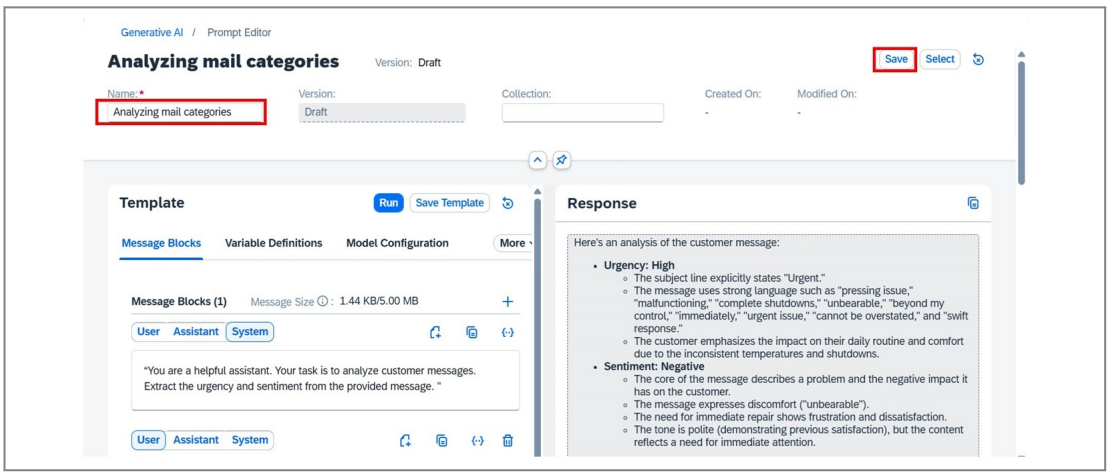


Figure 1: Ex1-6.pptx

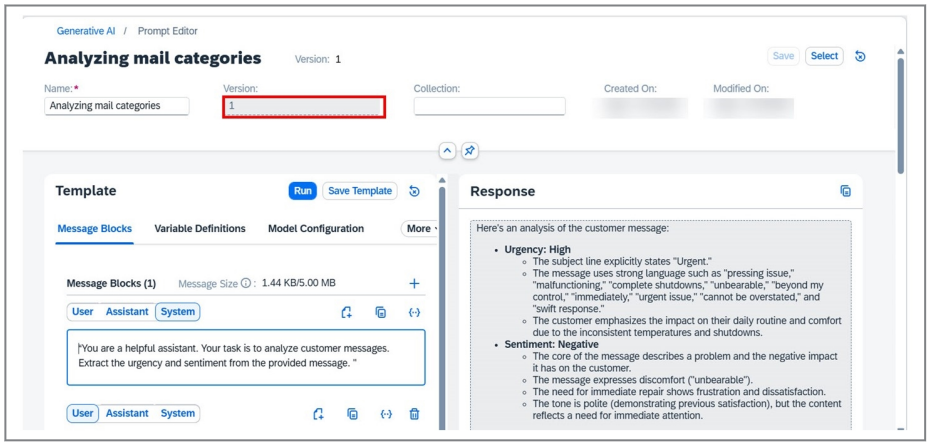
9. Click the **Run** button. You can see the Urgency and the Sentiment of the mail is generated in the **Response** text box. However, as we can see this response is too lengthy. Our objective is to assign urgency and sentiment to the mail that can be used as an input for software applications. This lengthy response is not useful for us as of now. Let us continue to develop it.



10. Give a proper name to your prompt and save it, as shown in the following screenshot.



11. The message is saved for future development with version 1.



Note:
You may get a slightly different response to the one shown here and in all the remaining responses of models shown in this learning journey.

When you execute the same prompt in your machine, an LLM produces varying outputs due to its probabilistic nature, temperature setting, and non-deterministic architecture, leading to different responses even with slight setting changes or internal state shifts.

Task 2: Assign Values to Urgency and Sentiment to the Existing Prompt

We will now assign some basic urgency and sentiment values and execute the prompt.

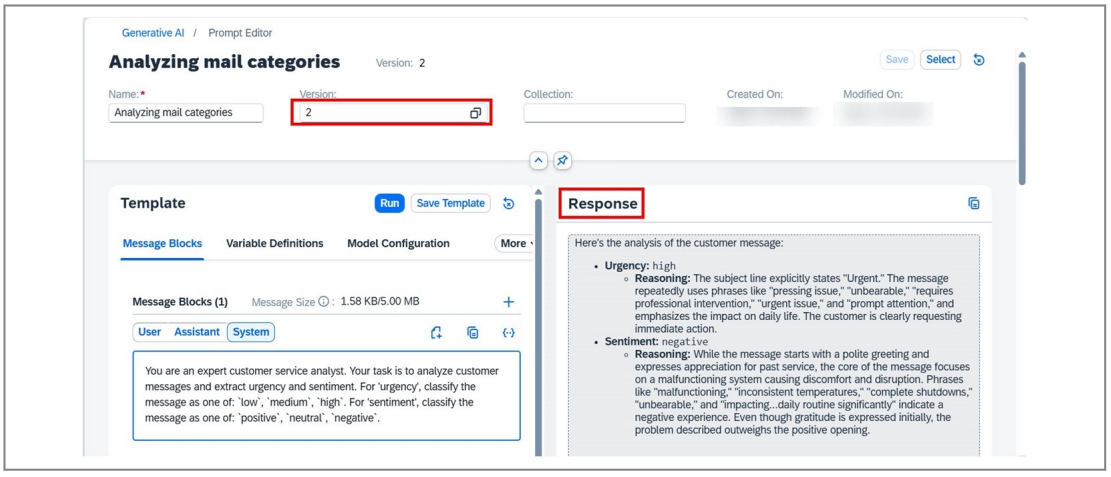
1. Update the saved prompt in the Prompt editor as to assign value to urgency and sentiment. Use the following prompt in the **System** role.

```
You are an expert customer service analyst. Your task is to analyze
customer messages and extract urgency and sentiment. For 'urgency',
classify the message as one of: `low`, `medium`, `high`. For
'sentiment', classify the message as one of: `positive`, `neutral`,
`negative`.
```

   Continue the same prompt in the **User** role.

2. Copy the prompt and paste it in the **System** role in the **Message Blocks** text box, and then click **Run.** You get a response in the **Response** field. You can see that the urgency and sentiment of the mail is generated. These values are based on assigned values.

3. Click **Save** button to save the updated version of the prompt. Version 2 of the prompt is saved.



Task 3: Generate JSON output

We need the output of these values in a format that can be used as an input for software. We will use the JSON output.

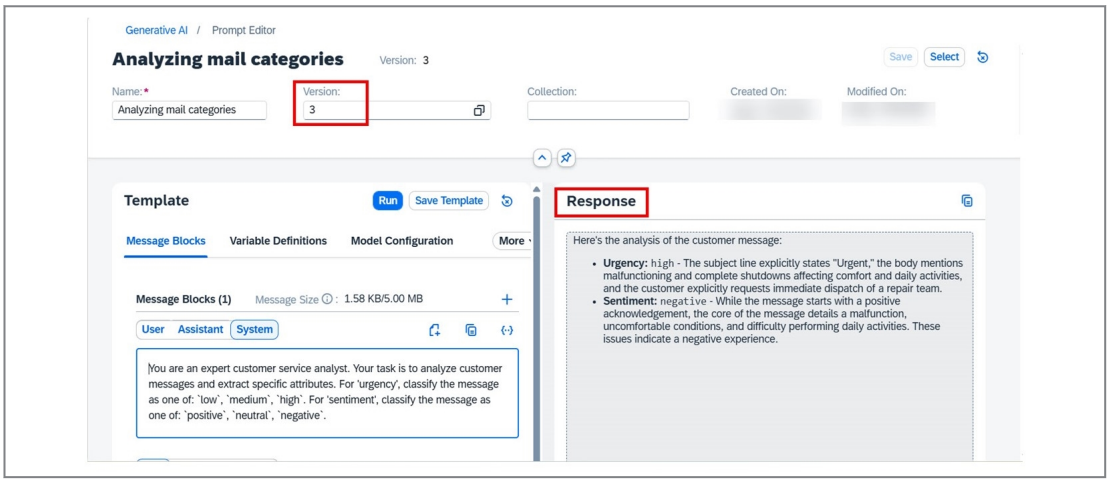
1. Update the **System** role prompt with the following text.

```
You are an expert customer service analyst. Your task is to analyze
customer messages and extract specific attributes. For 'urgency',
classify the message as one of: `low`, `medium`, `high`. For
'sentiment', classify the message as one of: `positive`, `neutral`,
`negative`.
```

   The prompt is the same as the previous prompt, but the last few sentences now request a JSON string.

2. Copy the prompt and paste it in the **System** role in the **Message Blocks** text box, and then click **Run**. You can see the JSON output.

3. Click Save. The following screenshot shows version 3 of the prompt.



Task 4: Ensure that the JSON formatting is correct

Even with instructions for JSON, LLMs can sometimes include extraneous characters like markdown code blocks (json ...) or extra whitespace, which can break automated parsing. This step adds explicit instructions to ensure a clean JSON string that can be parsed by an application.

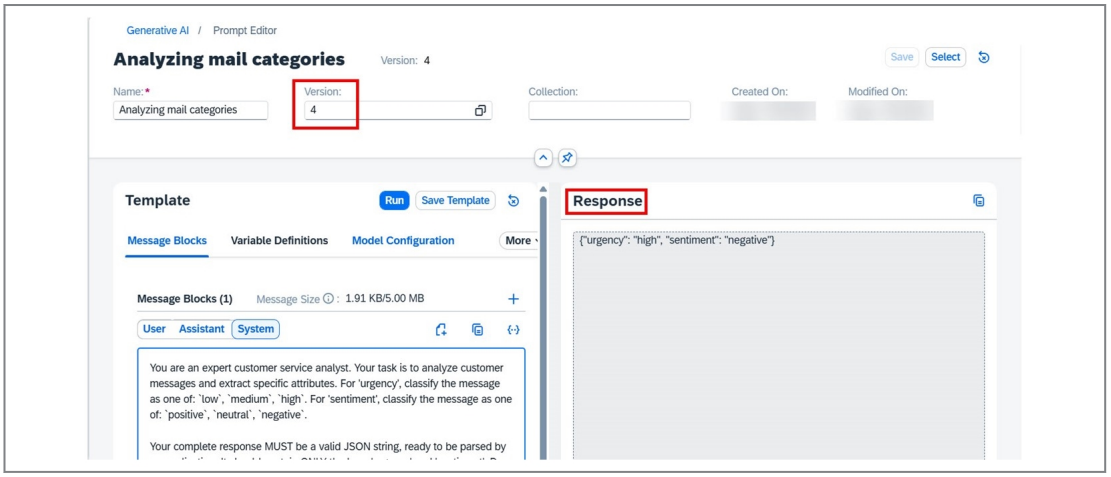
1. Update the **System** role prompt with the following text.

```
You are an expert customer service analyst. Your task is to analyze
customer messages and extract specific attributes. For 'urgency',
classify the message as one of: `low`, `medium`, `high`. For
'sentiment', classify the message as one of: `positive`, `neutral`,
`negative`.

Your complete response MUST be a valid JSON string, ready to be parsed
by an application. It should contain ONLY the keys 'urgency' and
'sentiment'. Do not include any other text, explanations, or
formatting like markdown code blocks (e.g., ```json). Ensure there are
no newlines or unnecessary whitespaces outside the JSON structure.
```

   This prompt gives clear instruction to provide a clean format without any quotes or whitespaces.

2. Copy the prompt and paste it in the **System** role in the **Message Blocks** text box, and then click **Run**. You can see that the JSON output is clear. This is ready for software consumption.

3. Click **Save.** The following screenshot shows version 4 of the prompt.



Task 5: Generate Simple Categories Based on Business Functions

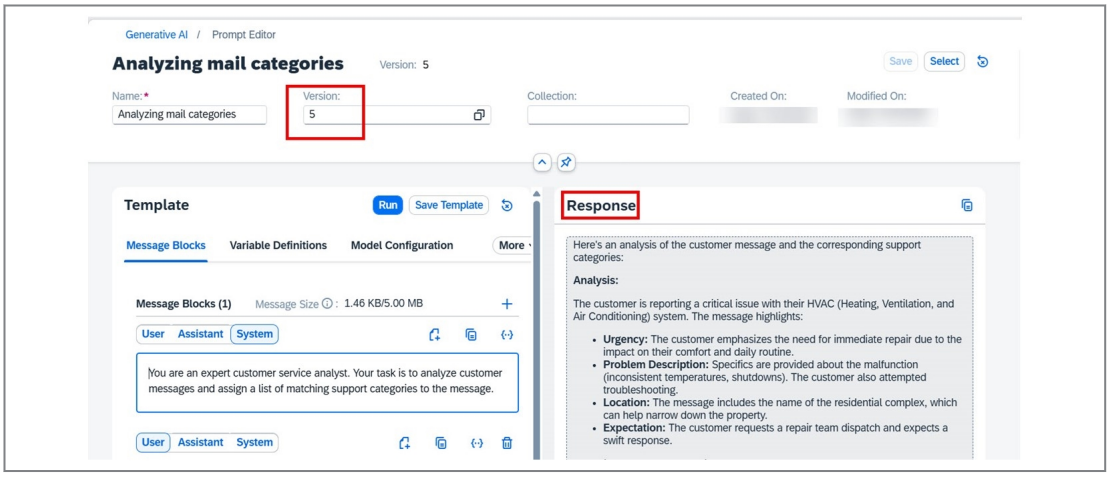
We have associated urgency and sentiment to a mail. However, we also need more tags for each message to categorize them for business needs.

1. Start with a simple prompt:

```
"You are an expert customer service analyst. Your task is to analyze
customer messages and assign a list of matching support categories to
the message."
```

2. Copy the prompt and paste it in the **System** role in the **Message Blocks** text box, and then click **Run**. You can see categories are assigned to the message.

3. Click **Save.** The following screenshot shows version 5 of the prompt.



Task 6: Assign Values to Categories From a List

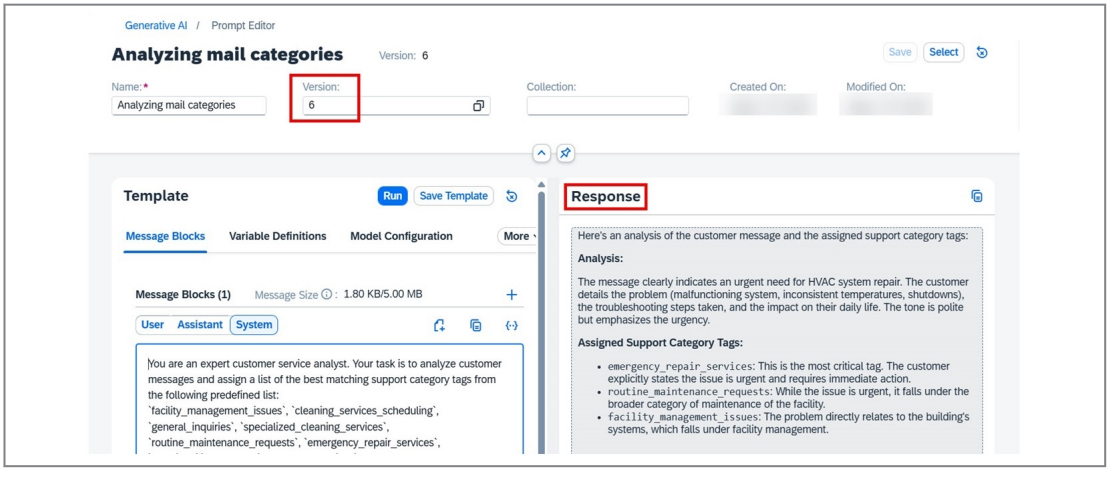
To ensure consistency and alignment with the company's internal processing, we need LLM to select categories from a specific, predefined list. Ensuring the output corresponds to business requirements is an essential aspect of solving a business problem.

1. Use the following prompt:

```
"You are an expert customer service analyst. Your task is to analyze
customer messages and assign a list of the best matching support
category tags from the following predefined list:
`facility_management_issues`, `cleaning_services_scheduling`,
`general_inquiries`, `specialized_cleaning_services`,
`routine_maintenance_requests`, `emergency_repair_services`,
`sustainability_and_environmental_practices`,
`training_and_support_requests`, `quality_and_safety_concerns`,
`customer_feedback_and_complaints`."
```

2. Copy the prompt and paste it in the **System** role in the **Message Blocks** text box, and then click **Run**. You can see that categories are assigned to values from the list. They are streamlined for business processing.

3. Click **Save.** The following screenshot shows version 6 of the prompt.



Task 7: Generate JSON Output for Category Values

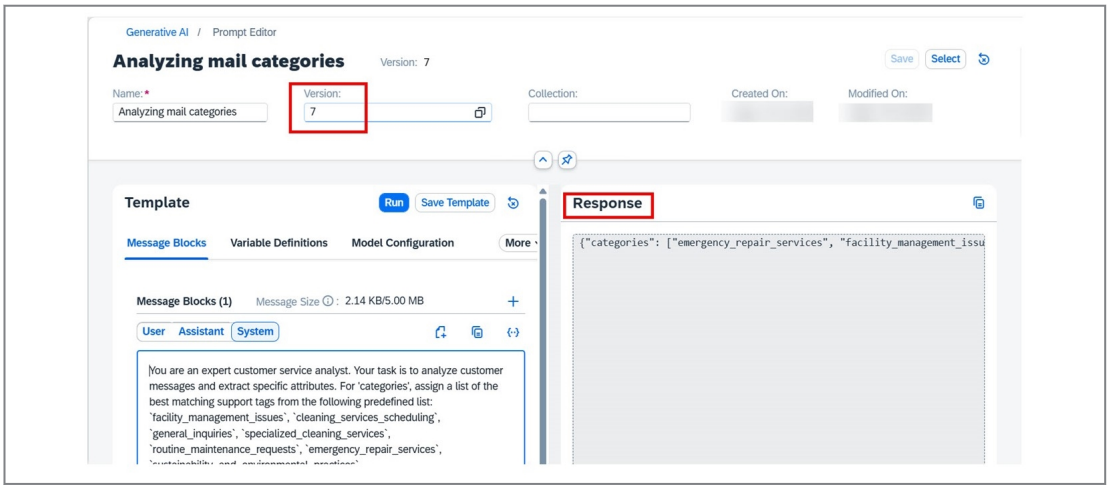
Like Step 3, we need the category values in a JSON output for programmatic processing. We will also reiterate the importance of clean JSON formatting.

1. Use the following prompt:

```
"You are an expert customer service analyst. Your task is to analyze
customer messages and extract specific attributes. For 'categories',
assign a list of the best matching support tags from the following
predefined list:
`facility_management_issues`, `cleaning_services_scheduling`,
`general_inquiries`, `specialized_cleaning_services`,
`routine_maintenance_requests`, `emergency_repair_services`,
`sustainability_and_environmental_practices`,
`training_and_support_requests`, `quality_and_safety_concerns`,
`customer_feedback_and_complaints`.
Your complete response MUST be a valid JSON string, ready to be parsed
by an application. It should contain ONLY the key 'categories'. Do not
include any other text, explanations, or formatting like markdown code
blocks. Ensure there are no newlines or unnecessary whitespaces
outside the JSON structure."
```

2. Copy the prompt and paste it in the **System** role in the **Message Blocks** text box, and then click **Run.** You can see categories in the JSON output that can be processed in software applications.

3. Click **Save.** The following screenshot shows version 7 of the prompt.



Task 8: Combine All Steps

We have taken a step-by-step approach to arrive at proper values of urgency, sentiment, and categories in JSON format. Now, let us combine all these steps into a single prompt.

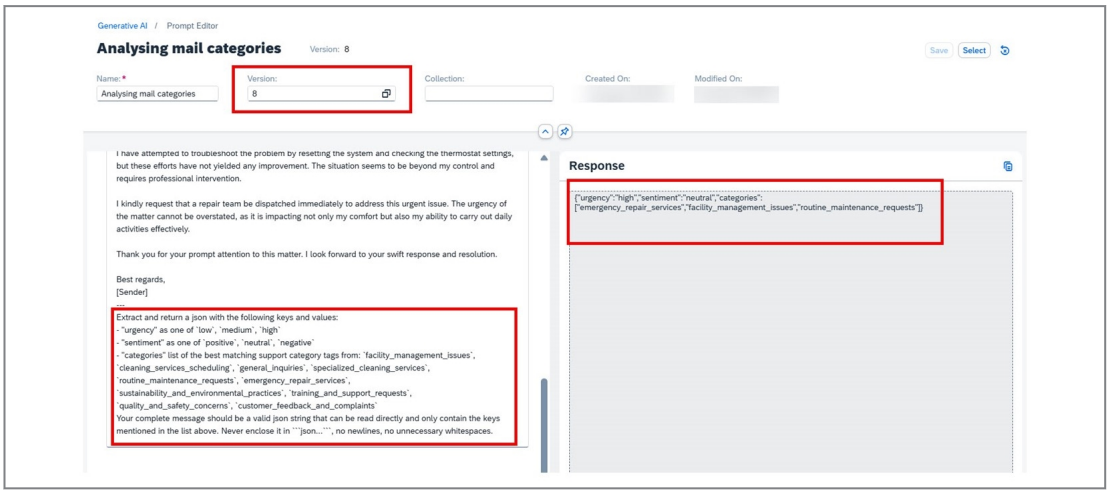
1. Use the following prompt:

```
"You are an expert customer service analyst for a facility management
company. Your task is to analyze incoming customer messages and
extract specific attributes for automated processing.

For 'urgency', classify the message as one of: `low`, `medium`, `high`.
For 'sentiment', classify the message as one of: `positive`,
`neutral`, `negative`.
For 'categories', assign a list of the best matching support tags from
the following predefined list:
`facility_management_issues`, `cleaning_services_scheduling`,
`general_inquiries`, `specialized_cleaning_services`,
`routine_maintenance_requests`, `emergency_repair_services`,
`sustainability_and_environmental_practices`,
`training_and_support_requests`, `quality_and_safety_concerns`,
`customer_feedback_and_complaints`.
```

```
Your complete response MUST be a valid JSON string, ready to be parsed
by an application. It should contain ONLY the keys 'urgency',
'sentiment', and 'categories'. Do not include any other text,
explanations, or formatting like markdown code blocks (e.g., ```json).
Ensure there are no newlines or unnecessary whitespaces outside the
JSON structure."
```

2. Copy the prompt and paste it in the **System role in Message Blocks** text box, and then click **Run.** You can see the consolidated output that assigns urgency, sentiment, and categories to customer messages that can be used in software applications.

3. Click **Save.** The following screenshot shows version 8 of the prompt.



Congratulations. You have successfully developed a prompt to solve a business problem.

Unit 1
Solution 1

Develop a Prompt in SAP AI Launchpad

Business Example

Facility solutions company receives thousands of mails pertaining to customers' requests, complaints, and other messages. The company maintains internal applications to address customer requests and complaints and prioritize them to address these requests in a timely and correct manner.

However, categorizing these mails include manual process to transfer data from mails to internal applications, categorizing, and then prioritizing these tasks.

They turn towards generative AI hub to address this problem.

Developing a prompt to solve a business problem is a stepwise ideation process.

In this exercise, we will develop a prompt to address this problem.

Prerequisites

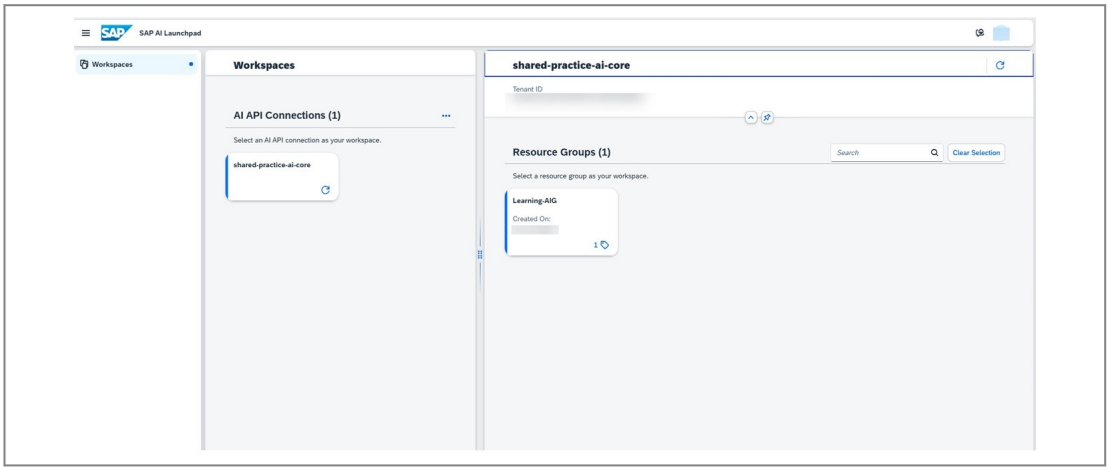
Login credentials for SAP AI Launchpad. Refer to the SSG.

> Caution:
> This SAP's generative AI hub use is limited to the set exercises in the handbook provided by SAP practice system; all other activities are not permitted. User activity may be checked at any time for unauthorized activities. Please read SAP AI Terms and Conditions here .

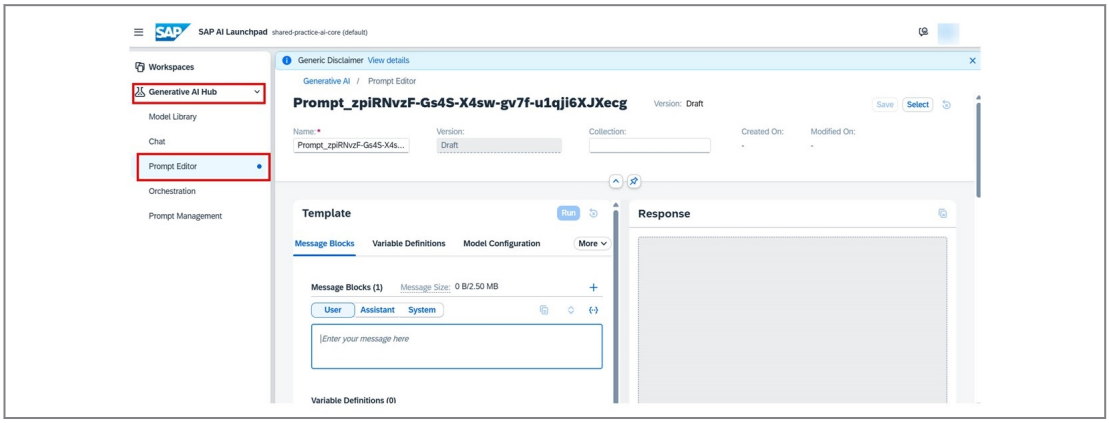
Task 1: Develop a Basic Prompt

We will now find the urgency and sentiment in an incoming mail.

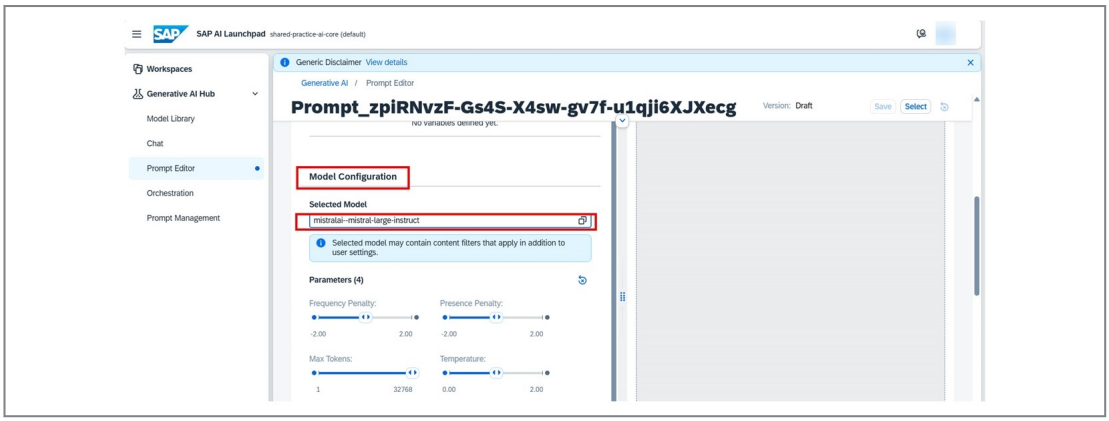
1. Log into Generative AI Hub , you will see the SAP AI Launchpad. Ensure that the **Learning-AIG** resource group is selected. After a few moments, Generative AI Hub option is displayed in the left pane.



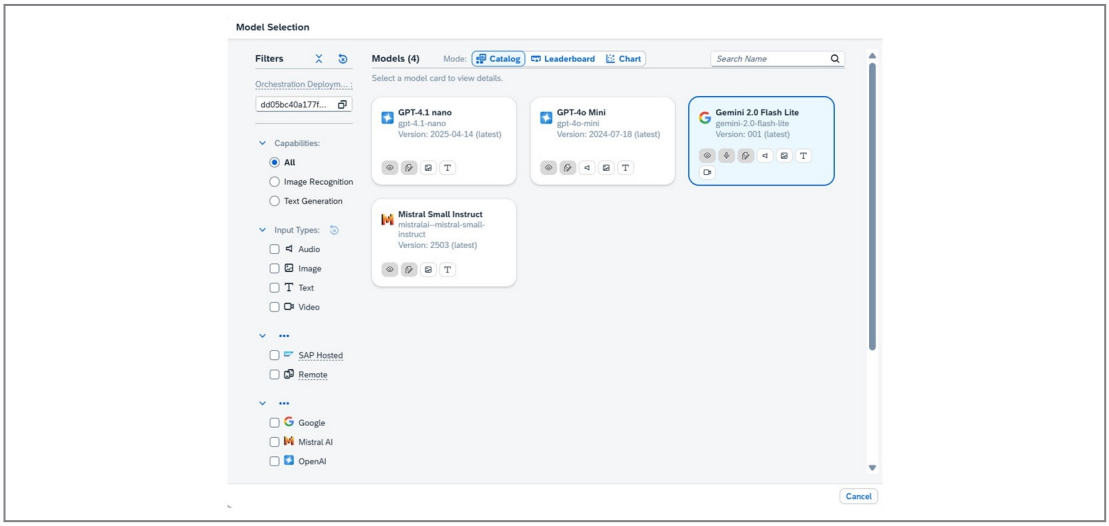
2. Expand the Generative AI Hub and then select *Prompt Editor* in the left pane.



