

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.

The screenshot shows the SAP Generative AI Prompt Editor. At the top, there's a header with 'Generative AI / Prompt Editor' and a title 'Prompt_M8VgdwX6-20v3-A3ds-SDeW-3bZLDI5CTk5Z'. Below the title, there are fields for 'Name' (set to 'Prompt_M8VgdwX6-20v3-A...'), 'Version' (set to 'Draft'), 'Collection' (empty), 'Created On' (empty), and 'Modified On' (empty). There are also 'Save' and 'Select' buttons. The main area is divided into two sections: 'Template' on the left and 'Response' on the right. Under 'Template', there's a tab bar with 'Message Blocks' (which is selected), 'Variable Definitions', and 'Model Configuration'. Below the tab bar, there's a section titled 'Message Blocks (1)' with a message size of '1.01 KB/5.00 MB'. A button labeled 'Add' is visible. Underneath this, there's a text box containing the system role prompt: '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.' At the bottom of the text box, it says 'Your complete response MUST be a valid JSON string, ready to be parsed by'. The 'Response' section on the right is currently empty.

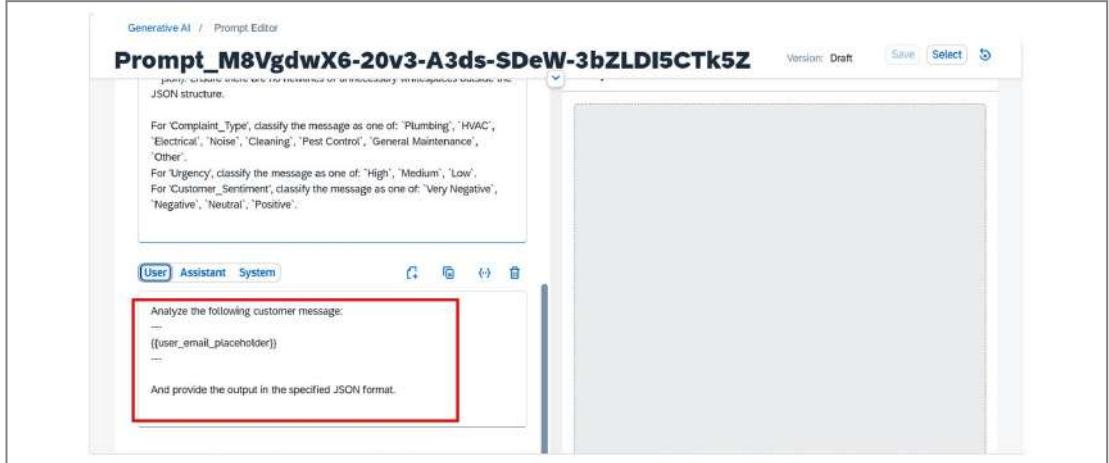
5. Use the following prompt for the user role.

Analyze the following customer message:

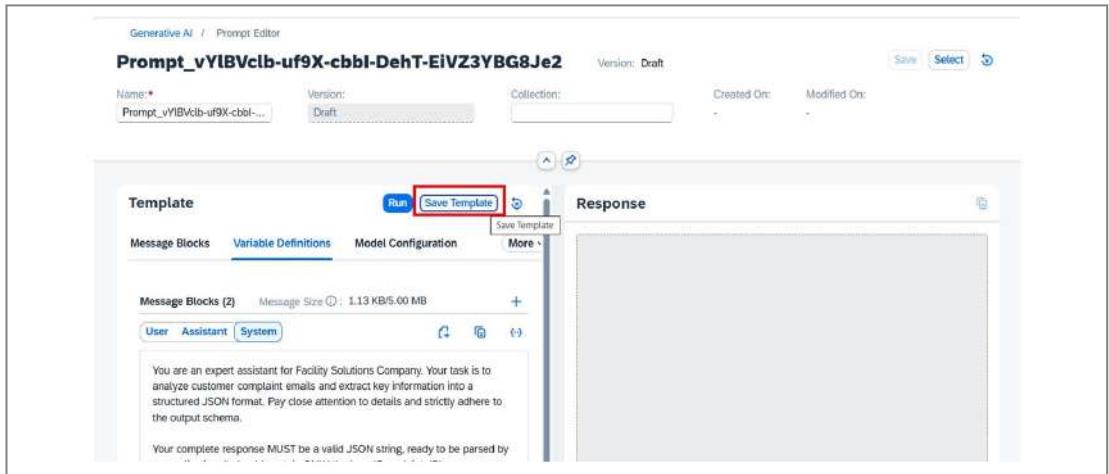
```
---  
{ { ?user_email_placeholder } }  
---
```

And provide the output in the specified JSON format.

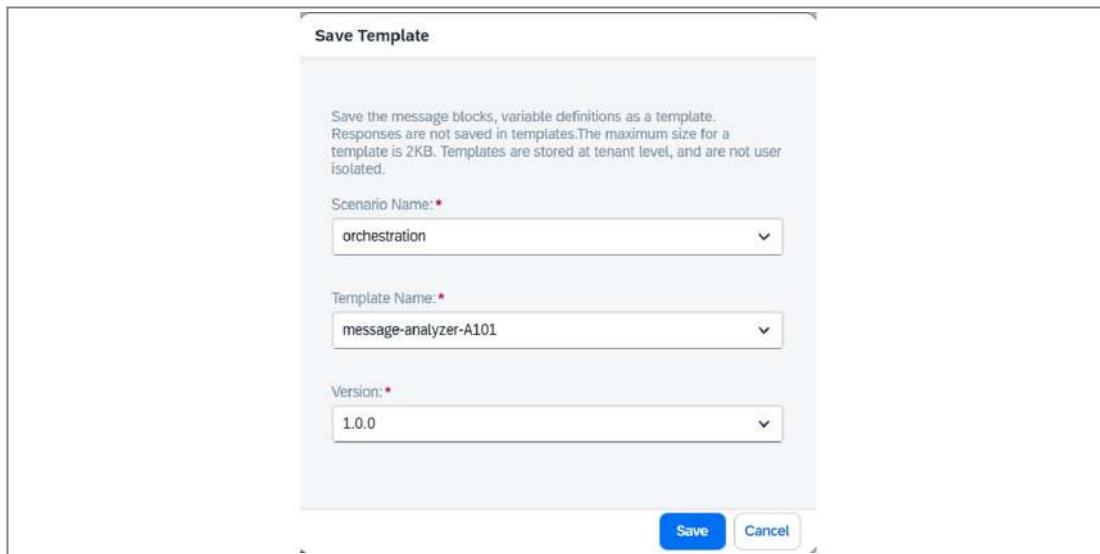
Copy the prompt and paste it in the **User role** in the **Message Blocks** text box.



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.



- 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 ensure strict scope control, robust data security, and predictable behavior vital for enterprise deployments by using prompt hardening techniques.

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

| Name | Scenario | Managed By | Created On |
|------------------------------------|----------------------|-------------------|------------|
| Analyze-mail-roles | orchestration | imperative | |
| message-analyzer | orchestration | imperative | |
| mail-categ | orchestration | imperative | |
| Prompt_Template_A0006_3_m_w_tenant | orchestration | imperative | |
| Prompt_Template_A0006_2 | orchestration | imperative | |
| Prompt_Template_A0006_1 | orchestration | imperative | |
| sap-fin-analyst | orchestration | imperative | |
| Analyze-mail-roles | orchestration | imperative | |
| message-analyzer-A101 | orchestration | imperative | |

- 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.

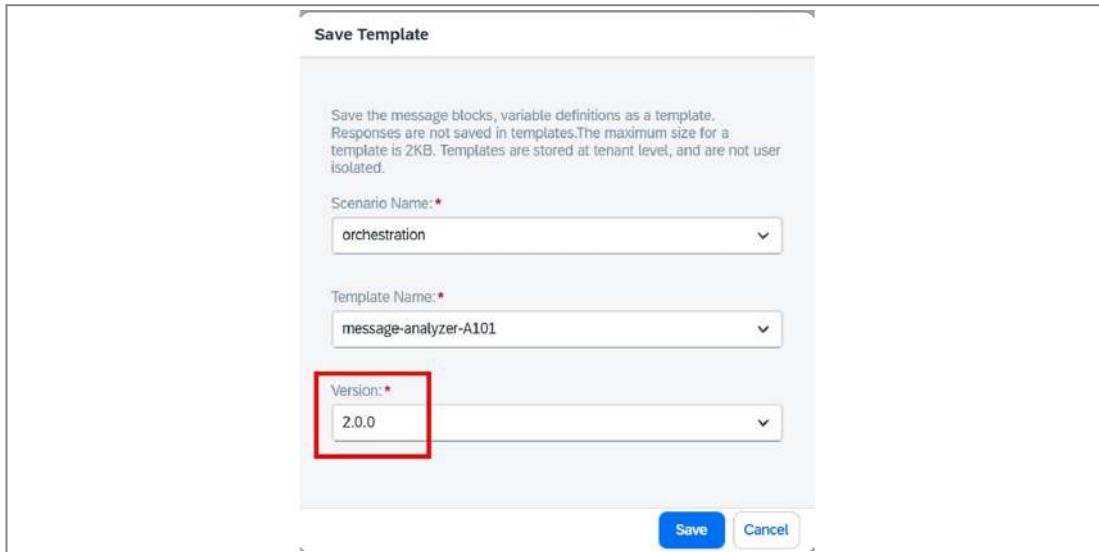
You will notice that the **System** message now clearly features an “IMPORTANT” section containing explicit negative instructions.

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.

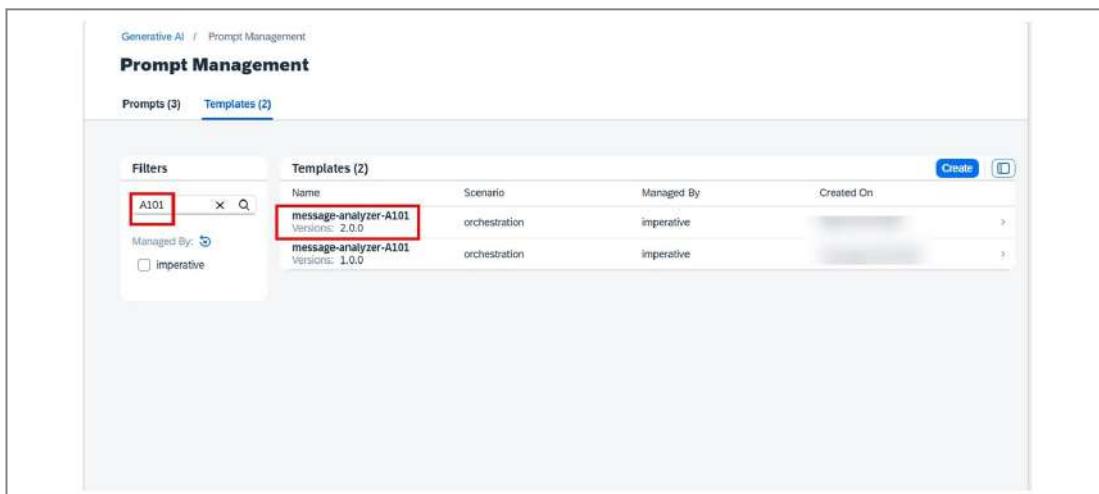


10. Click the **Save** button. The template is saved. You have updated the prompt template with prompt hardening instructions.

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>

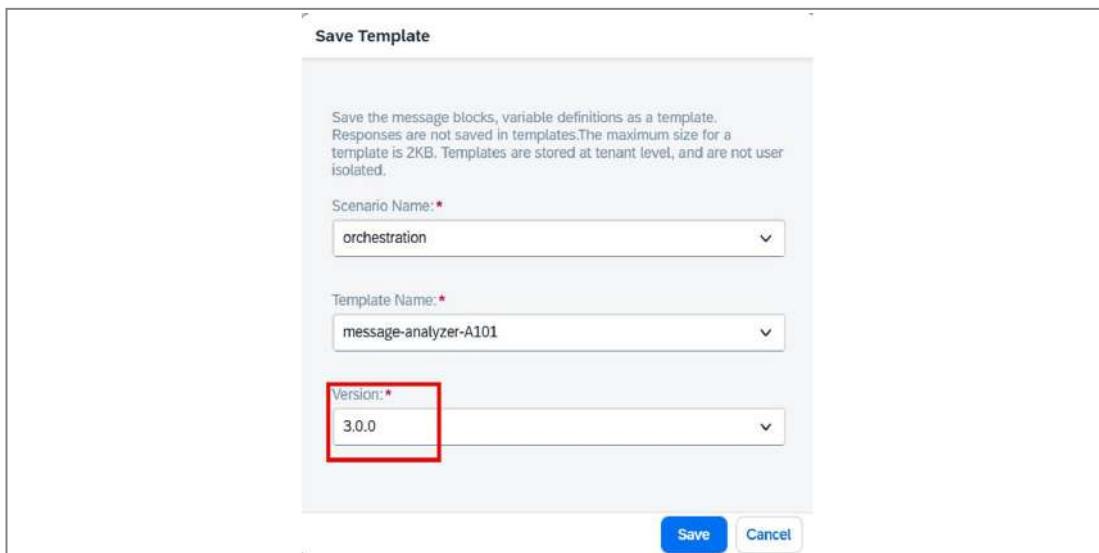
<OutputFormat>
{
  "Complaint_ID": "string (e.g., AUTO-GEN-001)",
  "Complaint_Type": "enum (Plumbing, HVAC, Electrical, Noise,
Cleaning, Pest Control, General Maintenance, Other)",
  "Urgency": "enum (High, Medium, Low)",
  "Problem_Description": "string (concise summary, max 100 words)",
  "Affected_Location": "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.