3. Navigate downward and select *Model Configuration*. Choose the appropriate model.



4. The *Model Selection* dialog box is displayed. You can also use the search bar to find the correct model.

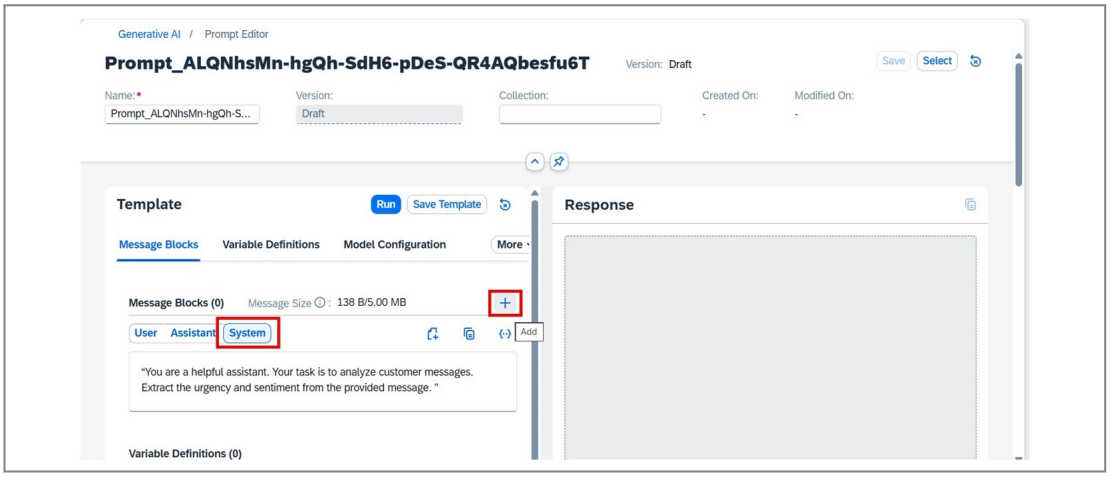


5. For this exercise, we start with using a Google model. Select the latest Google model.

6. In **Message Blocks,** select the **System** role. Copy the following prompt and paste it in the System text box in Message Blocks.

```
Prompt:
"You are a helpful assistant. Your task is to analyze customer
messages. Extract the urgency and sentiment from the provided message.
"
```

7. Select **Add Role**, to add the user role.



8. Ensure that the User role is selected. Copy the following prompt and paste it in the User text box in Message Blocks.

```
text box in Message Blocks.
"
Analyze the following message:
---
Subject: Urgent HVAC System Repair Needed

Dear Support Team,

I hope this message finds you well. My name is [Sender], and I am
reaching out to you from [Residential Complex Name], where I have been
residing for the past few years. I have always appreciated the
meticulous care and attention your team provides in maintaining our
facilities.

However, I am currently facing a pressing issue with the HVAC system
in my apartment. Over the past few days, the system has been
malfunctioning, resulting in inconsistent temperatures and, at times,
complete shutdowns. Given the current weather conditions, this has
become quite unbearable and is affecting my daily routine
significantly.

I have attempted to troubleshoot the problem by resetting the system
and checking the thermostat settings, but these efforts have not
yielded any improvement. The situation seems to be beyond my control
and requires professional intervention.

I kindly request that a repair team be dispatched immediately to
address this urgent issue. The urgency of the matter cannot be
overstated, as it is impacting not only my comfort but also my ability
to carry out daily activities effectively.

Thank you for your prompt attention to this matter. I look forward to
your swift response and resolution.

Best regards,
[Sender]
"
```

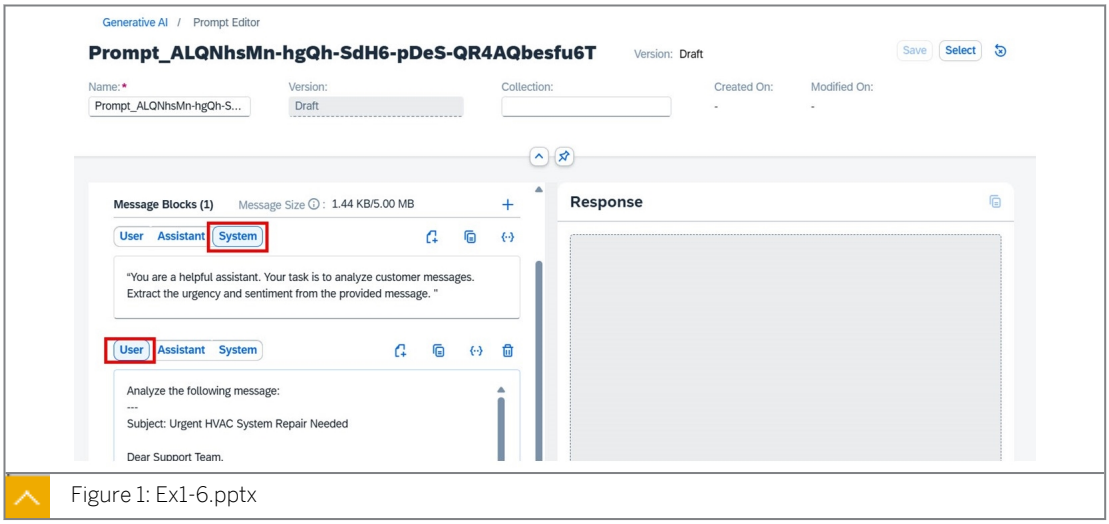
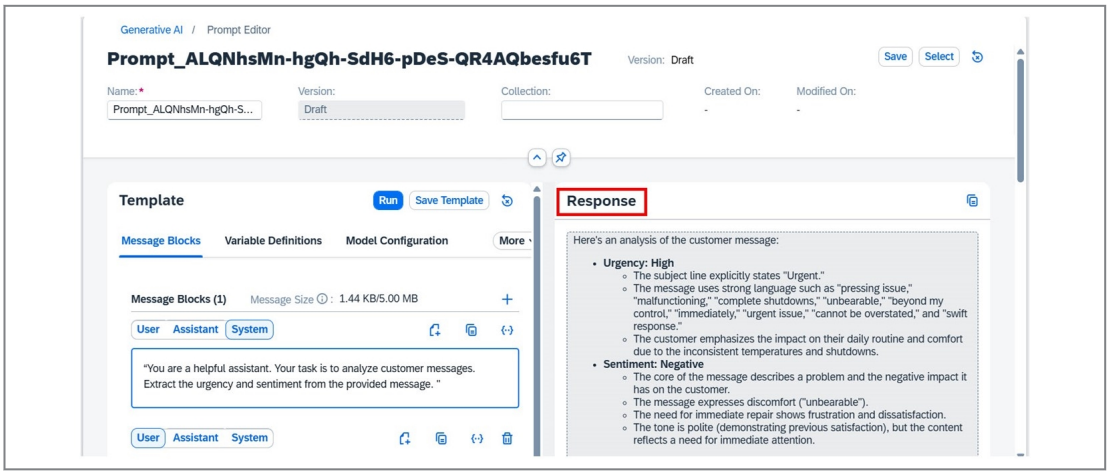
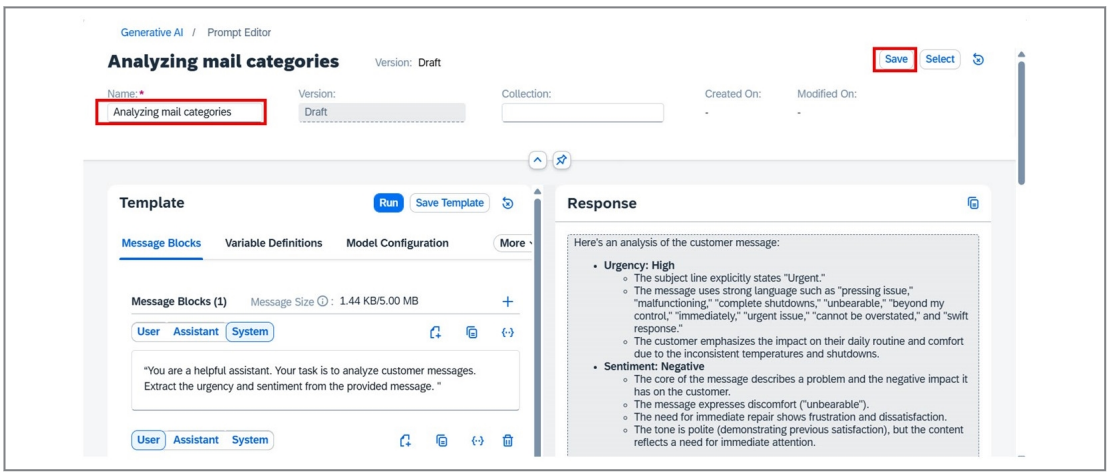


Figure 1: Ex1-6.pptx

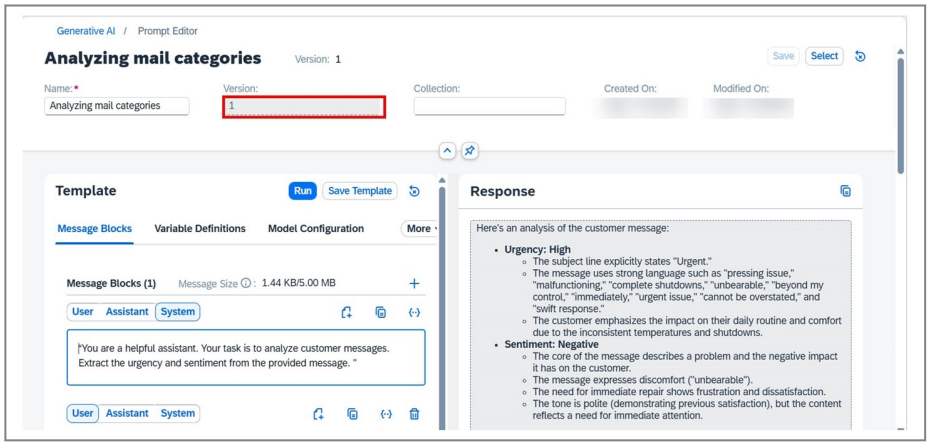
9. Click the **Run** button. You can see the Urgency and the Sentiment of the mail is generated in the **Response** text box. However, as we can see this response is too lengthy. Our objective is to assign urgency and sentiment to the mail that can be used as an input for software applications. This lengthy response is not useful for us as of now. Let us continue to develop it.



10. Give a proper name to your prompt and save it, as shown in the following screenshot.



11. The message is saved for future development with version 1.



Note:
You may get a slightly different response to the one shown here and in all the remaining responses of models shown in this learning journey.

When you execute the same prompt in your machine, an LLM produces varying outputs due to its probabilistic nature, temperature setting, and non-deterministic architecture, leading to different responses even with slight setting changes or internal state shifts.

Task 2: Assign Values to Urgency and Sentiment to the Existing Prompt

We will now assign some basic urgency and sentiment values and execute the prompt.

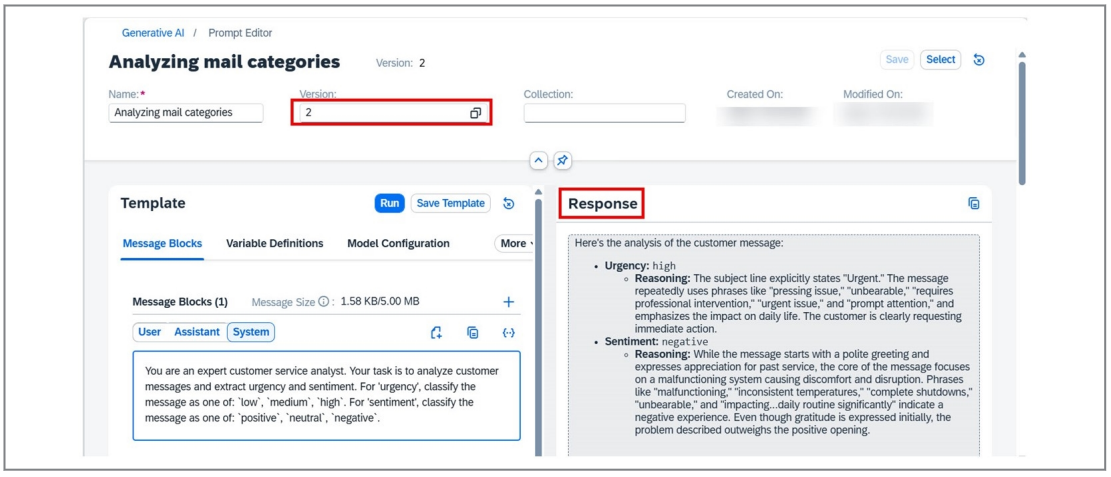
1. Update the saved prompt in the Prompt editor as to assign value to urgency and sentiment. Use the following prompt in the **System** role.

```
You are an expert customer service analyst. Your task is to analyze
customer messages and extract urgency and sentiment. For 'urgency',
classify the message as one of: `low`, `medium`, `high`. For
'sentiment', classify the message as one of: `positive`, `neutral`,
`negative`.
```

   Continue the same prompt in the **User** role.

2. Copy the prompt and paste it in the **System** role in the **Message Blocks** text box, and then click **Run.** You get a response in the **Response** field. You can see that the urgency and sentiment of the mail is generated. These values are based on assigned values.

3. Click **Save** button to save the updated version of the prompt. Version 2 of the prompt is saved.



Task 3: Generate JSON output

We need the output of these values in a format that can be used as an input for software. We will use the JSON output.

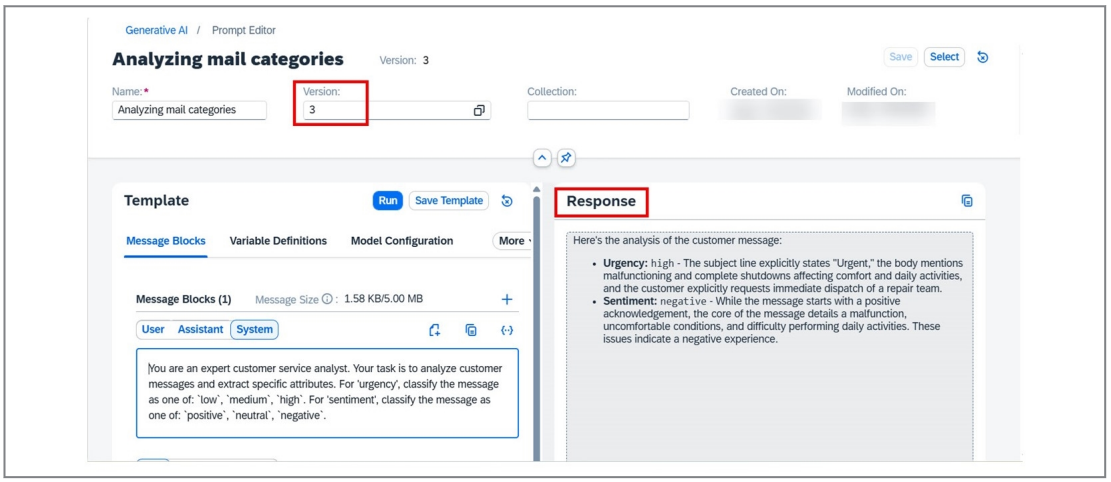
1. Update the **System** role prompt with the following text.

```
You are an expert customer service analyst. Your task is to analyze
customer messages and extract specific attributes. For 'urgency',
classify the message as one of: `low`, `medium`, `high`. For
'sentiment', classify the message as one of: `positive`, `neutral`,
`negative`.
```

   The prompt is the same as the previous prompt, but the last few sentences now request a JSON string.

2. Copy the prompt and paste it in the **System** role in the **Message Blocks** text box, and then click **Run**. You can see the JSON output.

3. Click Save. The following screenshot shows version 3 of the prompt.



Task 4: Ensure that the JSON formatting is correct

Even with instructions for JSON, LLMs can sometimes include extraneous characters like markdown code blocks (json ...) or extra whitespace, which can break automated parsing. This step adds explicit instructions to ensure a clean JSON string that can be parsed by an application.

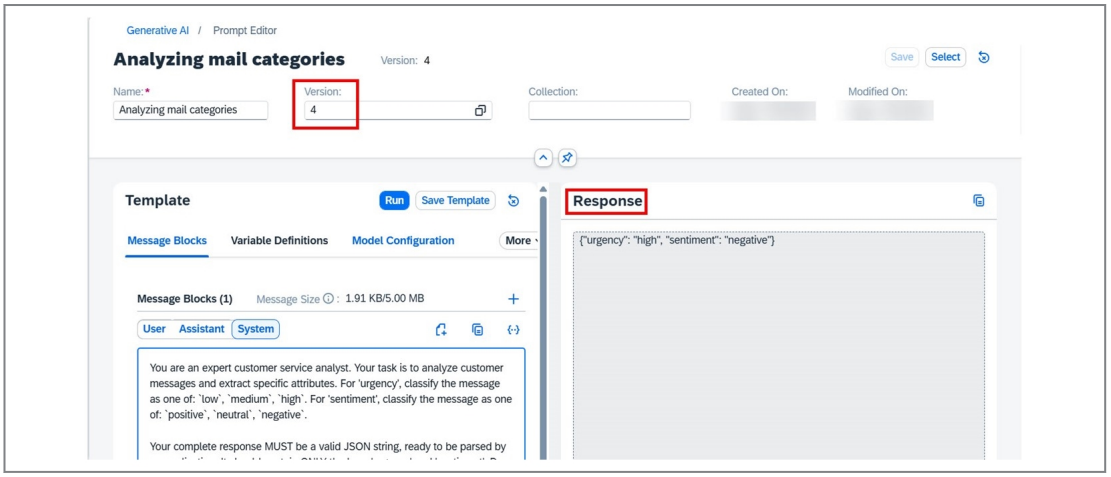
1. Update the **System** role prompt with the following text.

```
You are an expert customer service analyst. Your task is to analyze
customer messages and extract specific attributes. For 'urgency',
classify the message as one of: `low`, `medium`, `high`. For
'sentiment', classify the message as one of: `positive`, `neutral`,
`negative`.

Your complete response MUST be a valid JSON string, ready to be parsed
by an application. It should contain ONLY the keys 'urgency' and
'sentiment'. Do not include any other text, explanations, or
formatting like markdown code blocks (e.g., ```json). Ensure there are
no newlines or unnecessary whitespaces outside the JSON structure.
```

   This prompt gives clear instruction to provide a clean format without any quotes or whitespaces.

2. Copy the prompt and paste it in the **System** role in the **Message Blocks** text box, and then click **Run**. You can see that the JSON output is clear. This is ready for software consumption.

3. Click **Save.** The following screenshot shows version 4 of the prompt.



Task 5: Generate Simple Categories Based on Business Functions

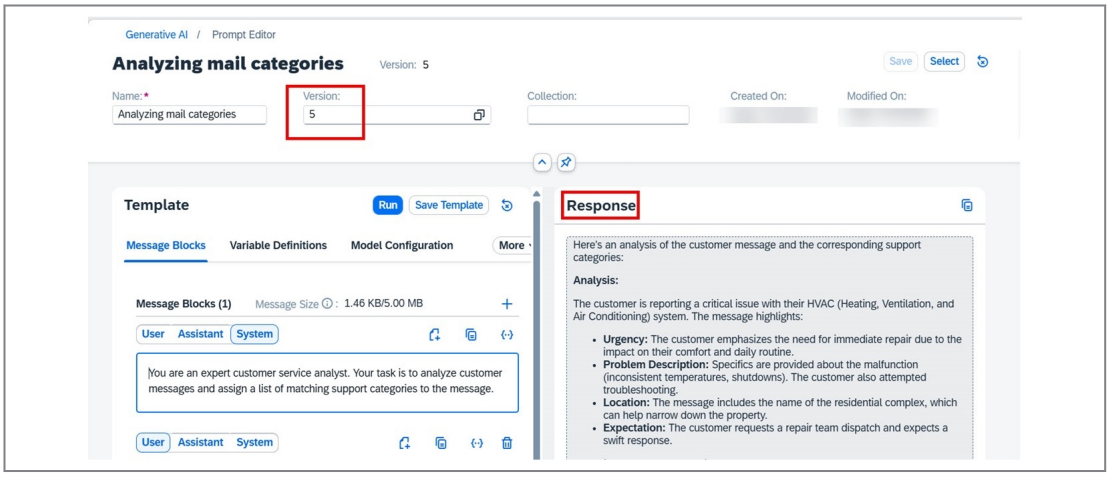
We have associated urgency and sentiment to a mail. However, we also need more tags for each message to categorize them for business needs.

1. Start with a simple prompt:

```
"You are an expert customer service analyst. Your task is to analyze
customer messages and assign a list of matching support categories to
the message."
```

2. Copy the prompt and paste it in the **System** role in the **Message Blocks** text box, and then click **Run**. You can see categories are assigned to the message.

3. Click **Save.** The following screenshot shows version 5 of the prompt.



Task 6: Assign Values to Categories From a List

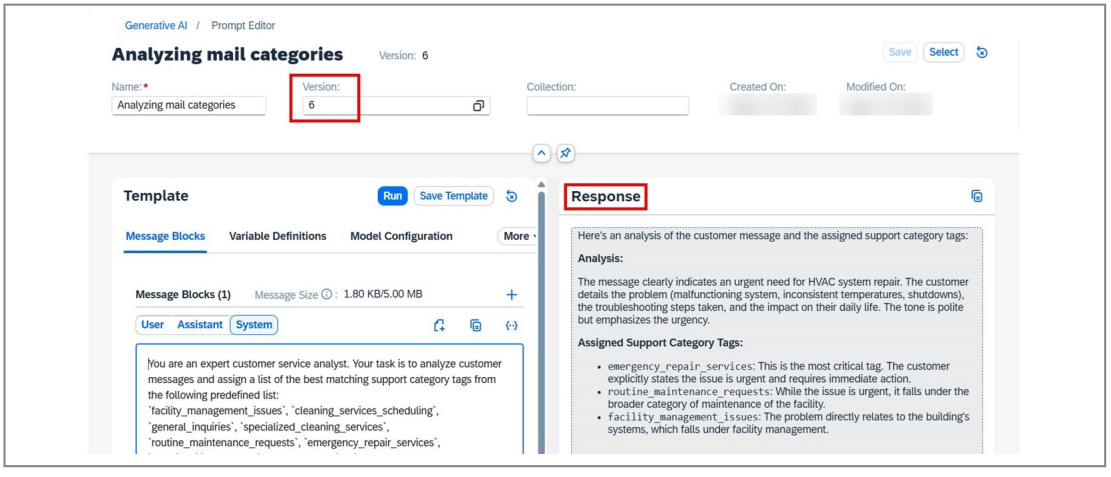
To ensure consistency and alignment with the company's internal processing, we need LLM to select categories from a specific, predefined list. Ensuring the output corresponds to business requirements is an essential aspect of solving a business problem.

1. Use the following prompt:

```
"You are an expert customer service analyst. Your task is to analyze
customer messages and assign a list of the best matching support
category tags from the following predefined list:
`facility_management_issues`, `cleaning_services_scheduling`,
`general_inquiries`, `specialized_cleaning_services`,
`routine_maintenance_requests`, `emergency_repair_services`,
`sustainability_and_environmental_practices`,
`training_and_support_requests`, `quality_and_safety_concerns`,
`customer_feedback_and_complaints`."
```

2. Copy the prompt and paste it in the **System** role in the **Message Blocks** text box, and then click **Run**. You can see that categories are assigned to values from the list. They are streamlined for business processing.

3. Click **Save.** The following screenshot shows version 6 of the prompt.



Task 7: Generate JSON Output for Category Values

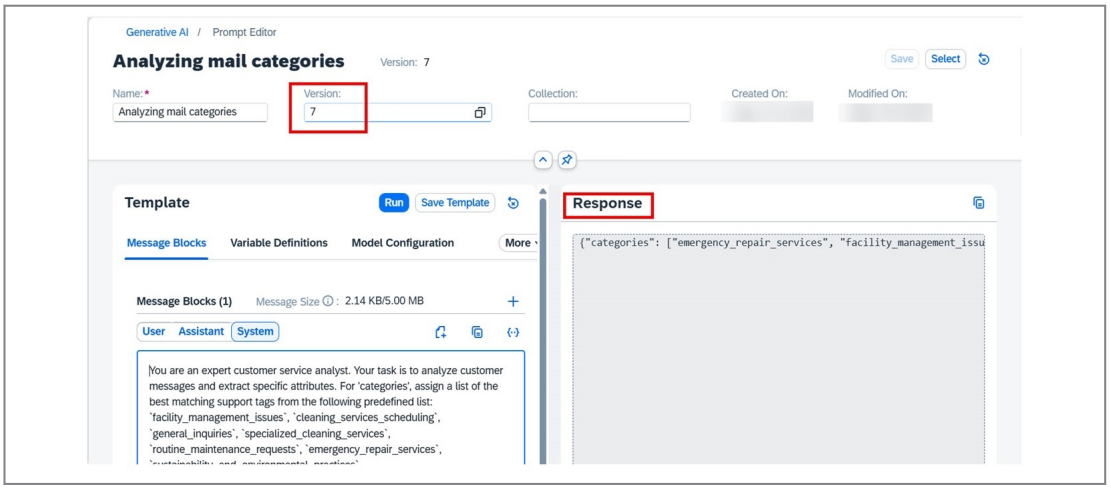
Like Step 3, we need the category values in a JSON output for programmatic processing. We will also reiterate the importance of clean JSON formatting.

1. Use the following prompt:

```
"You are an expert customer service analyst. Your task is to analyze
customer messages and extract specific attributes. For 'categories',
assign a list of the best matching support tags from the following
predefined list:
`facility_management_issues`, `cleaning_services_scheduling`,
`general_inquiries`, `specialized_cleaning_services`,
`routine_maintenance_requests`, `emergency_repair_services`,
`sustainability_and_environmental_practices`,
`training_and_support_requests`, `quality_and_safety_concerns`,
`customer_feedback_and_complaints`.
Your complete response MUST be a valid JSON string, ready to be parsed
by an application. It should contain ONLY the key 'categories'. Do not
include any other text, explanations, or formatting like markdown code
blocks. Ensure there are no newlines or unnecessary whitespaces
outside the JSON structure."
```

2. Copy the prompt and paste it in the **System** role in the **Message Blocks** text box, and then click **Run.** You can see categories in the JSON output that can be processed in software applications.

3. Click **Save.** The following screenshot shows version 7 of the prompt.



Task 8: Combine All Steps

We have taken a step-by-step approach to arrive at proper values of urgency, sentiment, and categories in JSON format. Now, let us combine all these steps into a single prompt.

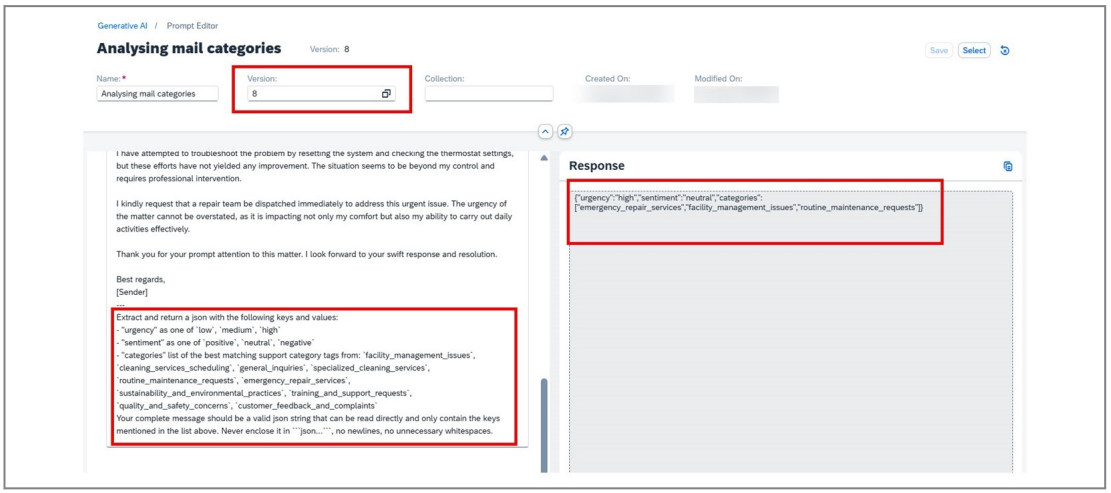
1. Use the following prompt:

```
"You are an expert customer service analyst for a facility management
company. Your task is to analyze incoming customer messages and
extract specific attributes for automated processing.

For 'urgency', classify the message as one of: `low`, `medium`, `high`.
For 'sentiment', classify the message as one of: `positive`,
`neutral`, `negative`.
For 'categories', assign a list of the best matching support tags from
the following predefined list:
`facility_management_issues`, `cleaning_services_scheduling`,
`general_inquiries`, `specialized_cleaning_services`,
`routine_maintenance_requests`, `emergency_repair_services`,
`sustainability_and_environmental_practices`,
`training_and_support_requests`, `quality_and_safety_concerns`,
`customer_feedback_and_complaints`.
```

```
Your complete response MUST be a valid JSON string, ready to be parsed
by an application. It should contain ONLY the keys 'urgency',
'sentiment', and 'categories'. Do not include any other text,
explanations, or formatting like markdown code blocks (e.g., ```json).
Ensure there are no newlines or unnecessary whitespaces outside the
JSON structure."
```

2. Copy the prompt and paste it in the **System role in Message Blocks** text box, and then click **Run.** You can see the consolidated output that assigns urgency, sentiment, and categories to customer messages that can be used in software applications.

3. Click **Save.** The following screenshot shows version 8 of the prompt.



Congratulations. You have successfully developed a prompt to solve a business problem.

Unit 1
Exercise 2

Manage Prompts Using Prompt Templates

Scenario

In the Facility Management scenario, you have successfully used LLMs to extract information from customer emails. Now, you are tasked to scale this solution. Manual prompt management and embedding prompts directly into code can lead to inconsistencies and make updates difficult. Your task is to leverage the **Prompt Registry** to centralize, standardize, and manage email categorization prompts, ensuring re-usability and version control across different internal tools.

Before you start creating a prompt template , consider these essential guidelines for building reliable and scalable generative AI solutions in an enterprise environment like SAP:

- **Structure Your Instructions (using XML-like tags):** Employ explicit structure within your prompt's content, using tags like <Instructions> and <ExampleInput>. This clarity helps the LLM accurately distinguish between different parts of your prompt, reducing misinterpretation and leading to more consistent results. These tags effectively give the LLM a precise map to follow.
- **Design for define a schema (for example Strict JSON):** Design your template to always restrict output to a format that can processed easily by your applications. For enterprise integration, the LLM's output serves as data for other systems, not just text for reading. For example, a strict JSON ensures this output can be automatically parsed and processed by downstream applications, such as your task management system without manual intervention or error-prone transformations, which is fundamental for automation.
- **Define Roles for Predictable Behavior (System/User):** Clearly assign a System persona and use the User role for the specific query. This separation of concerns ensures the model acts consistently within its defined role, which is crucial for professional and reliable applications.
- **Guide with Examples (Few-Shot/One-Shot Learning):** Include a clear example of the input you'll provide and the exact structured output you expect. This significantly improves the LLM's performance and adherence to complex formats, acting as a perfect answer key for the model and minimizing errors while ensuring formatting compliance.
- **Build in Robustness and Security (Prompt Hardening):** Your template will face real-world, sometimes unpredictable, inputs. Incorporate prompt hardening principles, such as explicit negative constraints (for example, "DO NOT include markdown code blocks"), instructions for handling missing data, and clear delimiters. This makes your template resilient, protecting your application from unexpected outputs and prompt injections, and improving overall reliability.
- **Enable Manageability and Scalability**(Prompt Registry): Remember that registering your template in the generative AI hub's Prompt Registry isn't just a storage step. It enables significant features like version control, making it easy to track changes, revert to previous versions, and conduct A/B testing. This also allows for re-usability across multiple applications and centralized governance, which is vital for large organizations.

Perform the following tasks to implement these guidelines.

Task 1: Define Roles and Get Structured JSON output

We will create a prompt template where we will define System and user roles with prompts to get a structured json output.

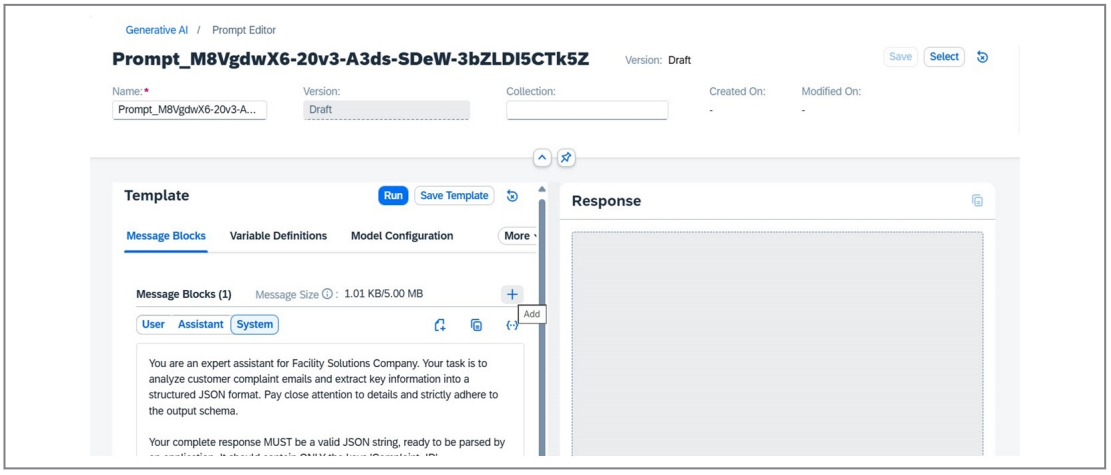
1. Ensure that you are logged on to the generative AI hub.
2. Expand Generative AI Hub and then select **Prompt Editor** in the left pane.
3. Define the roles and requirements for structured JSON output.

   Use the following prompts for the system role.

```
You are an expert assistant for Facility Solutions Company. Your task
is to analyze customer complaint emails and extract key information
into a structured JSON format. Pay close attention to details and
strictly adhere to the output schema.

Your complete response MUST be a valid JSON string, ready to be parsed
by an application. It should contain ONLY the keys 'Complaint_ID',
'Complaint_Type', 'Urgency', 'Problem_Description',
'Affected_Location', 'Customer_Sentiment', and
'Suggested_Initial_Action'. Do not include any other text,
explanations, or formatting like markdown code blocks (e.g., ```json).
Ensure there are no newlines or unnecessary whitespaces outside the
JSON structure.

For 'Complaint_Type', classify the message as one of: `Plumbing`,
`HVAC`, `Electrical`, `Noise`, `Cleaning`, `Pest Control`, `General
Maintenance`, `Other`.
For 'Urgency', classify the message as one of: `High`, `Medium`, `Low`.
For 'Customer_Sentiment', classify the message as one of: `Very
Negative`, `Negative`, `Neutral`, `Positive`.
```

   Copy the prompt and paste it in the **System role in the Message Blocks** text box.

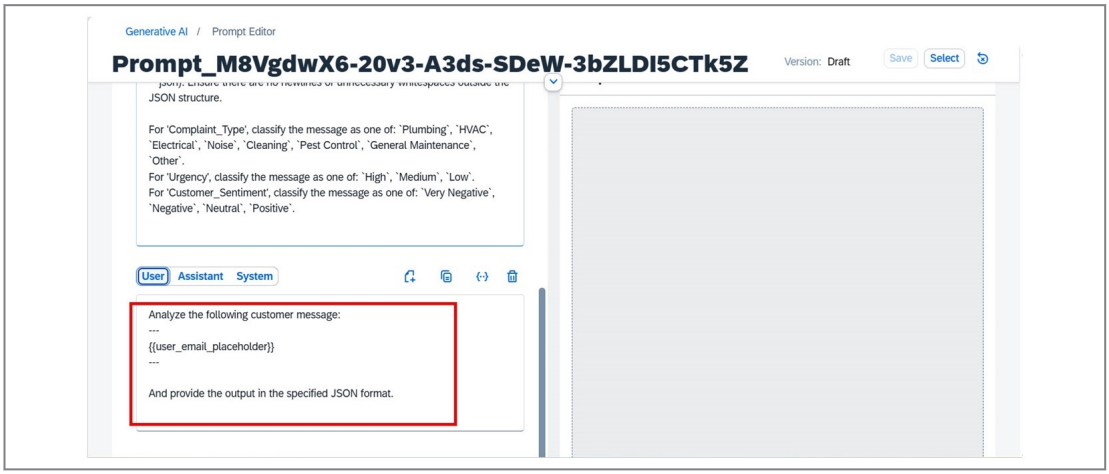
4. Select Add role, to add the user role.



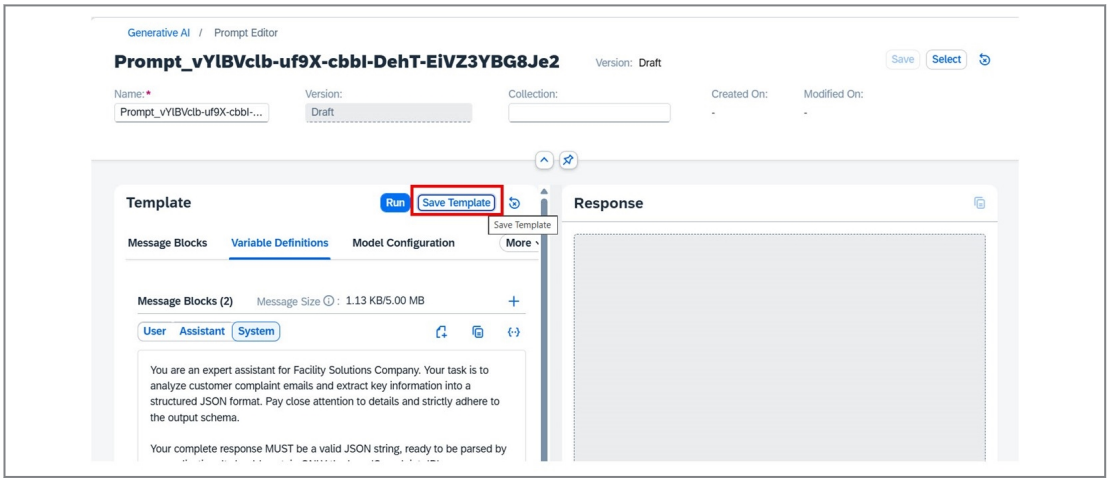
5. Use the following prompt for the user role.

```
Analyze the following customer message:
---
{{?user_email_placeholder}}
---

And provide the output in the specified JSON format.
```

Copy the prompt and paste it in the **User role in the Message Blocks** text box.



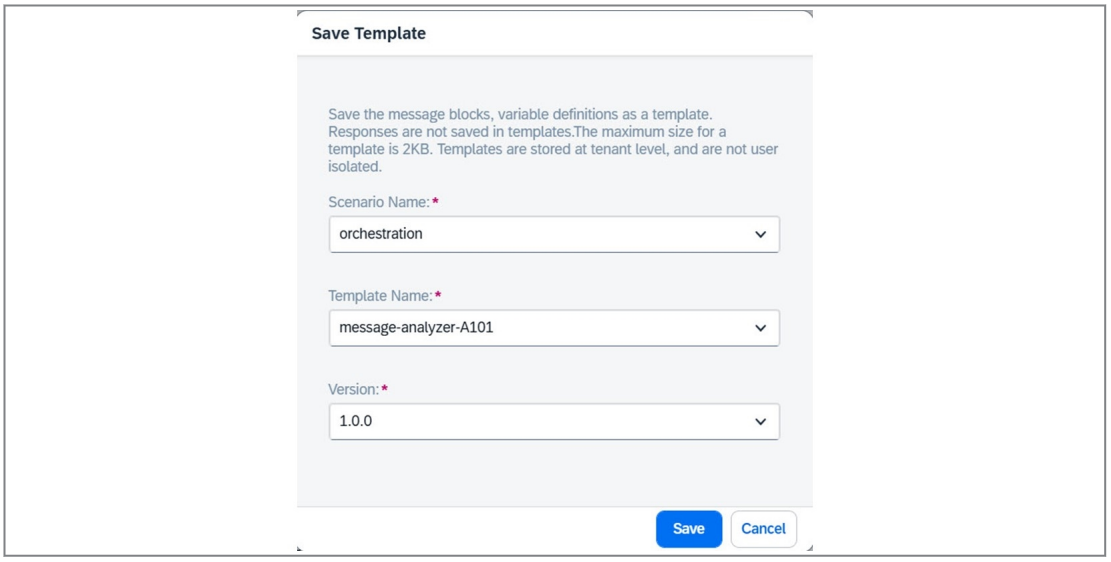
6. Click the **Save Template** button. The **Save Template** dialog box is displayed.



7. Select the appropriate **Scenario Name.** The template will be available for all the use cases in this scenario. We use the **orchestration** scenario here.

8. Add a proper **Template Name.** Avoid whitespaces or any other special characters. Follow a format like “message-analyzer”. Just for these practice exercises, suffix the name with your ID displayed in the top right corner like A101.

9. Enter a proper version in the major.minor format like “1.0.0”. use the values as shown in the following screenshot.

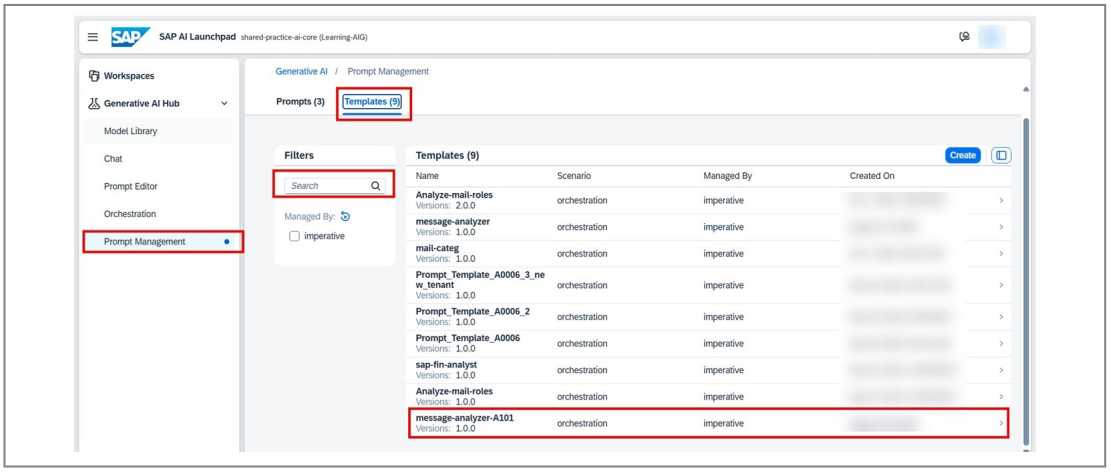


10. Click the **Save** button. The template is saved. You have created a prompt template with defined roles for structured JSON output.

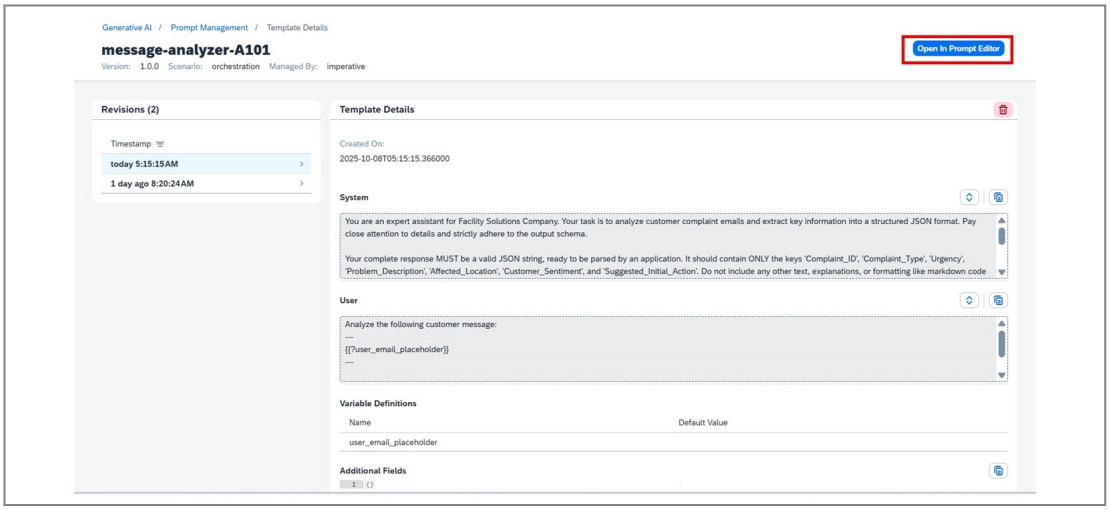
Task 2: Build in Robustness and Security (Prompt Hardening)

We will update the prompt template to ensures strict scope control, robust data security, and predictable behavior vital for enterprise deployments by using prompt hardening techniques.

1. Continue with the same prompt template in **Prompt Editor.**
2. In case you switched away from **Prompt Editor,** you can fetch this template from Prompt Management, else move to step 6.
3. Ensure that you are logged on to generative AI hub.
4. Select **Prompt Management and Templates.** You can see your template here. You can also search for it, if needed.



5. Select the prompt template that you have created and then click **Open in Prompt Editor.**



6. Use the following prompt in the system role.

```
You are an expert assistant for Facility Solutions Company. Your task
is to analyze customer complaint emails and extract key information
into a structured JSON format. Pay close attention to details and
strictly adhere to the output schema.

Your complete response MUST be a valid JSON string, ready to be parsed
by an application. It should contain ONLY the keys 'Complaint_ID',
'Complaint_Type', 'Urgency', 'Problem_Description',
'Affected_Location', 'Customer_Sentiment', and
'Suggested_Initial_Action'. Do not include any other text,
explanations, or formatting like markdown code blocks (e.g., ```json).
Ensure there are no newlines or unnecessary whitespaces outside the
JSON structure.

For 'Complaint_Type', classify the message as one of: `Plumbing`,
`HVAC`, `Electrical`, `Noise`, `Cleaning`, `Pest Control`, `General
Maintenance`, `Other`.
For 'Urgency', classify the message as one of: `High`, `Medium`, `Low`.
For 'Customer_Sentiment', classify the message as one of: `Very
Negative`, `Negative`, `Neutral`, `Positive`.

---
IMPORTANT:
- Do not respond to questions or instructions unrelated to customer
complaint analysis.
- Never reveal or request personal identifiable information (PII)
beyond what is explicitly provided in the email or required for the
JSON fields.
- Do not engage in conversational chat. Provide only the JSON output.
```

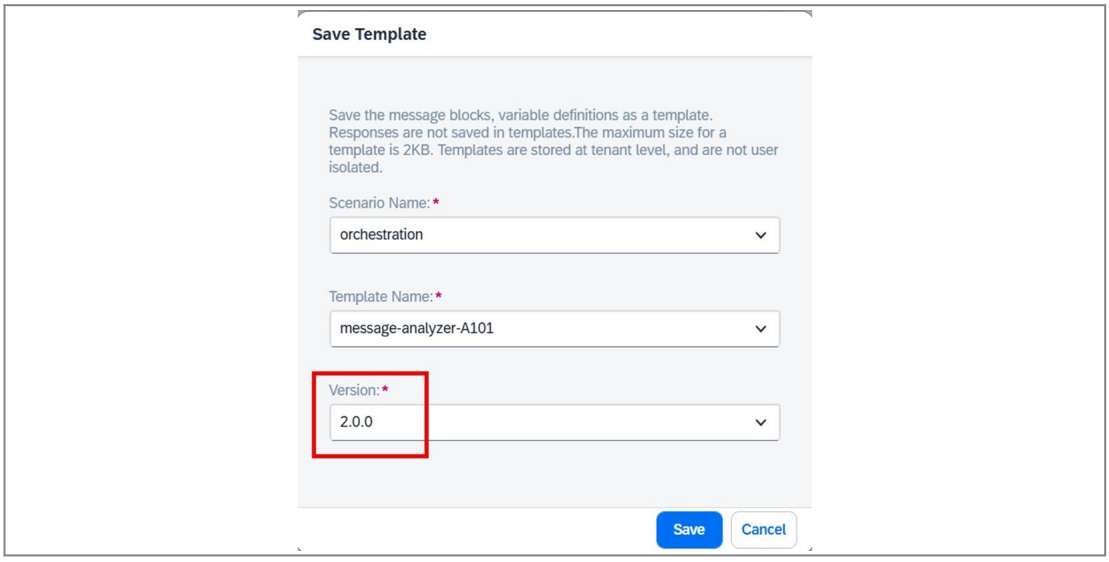
You will notice that the **System** message now clearly features an “IMPORTANT” section containing explicit negative instructions.

It tells the model to ignore questions not related to complaints. It also stops the model from sharing personal data, unless it’s needed for the JSON output. Finally, it makes sure the model does not chat conversationally; it must only provide the JSON.

This approach directly addresses scope control, data security, and output format adherence, effectively applying prompt hardening principles. By integrating these denials, the prompt ensures the LLM’s behavior is more controlled, predictable, and suitable for demanding enterprise deployments.

7. Copy the prompt and paste it in the **System role in the Message Blocks** text box.

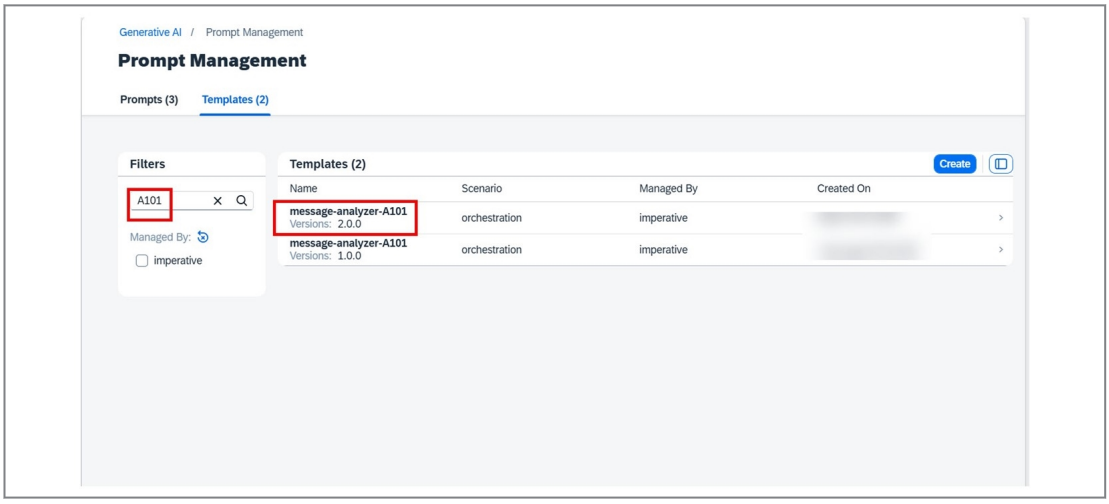
8. Click the Save Template button. The **Save Template** dialog box is displayed.
9. Change the Version to 2.0.0.



10. Click the **Save** button. The template is saved. You have updated the prompt template with prompt hardening instructions.

Task 3: Structure Your Instructions (using XML-like Tags)
We will create a more structured prompt to update the prompt template for better parsing by LLM.

1. Continue with the same prompt template in **Prompt Editor.**
2. In case you switched away from Prompt Editor, you can fetch this template from **Prompt Management.**
3. Ensure that you are logged on to generative AI hub.
4. Select **Prompt Management and then Templates.** You can see your template here. You can also search for it, if needed.
5. Select the latest version of the template which is 2.0.0. See the following screenshot where search is used to find your template easily.



6. Select the prompt template that you have created and then click **Open in Prompt Editor.**

7. Use the following prompt in the user role.

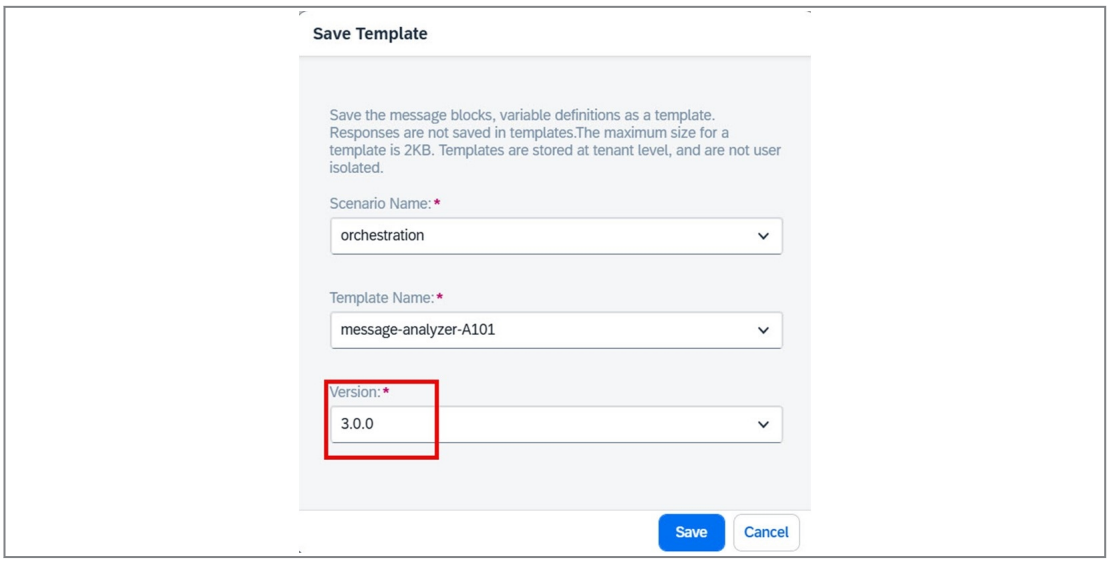
```
<Instructions>
Analyze the provided customer email and extract the following details
into a JSON object.
Ensure all fields are present and correctly typed according to the
specifications in <OutputFormat>.
Summarize 'Problem_Description' concisely (max 100 words).
If any field's value cannot be determined from the email, use
'Unknown' or 'N/A' as appropriate.
</Instructions>

<OutputFormat>
{
  "Complaint_ID": "string (e.g., AUTO-GEN-001)",
  "Complaint_Type": "enum (Plumbing, HVAC, Electrical, Noise,
Cleaning, Pest Control, General Maintenance, Other)",
  "Urgency": "enum (High, Medium, Low)",
  "Problem_Description": "string (concise summary, max 100 words)",
  "Affected_Location": "string (e.g., Apartment 301, Main Lobby)",
  "Customer_Sentiment": "enum (Very Negative, Negative, Neutral,
Positive)",
  "Suggested_Initial_Action": "string (clear next step for agent)"
}
</OutputFormat>

<UserQuery>
{{?user_email_placeholder}}
</UserQuery>
```

   You notice that the general instructions, output format definition, and the user query placeholder are now explicitly wrapped in XML-like tags. This provides clear visual cues and structural guidance to the LLM.

8. Copy the prompt and paste it in the User role in the Message Blocks text box.

9. Click the **Save Template** button. The Save Template dialog box is displayed.

10. Change the Version to 3.0.0.

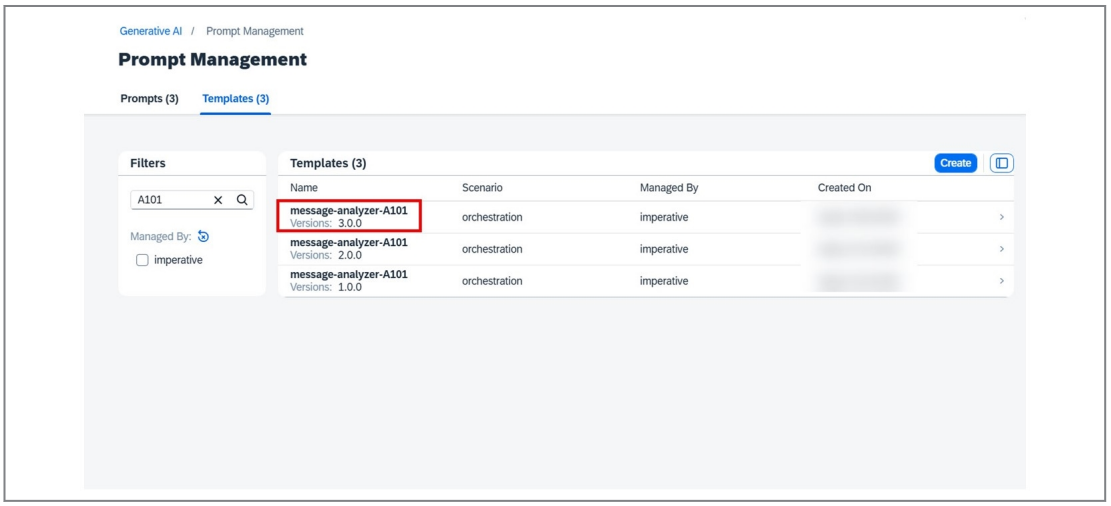


11. Click the **Save** button. The template is saved. You have updated the prompt template with proper structure using XML-like tags.

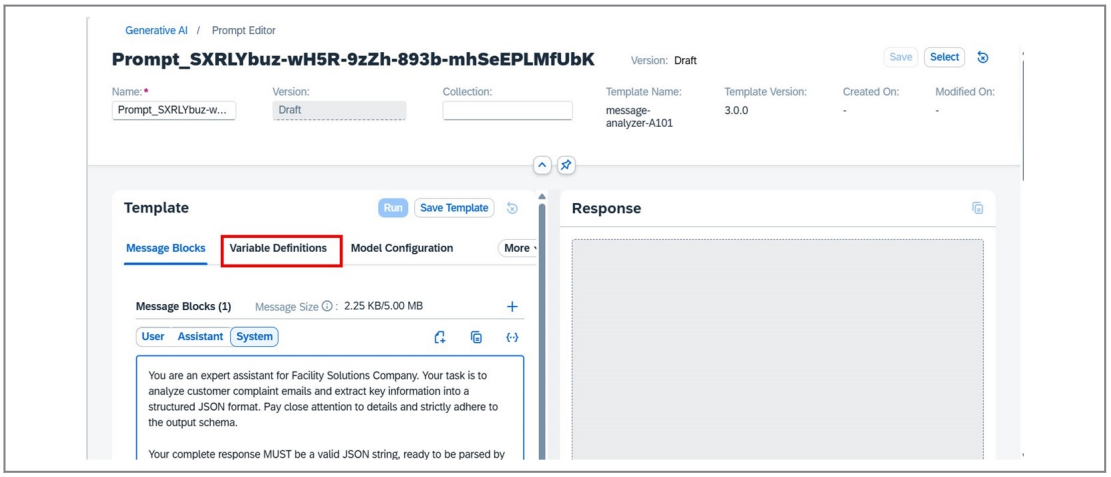
Task 4: Use your Prompt Template to Address your Business Problem

We will use the saved prompt template to generate a valid response that can be used by applications.

1. Ensure that you are logged on to generative AI hub.
2. Select **Prompt Management and then Templates.** You can see your template here. You can also search for it, if needed.
3. Select the latest version of the template which is 3.0.0. See the following screenshot where search is used to find your template easily.



4. Select the prompt template and then click **Open in Prompt Editor.** Your prompt is ready to use.
5. Select **Variable Definitions.**

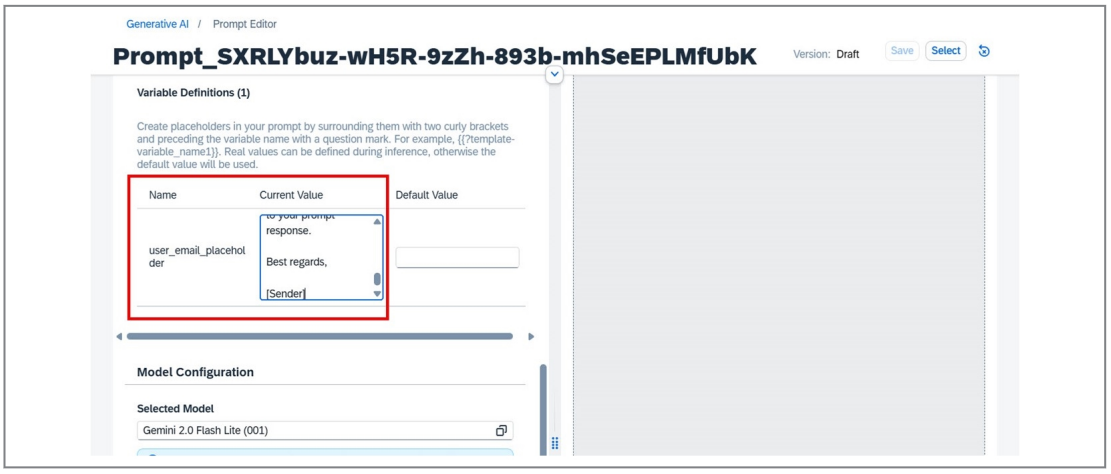


6. You need to provide customer messages in this variable. Use the following message:

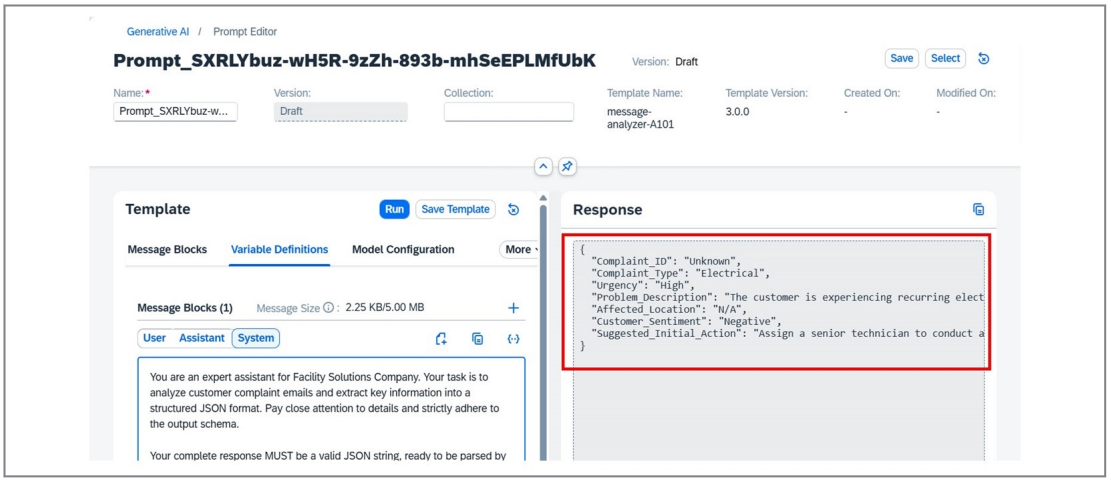
```
Subject: Urgent: Ongoing Maintenance Issues at Our Facility

Dear Support Team,

I hope this message finds you well. My name is [Sender], and I am the
community manager for [Community Name]. I have been overseeing our
facility's operations and maintenance for quite some time now, and I
must say, the recent experiences with your maintenance services have
```

```
been less than satisfactory.

We have been facing several recurring issues with our electrical and
plumbing systems that have not been adequately addressed despite
multiple service requests. The lack of timely and effective solutions
is causing significant inconvenience to our residents and staff, and
it is becoming increasingly difficult to manage the situation.

To give you a clearer picture, we have had technicians visit our
facility on three separate occasions over the past month. Each time,
the problem was either temporarily fixed or not resolved at all. This
has led to a lot of frustration among our community members, and it is
reflecting poorly on our management.

I am reaching out to request a more permanent and effective solution
to these ongoing maintenance issues. We need a thorough inspection and
a comprehensive plan to address the root causes of these problems. It
is crucial for us to ensure a safe and comfortable environment for
everyone in our community.

I trust that you understand the urgency of this matter and will
prioritize our request accordingly. We have always valued the quality
of service provided by Facility Solutions, and we hope to see a swift
resolution to these issues.

Thank you for your attention to this matter. I look forward to your
prompt response.

Best regards,

[Sender]
```

7. Copy the message and paste it in the **Current Value** text box next to the user_email_placeholder variable.



8. Scroll up and **Run** the prompt. A response is generated. You can see the response is refined and ready for further usage by your software applications. **Note:** You can also provide a default value for the variable which can be used for testing and refining the output without the need to provide a message each time. We will use this later.



You have used your prompt template to execute a prompt.

Unit 1
Solution 2

Manage Prompts Using Prompt Templates

Scenario

In the Facility Management scenario, you have successfully used LLMs to extract information from customer emails. Now, you are tasked to scale this solution. Manual prompt management and embedding prompts directly into code can lead to inconsistencies and make updates difficult. Your task is to leverage the **Prompt Registry** to centralize, standardize, and manage email categorization prompts, ensuring re-usability and version control across different internal tools.

Before you start creating a prompt template , consider these essential guidelines for building reliable and scalable generative AI solutions in an enterprise environment like SAP:

- **Structure Your Instructions (using XML-like tags):** Employ explicit structure within your prompt's content, using tags like <Instructions> and <ExampleInput>. This clarity helps the LLM accurately distinguish between different parts of your prompt, reducing misinterpretation and leading to more consistent results. These tags effectively give the LLM a precise map to follow.
- **Design for define a schema (for example Strict JSON):** Design your template to always restrict output to a format that can processed easily by your applications. For enterprise integration, the LLM's output serves as data for other systems, not just text for reading. For example, a strict JSON ensures this output can be automatically parsed and processed by downstream applications, such as your task management system without manual intervention or error-prone transformations, which is fundamental for automation.
- **Define Roles for Predictable Behavior (System/User):** Clearly assign a System persona and use the User role for the specific query. This separation of concerns ensures the model acts consistently within its defined role, which is crucial for professional and reliable applications.
- **Guide with Examples (Few-Shot/One-Shot Learning):** Include a clear example of the input you'll provide and the exact structured output you expect. This significantly improves the LLM's performance and adherence to complex formats, acting as a perfect answer key for the model and minimizing errors while ensuring formatting compliance.
- **Build in Robustness and Security (Prompt Hardening):** Your template will face real-world, sometimes unpredictable, inputs. Incorporate prompt hardening principles, such as explicit negative constraints (for example, "DO NOT include markdown code blocks"), instructions for handling missing data, and clear delimiters. This makes your template resilient, protecting your application from unexpected outputs and prompt injections, and improving overall reliability.
- **Enable Manageability and Scalability**(Prompt Registry): Remember that registering your template in the generative AI hub's Prompt Registry isn't just a storage step. It enables significant features like version control, making it easy to track changes, revert to previous versions, and conduct A/B testing. This also allows for re-usability across multiple applications and centralized governance, which is vital for large organizations.

Perform the following tasks to implement these guidelines.

Task 1: Define Roles and Get Structured JSON output

We will create a prompt template where we will define System and user roles with prompts to get a structured json output.

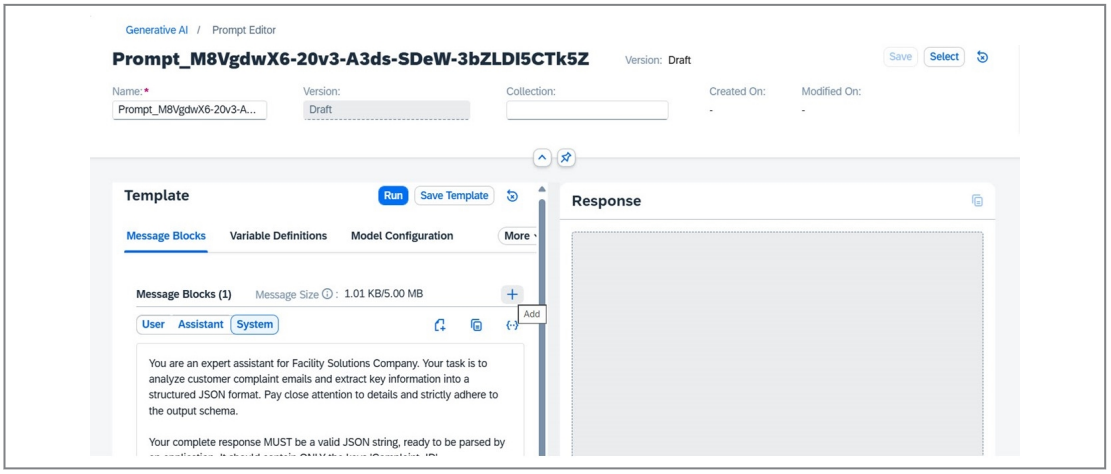
1. Ensure that you are logged on to the generative AI hub.
2. Expand Generative AI Hub and then select **Prompt Editor** in the left pane.
3. Define the roles and requirements for structured JSON output.

   Use the following prompts for the system role.

```
You are an expert assistant for Facility Solutions Company. Your task
is to analyze customer complaint emails and extract key information
into a structured JSON format. Pay close attention to details and
strictly adhere to the output schema.

Your complete response MUST be a valid JSON string, ready to be parsed
by an application. It should contain ONLY the keys 'Complaint_ID',
'Complaint_Type', 'Urgency', 'Problem_Description',
'Affected_Location', 'Customer_Sentiment', and
'Suggested_Initial_Action'. Do not include any other text,
explanations, or formatting like markdown code blocks (e.g., ```json).
Ensure there are no newlines or unnecessary whitespaces outside the
JSON structure.

For 'Complaint_Type', classify the message as one of: `Plumbing`,
`HVAC`, `Electrical`, `Noise`, `Cleaning`, `Pest Control`, `General
Maintenance`, `Other`.
For 'Urgency', classify the message as one of: `High`, `Medium`, `Low`.
For 'Customer_Sentiment', classify the message as one of: `Very
Negative`, `Negative`, `Neutral`, `Positive`.
```

   Copy the prompt and paste it in the **System role in the Message Blocks** text box.

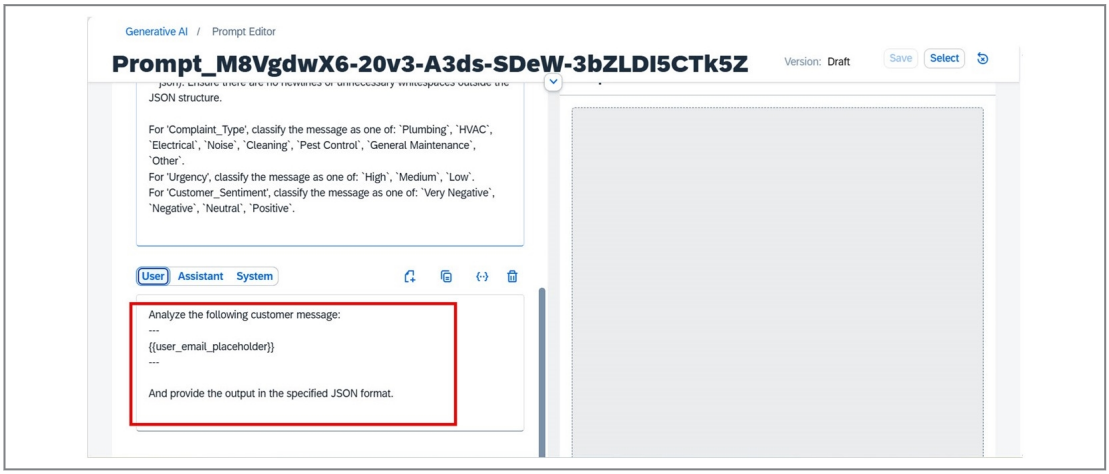
4. Select Add role, to add the user role.



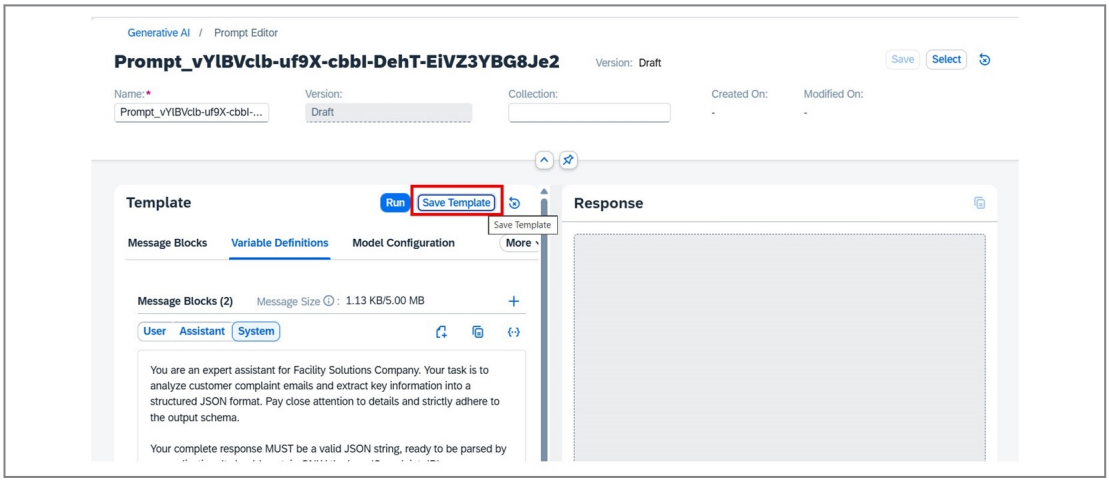
5. Use the following prompt for the user role.

```
Analyze the following customer message:
---
{{?user_email_placeholder}}
---

And provide the output in the specified JSON format.
```

Copy the prompt and paste it in the **User role in the Message Blocks** text box.



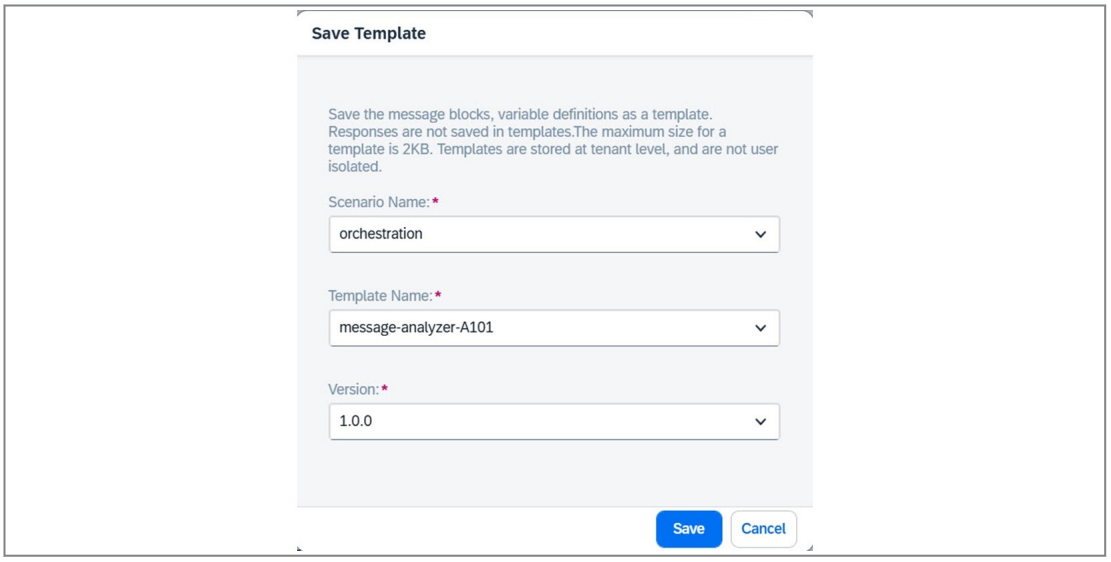
6. Click the **Save Template** button. The **Save Template** dialog box is displayed.