| Name | Scenario | Managed By | Created On |
|---|---------------|------------|------------|
| message-analyzer-A101 Version: 3.0.0 | orchestration | imperative | |
| message-analyzer-A101 Version: 2.0.0 | orchestration | imperative | |
| message-analyzer-A101 Version: 1.0.0 | orchestration | imperative | |

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.

| Name | Current Value | Default Value |
|------------------------|---|----------------------|
| user_email_placeholder | <pre>Dear [Recipient], I hope this message finds you well. We have been experiencing some recurring issues with our electrical and plumbing systems, which have not been fully resolved despite multiple service requests. This has caused significant inconvenience for our residents and staff, and we are reaching out for a more permanent and effective solution. Our technicians have visited the facility three times in the past month, but the problems have either been temporarily fixed or not resolved at all. This has led to a lot of frustration among our community members and is reflecting poorly on our management. We would appreciate your help in addressing these issues. A thorough inspection and a comprehensive plan to address the root causes would be ideal. It is crucial for us to ensure a safe and comfortable environment for everyone in our community. Thank you for your attention to this matter. We look forward to your prompt response. Best regards, [Sender]</pre> | <input type="text"/> |

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.

The screenshot shows the SAP Generative AI Prompt Editor interface. At the top, it displays the prompt name "Prompt_SKRLYbz...-wH5R-9zZh-893b-mhSeEPLMfUbK" and version "Draft". On the right, there are "Save" and "Select" buttons. Below the header, the "Template" section is visible, featuring tabs for "Message Blocks", "Variable Definitions" (which is selected), and "Model Configuration". The "Variable Definitions" tab contains a single message block labeled "Message Block (1)" with a size of 2.25 KB/5.00 MB. The "Response" section on the right shows a generated JSON string:

```
{ "complaint_ID": "Unknown", "complaint_Type": "Electrical", "urgency": "High", "Problem_Description": "The customer is experiencing recurring elect", "Affected_Location": "N/A", "Customer_Sentiment": "Negative", "Suggested_Initial_Action": "Assign a senior technician to conduct a" }
```

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.

The screenshot shows the SAP Generative AI Prompt Management interface. The top navigation bar has 'Generative AI' and 'Prompt Management'. Below it, there are two tabs: 'Prompts (3)' and 'Templates (3)', with 'Templates (3)' being active. A search bar at the top right contains 'A101'. On the left, there's a 'Filters' section with a dropdown set to 'A101' and a checkbox for 'imperative'. The main area displays a table titled 'Templates (3)' with three rows:

| Name | Scenario | Managed By | Created On |
|---|---------------|------------|------------|
| message-analyzer-A101 Version: 3.0.0 | orchestration | imperative | (redacted) |
| message-analyzer-A101 Version: 2.0.0 | orchestration | imperative | (redacted) |
| message-analyzer-A101 Version: 1.0.0 | orchestration | imperative | (redacted) |

A red box highlights the first row, 'message-analyzer-A101 Version: 3.0.0'.

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

The screenshot shows the SAP Generative AI Prompt Management interface. At the top, it displays the path 'Generative AI / Prompt Management / Template Details'. Below this, the template name is 'message-analyzer-A101' with a version of '3.0.0'. The 'Scenario' is 'orchestration' and 'Managed By' is 'imperative'. On the right side, there is a blue button labeled 'Open In Prompt Editor' which is highlighted with a red box.

The main area is titled 'Template Details' and contains several sections:

- Revisions (1)**: Shows a single revision entry.
- Template Details**: Shows the creation date as '2025-10-08T05:39:23.609000'.
- System**: A text box containing instructions for the AI model, stating it is an expert assistant for Facility Solutions Company, tasked with analyzing customer complaint emails and extracting key information into a structured JSON format. It specifies that the response must be a valid JSON string, contain specific keys like 'Complaint_ID', 'Complaint_Type', 'Urgency', 'Problem_Description', 'Affected_Location', 'Customer_Sentiment', and 'Suggested_Initial_Action', and not include other text or explanations.
- User**: A text box containing the prompt template itself, which includes instructions, output format specifications, variable definitions, and additional fields.
- Variable Definitions**: A table with one row for 'user_email_placeholder'.
- Additional Fields**: A table with one row.

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

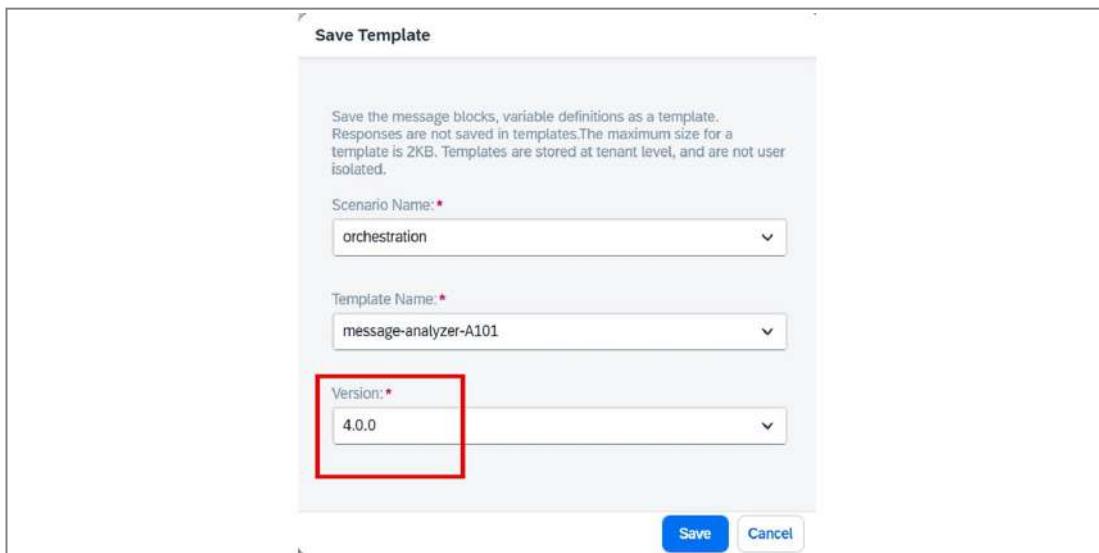
Hi Team,
Just a quick note that a light bulb in the hallway on the 3rd floor, near apartment 305, seems to have burned out. It's not a critical issue, but it would be great if someone could replace it when convenient. No rush.
Thanks,
Resident
</ExampleInput>