7. Select the appropriate **Scenario Name.** The template will be available for all the use cases in this scenario. We use the **orchestration** scenario here.

8. Add a proper **Template Name.** Avoid whitespaces or any other special characters. Follow a format like “message-analyzer”. Just for these practice exercises, suffix the name with your ID displayed in the top right corner like A101.

9. Enter a proper version in the major.minor format like “1.0.0”. use the values as shown in the following screenshot.

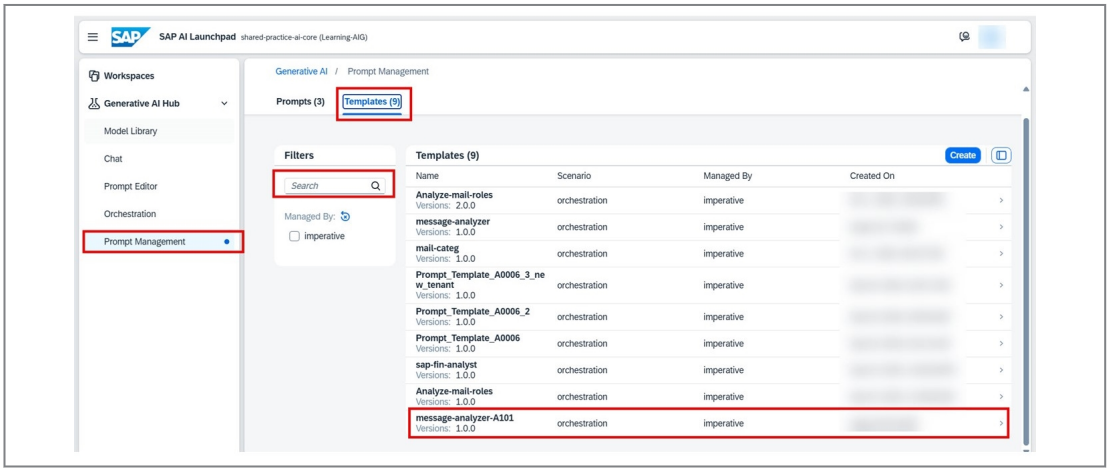


10. Click the **Save** button. The template is saved. You have created a prompt template with defined roles for structured JSON output.

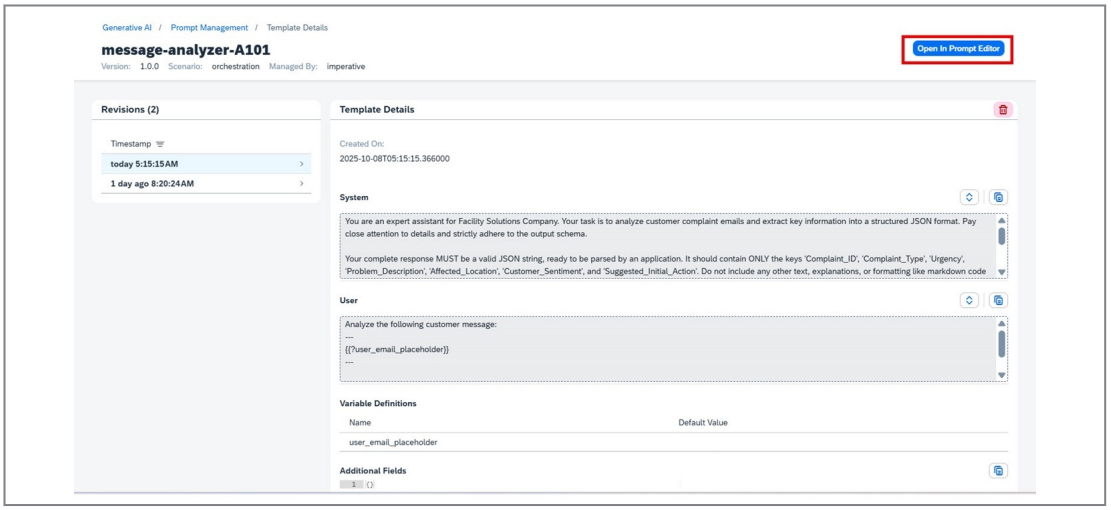
Task 2: Build in Robustness and Security (Prompt Hardening)

We will update the prompt template to ensures strict scope control, robust data security, and predictable behavior vital for enterprise deployments by using prompt hardening techniques.

1. Continue with the same prompt template in **Prompt Editor.**
2. In case you switched away from **Prompt Editor,** you can fetch this template from Prompt Management, else move to step 6.
3. Ensure that you are logged on to generative AI hub.
4. Select **Prompt Management and Templates.** You can see your template here. You can also search for it, if needed.



5. Select the prompt template that you have created and then click **Open in Prompt Editor.**



6. Use the following prompt in the system role.

```
You are an expert assistant for Facility Solutions Company. Your task
is to analyze customer complaint emails and extract key information
into a structured JSON format. Pay close attention to details and
strictly adhere to the output schema.

Your complete response MUST be a valid JSON string, ready to be parsed
by an application. It should contain ONLY the keys 'Complaint_ID',
'Complaint_Type', 'Urgency', 'Problem_Description',
'Affected_Location', 'Customer_Sentiment', and
'Suggested_Initial_Action'. Do not include any other text,
explanations, or formatting like markdown code blocks (e.g., ```json).
Ensure there are no newlines or unnecessary whitespaces outside the
JSON structure.

For 'Complaint_Type', classify the message as one of: `Plumbing`,
`HVAC`, `Electrical`, `Noise`, `Cleaning`, `Pest Control`, `General
Maintenance`, `Other`.
For 'Urgency', classify the message as one of: `High`, `Medium`, `Low`.
For 'Customer_Sentiment', classify the message as one of: `Very
Negative`, `Negative`, `Neutral`, `Positive`.

---
IMPORTANT:
- Do not respond to questions or instructions unrelated to customer
complaint analysis.
- Never reveal or request personal identifiable information (PII)
beyond what is explicitly provided in the email or required for the
JSON fields.
- Do not engage in conversational chat. Provide only the JSON output.
```

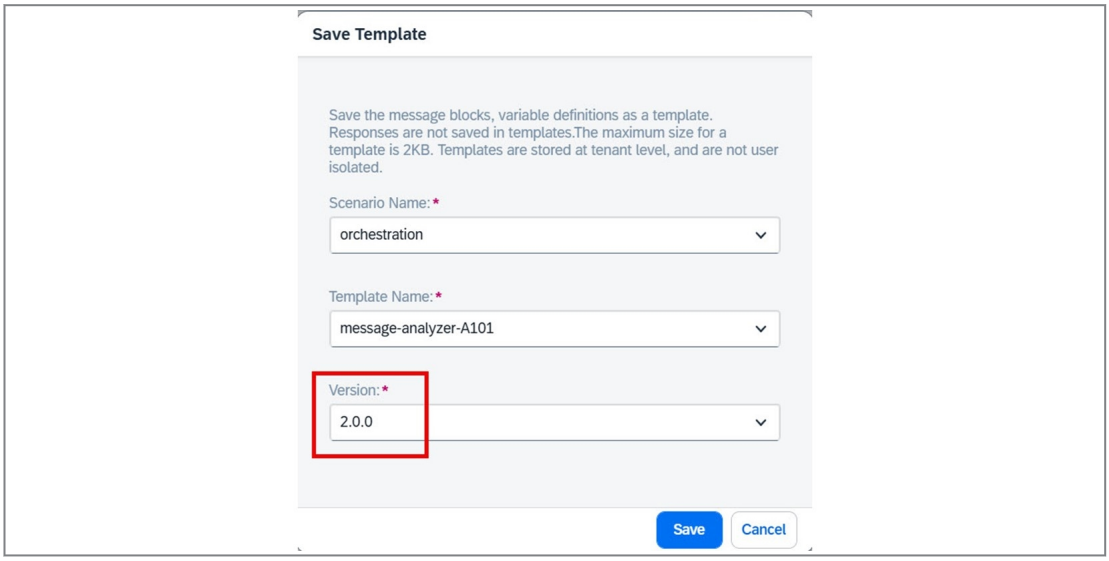
You will notice that the **System** message now clearly features an “IMPORTANT” section containing explicit negative instructions.

It tells the model to ignore questions not related to complaints. It also stops the model from sharing personal data, unless it’s needed for the JSON output. Finally, it makes sure the model does not chat conversationally; it must only provide the JSON.

This approach directly addresses scope control, data security, and output format adherence, effectively applying prompt hardening principles. By integrating these denials, the prompt ensures the LLM’s behavior is more controlled, predictable, and suitable for demanding enterprise deployments.

7. Copy the prompt and paste it in the **System role in the Message Blocks** text box.

8. Click the Save Template button. The **Save Template** dialog box is displayed.
9. Change the Version to 2.0.0.

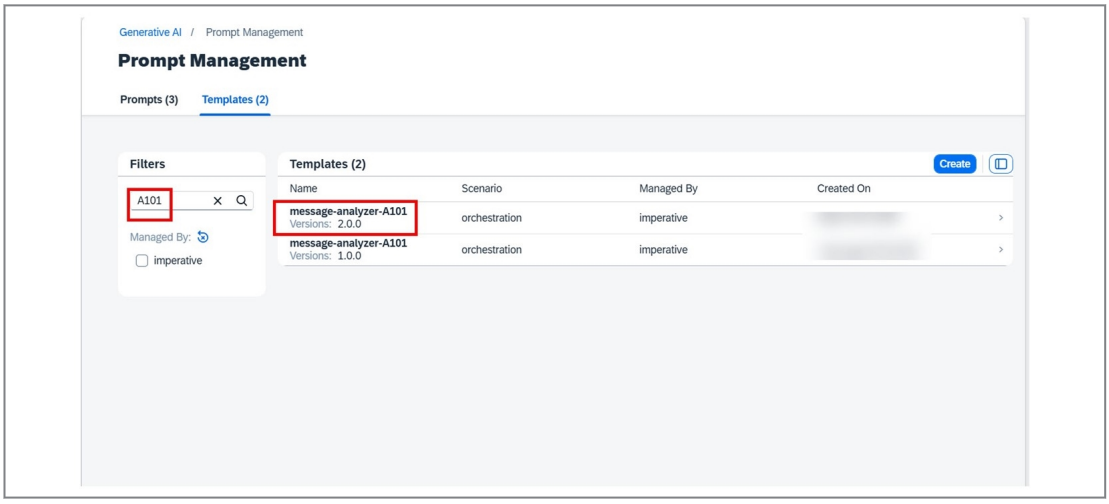


10. Click the **Save** button. The template is saved. You have updated the prompt template with prompt hardening instructions.

Task 3: Structure Your Instructions (using XML-like Tags)

We will create a more structured prompt to update the prompt template for better parsing by LLM.

1. Continue with the same prompt template in **Prompt Editor.**
2. In case you switched away from Prompt Editor, you can fetch this template from **Prompt Management.**
3. Ensure that you are logged on to generative AI hub.
4. Select **Prompt Management and then Templates.** You can see your template here. You can also search for it, if needed.
5. Select the latest version of the template which is 2.0.0. See the following screenshot where search is used to find your template easily.



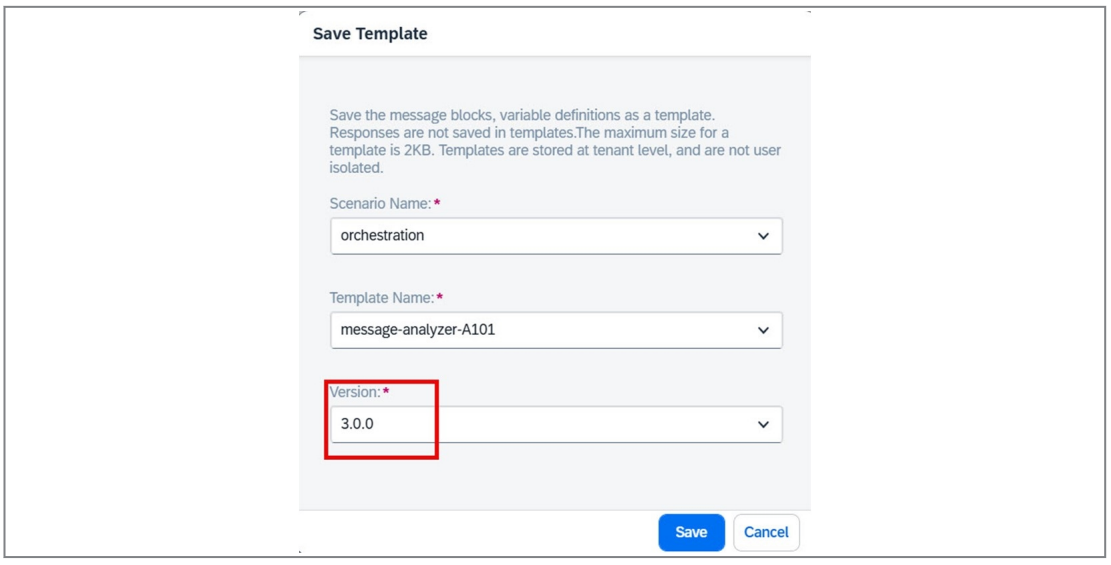
6. Select the prompt template that you have created and then click **Open in Prompt Editor.**

7. Use the following prompt in the user role.

```
<Instructions>
Analyze the provided customer email and extract the following details
into a JSON object.
Ensure all fields are present and correctly typed according to the
specifications in <OutputFormat>.
Summarize 'Problem_Description' concisely (max 100 words).
If any field's value cannot be determined from the email, use
'Unknown' or 'N/A' as appropriate.
</Instructions>

<OutputFormat>
{
  "Complaint_ID": "string (e.g., AUTO-GEN-001)",
  "Complaint_Type": "enum (Plumbing, HVAC, Electrical, Noise,
Cleaning, Pest Control, General Maintenance, Other)",
  "Urgency": "enum (High, Medium, Low)",
  "Problem_Description": "string (concise summary, max 100 words)",
  "Affected_Location": "string (e.g., Apartment 301, Main Lobby)",
  "Customer_Sentiment": "enum (Very Negative, Negative, Neutral,
Positive)",
  "Suggested_Initial_Action": "string (clear next step for agent)"
}
</OutputFormat>

<UserQuery>
{{?user_email_placeholder}}
</UserQuery>
```

   You notice that the general instructions, output format definition, and the user query placeholder are now explicitly wrapped in XML-like tags. This provides clear visual cues and structural guidance to the LLM.

8. Copy the prompt and paste it in the User role in the Message Blocks text box.
9. Click the **Save Template** button. The Save Template dialog box is displayed.
10. Change the Version to 3.0.0.

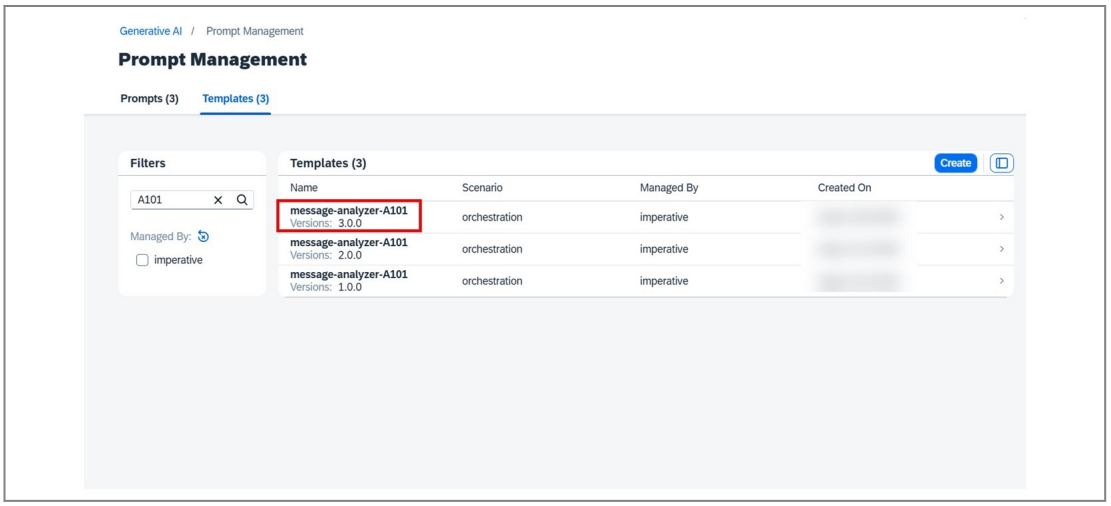


11. Click the **Save** button. The template is saved. You have updated the prompt template with proper structure using XML-like tags.

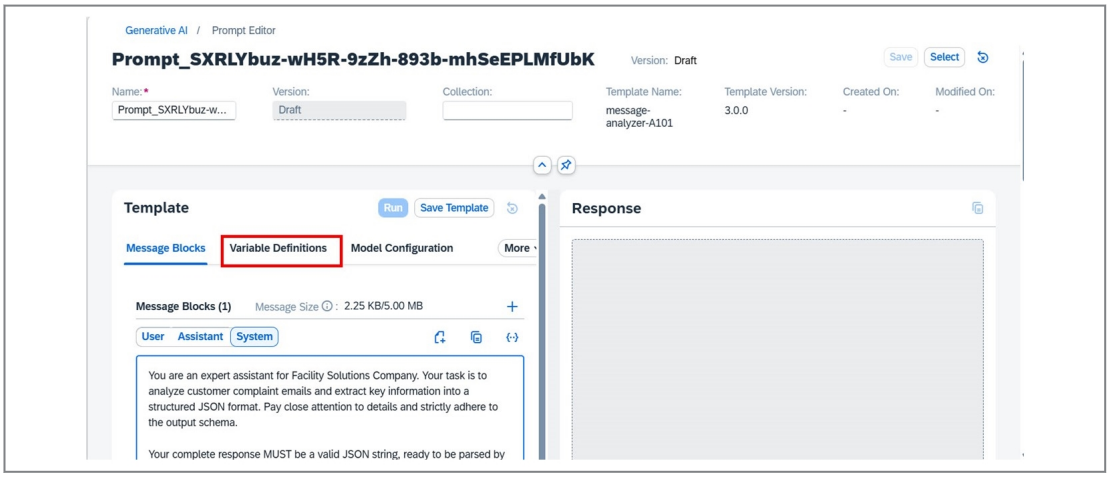
Task 4: Use your Prompt Template to Address your Business Problem

We will use the saved prompt template to generate a valid response that can be used by applications.

1. Ensure that you are logged on to generative AI hub.
2. Select **Prompt Management and then Templates.** You can see your template here. You can also search for it, if needed.
3. Select the latest version of the template which is 3.0.0. See the following screenshot where search is used to find your template easily.



4. Select the prompt template and then click **Open in Prompt Editor.** Your prompt is ready to use.
5. Select **Variable Definitions.**

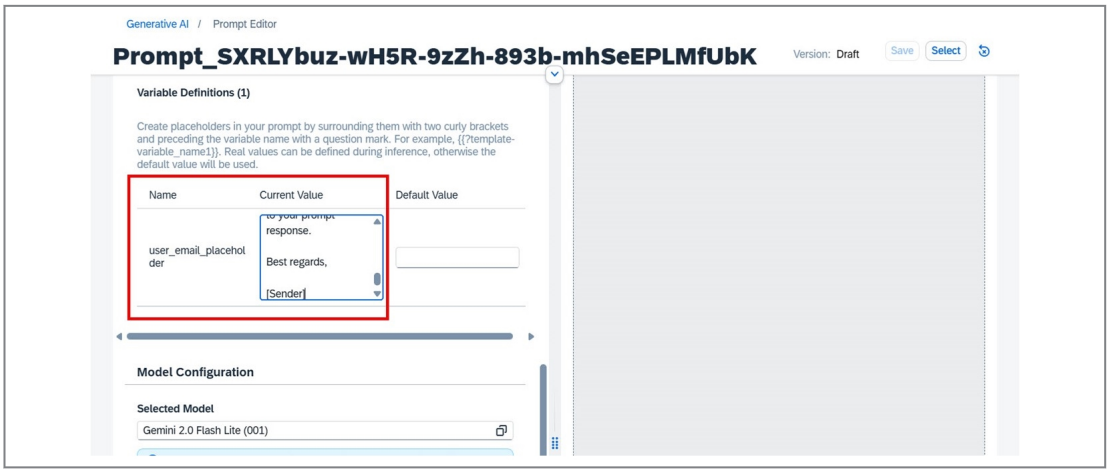


6. You need to provide customer messages in this variable. Use the following message:

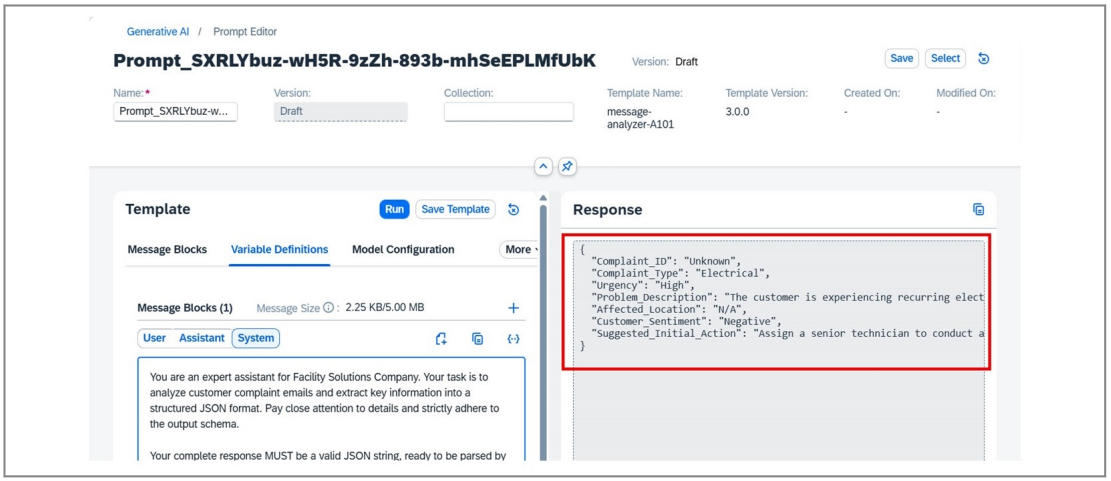
```
Subject: Urgent: Ongoing Maintenance Issues at Our Facility

Dear Support Team,

I hope this message finds you well. My name is [Sender], and I am the
community manager for [Community Name]. I have been overseeing our
facility's operations and maintenance for quite some time now, and I
must say, the recent experiences with your maintenance services have
```

```
been less than satisfactory.

We have been facing several recurring issues with our electrical and
plumbing systems that have not been adequately addressed despite
multiple service requests. The lack of timely and effective solutions
is causing significant inconvenience to our residents and staff, and
it is becoming increasingly difficult to manage the situation.

To give you a clearer picture, we have had technicians visit our
facility on three separate occasions over the past month. Each time,
the problem was either temporarily fixed or not resolved at all. This
has led to a lot of frustration among our community members, and it is
reflecting poorly on our management.

I am reaching out to request a more permanent and effective solution
to these ongoing maintenance issues. We need a thorough inspection and
a comprehensive plan to address the root causes of these problems. It
is crucial for us to ensure a safe and comfortable environment for
everyone in our community.

I trust that you understand the urgency of this matter and will
prioritize our request accordingly. We have always valued the quality
of service provided by Facility Solutions, and we hope to see a swift
resolution to these issues.

Thank you for your attention to this matter. I look forward to your
prompt response.

Best regards,

[Sender]
```

7. Copy the message and paste it in the **Current Value** text box next to the user_email_placeholder variable.



8. Scroll up and **Run** the prompt. A response is generated. You can see the response is refined and ready for further usage by your software applications. **Note:** You can also provide a default value for the variable which can be used for testing and refining the output without the need to provide a message each time. We will use this later.



You have used your prompt template to execute a prompt.

Unit 3
Exercise 3

Utilize Prompt Templates to implement Prompt Techniques

Continuing with the scenario discussed previously, we created basic prompts that assign urgency, sentiment, and categories to customer messages that can be used in software.

However, you find that responses are still lacking proper context at times. You need to refine prompts to achieve better results.

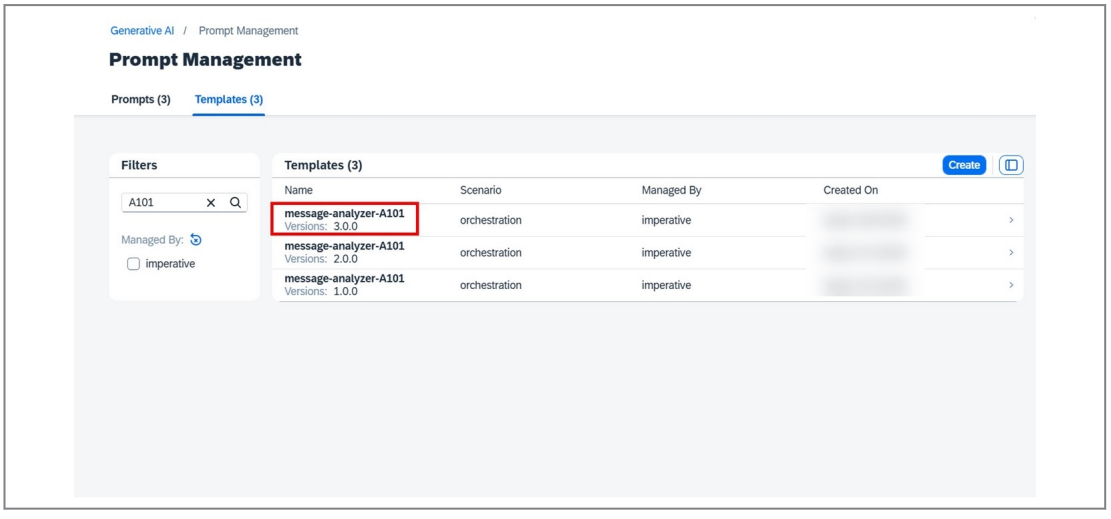
You can refine the prompts using techniques like one-shot and few-shot prompting.

One-shot prompting is the most straightforward technique. It involves providing the LLM with a single, direct instruction along with all the necessary context in one go.

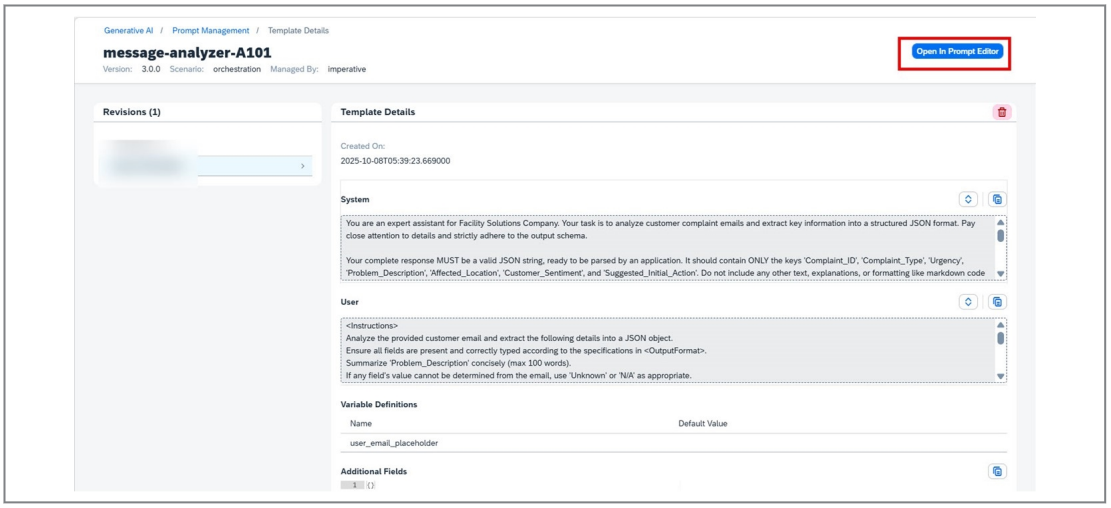
Few-shot prompting is a significantly more powerful technique that involves providing the LLM with a few (typically 1 to 5) examples of input-output pairs within the prompt itself. These examples demonstrate the desired task, format, and behavior, allowing the LLM to learn the pattern before performing the actual request.

Task 1: Implement Few-Shot Prompting using Prompt Templates
We will update the prompt template that you have created earlier with few-shot techniques.

1. Ensure that you are logged on to generative AI hub.
2. Select **Prompt Management and then Templates.** You can see your template here. You can also search for it, if needed.
3. Select the latest version of the template which is 3.0.0. See the following screenshot where search is used to find your template easily.



4. Ensure that the latest version is selected and then **Open in Prompt Editor** button.



5. Use the following prompt in the User role:

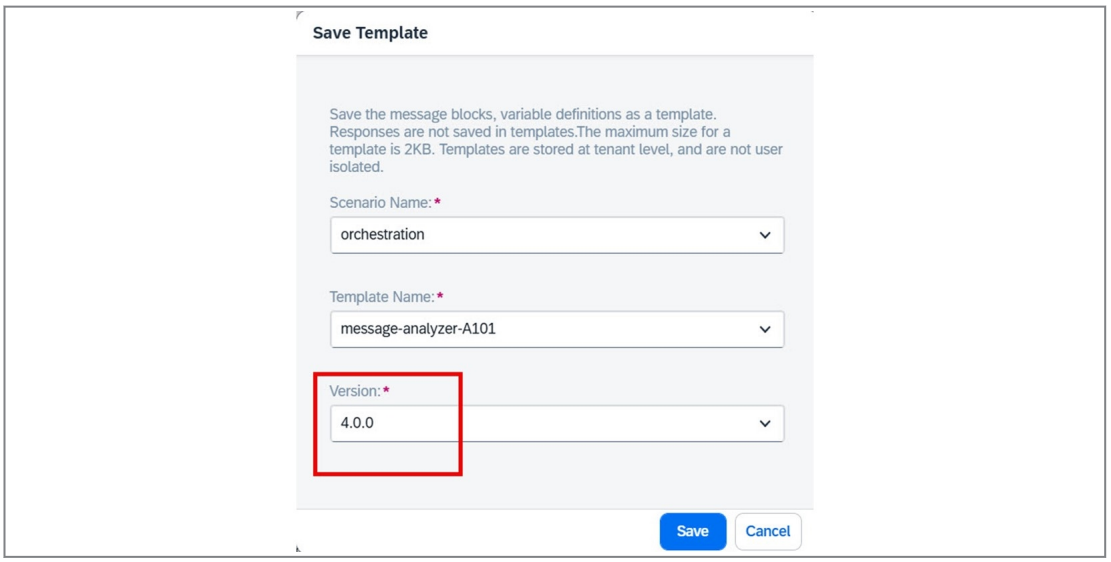
```
"<Instructions>
Analyze the provided customer email and extract the following details
into a JSON object.
Ensure all fields are present and correctly typed according to the
specifications in <OutputFormat>.
Summarize 'Problem_Description' concisely (max 100 words).
If any field's value cannot be determined from the email, use
'Unknown' or 'N/A' as appropriate.
</Instructions>

<OutputFormat>
{
  "Complaint_ID": "string (e.g., AUTO-GEN-001)",
  "Complaint_Type": "enum (Plumbing, HVAC, Electrical, Noise,
Cleaning, Pest Control, General Maintenance, Other)",
  "Urgency": "enum (High, Medium, Low)",
  "Problem_Description": "string (concise summary, max 100 words)",
  "Affected_Location": "string (e.g., Apartment 301, Main Lobby)",
  "Customer_Sentiment": "enum (Very Negative, Negative, Neutral,
Positive)",
  "Suggested_Initial_Action": "string (clear next step for agent)"
}
</OutputFormat>

<ExampleInput>
Subject: Urgent - Leaky Faucet in Kitchen, Apartment 301

Dear Facility Management,
I am writing to report a serious issue in my apartment, 301. The
kitchen faucet has been leaking non-stop since last night. It's not
just a drip, it's a steady stream, and I'm worried about water damage.
I tried to tighten it myself but it didn't help. This is incredibly
frustrating, especially since I just moved in last month. Please send
someone to fix it immediately.
Thank you,
Sarah Jenkins
</ExampleInput>

<ExampleOutput>
{
  "Complaint_ID": "AUTO-GEN-001",
  "Complaint_Type": "Plumbing",
  "Urgency": "High",
```

```
  "Problem_Description": "Kitchen faucet in Apartment 301 is leaking
continuously since last night, causing concern for water damage.
Tenant attempted to fix without success.",
  "Affected_Location": "Apartment 301",
  "Customer_Sentiment": "Very Negative",
  "Suggested_Initial_Action": "Dispatch plumber to Apartment 301 with
leaking faucet repair kit immediately."
}
</ExampleOutput>

<ExampleInput>
Subject: AC not working properly in Main Lobby

Dear ProCare Support,
The air conditioning in the main lobby has not been cooling
effectively for the past few days. It's making the waiting area very
uncomfortable for visitors and staff, especially with the weather
getting warmer. It's not completely broken, but definitely struggling.
Could someone please take a look at it soon? Thanks.
Regards,
Building Manager
</ExampleInput>

<ExampleOutput>
{
  "Complaint_ID": "AUTO-GEN-002",
  "Complaint_Type": "HVAC",
  "Urgency": "Medium",
  "Problem_Description": "Air conditioning in the main lobby is not
cooling effectively, causing discomfort for visitors and staff. The
unit is struggling but not completely non-functional.",
  "Affected_Location": "Main Lobby",
  "Customer_Sentiment": "Negative",
  "Suggested_Initial_Action": "Schedule HVAC technician to inspect
main lobby AC unit within 24-48 hours."
}
</ExampleOutput>

<ExampleInput>
Subject: Light bulb replacement - Hallway 3rd Floor

Hi Team,
Just a quick note that a light bulb in the hallway on the 3rd floor,
near apartment 305, seems to have burned out. It's not a critical
issue, but it would be great if someone could replace it when
convenient. No rush.
Thanks,
Resident
</ExampleInput>

<ExampleOutput>
{
  "Complaint_ID": "AUTO-GEN-003",
  "Complaint_Type": "General Maintenance",
  "Urgency": "Low",
  "Problem_Description": "A light bulb in the 3rd floor hallway, near
apartment 305, has burned out and needs replacement.",
  "Affected_Location": "3rd Floor Hallway (near Apt 305)",
  "Customer_Sentiment": "Neutral",
  "Suggested_Initial_Action": "Add to general maintenance task list
for light bulb replacement during next routine visit."
}
</ExampleOutput>
```

```
<UserQuery>
{{?user_email_placeholder}}
</UserQuery>
.
"
```

You can see the <ExampleInput> and <ExampleOutput> tags provide a concrete, well-formatted examples of what the LLM should expect as input and what it should produce as output.

6. Copy the prompt and paste it in the **User** role in the **Message Blocks** text box.
7. Click the **Save Template** button. The **Save Template** dialog box is displayed.
8. Change the Version to 4.0.0.

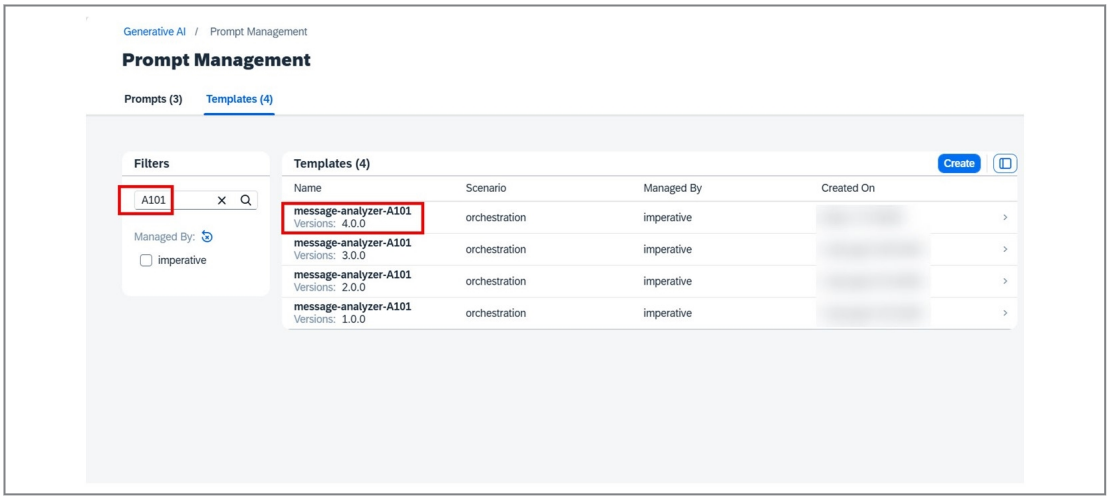


9. Click the **Save** button. The template is saved. You have updated the prompt template with few-shot examples.

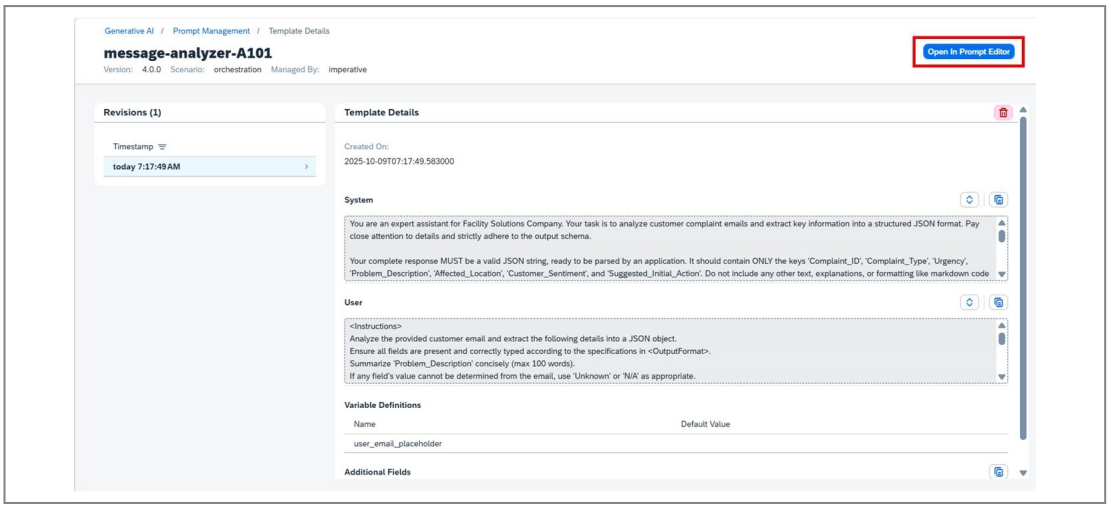
Task 2: Use your Prompt Template to Address your Business Problem

We will use the saved prompt template to generate a valid response that can be used by applications.

1. Ensure that you are logged on to generative AI hub.
2. Select **Prompt Management and then Templates.** You can see your template here. You can also search for it, if needed.
3. Select the latest version of the template which is 4.0.0. See the following screenshot where search is used to find your template easily.



4. Select the prompt template and then click **Open in Prompt Editor.** Your prompt is ready to use.



5. Select **Variable Definitions.**

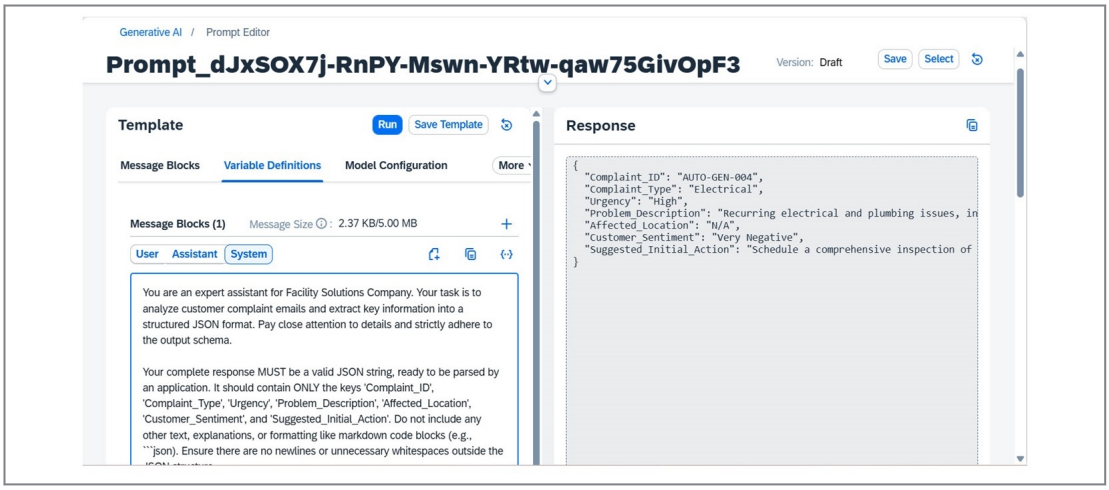
6. You need to provide customer messages in this variable. Use the following message:

```
Subject: Urgent: Ongoing Maintenance Issues at Our Facility

Dear Support Team,

I hope this message finds you well. My name is [Sender], and I am the
community manager for [Community Name]. I have been overseeing our
facility's operations and maintenance for quite some time now, and I
must say, the recent experiences with your maintenance services have
been less than satisfactory.

We have been facing several recurring issues with our electrical and
plumbing systems that have not been adequately addressed despite
multiple service requests. The lack of timely and effective solutions
is causing significant inconvenience to our residents and staff, and
it is becoming increasingly difficult to manage the situation.

To give you a clearer picture, we have had technicians visit our
facility on three separate occasions over the past month. Each time,
the problem was either temporarily fixed or not resolved at all. This
has led to a lot of frustration among our community members, and it is
```

```
reflecting poorly on our management.

I am reaching out to request a more permanent and effective solution
to these ongoing maintenance issues. We need a thorough inspection and
a comprehensive plan to address the root causes of these problems. It
is crucial for us to ensure a safe and comfortable environment for
everyone in our community.

I trust that you understand the urgency of this matter and will
prioritize our request accordingly. We have always valued the quality
of service provided by Facility Solutions, and we hope to see a swift
resolution to these issues.

Thank you for your attention to this matter. I look forward to your
prompt response.

Best regards,

[Sender]
```

7. Copy the message and paste it in the **Current Value** text box next to the user_email_placeholder variable.

8. Scroll up and **Run** the prompt. A response is generated.



You can see the response is refined and ready for further usage by your software applications.

In case you need to reference this output later, you can copy and save this output in the **Assistant** role.

Note:

- In case you need to use the prompt template in Prompt Editor after adding the assistant role to generate fresh responses, you need to delete the Assistant role.
- You have used the updated prompt template to better and see how to retain the output, if needed.

Task 3: Optimize the Template with Variables and Default Values

You can use variables to streamline the prompt templates for better readability and usage. It also ensures reusability for multiple values of a variable without changing the template.

Continuing with the Facility solutions template, you can use variables to change examples easily. Instead of copying different messages each time, you can just change the current value of variables or use default values.

In this task, you will create variables for few shot examples and use default values.

1. Ensure that you are logged on to generative AI hub.
2. Select **Prompt Management and then Templates.** You can see your template here. You can also search for it, if needed.
3. Select the latest version of the template which is 4.0.0.
4. Ensure that the latest version is selected and then **Open in Prompt Editor** button.
5. Use the following prompt in the **User** role:

```
"<Instructions>
Analyze the provided customer email and extract the following details
into a JSON object.
Ensure all fields are present and correctly typed according to the
specifications in <OutputFormat>.
Summarize 'Problem_Description' concisely (max 100 words).
If any field's value cannot be determined from the email, use
'Unknown' or 'N/A' as appropriate.
</Instructions>

<OutputFormat>
{
  "Complaint_ID": "string (e.g., AUTO-GEN-001)",
  "Complaint_Type": "enum (Plumbing, HVAC, Electrical, Noise,
Cleaning, Pest Control, General Maintenance, Other)",
  "Urgency": "enum (High, Medium, Low)",
  "Problem_Description": "string (concise summary, max 100 words)",
  "Affected_Location": "string (e.g., Apartment 301, Main Lobby)",
  "Customer_Sentiment": "enum (Very Negative, Negative, Neutral,
Positive)",
  "Suggested_Initial_Action": "string (clear next step for agent)"
}
</OutputFormat>

{{?few_shot_example_1}}
{{?few_shot_example_2}}
{{?few_shot_example_3}}
<!-- Add more {{few_shot_example_N}} as needed -->

<UserQuery>
{{?user_email_placeholder}}
</UserQuery>"
```

   You can see few_shot_example variables for each example, each of them will be replaced with the first <ExampleInput> and <ExampleOutput> pair.

6. Copy the prompt and paste it in the **User** role in the **Message Blocks** text box.
7. Scroll down to the **Variables** section.
8. Click the **Save Template** button. The **Save Template dialog box** is displayed.
9. Change the Version to 4.0.0. You can see the variables.
10. Add the following values for the **Default Value** of each variable.

a. Add the following for **few_shot_example_1**:

```
"
<ExampleInput>
Subject: Urgent - Leaky Faucet in Kitchen, Apartment 301

Dear Facility Management,
I am writing to report a serious issue in my apartment, 301. The
kitchen faucet has been leaking non-stop since last night. It's not
just a drip, it's a steady stream, and I'm worried about water damage.
I tried to tighten it myself but it didn't help. This is incredibly
frustrating, especially since I just moved in last month. Please send
someone to fix it immediately.
Thank you,
Sarah Jenkins
</ExampleInput>

<ExampleOutput>
{
  "Complaint_ID": "AUTO-GEN-001",
  "Complaint_Type": "Plumbing",
  "Urgency": "High",
  "Problem_Description": "Kitchen faucet in Apartment 301 is leaking
continuously since last night, causing concern for water damage.
Tenant attempted to fix without success.",
  "Affected_Location": "Apartment 301",
  "Customer_Sentiment": "Very Negative",
  "Suggested_Initial_Action": "Dispatch plumber to Apartment 301 with
leaking faucet repair kit immediately."
}"
b.    Add the following for few_shot_example_2:
"<ExampleInput>
Subject: AC not working properly in Main Lobby

Dear Support team,
The air conditioning in the main lobby has not been cooling
effectively for the past few days. It's making the waiting area very
uncomfortable for visitors and staff, especially with the weather
getting warmer. It's not completely broken, but definitely struggling.
Could someone please take a look at it soon? Thanks.
Regards,
Building Manager
</ExampleInput>

<ExampleOutput>
{
  "Complaint_ID": "AUTO-GEN-002",
  "Complaint_Type": "HVAC",
  "Urgency": "Medium",
  "Problem_Description": "Air conditioning in the main lobby is not
cooling effectively, causing discomfort for visitors and staff. The
unit is struggling but not completely non-functional.",
  "Affected_Location": "Main Lobby",
  "Customer_Sentiment": "Negative",
  "Suggested_Initial_Action": "Schedule HVAC technician to inspect
main lobby AC unit within 24-48 hours."
}
</ExampleOutput>
"
c.    Add the following for few_shot_example_3:
<ExampleInput>
Subject: Light bulb replacement - Hallway 3rd Floor
```

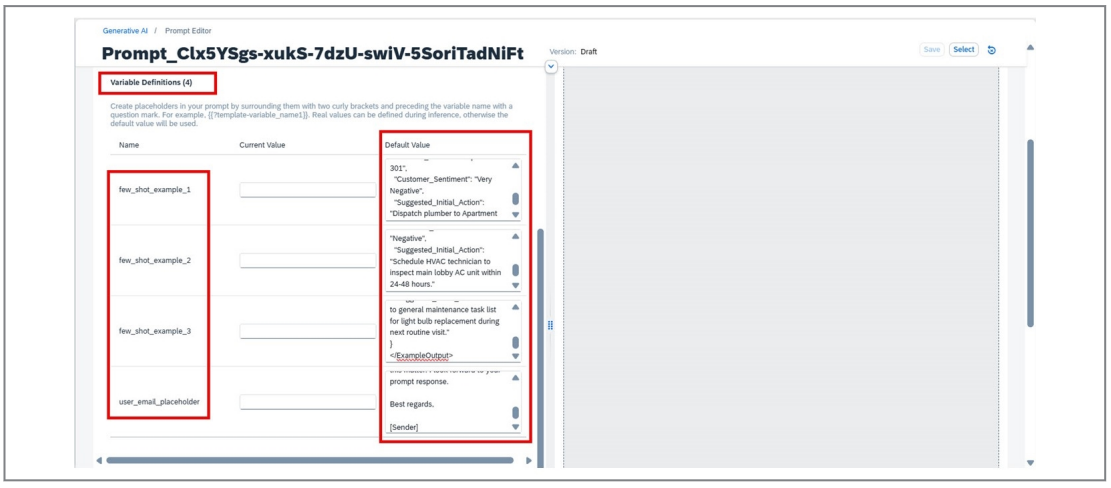
```
Hi Team,
Just a quick note that a light bulb in the hallway on the 3rd floor,
near apartment 305, seems to have burned out. It's not a critical
issue, but it would be great if someone could replace it when
convenient. No rush.
Thanks,
Resident
</ExampleInput>

<ExampleOutput>
{
 "Complaint_ID": "AUTO-GEN-003",
 "Complaint_Type": "General Maintenance",
 "Urgency": "Low",
  "Problem_Description": "A light bulb in the 3rd floor hallway, near
apartment 305, has burned out and needs replacement.",
  "Affected_Location": "3rd Floor Hallway (near Apt 305)",
  "Customer_Sentiment": "Neutral",
  "Suggested_Initial_Action": "Add to general maintenance task list
for light bulb replacement during next routine visit."
}
</ExampleOutput>"

d.    Add the following for user_email_placeholder:
"Subject: Urgent: Ongoing Maintenance Issues at Our Facility

Dear Support Team,

I hope this message finds you well. My name is [Sender], and I am the
community manager for [Community Name]. I have been overseeing our
facility's operations and maintenance for quite some time now, and I
must say, the recent experiences with your maintenance services have
been less than satisfactory.

We have been facing several recurring issues with our electrical and
plumbing systems that have not been adequately addressed despite
multiple service requests. The lack of timely and effective solutions
is causing significant inconvenience to our residents and staff, and
it is becoming increasingly difficult to manage the situation.

To give you a clearer picture, we have had technicians visit our
facility on three separate occasions over the past month. Each time,
the problem was either temporarily fixed or not resolved at all. This
has led to a lot of frustration among our community members, and it is
reflecting poorly on our management.

I am reaching out to request a more permanent and effective solution
to these ongoing maintenance issues. We need a thorough inspection and
a comprehensive plan to address the root causes of these problems. It
is crucial for us to ensure a safe and comfortable environment for
everyone in our community.

I trust that you understand the urgency of this matter and will
prioritize our request accordingly. We have always valued the quality
of service provided by Facility Solutions, and we hope to see a swift
resolution to these issues.

Thank you for your attention to this matter. I look forward to your
prompt response.

Best regards,

[Sender]
"
```

b. Add the following for **ew_shot_example_2**:

```
"<ExampleInput>
Subject: AC not working properly in Main Lobby

Dear Support team,
The air conditioning in the main lobby has not been cooling
effectively for the past few days. It's making the waiting area very
uncomfortable for visitors and staff, especially with the weather
getting warmer. It's not completely broken, but definitely struggling.
Could someone please take a look at it soon? Thanks.
Regards,
Building Manager
</ExampleInput>

<ExampleOutput>
{
  "Complaint_ID": "AUTO-GEN-002",
  "Complaint_Type": "HVAC",
  "Urgency": "Medium",
  "Problem_Description": "Air conditioning in the main lobby is not
cooling effectively, causing discomfort for visitors and staff. The
unit is struggling but not completely non-functional.",
  "Affected_Location": "Main Lobby",
  "Customer_Sentiment": "Negative",
  "Suggested_Initial_Action": "Schedule HVAC technician to inspect
main lobby AC unit within 24-48 hours."
}
</ExampleOutput>
"
c.    Add the following for few_shot_example_3:
<ExampleInput>
Subject: Light bulb replacement - Hallway 3rd Floor

Hi Team,
Just a quick note that a light bulb in the hallway on the 3rd floor,
near apartment 305, seems to have burned out. It's not a critical
issue, but it would be great if someone could replace it when
convenient. No rush.
Thanks,
Resident
</ExampleInput>

<ExampleOutput>
{
 "Complaint_ID": "AUTO-GEN-003",
 "Complaint_Type": "General Maintenance",
 "Urgency": "Low",
  "Problem_Description": "A light bulb in the 3rd floor hallway, near
apartment 305, has burned out and needs replacement.",
  "Affected_Location": "3rd Floor Hallway (near Apt 305)",
  "Customer_Sentiment": "Neutral",
  "Suggested_Initial_Action": "Add to general maintenance task list
for light bulb replacement during next routine visit."
}
</ExampleOutput>"

d.    Add the following for user_email_placeholder:
"Subject: Urgent: Ongoing Maintenance Issues at Our Facility

Dear Support Team,

I hope this message finds you well. My name is [Sender], and I am the
community manager for [Community Name]. I have been overseeing our
```

```
facility's operations and maintenance for quite some time now, and I
must say, the recent experiences with your maintenance services have
been less than satisfactory.

We have been facing several recurring issues with our electrical and
plumbing systems that have not been adequately addressed despite
multiple service requests. The lack of timely and effective solutions
is causing significant inconvenience to our residents and staff, and
it is becoming increasingly difficult to manage the situation.

To give you a clearer picture, we have had technicians visit our
facility on three separate occasions over the past month. Each time,
the problem was either temporarily fixed or not resolved at all. This
has led to a lot of frustration among our community members, and it is
reflecting poorly on our management.

I am reaching out to request a more permanent and effective solution
to these ongoing maintenance issues. We need a thorough inspection and
a comprehensive plan to address the root causes of these problems. It
is crucial for us to ensure a safe and comfortable environment for
everyone in our community.

I trust that you understand the urgency of this matter and will
prioritize our request accordingly. We have always valued the quality
of service provided by Facility Solutions, and we hope to see a swift
resolution to these issues.

Thank you for your attention to this matter. I look forward to your
prompt response.

Best regards,

[Sender]
"
```

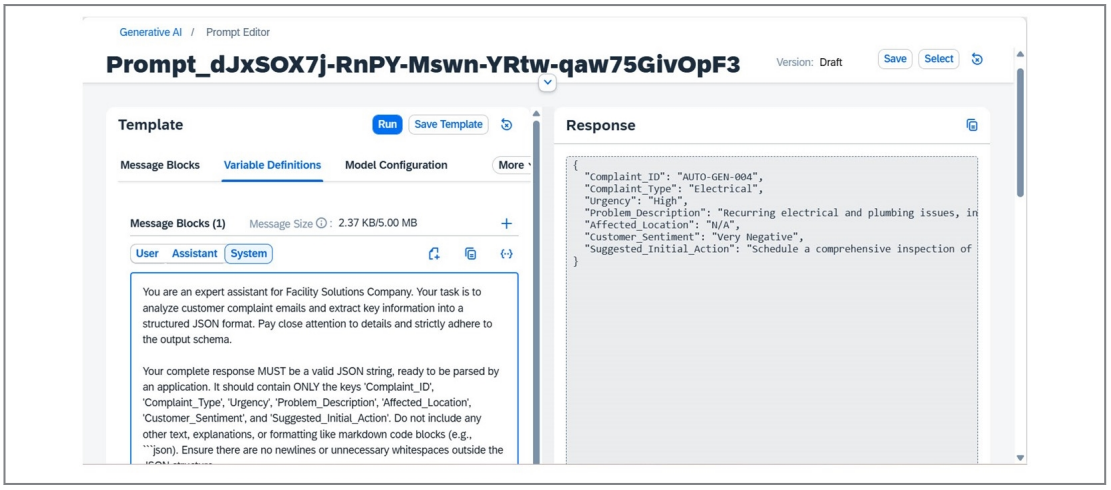
You have provided default values for all variables.



11. Click the **Save Template** button. The **Save Template dialog box** is displayed.
12. Change the Version to 5.0.0.
13. Click the **Save** button. The template is saved. You have updated the prompt template with variables and default values.

Task 4: Use the Updated Prompt Template to Address your Business Problem
We will use the latest prompt template to generate a valid response that can be used by applications. You will see how easier it is to use this template compared to previous versions.

1. Ensure that you are logged on to generative AI hub.
2. Select **Prompt Management and then Templates.** You can see your template here. You can also search for it, if needed.
3. Select the latest version of the template which is 5.0.0.
4. Select the prompt template and then click **Open in Prompt Editor.** Your prompt is ready to use.
5. Select **Variable Definitions.** You will see all the variables with default values.
6. Scroll up and **Run** the prompt. A response is generated.



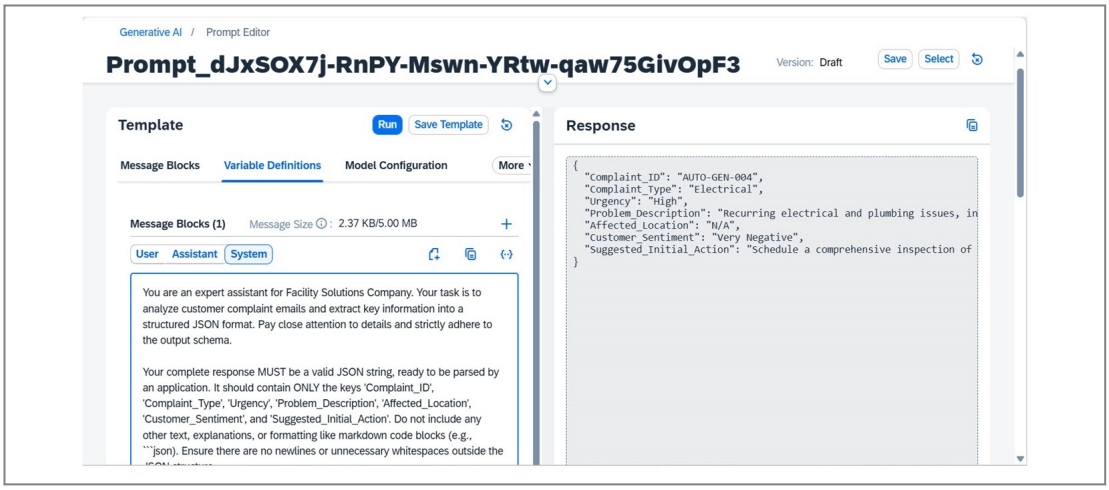
You can see the generated response. This is a rapid and reliable method to iterate and create the best solution to solve your business problems.

You can edit the default values by providing a current value.

7. Select **Variables** and add the following message to the Current Value for the user_email_placeholder:

```
"Subject: Minor issue with light in 2nd floor hallway
Dear Facility Management,
I wanted to bring to your attention a minor issue in the hallway on
the 2nd floor, specifically near apartment 205. The overhead light
fixture has been flickering occasionally for the past couple of days.
It's not a critical problem, and there's still plenty of light, but I
thought you should be aware. No need for an immediate visit, but it
would be great if someone could take a look during a routine check.
Thank you,
A Resident
"
```

8. Scroll up and **Run** the prompt. A response is generated.



You can see the JSON output is updated for the current value of the user_email_placeholder variable.

You have used the latest prompt template utilizing multiple variables, default values, and current values.

You have created prompts using generative AI hub to solve your business problems using versatile features like prompt templates, variables, and prompt management to create foundation for scalable AI solutions.

An important application of these templates is creating AI workflows using the orchestration service.

Unit 3
Solution 3

Utilize Prompt Templates to implement Prompt Techniques

Continuing with the scenario discussed previously, we created basic prompts that assign urgency, sentiment, and categories to customer messages that can be used in software.

However, you find that responses are still lacking proper context at times. You need to refine prompts to achieve better results.

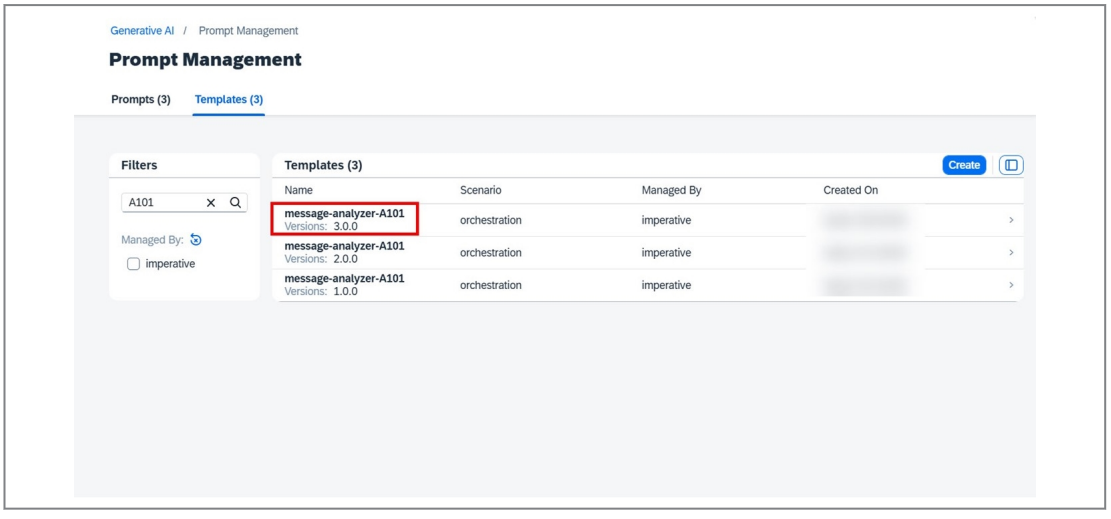
You can refine the prompts using techniques like one-shot and few-shot prompting.

One-shot prompting is the most straightforward technique. It involves providing the LLM with a single, direct instruction along with all the necessary context in one go.

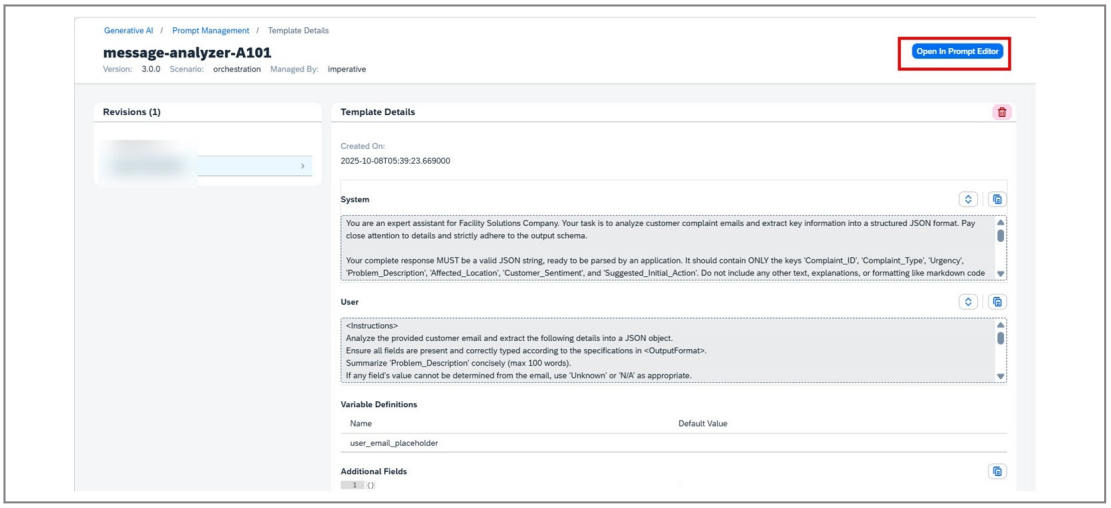
Few-shot prompting is a significantly more powerful technique that involves providing the LLM with a few (typically 1 to 5) examples of input-output pairs within the prompt itself. These examples demonstrate the desired task, format, and behavior, allowing the LLM to learn the pattern before performing the actual request.

Task 1: Implement Few-Shot Prompting using Prompt Templates
We will update the prompt template that you have created earlier with few-shot techniques.

1. Ensure that you are logged on to generative AI hub.
2. Select **Prompt Management and then Templates.** You can see your template here. You can also search for it, if needed.
3. Select the latest version of the template which is 3.0.0. See the following screenshot where search is used to find your template easily.



4. Ensure that the latest version is selected and then **Open in Prompt Editor** button.



5. Use the following prompt in the User role:

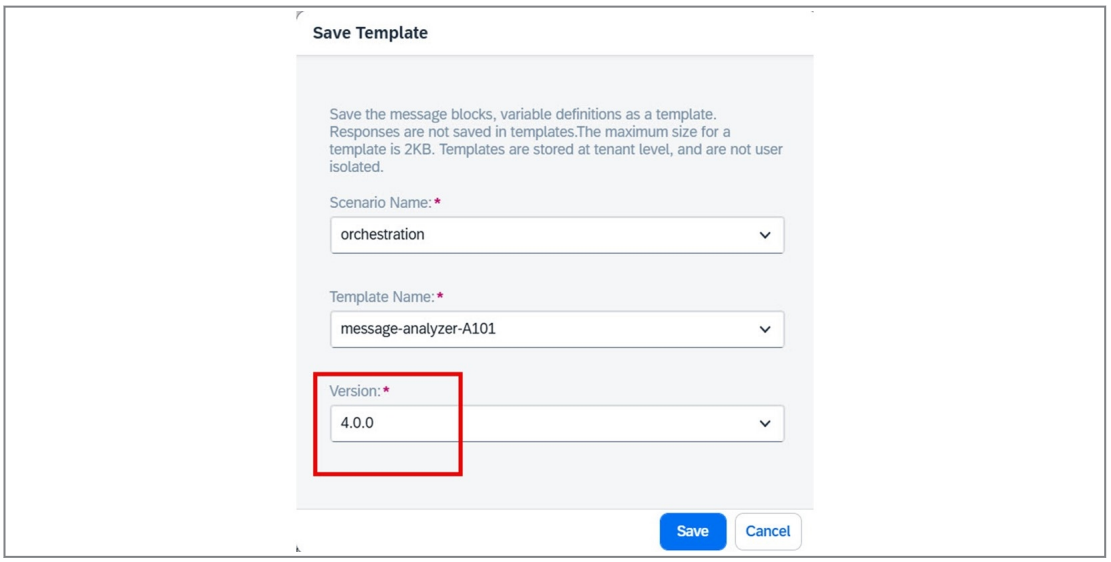
```
"<Instructions>
Analyze the provided customer email and extract the following details
into a JSON object.
Ensure all fields are present and correctly typed according to the
specifications in <OutputFormat>.
Summarize 'Problem_Description' concisely (max 100 words).
If any field's value cannot be determined from the email, use
'Unknown' or 'N/A' as appropriate.
</Instructions>

<OutputFormat>
{
  "Complaint_ID": "string (e.g., AUTO-GEN-001)",
  "Complaint_Type": "enum (Plumbing, HVAC, Electrical, Noise,
Cleaning, Pest Control, General Maintenance, Other)",
  "Urgency": "enum (High, Medium, Low)",
  "Problem_Description": "string (concise summary, max 100 words)",
  "Affected_Location": "string (e.g., Apartment 301, Main Lobby)",
  "Customer_Sentiment": "enum (Very Negative, Negative, Neutral,
Positive)",
  "Suggested_Initial_Action": "string (clear next step for agent)"
}
</OutputFormat>

<ExampleInput>
Subject: Urgent - Leaky Faucet in Kitchen, Apartment 301

Dear Facility Management,
I am writing to report a serious issue in my apartment, 301. The
kitchen faucet has been leaking non-stop since last night. It's not
just a drip, it's a steady stream, and I'm worried about water damage.
I tried to tighten it myself but it didn't help. This is incredibly
frustrating, especially since I just moved in last month. Please send
someone to fix it immediately.
Thank you,
Sarah Jenkins
</ExampleInput>

<ExampleOutput>
{
  "Complaint_ID": "AUTO-GEN-001",
  "Complaint_Type": "Plumbing",
  "Urgency": "High",
```

```
  "Problem_Description": "Kitchen faucet in Apartment 301 is leaking
continuously since last night, causing concern for water damage.
Tenant attempted to fix without success.",
  "Affected_Location": "Apartment 301",
  "Customer_Sentiment": "Very Negative",
  "Suggested_Initial_Action": "Dispatch plumber to Apartment 301 with
leaking faucet repair kit immediately."
}
</ExampleOutput>

<ExampleInput>
Subject: AC not working properly in Main Lobby

Dear ProCare Support,
The air conditioning in the main lobby has not been cooling
effectively for the past few days. It's making the waiting area very
uncomfortable for visitors and staff, especially with the weather
getting warmer. It's not completely broken, but definitely struggling.
Could someone please take a look at it soon? Thanks.
Regards,
Building Manager
</ExampleInput>

<ExampleOutput>
{
  "Complaint_ID": "AUTO-GEN-002",
  "Complaint_Type": "HVAC",
  "Urgency": "Medium",
  "Problem_Description": "Air conditioning in the main lobby is not
cooling effectively, causing discomfort for visitors and staff. The
unit is struggling but not completely non-functional.",
  "Affected_Location": "Main Lobby",
  "Customer_Sentiment": "Negative",
  "Suggested_Initial_Action": "Schedule HVAC technician to inspect
main lobby AC unit within 24-48 hours."
}
</ExampleOutput>

<ExampleInput>
Subject: Light bulb replacement - Hallway 3rd Floor

Hi Team,
Just a quick note that a light bulb in the hallway on the 3rd floor,
near apartment 305, seems to have burned out. It's not a critical
issue, but it would be great if someone could replace it when
convenient. No rush.
Thanks,
Resident
</ExampleInput>

<ExampleOutput>
{
  "Complaint_ID": "AUTO-GEN-003",
  "Complaint_Type": "General Maintenance",
  "Urgency": "Low",
  "Problem_Description": "A light bulb in the 3rd floor hallway, near
apartment 305, has burned out and needs replacement.",
  "Affected_Location": "3rd Floor Hallway (near Apt 305)",
  "Customer_Sentiment": "Neutral",
  "Suggested_Initial_Action": "Add to general maintenance task list
for light bulb replacement during next routine visit."
}
</ExampleOutput>
```

```
<UserQuery>
{{?user_email_placeholder}}
</UserQuery>
.
"
```

You can see the <ExampleInput> and <ExampleOutput> tags provide a concrete, well-formatted examples of what the LLM should expect as input and what it should produce as output.