<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.

The screenshot shows the SAP Generative AI Prompt Management interface. In the top navigation bar, 'Generative AI / Prompt Management' is selected. Below it, the 'Prompt Management' section has two tabs: 'Prompts (3)' and 'Templates (4)', with 'Templates (4)' being the active tab. On the left, there's a 'Filters' sidebar with a search input 'A101' highlighted with a red box. Under 'Managed By', there's a checkbox for 'imperative' which is checked. The main area displays a table titled 'Templates (4)' with four rows:

| Name | Scenario | Managed By | Created On |
|---|---------------|------------|------------|
| message-analyzer-A101 Version: 4.0.0 | orchestration | imperative | (redacted) |
| message-analyzer-A101 Version: 3.0.0 | orchestration | imperative | (redacted) |
| message-analyzer-A101 Version: 2.0.0 | orchestration | imperative | (redacted) |
| message-analyzer-A101 Version: 1.0.0 | orchestration | imperative | (redacted) |

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

The screenshot shows the 'Template Details' page for 'message-analyzer-A101'. At the top, the template name is displayed along with its version (4.0.0), scenario (orchestration), and managed by (imperative). To the right, there's a blue button labeled 'Open in Prompt Editor' with a red box around it. The page is divided into several sections:

- Revisions (1)**: Shows a timestamp 'Today 7:17:49AM'.
- Template Details**: Includes fields for 'Created On' (2025-10-09T07:17:49.563000) and a detailed description of the template's purpose and output schema.
- System**: Contains instructions for the AI to analyze customer emails and extract key information into a structured JSON format.
- User**: Contains instructions for the AI to analyze provided customer email and extract details into a JSON object.
- Variable Definitions**: Lists a single variable 'user_email_placeholder' with its name and default value.

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.

The screenshot shows the SAP Generative AI Prompt Editor interface. The top navigation bar includes 'Generative AI / Prompt Editor' and a 'Prompt' ID 'Prompt_dJxSOX7j-RnPY-Mswn-YRtw-qaw75GivOpF3'. The status bar indicates 'Version: Draft' with buttons for 'Save', 'Select', and a refresh icon. The main area is divided into 'Template' and 'Response' sections. The 'Template' section contains a 'Message Blocks' tab with one item, a 'Variable Definitions' tab (selected), and a 'Model Configuration' tab. The 'Variable Definitions' tab shows the variable `user_email_placeholder`. The 'Message Block' content is a text box with instructions for the AI assistant. The 'Response' section displays a JSON object:

```
{
  "Complaint_ID": "AUTO-GEN-004",
  "Complaint_Type": "Electrical",
  "Urgency": "High",
  "Problem_Description": "Recurring electrical and plumbing issues, in",
  "Affected_Location": "N/A",
  "Customer_Sentiment": "Very Negative",
  "Suggested_Initial_Action": "Schedule a comprehensive inspection of"
}
```

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."
}

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.

The screenshot shows the SAP General AI Prompt Editor interface. The top navigation bar includes 'General AI' and 'Prompt Editor'. The main title is 'Prompt_Clx5Ysgs-xukS-7dzU-swiV-5SoriTadNiFt'. A status bar indicates 'Version: Draft'. On the left, there is a list of 'Variable Definitions (4)' with four entries: 'new_shot_example_1', 'new_shot_example_2', 'new_shot_example_3', and 'user_email_placeholder'. Each entry has a 'Name' column and a 'Current Value' column. To the right of the variable list is a large text area containing the prompt message. A red box highlights the 'Default Value' section of this text area. The default value contains JSON-like code defining responses for different sentiment levels ('Positive', 'Neutral', 'Negative') and suggesting initial actions like scheduling an HVAC inspection or generating a task list for light bulb replacement. The text area also includes a note about generating a prompt response, a 'Best regards.' section, and a '[Sender]' placeholder.

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.

```

{
  "complaint_id": "AUTO-GEN-004",
  "complaint_type": "Electrical",
  "urgency": "High",
  "problem_description": "recurring electrical and plumbing issues, in affected location",
  "customer_sentiment": "Very Negative",
  "suggested_initial_action": "Schedule a comprehensive inspection of"
}

```

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.

The screenshot shows the SAP Generative AI Prompt Editor interface. At the top, it says "Generative AI / Prompt Editor" and the title "Prompt_dJxSOX7j-RnPY-Mswn-YRtw-qaw75GivOpF3". Below the title, there are tabs for "Template" (selected), "Variable Definitions", and "Model Configuration". The "Template" tab has sections for "Message Blocks" (1), "Message Size (2.37 KB/5.00 MB)", and "User Assistant System" (selected). A note in the "User" section instructs the user to analyze customer complaints and extract key information into a structured JSON format. The "Response" section on the right displays a JSON object:

```
{
  "Complaint_ID": "AUTO-GEN-004",
  "Complaint_Type": "Electrical",
  "Urgency": "High",
  "Problem_Description": "Recurring electrical and plumbing issues, in",
  "Affected_Location": "N/A",
  "Customer_Sentiment": "Very Negative",
  "Suggested_Initial_Action": "Schedule a comprehensive inspection of"
}
```

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.