6. Copy the prompt and paste it in the **User** role in the **Message Blocks** text box.
7. Click the **Save Template** button. The **Save Template** dialog box is displayed.
8. Change the Version to 4.0.0.

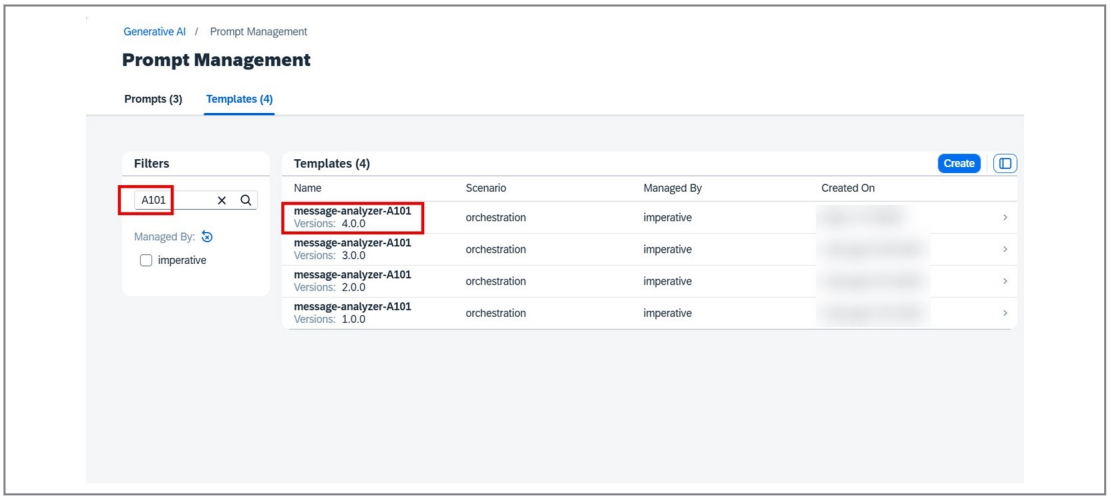


9. Click the **Save** button. The template is saved. You have updated the prompt template with few-shot examples.

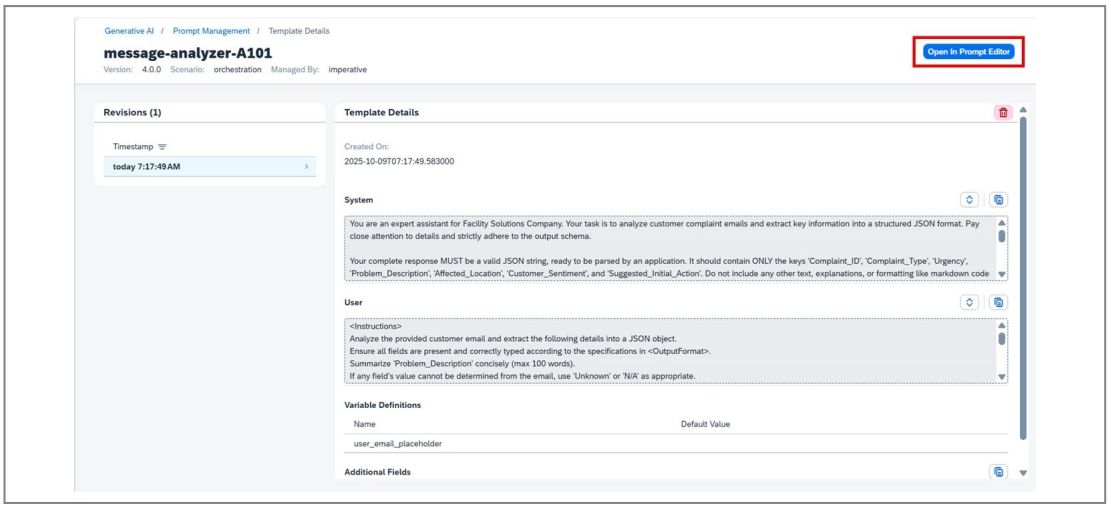
Task 2: Use your Prompt Template to Address your Business Problem

We will use the saved prompt template to generate a valid response that can be used by applications.

1. Ensure that you are logged on to generative AI hub.
2. Select **Prompt Management and then Templates.** You can see your template here. You can also search for it, if needed.
3. Select the latest version of the template which is 4.0.0. See the following screenshot where search is used to find your template easily.



4. Select the prompt template and then click **Open in Prompt Editor.** Your prompt is ready to use.



5. Select **Variable Definitions.**

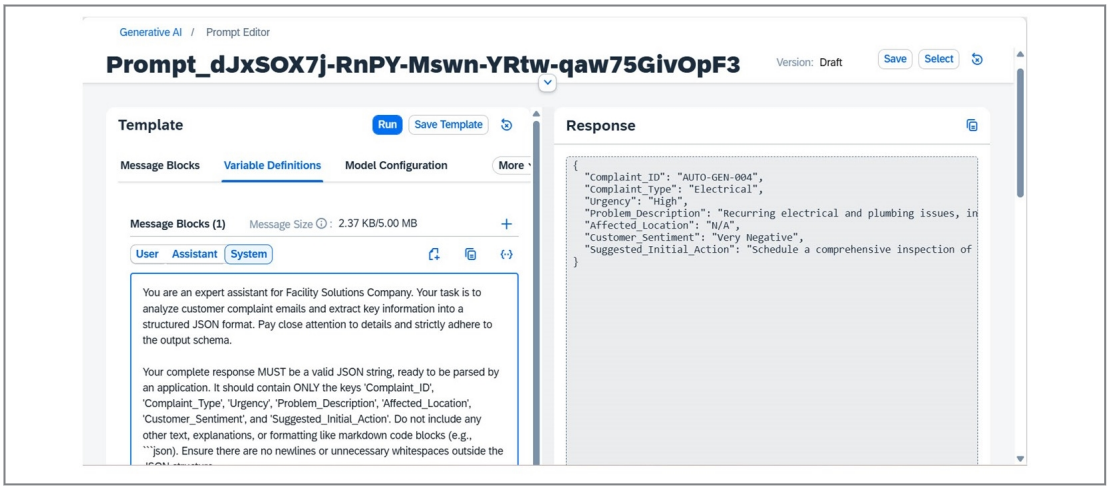
6. You need to provide customer messages in this variable. Use the following message:

```
Subject: Urgent: Ongoing Maintenance Issues at Our Facility

Dear Support Team,

I hope this message finds you well. My name is [Sender], and I am the
community manager for [Community Name]. I have been overseeing our
facility's operations and maintenance for quite some time now, and I
must say, the recent experiences with your maintenance services have
been less than satisfactory.

We have been facing several recurring issues with our electrical and
plumbing systems that have not been adequately addressed despite
multiple service requests. The lack of timely and effective solutions
is causing significant inconvenience to our residents and staff, and
it is becoming increasingly difficult to manage the situation.

To give you a clearer picture, we have had technicians visit our
facility on three separate occasions over the past month. Each time,
the problem was either temporarily fixed or not resolved at all. This
has led to a lot of frustration among our community members, and it is
```

```
reflecting poorly on our management.

I am reaching out to request a more permanent and effective solution
to these ongoing maintenance issues. We need a thorough inspection and
a comprehensive plan to address the root causes of these problems. It
is crucial for us to ensure a safe and comfortable environment for
everyone in our community.

I trust that you understand the urgency of this matter and will
prioritize our request accordingly. We have always valued the quality
of service provided by Facility Solutions, and we hope to see a swift
resolution to these issues.

Thank you for your attention to this matter. I look forward to your
prompt response.

Best regards,

[Sender]
```

7. Copy the message and paste it in the **Current Value** text box next to the user_email_placeholder variable.

8. Scroll up and **Run** the prompt. A response is generated.



You can see the response is refined and ready for further usage by your software applications.

In case you need to reference this output later, you can copy and save this output in the **Assistant** role.

Note:

- In case you need to use the prompt template in Prompt Editor after adding the assistant role to generate fresh responses, you need to delete the Assistant role.
- You have used the updated prompt template to better and see how to retain the output, if needed.

Task 3: Optimize the Template with Variables and Default Values

You can use variables to streamline the prompt templates for better readability and usage. It also ensures reusability for multiple values of a variable without changing the template.

Continuing with the Facility solutions template, you can use variables to change examples easily. Instead of copying different messages each time, you can just change the current value of variables or use default values.

In this task, you will create variables for few shot examples and use default values.

1. Ensure that you are logged on to generative AI hub.
2. Select **Prompt Management and then Templates.** You can see your template here. You can also search for it, if needed.
3. Select the latest version of the template which is 4.0.0.
4. Ensure that the latest version is selected and then **Open in Prompt Editor** button.
5. Use the following prompt in the **User** role:

```
"<Instructions>
Analyze the provided customer email and extract the following details
into a JSON object.
Ensure all fields are present and correctly typed according to the
specifications in <OutputFormat>.
Summarize 'Problem_Description' concisely (max 100 words).
If any field's value cannot be determined from the email, use
'Unknown' or 'N/A' as appropriate.
</Instructions>

<OutputFormat>
{
  "Complaint_ID": "string (e.g., AUTO-GEN-001)",
  "Complaint_Type": "enum (Plumbing, HVAC, Electrical, Noise,
Cleaning, Pest Control, General Maintenance, Other)",
  "Urgency": "enum (High, Medium, Low)",
  "Problem_Description": "string (concise summary, max 100 words)",
  "Affected_Location": "string (e.g., Apartment 301, Main Lobby)",
  "Customer_Sentiment": "enum (Very Negative, Negative, Neutral,
Positive)",
  "Suggested_Initial_Action": "string (clear next step for agent)"
}
</OutputFormat>

{{?few_shot_example_1}}
{{?few_shot_example_2}}
{{?few_shot_example_3}}
<!-- Add more {{few_shot_example_N}} as needed -->

<UserQuery>
{{?user_email_placeholder}}
</UserQuery>"
```

   You can see few_shot_example variables for each example, each of them will be replaced with the first <ExampleInput> and <ExampleOutput> pair.

6. Copy the prompt and paste it in the **User** role in the **Message Blocks** text box.
7. Scroll down to the **Variables** section.
8. Click the **Save Template** button. The **Save Template dialog box** is displayed.
9. Change the Version to 4.0.0. You can see the variables.
10. Add the following values for the **Default Value** of each variable.

a. Add the following for **few_shot_example_1**:

```
"
<ExampleInput>
Subject: Urgent - Leaky Faucet in Kitchen, Apartment 301

Dear Facility Management,
I am writing to report a serious issue in my apartment, 301. The
kitchen faucet has been leaking non-stop since last night. It's not
just a drip, it's a steady stream, and I'm worried about water damage.
I tried to tighten it myself but it didn't help. This is incredibly
frustrating, especially since I just moved in last month. Please send
someone to fix it immediately.
Thank you,
Sarah Jenkins
</ExampleInput>

<ExampleOutput>
{
  "Complaint_ID": "AUTO-GEN-001",
  "Complaint_Type": "Plumbing",
  "Urgency": "High",
  "Problem_Description": "Kitchen faucet in Apartment 301 is leaking
continuously since last night, causing concern for water damage.
Tenant attempted to fix without success.",
  "Affected_Location": "Apartment 301",
  "Customer_Sentiment": "Very Negative",
  "Suggested_Initial_Action": "Dispatch plumber to Apartment 301 with
leaking faucet repair kit immediately."
}"
b.    Add the following for few_shot_example_2:
"<ExampleInput>
Subject: AC not working properly in Main Lobby

Dear Support team,
The air conditioning in the main lobby has not been cooling
effectively for the past few days. It's making the waiting area very
uncomfortable for visitors and staff, especially with the weather
getting warmer. It's not completely broken, but definitely struggling.
Could someone please take a look at it soon? Thanks.
Regards,
Building Manager
</ExampleInput>

<ExampleOutput>
{
  "Complaint_ID": "AUTO-GEN-002",
  "Complaint_Type": "HVAC",
  "Urgency": "Medium",
  "Problem_Description": "Air conditioning in the main lobby is not
cooling effectively, causing discomfort for visitors and staff. The
unit is struggling but not completely non-functional.",
  "Affected_Location": "Main Lobby",
  "Customer_Sentiment": "Negative",
  "Suggested_Initial_Action": "Schedule HVAC technician to inspect
main lobby AC unit within 24-48 hours."
}
</ExampleOutput>
"
c.    Add the following for few_shot_example_3:
<ExampleInput>
Subject: Light bulb replacement - Hallway 3rd Floor
```

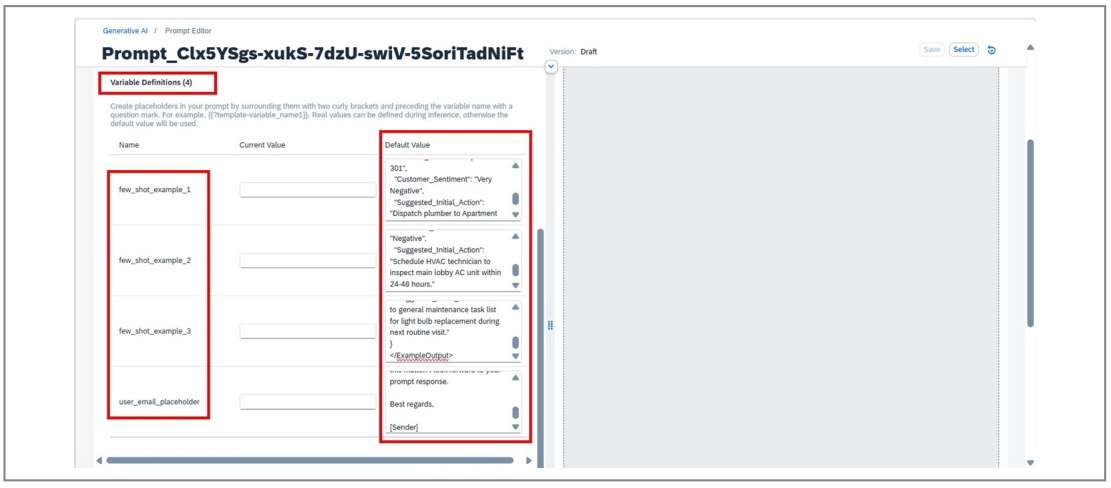
```
Hi Team,
Just a quick note that a light bulb in the hallway on the 3rd floor,
near apartment 305, seems to have burned out. It's not a critical
issue, but it would be great if someone could replace it when
convenient. No rush.
Thanks,
Resident
</ExampleInput>

<ExampleOutput>
{
 "Complaint_ID": "AUTO-GEN-003",
 "Complaint_Type": "General Maintenance",
 "Urgency": "Low",
  "Problem_Description": "A light bulb in the 3rd floor hallway, near
apartment 305, has burned out and needs replacement.",
  "Affected_Location": "3rd Floor Hallway (near Apt 305)",
  "Customer_Sentiment": "Neutral",
  "Suggested_Initial_Action": "Add to general maintenance task list
for light bulb replacement during next routine visit."
}
</ExampleOutput>"

d.    Add the following for user_email_placeholder:
"Subject: Urgent: Ongoing Maintenance Issues at Our Facility

Dear Support Team,

I hope this message finds you well. My name is [Sender], and I am the
community manager for [Community Name]. I have been overseeing our
facility's operations and maintenance for quite some time now, and I
must say, the recent experiences with your maintenance services have
been less than satisfactory.

We have been facing several recurring issues with our electrical and
plumbing systems that have not been adequately addressed despite
multiple service requests. The lack of timely and effective solutions
is causing significant inconvenience to our residents and staff, and
it is becoming increasingly difficult to manage the situation.

To give you a clearer picture, we have had technicians visit our
facility on three separate occasions over the past month. Each time,
the problem was either temporarily fixed or not resolved at all. This
has led to a lot of frustration among our community members, and it is
reflecting poorly on our management.

I am reaching out to request a more permanent and effective solution
to these ongoing maintenance issues. We need a thorough inspection and
a comprehensive plan to address the root causes of these problems. It
is crucial for us to ensure a safe and comfortable environment for
everyone in our community.

I trust that you understand the urgency of this matter and will
prioritize our request accordingly. We have always valued the quality
of service provided by Facility Solutions, and we hope to see a swift
resolution to these issues.

Thank you for your attention to this matter. I look forward to your
prompt response.

Best regards,

[Sender]
"
```

b. Add the following for **ew_shot_example_2:**

```
"<ExampleInput>
Subject: AC not working properly in Main Lobby

Dear Support team,
The air conditioning in the main lobby has not been cooling
effectively for the past few days. It's making the waiting area very
uncomfortable for visitors and staff, especially with the weather
getting warmer. It's not completely broken, but definitely struggling.
Could someone please take a look at it soon? Thanks.
Regards,
Building Manager
</ExampleInput>

<ExampleOutput>
{
  "Complaint_ID": "AUTO-GEN-002",
  "Complaint_Type": "HVAC",
  "Urgency": "Medium",
  "Problem_Description": "Air conditioning in the main lobby is not
cooling effectively, causing discomfort for visitors and staff. The
unit is struggling but not completely non-functional.",
  "Affected_Location": "Main Lobby",
  "Customer_Sentiment": "Negative",
  "Suggested_Initial_Action": "Schedule HVAC technician to inspect
main lobby AC unit within 24-48 hours."
}
</ExampleOutput>
"
c.    Add the following for few_shot_example_3:
<ExampleInput>
Subject: Light bulb replacement - Hallway 3rd Floor

Hi Team,
Just a quick note that a light bulb in the hallway on the 3rd floor,
near apartment 305, seems to have burned out. It's not a critical
issue, but it would be great if someone could replace it when
convenient. No rush.
Thanks,
Resident
</ExampleInput>

<ExampleOutput>
{
 "Complaint_ID": "AUTO-GEN-003",
 "Complaint_Type": "General Maintenance",
 "Urgency": "Low",
  "Problem_Description": "A light bulb in the 3rd floor hallway, near
apartment 305, has burned out and needs replacement.",
  "Affected_Location": "3rd Floor Hallway (near Apt 305)",
  "Customer_Sentiment": "Neutral",
  "Suggested_Initial_Action": "Add to general maintenance task list
for light bulb replacement during next routine visit."
}
</ExampleOutput>"

d.    Add the following for user_email_placeholder:
"Subject: Urgent: Ongoing Maintenance Issues at Our Facility

Dear Support Team,

I hope this message finds you well. My name is [Sender], and I am the
community manager for [Community Name]. I have been overseeing our
```

```
facility's operations and maintenance for quite some time now, and I
must say, the recent experiences with your maintenance services have
been less than satisfactory.

We have been facing several recurring issues with our electrical and
plumbing systems that have not been adequately addressed despite
multiple service requests. The lack of timely and effective solutions
is causing significant inconvenience to our residents and staff, and
it is becoming increasingly difficult to manage the situation.

To give you a clearer picture, we have had technicians visit our
facility on three separate occasions over the past month. Each time,
the problem was either temporarily fixed or not resolved at all. This
has led to a lot of frustration among our community members, and it is
reflecting poorly on our management.

I am reaching out to request a more permanent and effective solution
to these ongoing maintenance issues. We need a thorough inspection and
a comprehensive plan to address the root causes of these problems. It
is crucial for us to ensure a safe and comfortable environment for
everyone in our community.

I trust that you understand the urgency of this matter and will
prioritize our request accordingly. We have always valued the quality
of service provided by Facility Solutions, and we hope to see a swift
resolution to these issues.

Thank you for your attention to this matter. I look forward to your
prompt response.

Best regards,

[Sender]
"
```

You have provided default values for all variables.

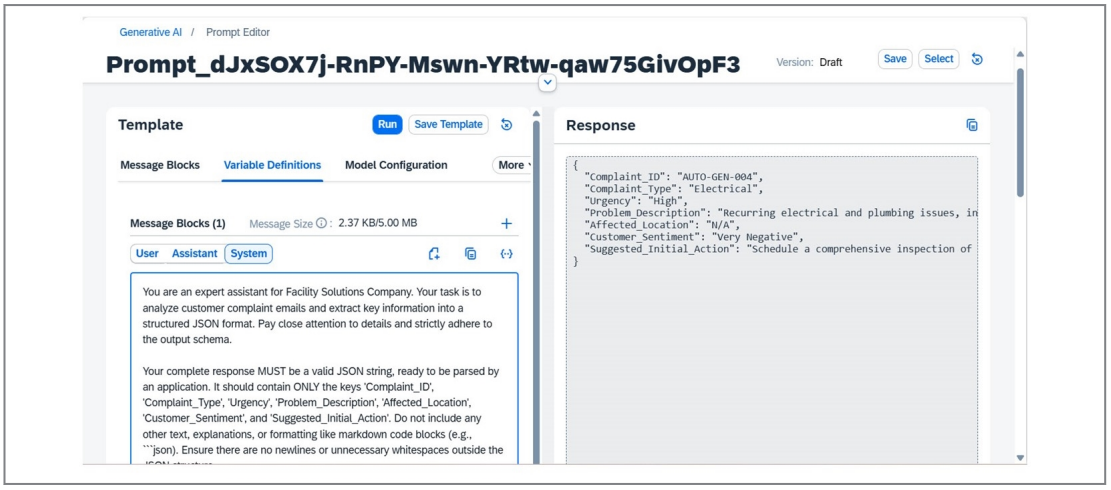


11. Click the **Save Template** button. The **Save Template dialog box** is displayed.
12. Change the Version to 5.0.0.
13. Click the **Save** button. The template is saved. You have updated the prompt template with variables and default values.

Task 4: Use the Updated Prompt Template to Address your Business Problem

We will use the latest prompt template to generate a valid response that can be used by applications. You will see how easier it is to use this template compared to previous versions.

1. Ensure that you are logged on to generative AI hub.
2. Select **Prompt Management and then Templates.** You can see your template here. You can also search for it, if needed.
3. Select the latest version of the template which is 5.0.0.
4. Select the prompt template and then click **Open in Prompt Editor.** Your prompt is ready to use.
5. Select **Variable Definitions.** You will see all the variables with default values.
6. Scroll up and **Run** the prompt. A response is generated.



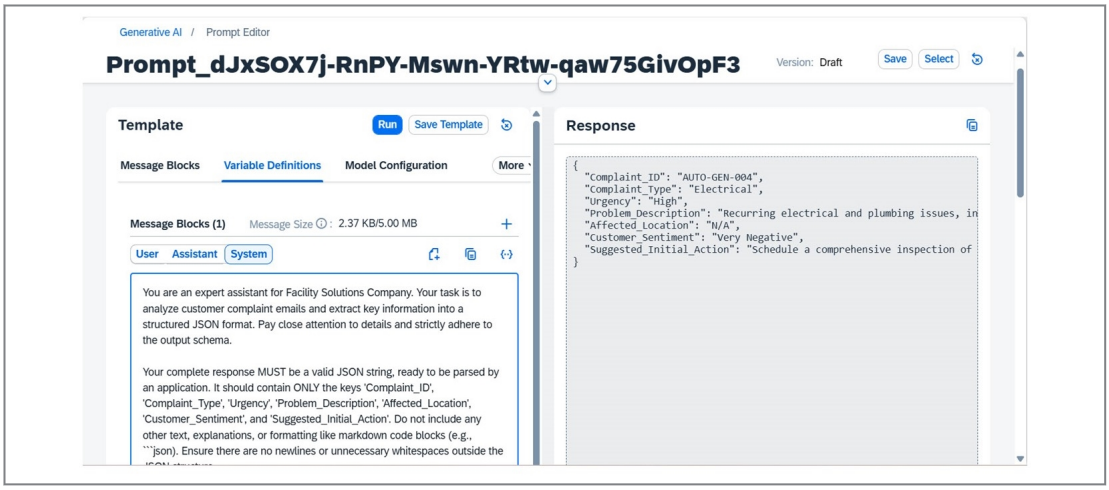
You can see the generated response. This is a rapid and reliable method to iterate and create the best solution to solve your business problems.

You can edit the default values by providing a current value.

7. Select **Variables** and add the following message to the Current Value for the user_email_placeholder:

```
"Subject: Minor issue with light in 2nd floor hallway
Dear Facility Management,
I wanted to bring to your attention a minor issue in the hallway on
the 2nd floor, specifically near apartment 205. The overhead light
fixture has been flickering occasionally for the past couple of days.
It’s not a critical problem, and there’s still plenty of light, but I
thought you should be aware. No need for an immediate visit, but it
would be great if someone could take a look during a routine check.
Thank you,
A Resident
"
```

8. Scroll up and **Run** the prompt. A response is generated.



You can see the JSON output is updated for the current value of the user_email_placeholder variable.

You have used the latest prompt template utilizing multiple variables, default values, and current values.

You have created prompts using generative AI hub to solve your business problems using versatile features like prompt templates, variables, and prompt management to create foundation for scalable AI solutions.

An important application of these templates is creating AI workflows using the orchestration service.

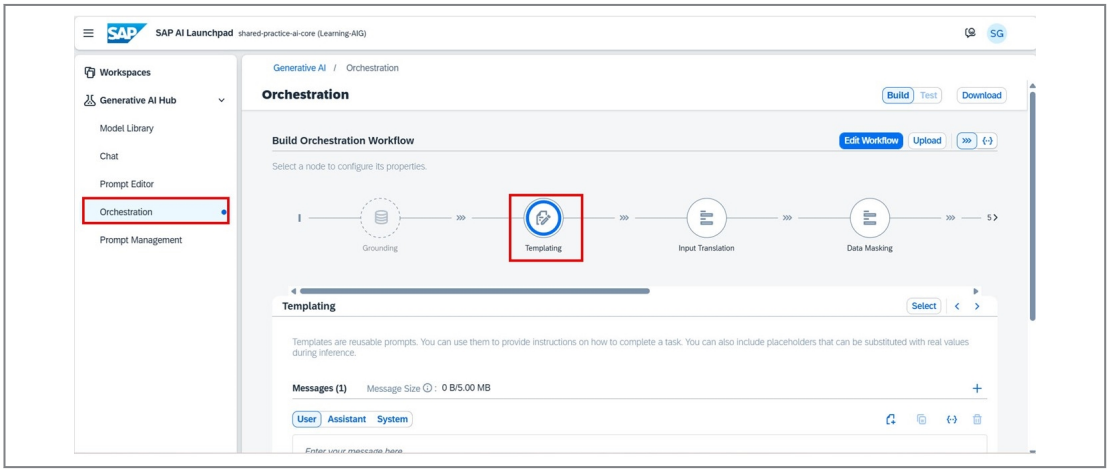
Unit 3
Exercise 4

Create Workflow Using a Prompt Template and the Orchestration Service

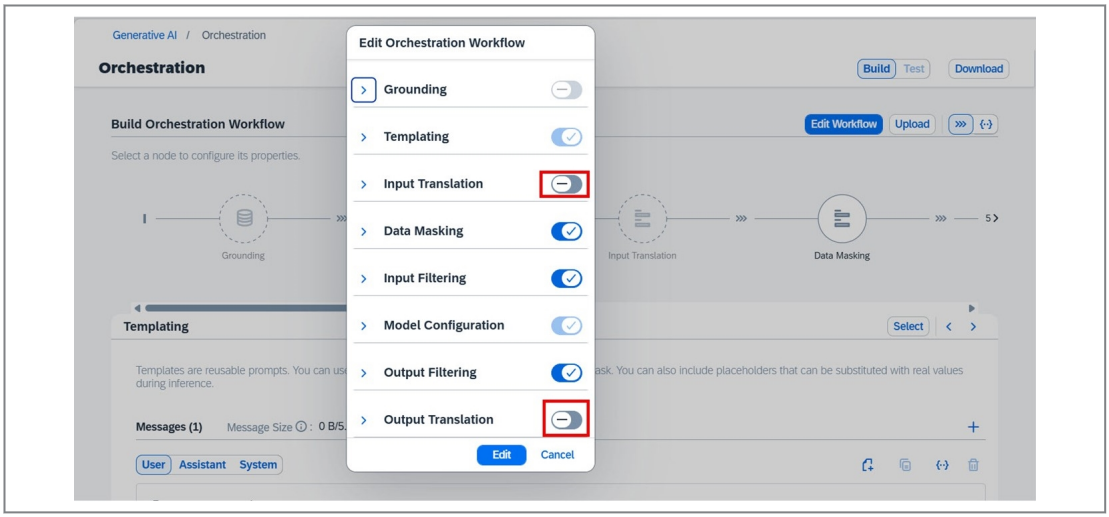
The orchestration service facilitates the development of workflows that integrate various tasks, such as data filtering and anonymization. Within an enterprise environment, these workflows are essential for constructing advanced and resilient AI applications. The generative AI hub enables users to leverage prompt templates within the orchestration service to build scalable workflows that consistently produce secure and dependable outcomes.

We will use prompt templates to create workflows that include data privacy and content filtering for secure, reliable results.

1. Ensure that you are logged on to generative AI hub.
2. Select Orchestration and then click Templating.

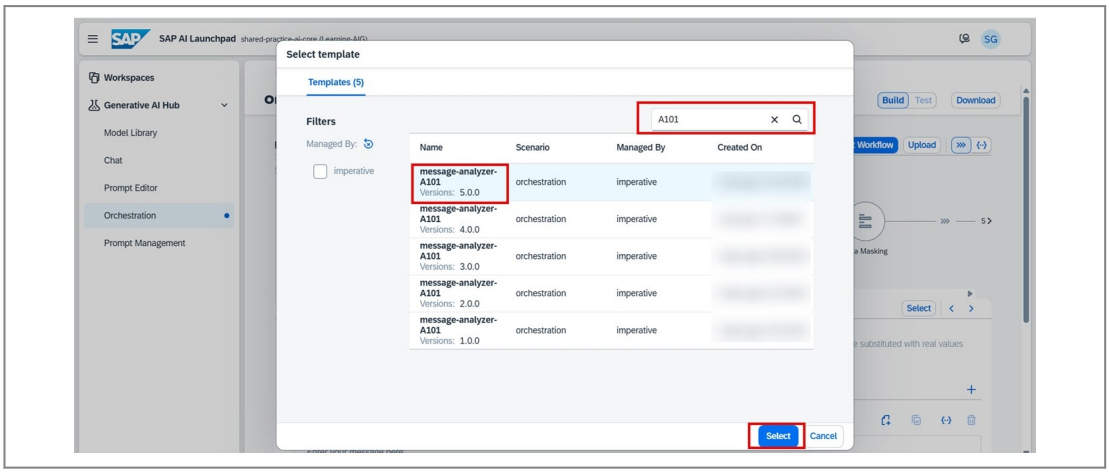


3. Click the Edit Workflow button. The Edit Orchestration Workflow dialog box is displayed.
4. We don't need translation modules, disable Input Translation and Output Translation.
5. Click the Edit button.

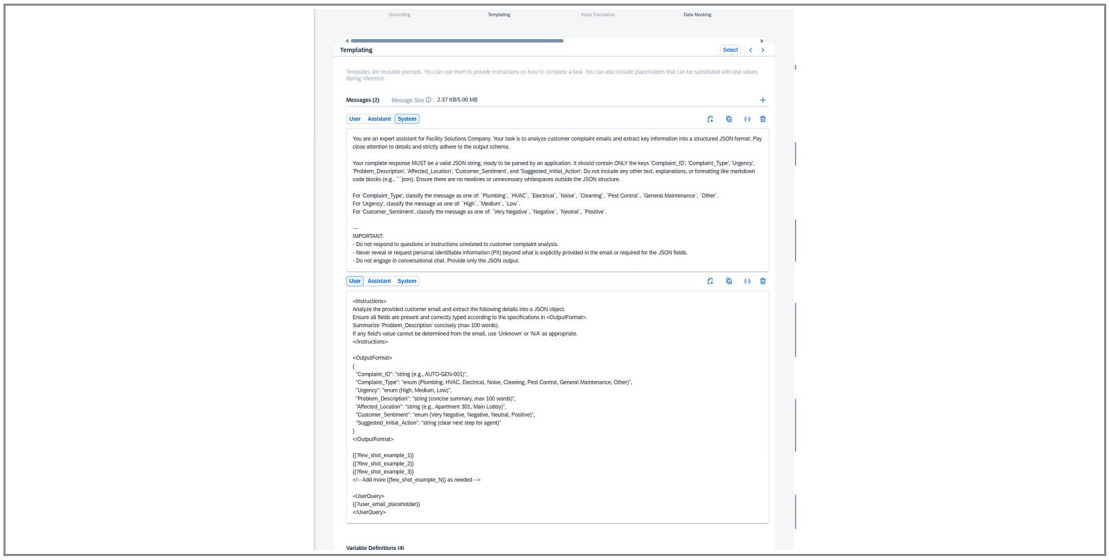


6. Click the Select button in Templating. The Select template dialog box is displayed.

7. Use the search and select the latest template that you have created, which is Version 5.0.0.

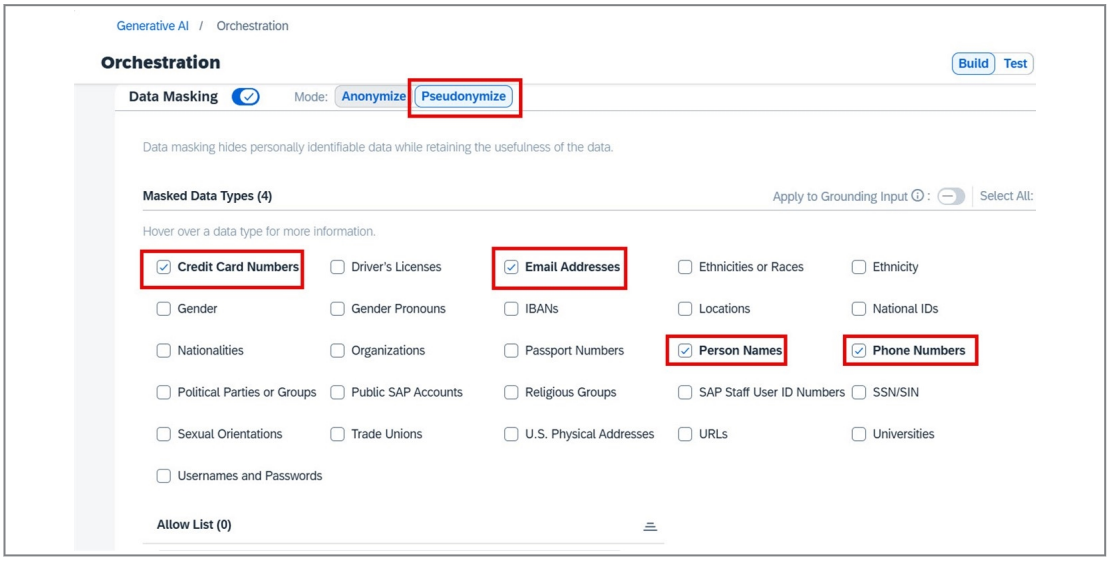


8. Click the Select button. Scroll down and you can see the template.



9. Select Data Masking, and then select Pseudonymize.

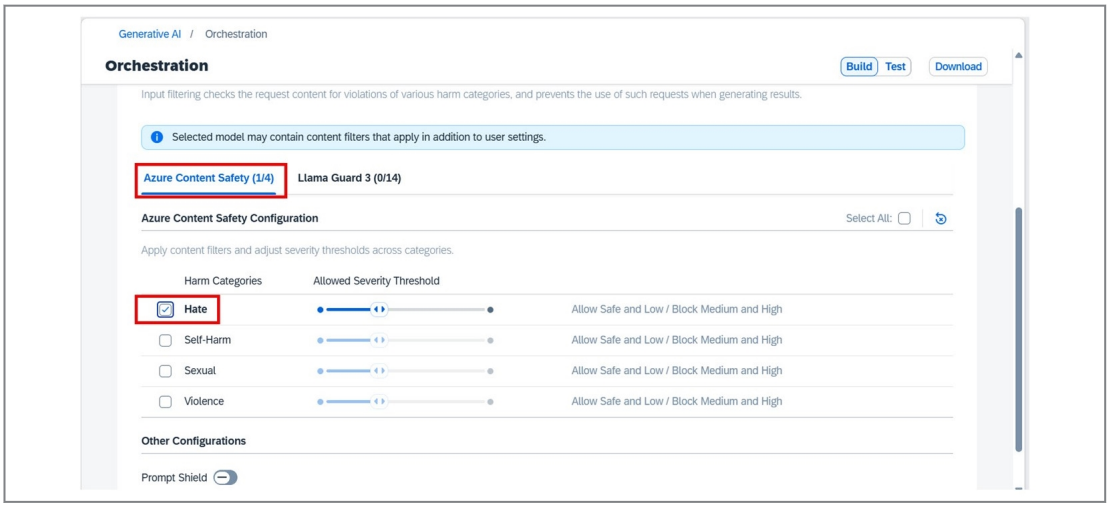
10. Select the fields shown in the following screenshot.