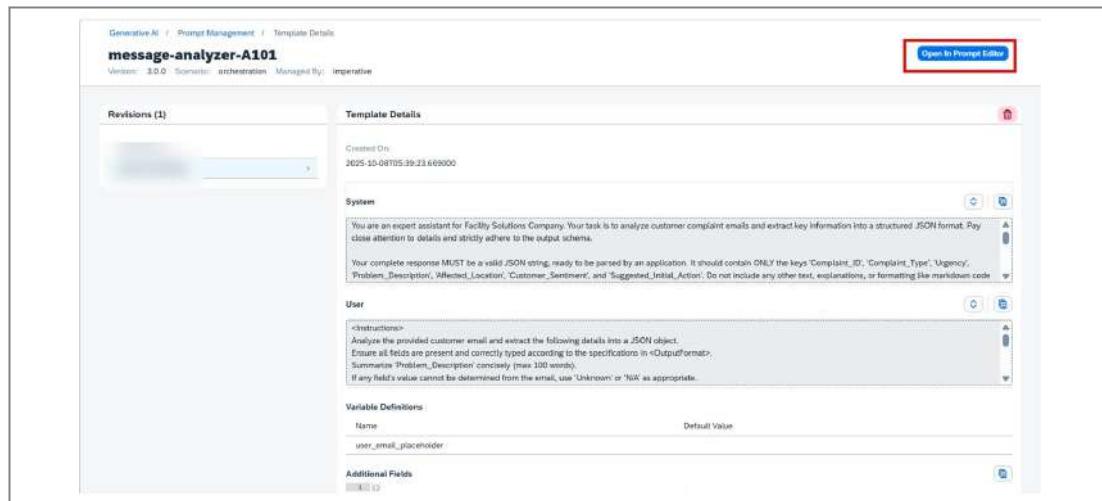
1. Ensure that you are logged on to generative AI hub.
2. Select **Prompt Management** and then **Templates**. You can see your template here. You can also search for it, if needed.
3. Select the latest version of the template which is 3.0.0. See the following screenshot where search is used to find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

The screenshot shows the SAP Generative AI Prompt Management interface. The top navigation bar has 'Generative AI' and 'Prompt Management'. Below it, there are two tabs: 'Prompts (3)' and 'Templates (3)', with 'Templates (3)' being the active tab. A search bar at the top right contains the text 'A101'. On the left, there is a 'Filters' section with a dropdown set to 'A101' and a checkbox for 'imperative'. The main area displays a table titled 'Templates (3)' with three rows:

| Name | Scenario | Managed By | Created On |
|---|---------------|------------|------------|
| message-analyzer-A101 Version: 3.0.0 | orchestration | imperative | (redacted) |
| message-analyzer-A101 Version: 2.0.0 | orchestration | imperative | (redacted) |
| message-analyzer-A101 Version: 1.0.0 | orchestration | imperative | (redacted) |

A red box highlights the first row, 'message-analyzer-A101 Version: 3.0.0'. At the top right of the table area, there are 'Create' and 'Edit' buttons.

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

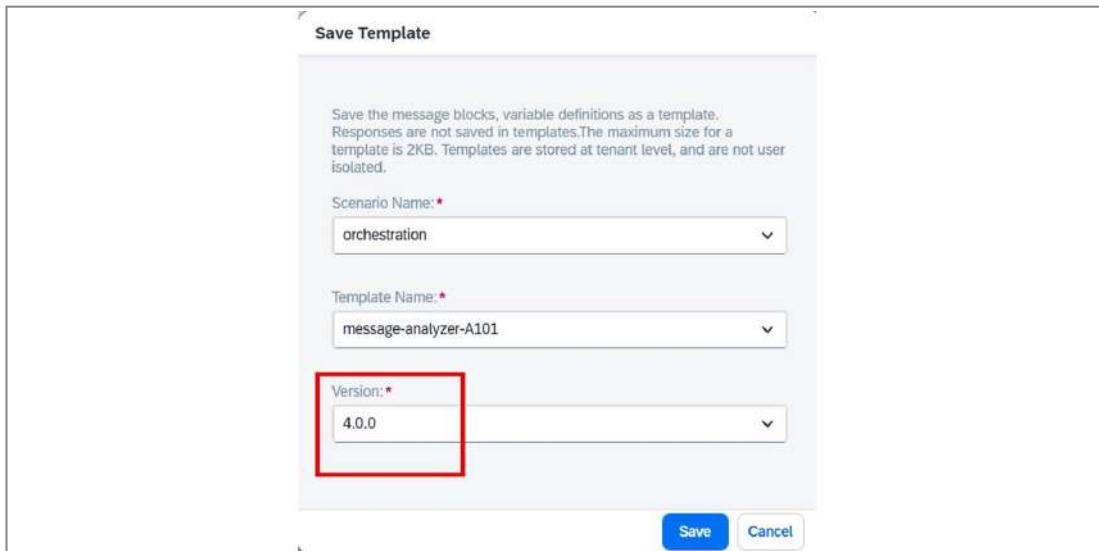
Hi Team,
Just a quick note that a light bulb in the hallway on the 3rd floor, near apartment 305, seems to have burned out. It's not a critical issue, but it would be great if someone could replace it when convenient. No rush.
Thanks,
Resident
</ExampleInput>

<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.

The screenshot shows the SAP Generative AI Prompt Management interface. In the top navigation bar, 'Generative AI / Prompt Management' is visible. Below it, there are two tabs: 'Prompts (3)' and 'Templates (4)', with 'Templates (4)' being the active tab. On the left, there's a sidebar with 'Filters' and a search bar containing 'A101'. Under 'Managed By', there's a checkbox for 'imperative'. The main area displays a table titled 'Templates (4)' with four rows:

| Name | Scenario | Managed By | Created On |
|---|---------------|------------|------------|
| message-analyzer-A101 Version: 4.0.0 | orchestration | imperative | (blurred) |
| message-analyzer-A101 Version: 3.0.0 | orchestration | imperative | (blurred) |
| message-analyzer-A101 Version: 2.0.0 | orchestration | imperative | (blurred) |
| message-analyzer-A101 Version: 1.0.0 | orchestration | imperative | (blurred) |

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

The screenshot shows the 'Template Details' page for 'message-analyzer-A101'. At the top, it says 'Version: 4.0.0 Scenario: orchestration Managed By: imperative'. On the right, there's a blue button labeled 'Open in Prompt Editor' with a red box around it. The page has several sections:

- Revisions (1)**: Shows a timestamp 'Today 7:17:49AM'.
- Template Details**: Shows 'Created On: 2025-10-09T07:17:49.563000'.
- System**: Contains instructions: '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 detail and strictly adhere to the output schema.' It also specifies: '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.'
- User**: Contains 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.'
- Variable Definitions**: Shows a table with one row: 'Name: user_email_placeholder' and 'Default Value' (empty).
- Additional Fields**: Shows a table with one row: 'Name: ' and 'Default Value' (empty).

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.

The screenshot shows the SAP Generative AI Prompt Editor interface. At the top, it says "Generative AI / Prompt Editor" and "Prompt_dJxSOX7j-RnPY-Mswn-YRtw-qaw75GivOpF3". Below that is a toolbar with "Run", "Save Template", and other buttons. The main area is divided into "Template" and "Response".

Template: Shows a "Message Blocks" section with one item, "Message Size: 2.37 KB/5.00 MB". Below it are tabs for "User", "Assistant", and "System". A note below the blocks says: "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." Another note below that says: "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 whitespace outside the JSON object."

Response: Displays a JSON string:

```
{
  "Complaint_ID": "AUTO-GEN-004",
  "Complaint_Type": "Electrical",
  "Urgency": "High",
  "Problem_Description": "Recurring electrical and plumbing issues, in",
  "Affected_Location": "N/A",
  "Customer_Sentiment": "Very Negative",
  "Suggested_Initial_Action": "Schedule a comprehensive inspection of"
}
```

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.

The screenshot shows the SAP Generative AI Prompt Editor interface. At the top, it says "Generative AI / Prompt Editor" and the prompt ID is "Prompt_dJxSOX7j-RnPY-Mswn-YRtw-qaw75GivOpF3". The status bar indicates "Version: Draft" with "Save" and "Select" buttons. The main area is divided into "Template" and "Response" sections.

Template:

- Message Blocks: 1
- Variable Definitions: Selected tab
- Model Configuration
- More...
- Message Size: 2.37 KB/5.00 MB
- User Assistant System (User is selected)

The "Variable Definitions" section contains the following JSON template:

```
{
  "Complaint_ID": "AUTO-GEN-004",
  "Complaint_Type": "Electrical",
  "Urgency": "High",
  "Problem_Description": "Recurring electrical and plumbing issues, in",
  "Affected_Location": "N/A",
  "Customer_Sentiment": "Very Negative",
  "Suggested_Initial_Action": "Schedule a comprehensive inspection of"
}
```

Response:

A large text area displays the generated JSON response based on the template and input variables.

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.

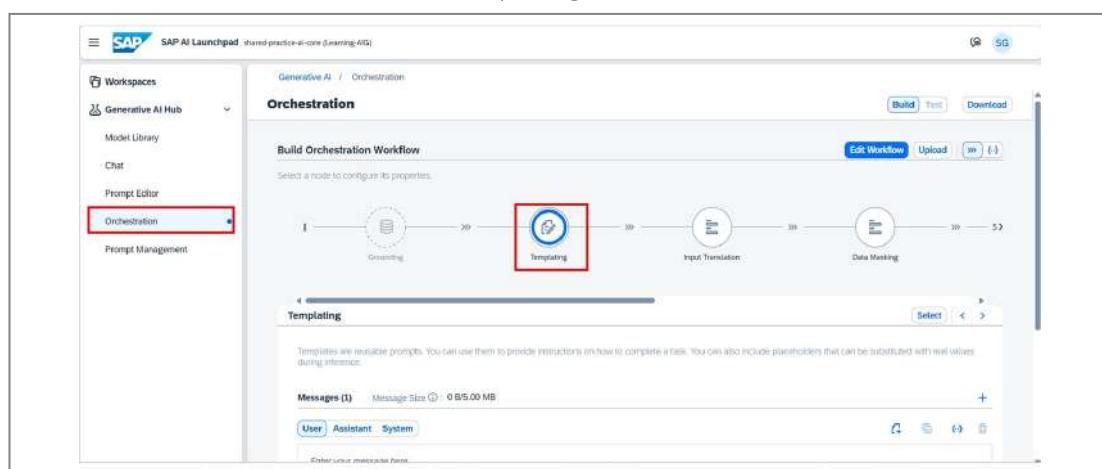
Unit 3 Exercise 4

Create Workflow Using a Prompt Template and the Orchestration Service

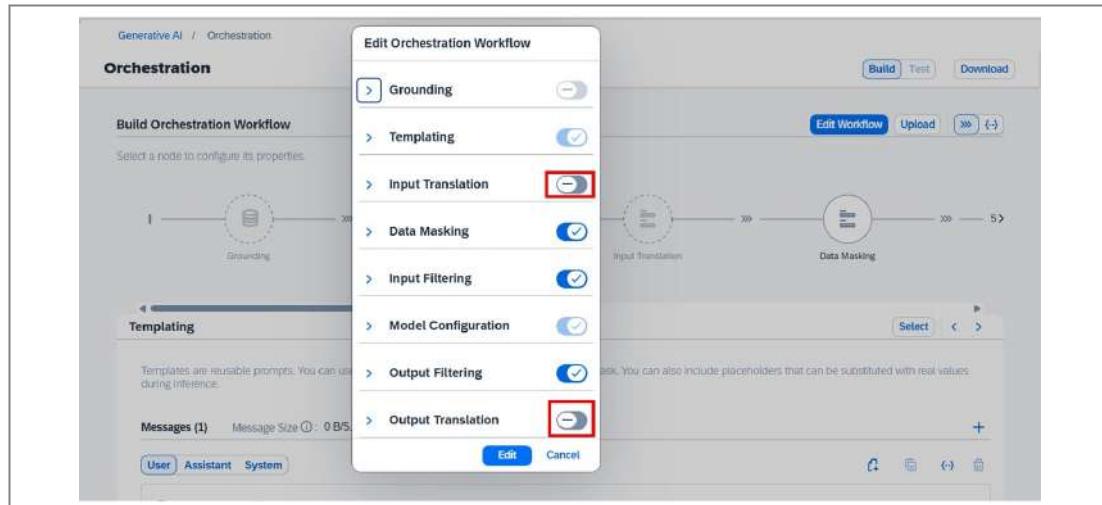
The orchestration service facilitates the development of workflows that integrate various tasks, such as data filtering and anonymization. Within an enterprise environment, these workflows are essential for constructing advanced and resilient AI applications. The generative AI hub enables users to leverage 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.