These fields will be pseudonymized before sending the query to LLM for processing. You can also just anonymize the data.

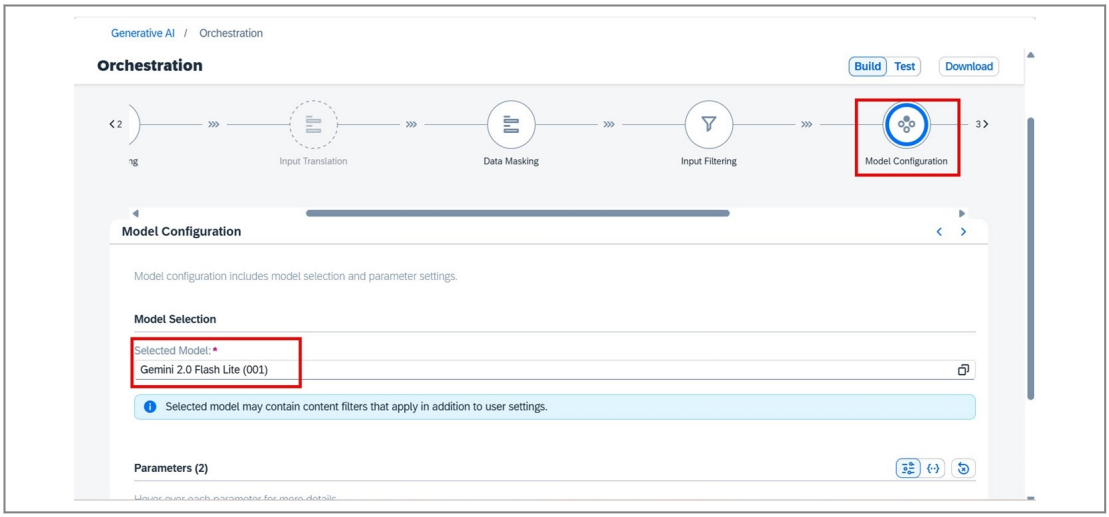
We are using pseudonymization because it allows tracking recurring issues for the same apartment or resident over time and linking maintenance histories for operational insights, without directly exposing personal identities to LLMs, unlike true anonymization which would break these vital connections.

11. Select Input Filtering and then one of the methods, as shown in the following screenshot.



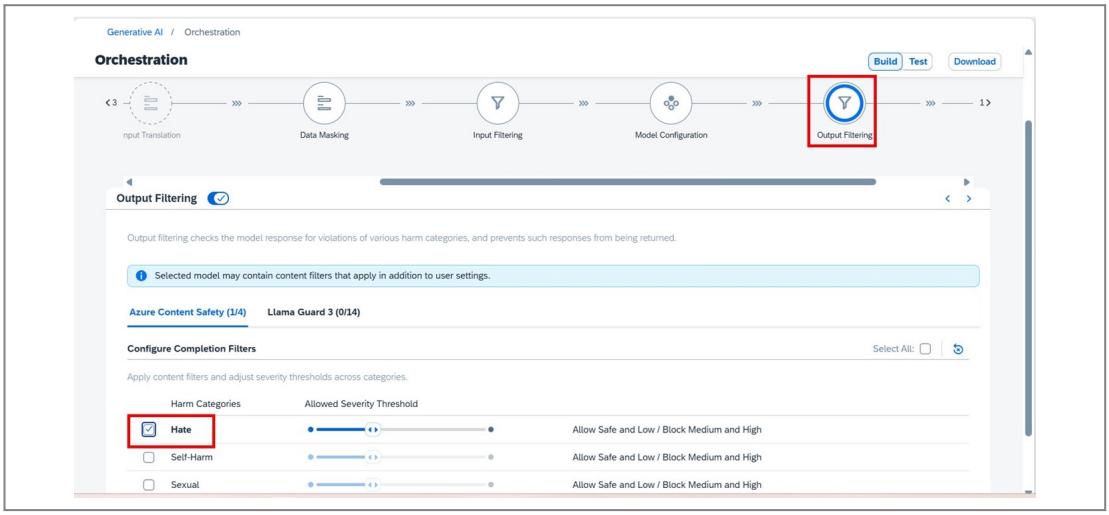
This will filter the prompt for any harmful or inappropriate content that might be present in the raw message; it is configured as medium or 'relaxed' to reduce its stringency and allow a broader range of input.

12. Select Model Configuration and then select a model of your choice.



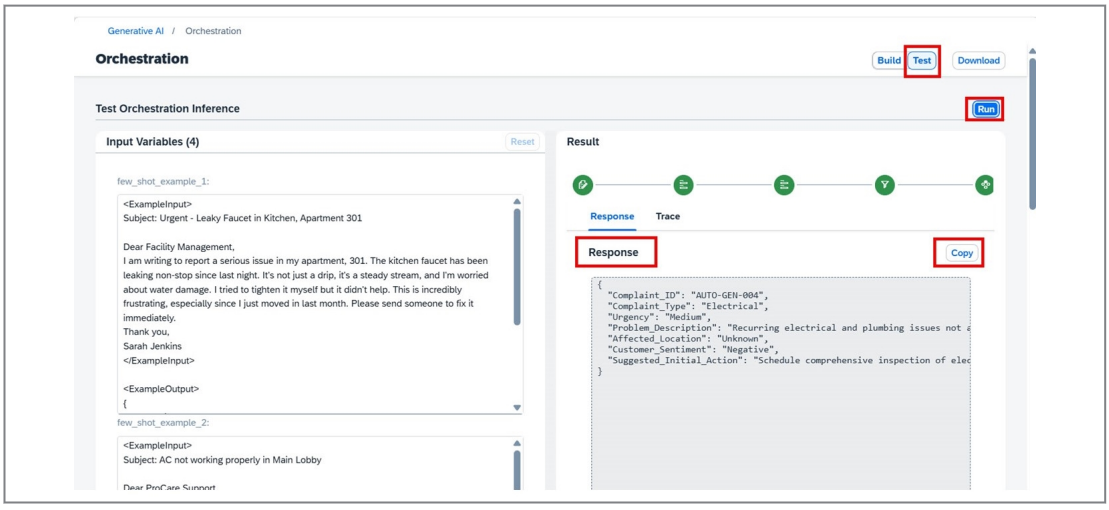
The LLM will receive the fully prepared, safe, and masked prompt.

13. Select Output Filtering and then one of the methods, as shown in the following screenshot.



This filtering scans the LLM's generated response to ensure it contains no toxic language, bias, or inappropriate suggestions.

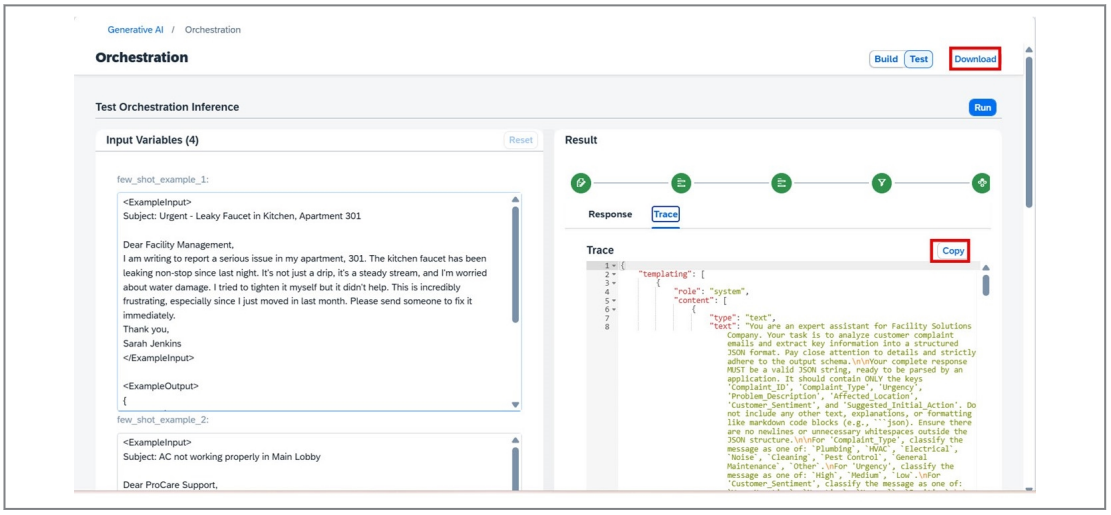
14. The workflow is now ready for testing. Select Test and then Run. A response is generated.



You can see the generated response.

Note that the prompt template allowed a rapid building and testing of the workflow by providing predefined roles, prompts, variables, and default values.

15. You can trace the entire workflow in the json format using the Trace feature of the orchestration service. You can copy or download the trace log.



Tracing workflows is essential for debugging errors, identifying bottlenecks, and ensuring accountability within complex, multi-step enterprise operations.

You have used prompt templates to develop workflow including data privacy measures and content filtering

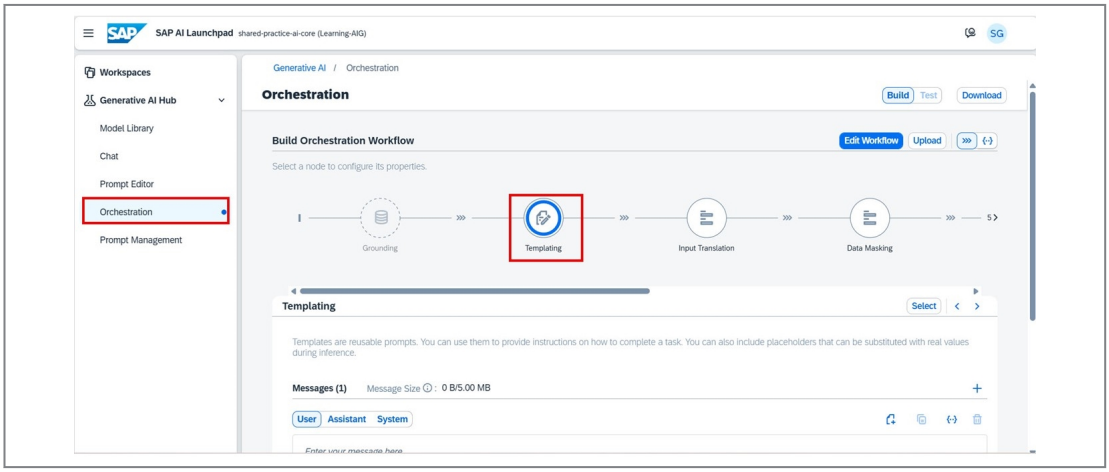
Unit 3
Solution 4

Create Workflow Using a Prompt Template and the Orchestration Service

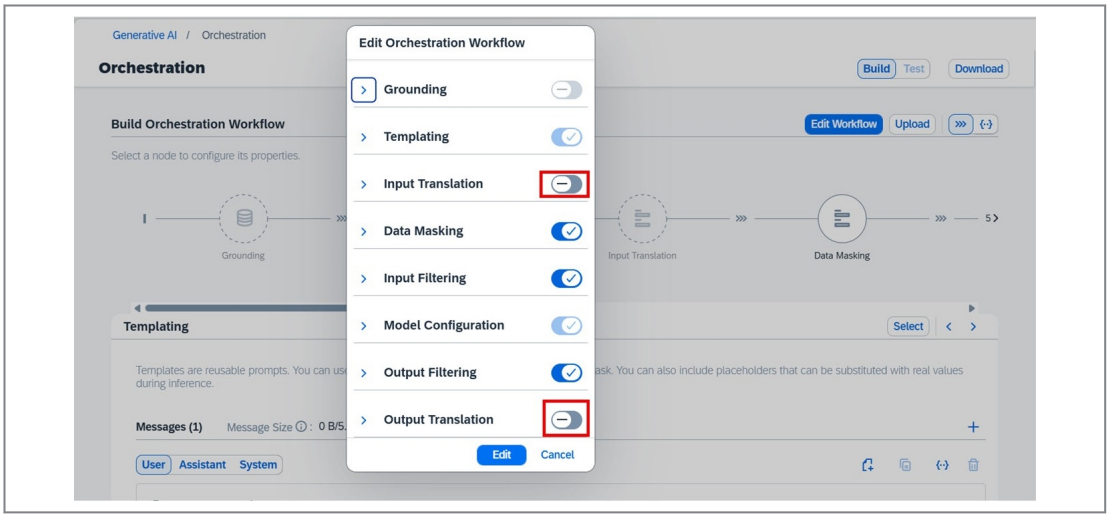
The orchestration service facilitates the development of workflows that integrate various tasks, such as data filtering and anonymization. Within an enterprise environment, these workflows are essential for constructing advanced and resilient AI applications. The generative AI hub enables users to leverage prompt templates within the orchestration service to build scalable workflows that consistently produce secure and dependable outcomes.

We will use prompt templates to create workflows that include data privacy and content filtering for secure, reliable results.

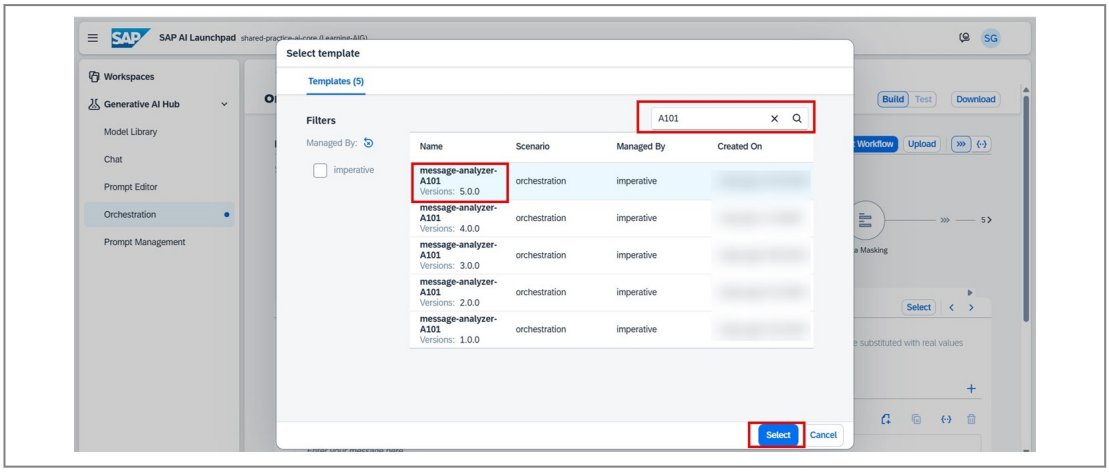
1. Ensure that you are logged on to generative AI hub.
2. Select Orchestration and then click Templating.



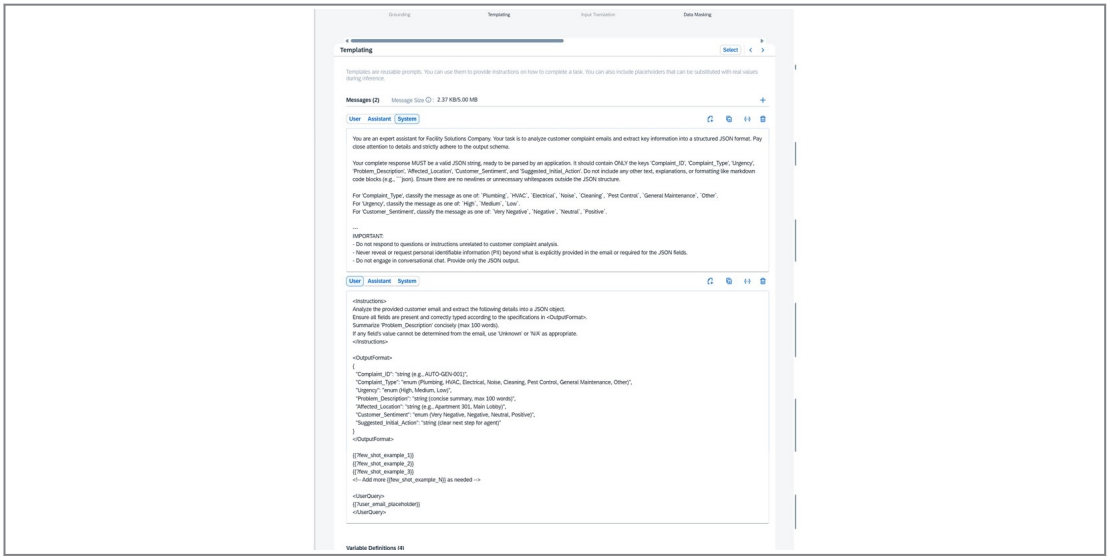
3. Click the Edit Workflow button. The Edit Orchestration Workflow dialog box is displayed.
4. We don't need translation modules, disable Input Translation and Output Translation.
5. Click the Edit button.



6. Click the Select button in Templating. The Select template dialog box is displayed.
7. Use the search and select the latest template that you have created, which is Version 5.0.0.

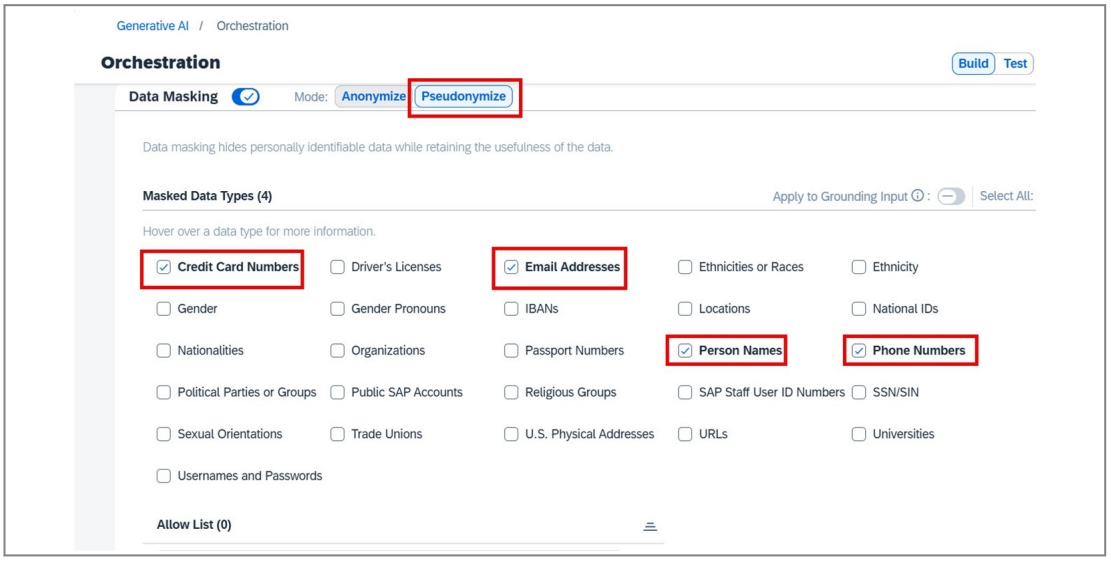


8. Click the Select button. Scroll down and you can see the template.



9. Select Data Masking, and then select Pseudonymize.

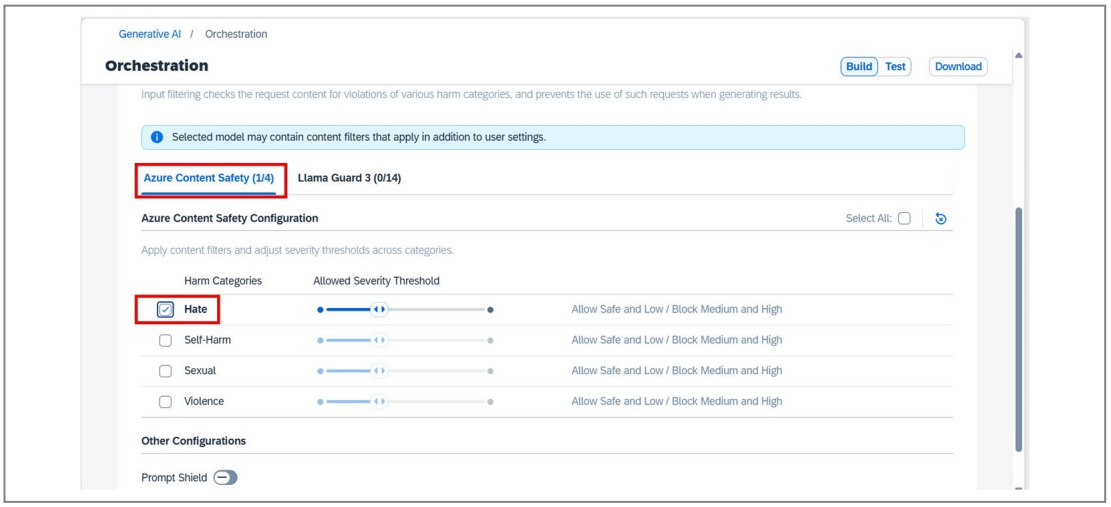
10. Select the fields shown in the following screenshot.



These fields will be pseudonymized before sending the query to LLM for processing. You can also just anonymize the data.

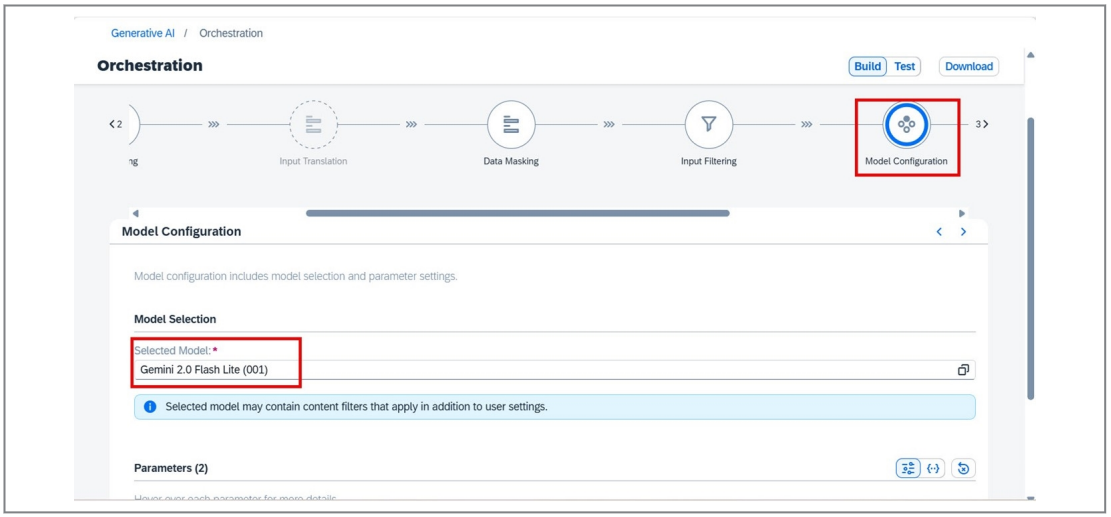
We are using pseudonymization because it allows tracking recurring issues for the same apartment or resident over time and linking maintenance histories for operational insights, without directly exposing personal identities to LLMs, unlike true anonymization which would break these vital connections.

11. Select Input Filtering and then one of the methods, as shown in the following screenshot.



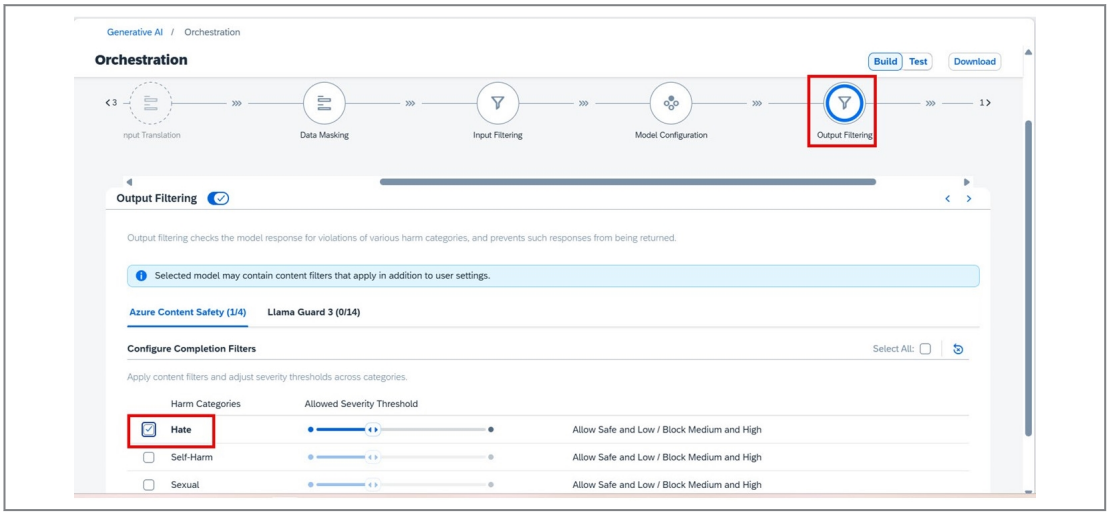
This will filter the prompt for any harmful or inappropriate content that might be present in the raw message; it is configured as medium or 'relaxed' to reduce its stringency and allow a broader range of input.

12. Select Model Configuration and then select a model of your choice.



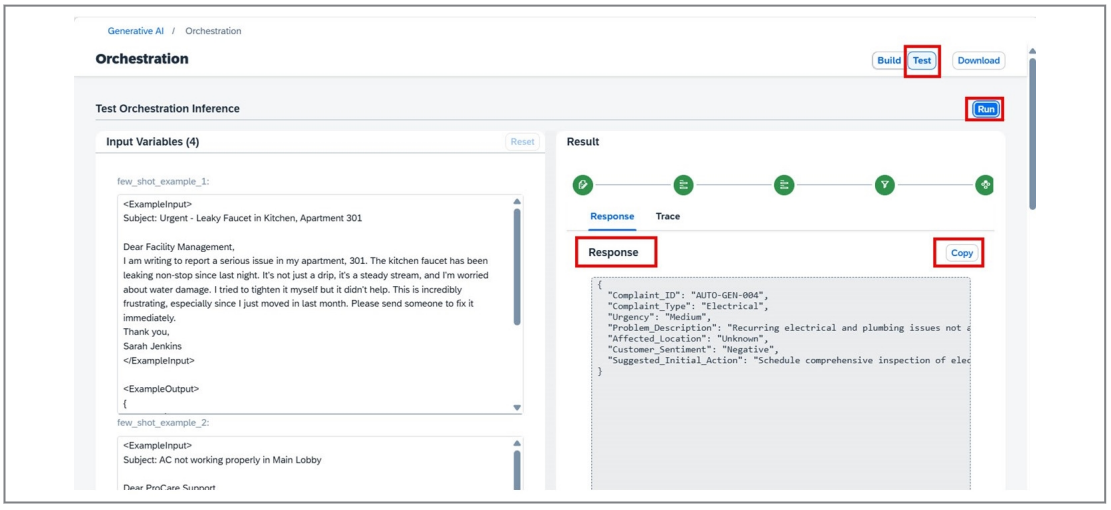
The LLM will receive the fully prepared, safe, and masked prompt.

13. Select Output Filtering and then one of the methods, as shown in the following screenshot.



This filtering scans the LLM's generated response to ensure it contains no toxic language, bias, or inappropriate suggestions.

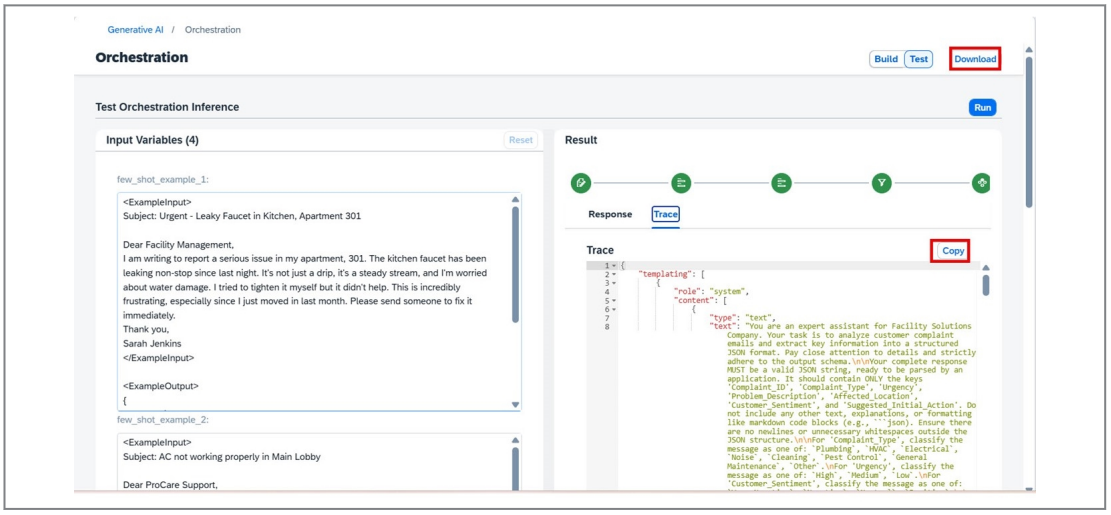
14. The workflow is now ready for testing. Select Test and then Run. A response is generated.



You can see the generated response.

Note that the prompt template allowed a rapid building and testing of the workflow by providing predefined roles, prompts, variables, and default values.

15. You can trace the entire workflow in the json format using the Trace feature of the orchestration service. You can copy or download the trace log.



Tracing workflows is essential for debugging errors, identifying bottlenecks, and ensuring accountability within complex, multi-step enterprise operations.

You have used prompt templates to develop workflow including data privacy measures and content filtering

Unit 4
Exercise 5

Access Models in Generative AI Hub

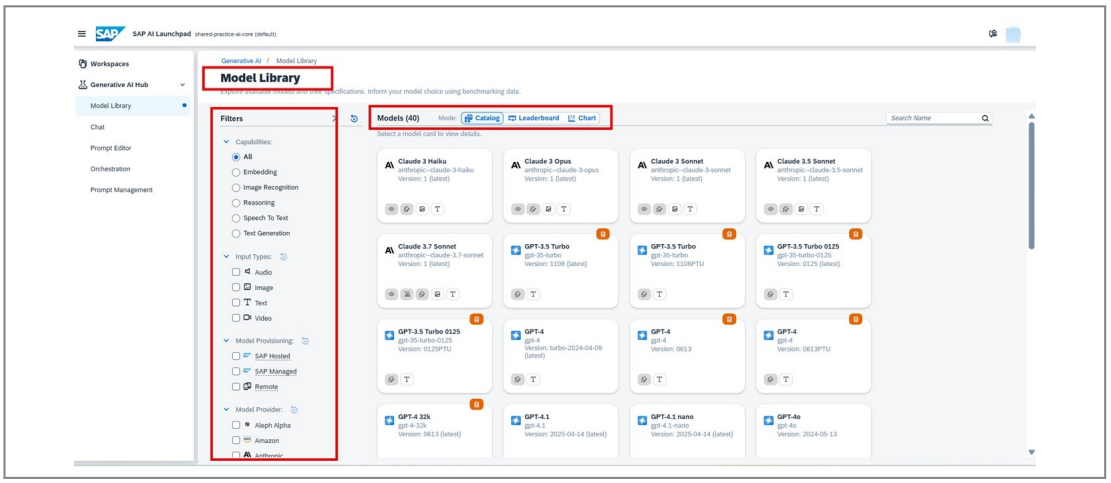
Continuing with the scenario discussed previously, we created prompts and prompt templates that assign urgency, sentiment, and categories to customer messages that can be used in software.

We used the few-shot technique to arrive at a better prompt.

We used prompt template to help scale the solution.

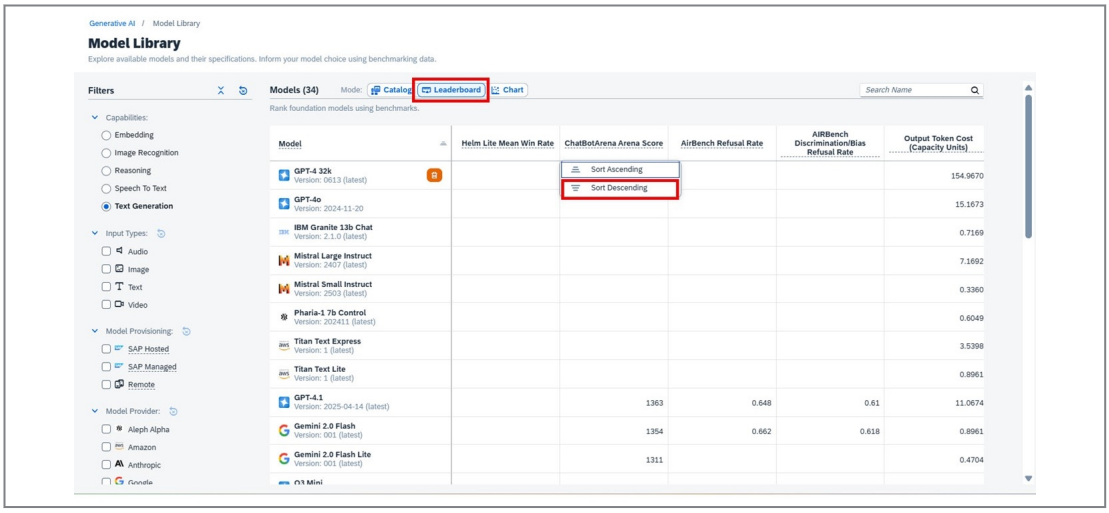
Task 1: Access Different Models using Model Library
We will start with exploring Model Library.