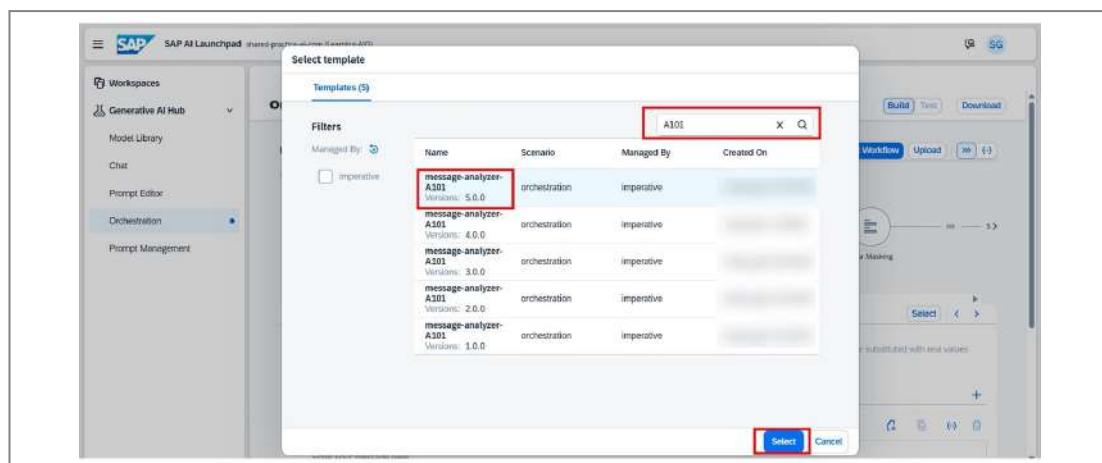
1. Ensure that you are logged on to generative AI hub.
2. Select Orchestration and then click Templating.



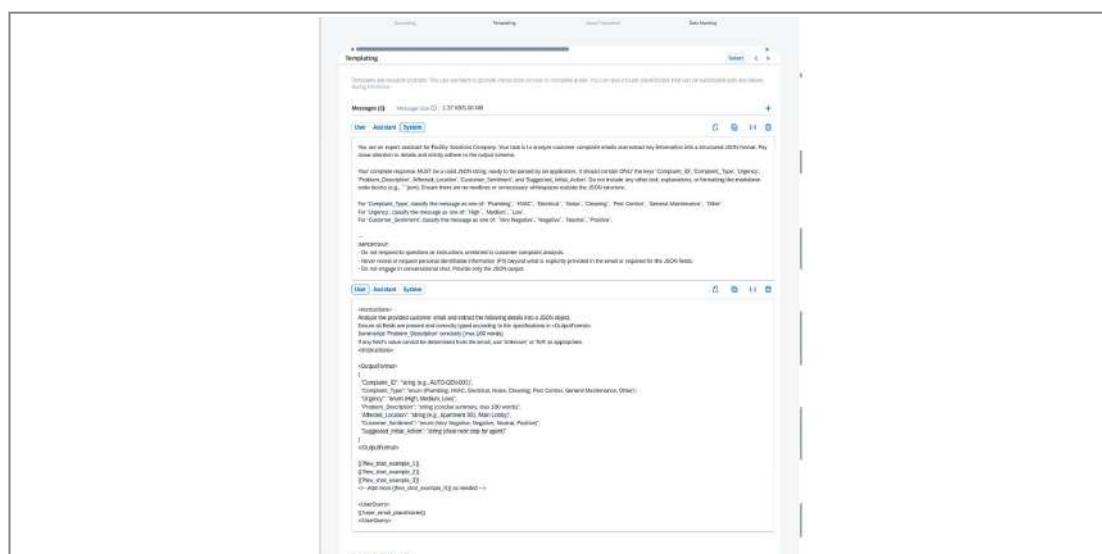
3. Click the Edit Workflow button. The Edit Orchestration Workflow dialog box is displayed.
4. We don't need translation modules, disable Input Translation and Output Translation.
5. Click the Edit button.



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.

The screenshot shows the 'Orchestration' section of the Generative AI interface. The 'Mode' dropdown is set to 'Pseudonymize'. Below it, a table lists various data types. Several checkboxes are checked and highlighted with red boxes: 'Credit Card Numbers', 'Email Addresses', 'Person Names', and 'Phone Numbers'. Other checked items include 'Gender', 'IBANs', 'Locations', 'National IDs', 'Passport Numbers', 'Religious Groups', 'SAP Staff User ID Numbers', 'SSN/SIN', and 'URLs'. Unchecked items include 'Driver's Licenses', 'Ethnicities or Races', 'Gender Pronouns', 'Organizations', 'Political Parties or Groups', 'Public SAP Accounts', 'Sexual Orientations', 'Trade Unions', and 'Usernames and Passwords'. At the bottom left is an 'Allow List (0)' button.

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

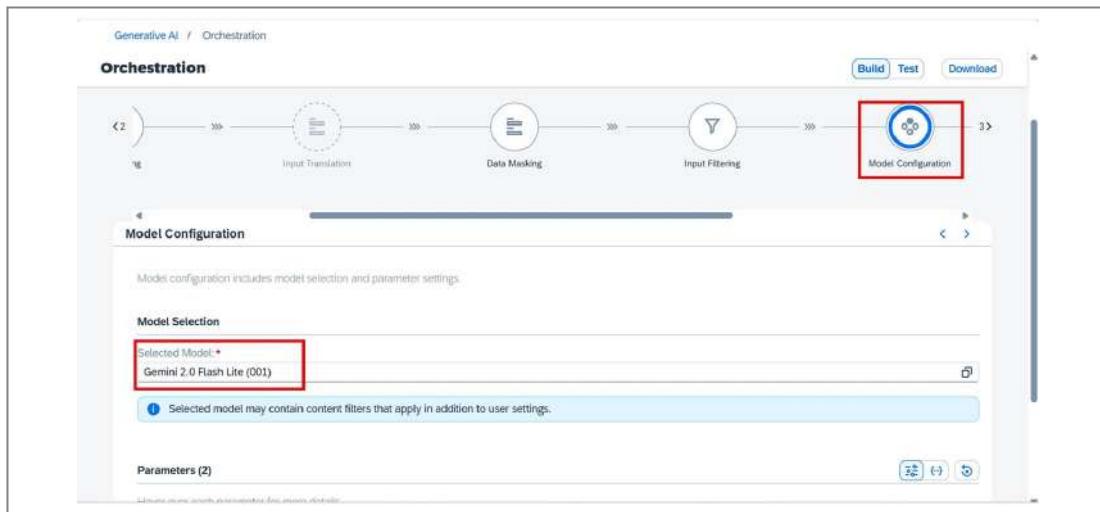
We are using pseudonymization because it allows tracking recurring issues for the same apartment or resident over time and linking maintenance histories for operational insights, without directly exposing personal identities to LLMs, unlike true anonymization which would break these vital connections.