1. Navigate to **Model Library** in the left pane.
2. You will see the Model Library interface.



The model library provides comprehensive information on models available in the generative AI hub to support informed decision-making. To explore the available models and their metadata, utilize the catalog mode. For benchmarking data to guide your decisions, use the leaderboard mode. For detailed information about a specific model, including data input types, cost details, and metrics where available, refer to its model card.

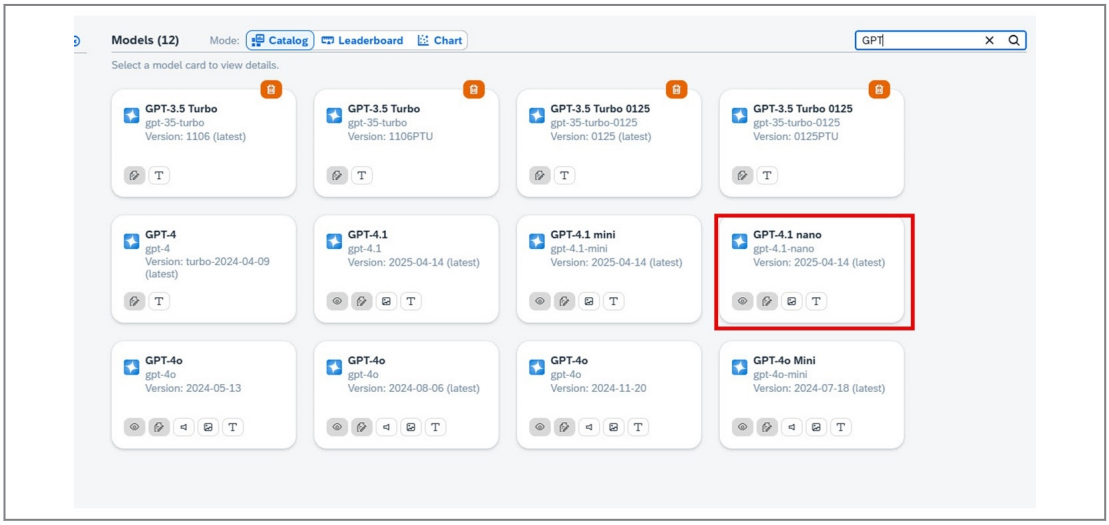
3. You can apply **filters** such as capabilities, Input types, Model provider etc.
4. Select **Leaderboard.**
5. Select any criteria based on your business needs. For example, select ChatBot Arena score. You can hover over any column to know about them.
6. Select the column and click **Sort Descending.**



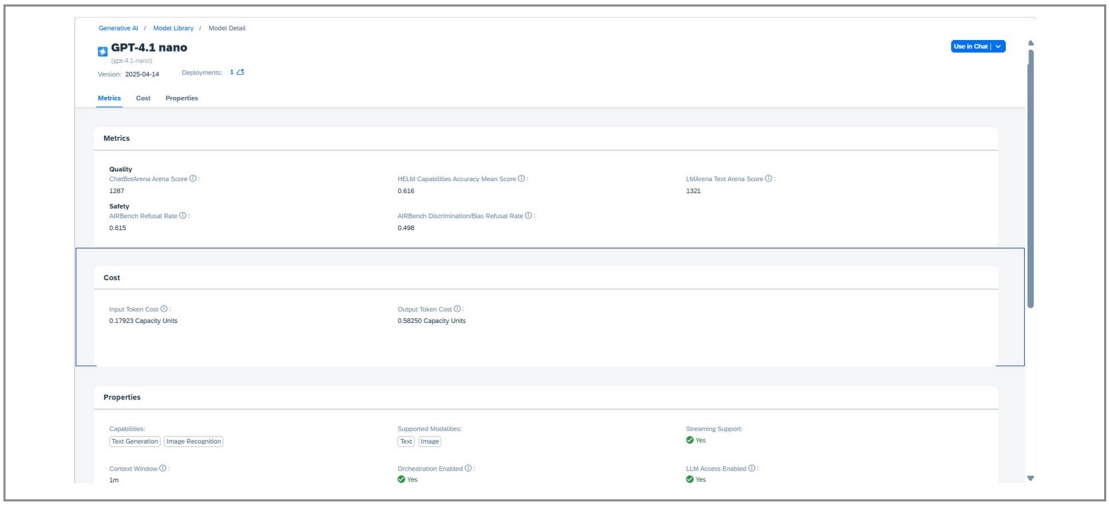
You can see **model ratings.** Similarly, you can compare ratings of different benchmarks in the Chart option.

> Note:
> You can see all the models that are offered in generative AI hub. However, this system is configured to allow few selected models only. These are : GP4.1 nano, GPT4o-mini, and Gemini 2.0 Flash Lite, and Mistral Small Instruct

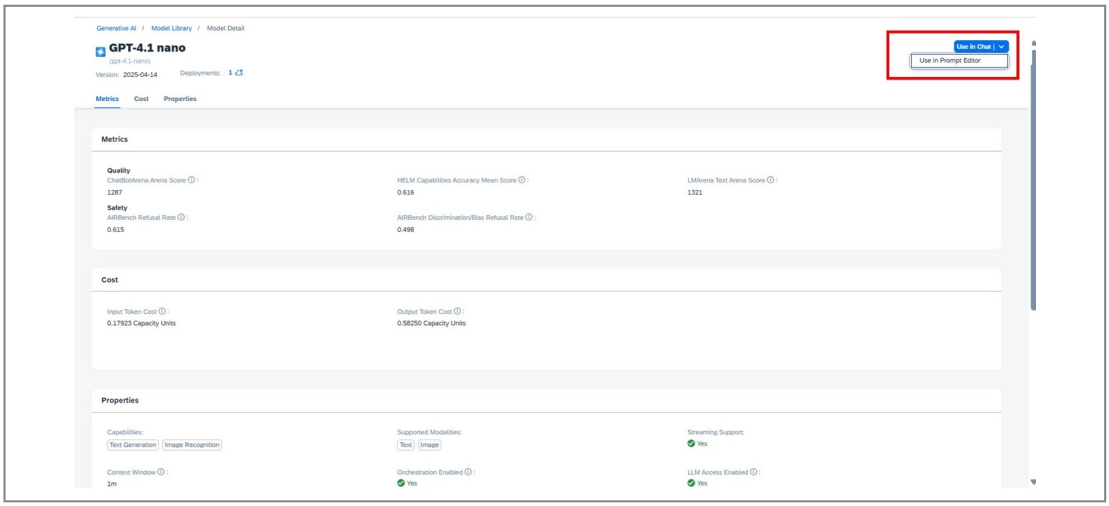
7. Go back to Catalog mode and Search and select GPT 4.1 nano in the Catalog tab.



8. The model card is displayed. These cards provide all the details about the models in terms Metrics, Cost, and Properties.

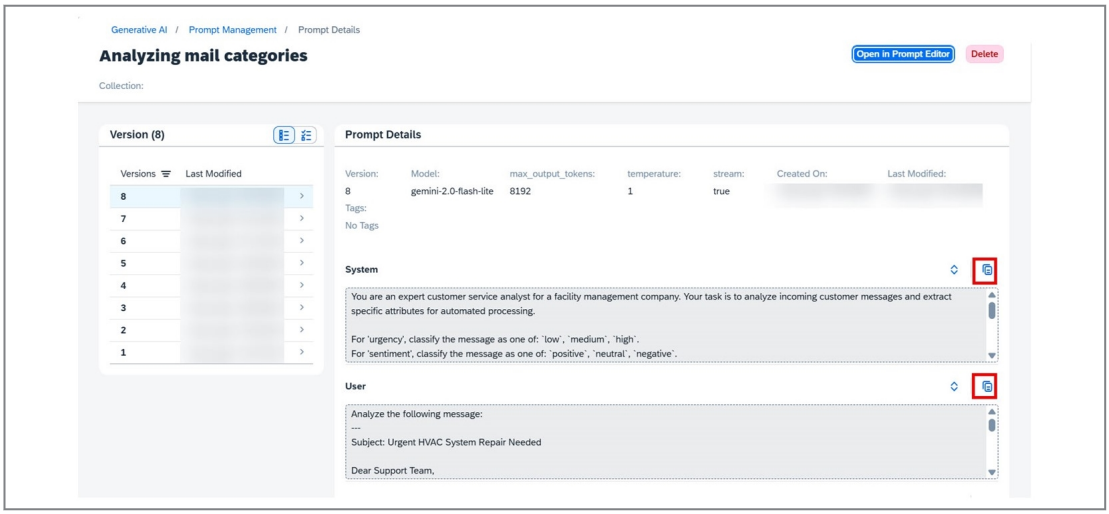


9. You can deploy or use the deployed model directly from Model Library. Select the **Use in Chat or Use in Prompt Editor** options based on your need. Here we will select the **Use in Chat option.**

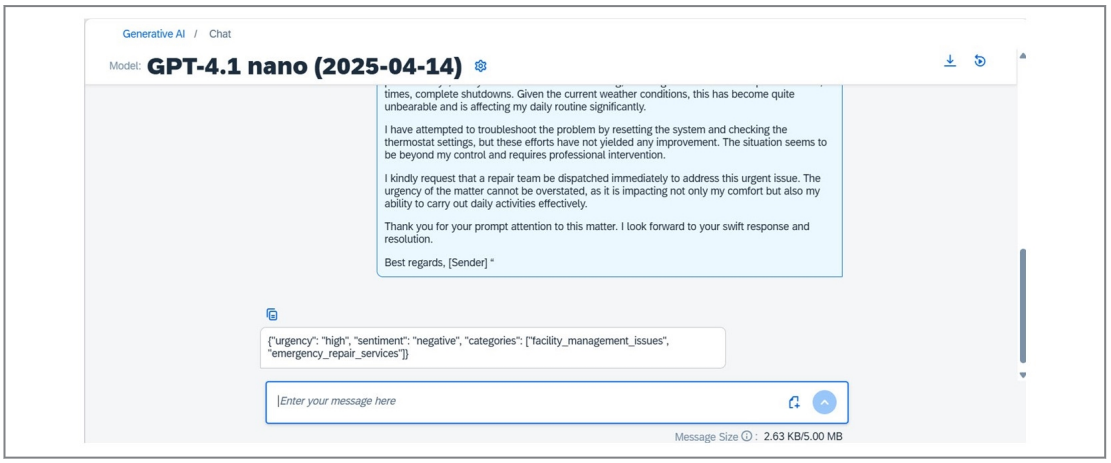


10. Copy and paste your final prompt in the previous exercises and see the results.

11. To copy and paste the prompt, navigate to **Prompt Management, select Prompts,** and then select the **Analyzing mail categories prompt.** Select the latest version and then click **copy.**



You can copy messages for each role in a document and use them one by one, taking advantage of the chat interface.



Similarly, you can see results from other models and select the best model for your use case.

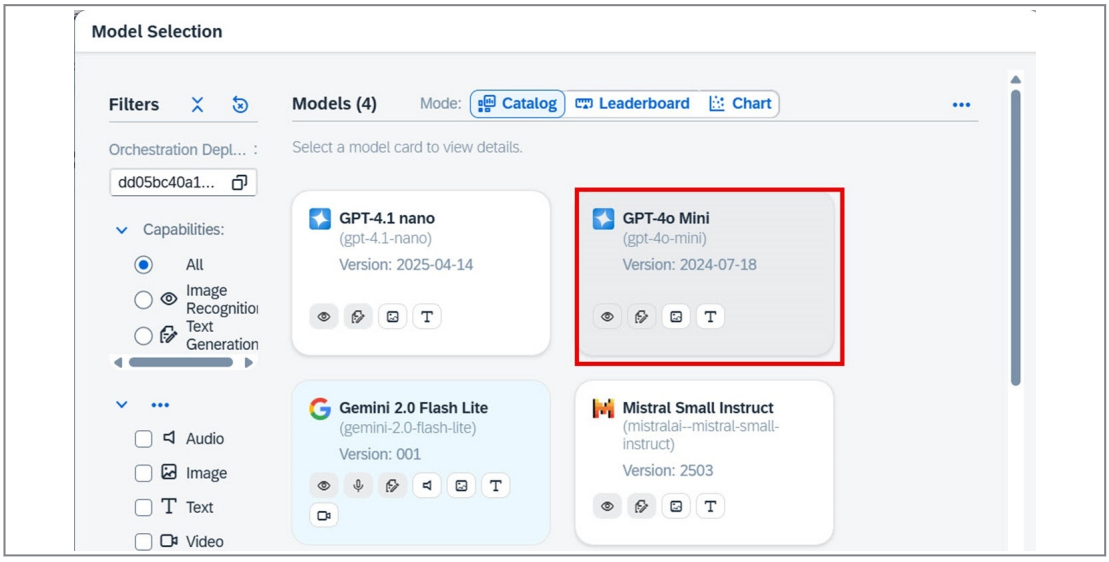
Task 2: Access Different Models using Prompt Editor

We will use latest version of the prompt template created in the previous exercises. This is the latest few-shot prompt version with variables and their default values. We will execute this prompt template with different models.

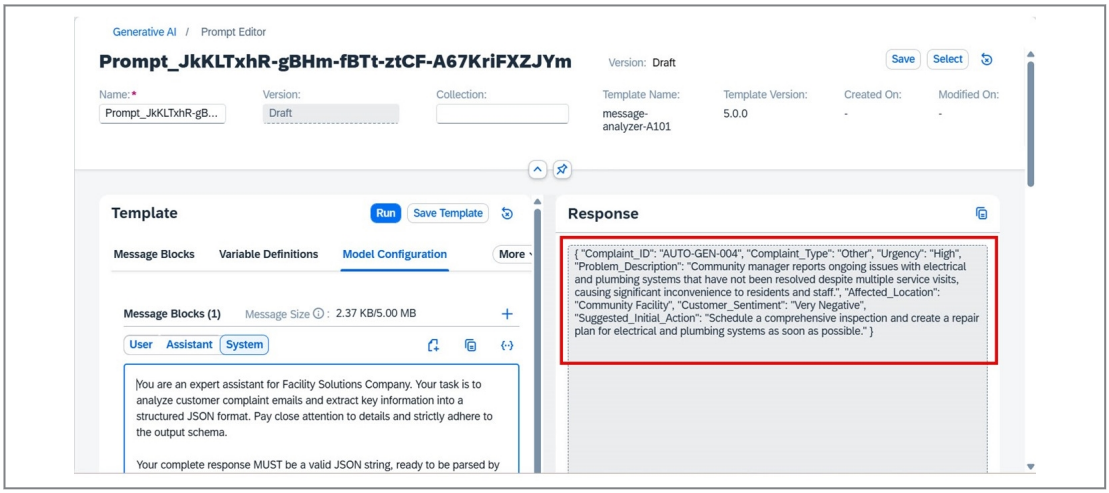
1. Ensure that you are logged on to generative AI hub.
2. Select **Prompt Management and then Templates.** You can see your template here. You can also search for it, if needed.
3. Select the latest version of the template which is 5.0.0.
4. Select the prompt template and then click **Open in Prompt Editor.** Your prompt is ready to use.
5. Scroll to the **Model Configuration tab.**
6. Click **Selected Model.**

7. The Model Selection dialog box is displayed.

8. Select **GPT-4o Mini.**



9. **Run** the prompt. Note the differences in the response.



10. Similarly, select **Mistral AI and GPT4o Nano** models and evaluate results.

You have used different models in generative AI hub.

You have tested various models in the generative AI hub and used a consistent prompt template to evaluate LLMs for cost, performance, scalability, and flexibility. You can evaluate results within a consistent framework and assess both cost and performance for enterprise generative AI tasks.

You need to weigh the cost of using advanced models against the expected return on investment. Refer to SAP notes 3437766 - Availability of Generative AI Models and 3505347 - Orchestration for pricing details in generative AI hub.

Unit 4
Solution 5

Access Models in Generative AI Hub

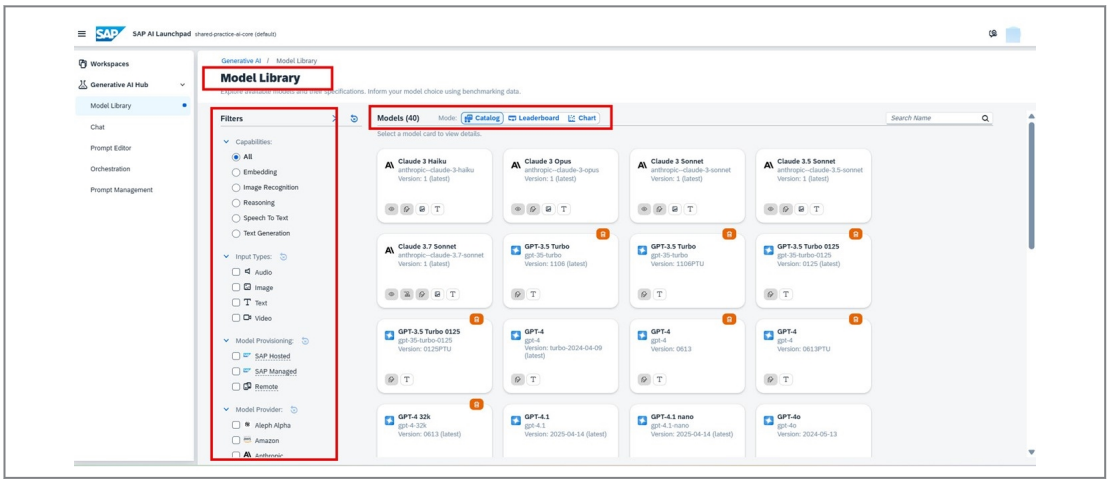
Continuing with the scenario discussed previously, we created prompts and prompt templates that assign urgency, sentiment, and categories to customer messages that can be used in software.

We used the few-shot technique to arrive at a better prompt.

We used prompt template to help scale the solution.

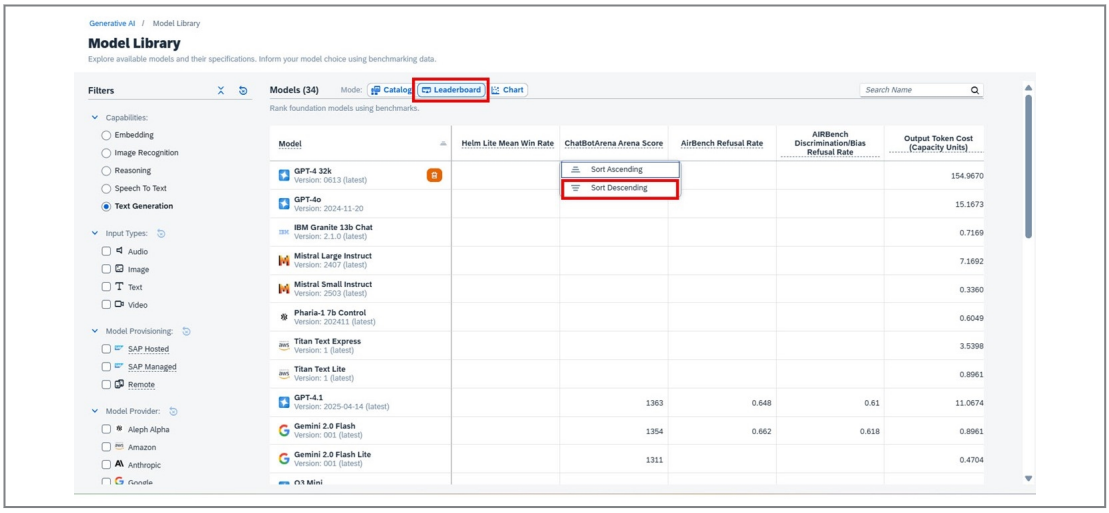
Task 1: Access Different Models using Model Library
We will start with exploring Model Library.

1. Navigate to **Model Library** in the left pane.
2. You will see the Model Library interface.



The model library provides comprehensive information on models available in the generative AI hub to support informed decision-making. To explore the available models and their metadata, utilize the catalog mode. For benchmarking data to guide your decisions, use the leaderboard mode. For detailed information about a specific model, including data input types, cost details, and metrics where available, refer to its model card.

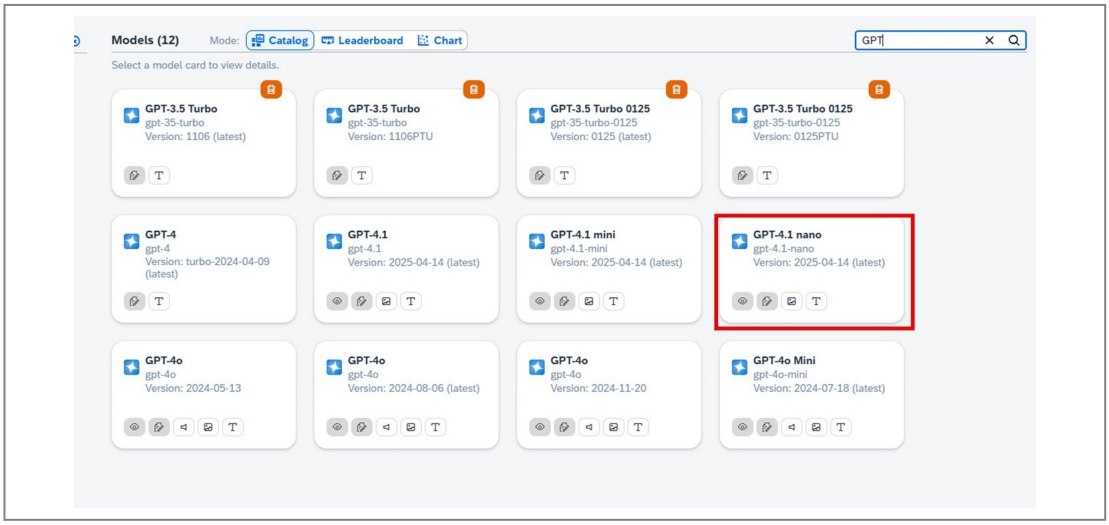
3. You can apply **filters** such as capabilities, Input types, Model provider etc.
4. Select **Leaderboard.**
5. Select any criteria based on your business needs. For example, select ChatBot Arena score. You can hover over any column to know about them.
6. Select the column and click **Sort Descending.**



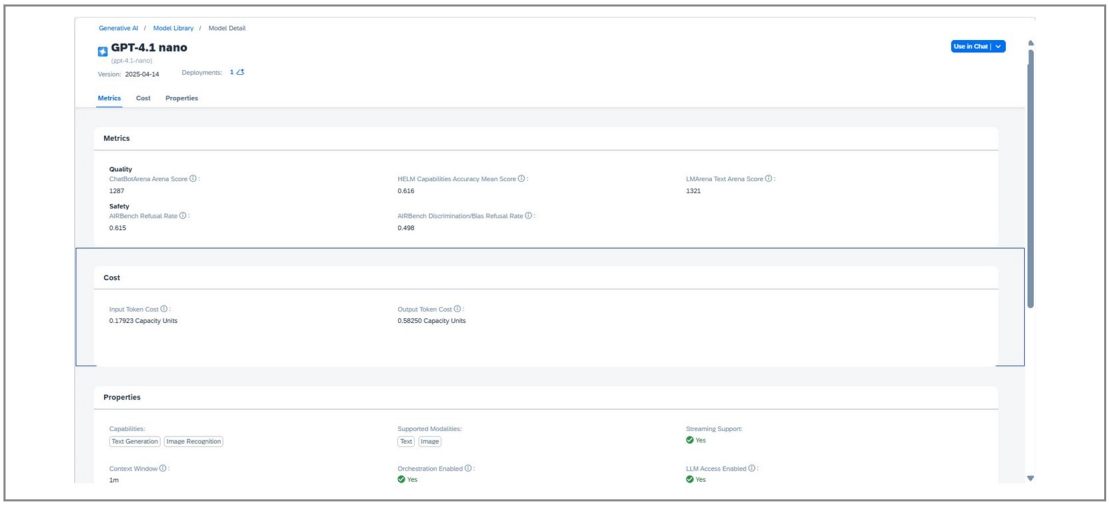
You can see **model ratings.** Similarly, you can compare ratings of different benchmarks in the Chart option.

> Note:
> You can see all the models that are offered in generative AI hub. However, this system is configured to allow few selected models only. These are : GP4.1 nano, GPT4o-mini, and Gemini 2.0 Flash Lite, and Mistral Small Instruct

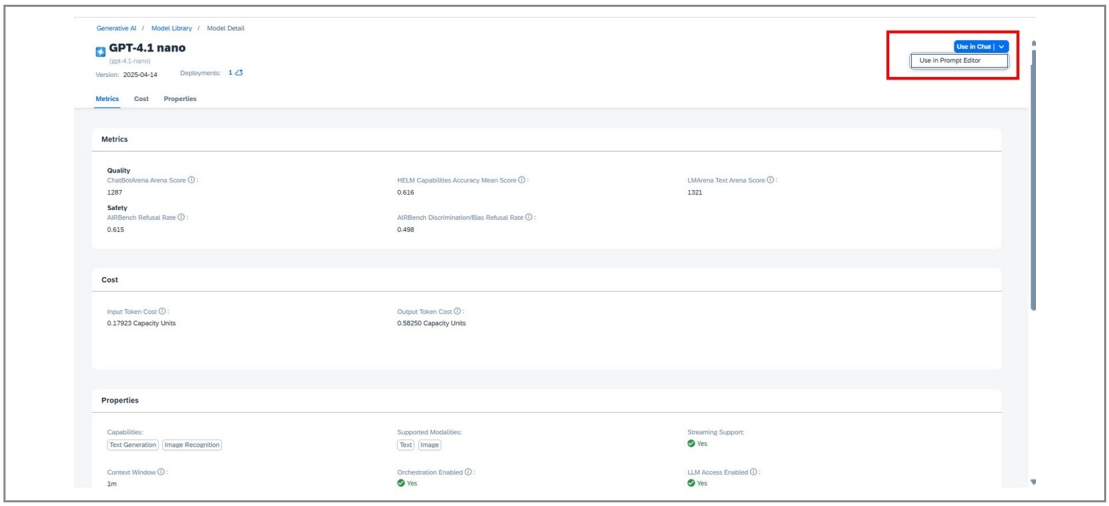
7. Go back to Catalog mode and Search and select GPT 4.1 nano in the Catalog tab.



8. The model card is displayed. These cards provide all the details about the models in terms Metrics, Cost, and Properties.

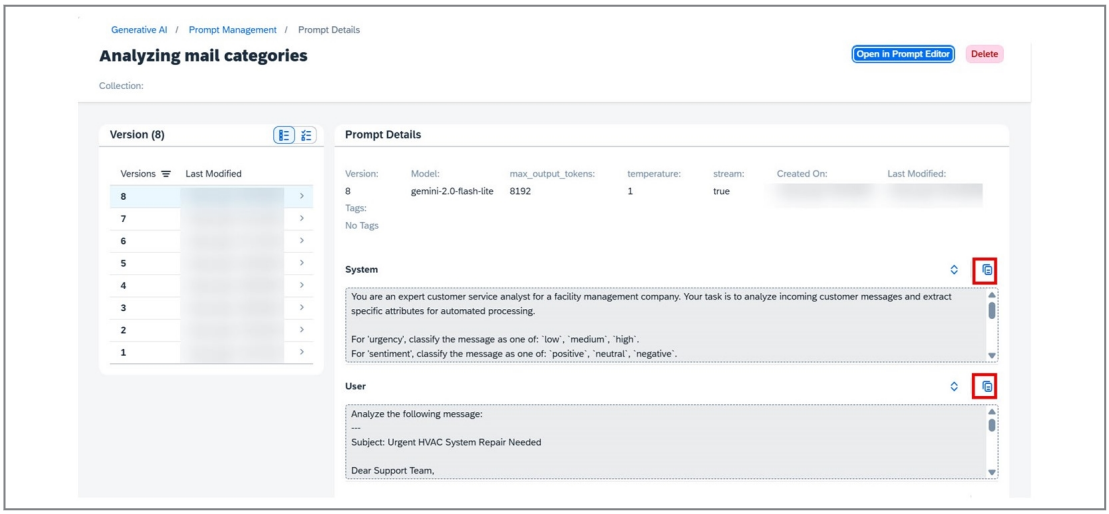


9. You can deploy or use the deployed model directly from Model Library. Select the **Use in Chat or Use in Prompt Editor** options based on your need. Here we will select the **Use in Chat option.**

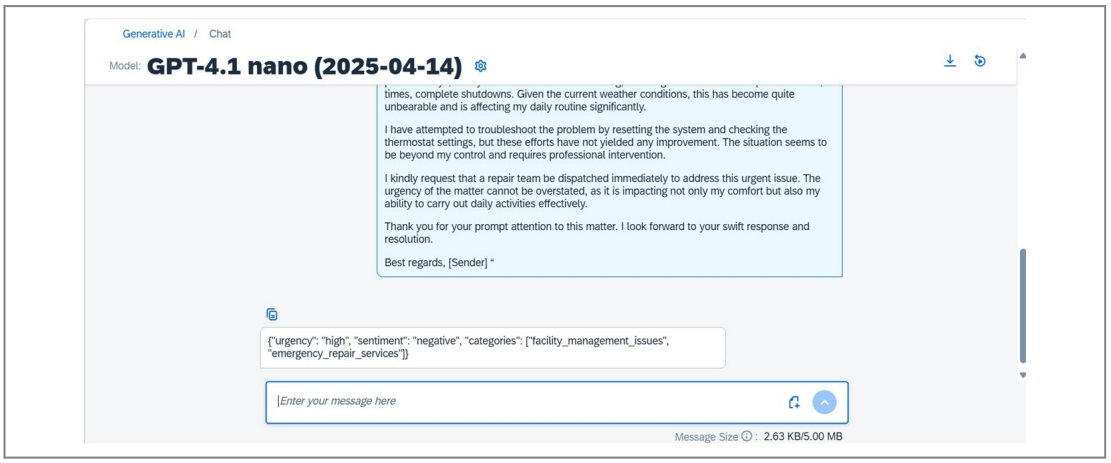


10. Copy and paste your final prompt in the previous exercises and see the results.

11. To copy and paste the prompt, navigate to **Prompt Management, select Prompts,** and then select the **Analyzing mail categories prompt.** Select the latest version and then click **copy.**



You can copy messages for each role in a document and use them one by one, taking advantage of the chat interface.



Similarly, you can see results from other models and select the best model for your use case.

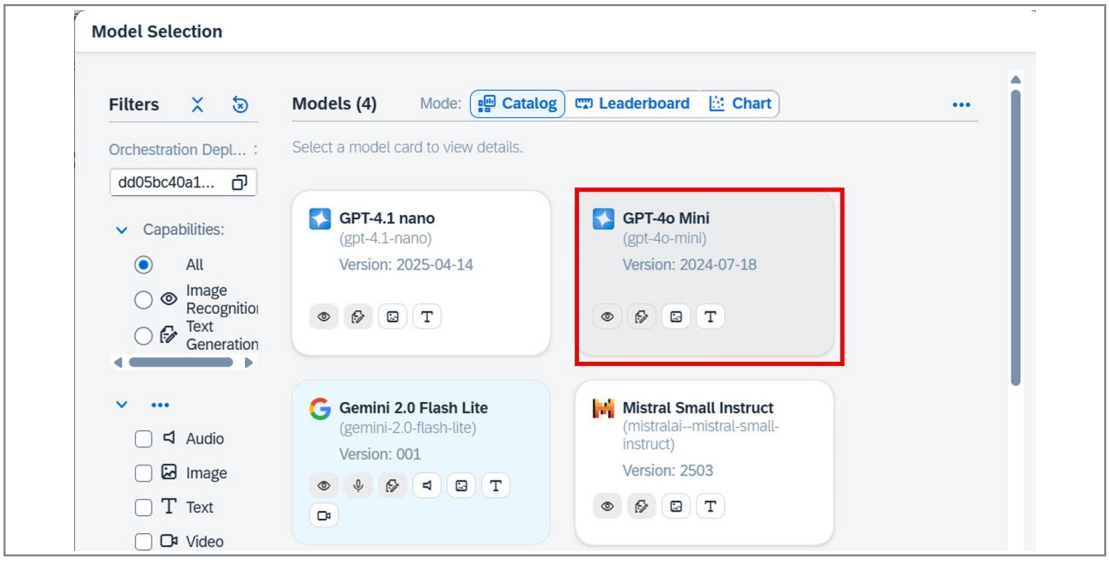
Task 2: Access Different Models using Prompt Editor

We will use latest version of the prompt template created in the previous exercises. This is the latest few-shot prompt version with variables and their default values. We will execute this prompt template with different models.

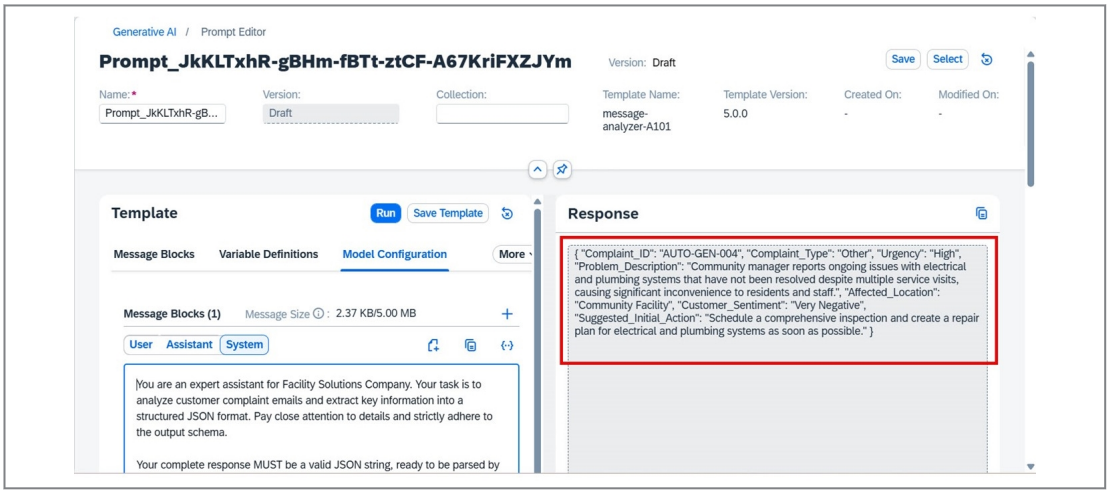
1. Ensure that you are logged on to generative AI hub.
2. Select **Prompt Management and then Templates.** You can see your template here. You can also search for it, if needed.
3. Select the latest version of the template which is 5.0.0.
4. Select the prompt template and then click **Open in Prompt Editor.** Your prompt is ready to use.
5. Scroll to the **Model Configuration tab.**
6. Click **Selected Model.**

7. The Model Selection dialog box is displayed.

8. Select **GPT-4o Mini.**



9. **Run** the prompt. Note the differences in the response.



10. Similarly, select **Mistral AI and GPT4o Nano** models and evaluate results.

You have used different models in generative AI hub.

You have tested various models in the generative AI hub and used a consistent prompt template to evaluate LLMs for cost, performance, scalability, and flexibility. You can evaluate results within a consistent framework and assess both cost and performance for enterprise generative AI tasks.

You need to weigh the cost of using advanced models against the expected return on investment. Refer to SAP notes 3437766 - Availability of Generative AI Models and 3505347 - Orchestration for pricing details in generative AI hub.