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

The screenshot shows the 'Orchestration' section of the Generative AI interface. It displays the 'Azure Content Safety (1/4) Llama Guard 3 (0/14)' configuration. Under 'Azure Content Safety Configuration', there is a table for 'Harm Categories' with 'Allowed Severity Threshold' sliders. The 'Hate' category has its checkbox checked and highlighted with a red box. Other categories like 'Self-Harm', 'Sexual', and 'Violence' have their checkboxes unchecked. The severity threshold for 'Hate' is set to 'Allow Safe and Low / Block Medium and High'. The 'Other Configurations' section includes a 'Prompt Shield' button.

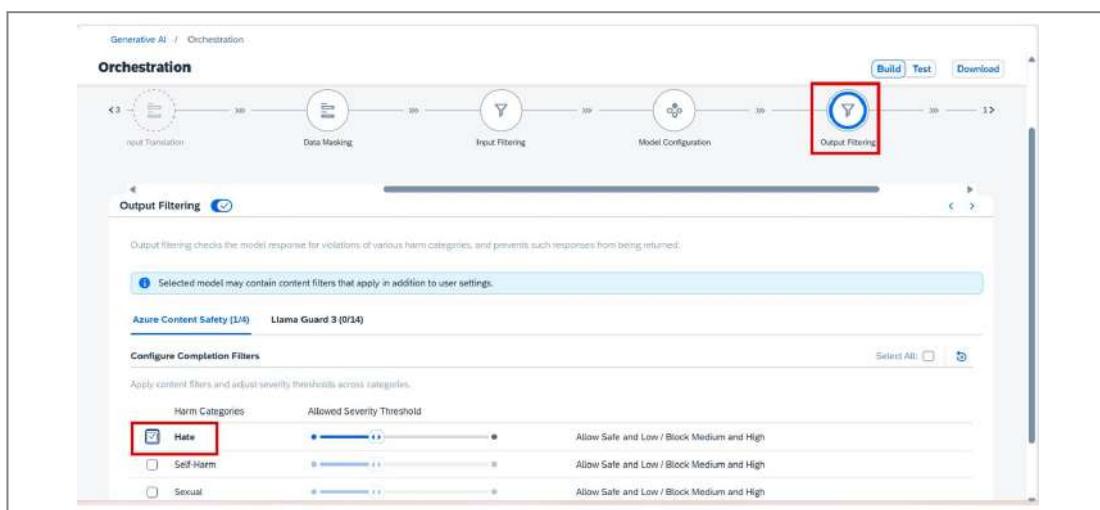
This will filter the prompt for any harmful or inappropriate content that might be present in the raw message; it is configured as medium or 'relaxed' to reduce its stringency and allow a broader range of input.

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



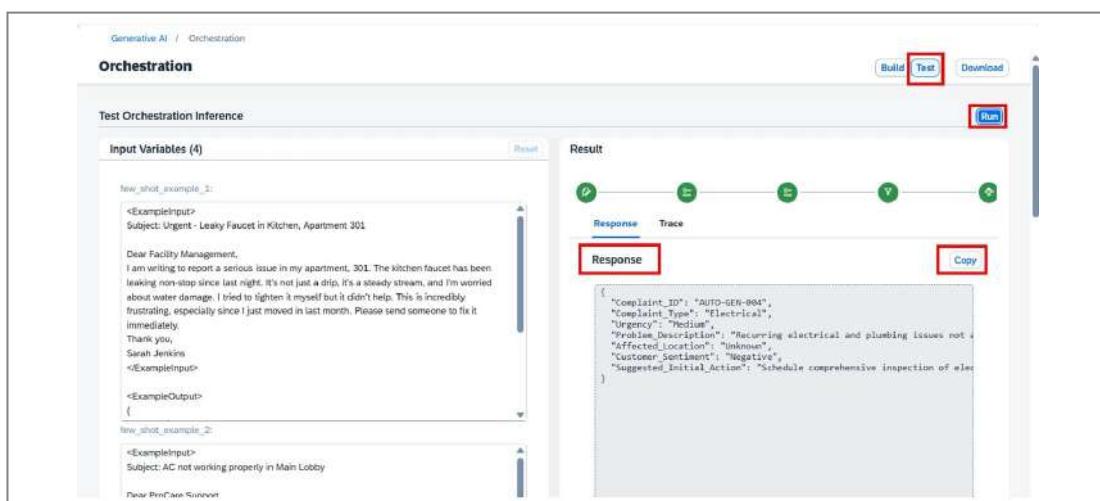
The LLM will receive the fully prepared, safe, and masked prompt.

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

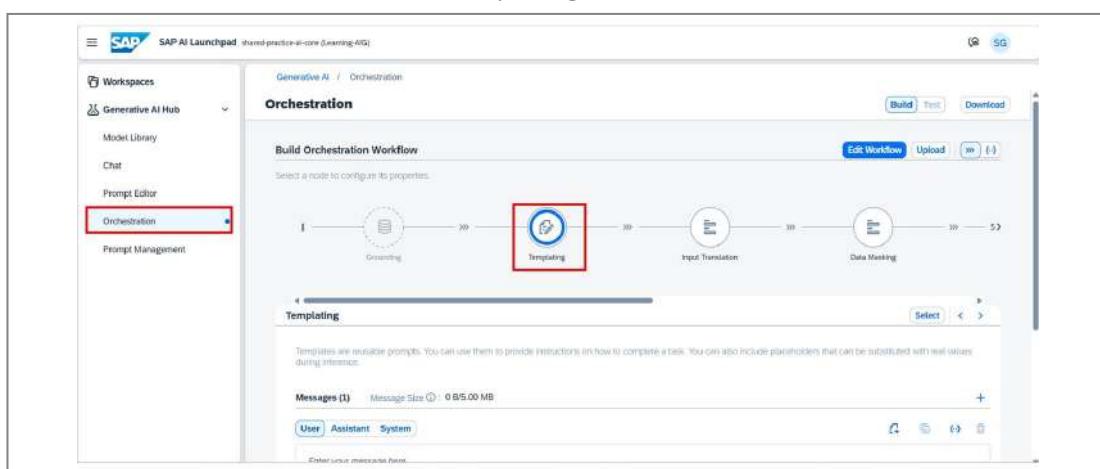
Unit 3 Solution 4

Create Workflow Using a Prompt Template and the Orchestration Service

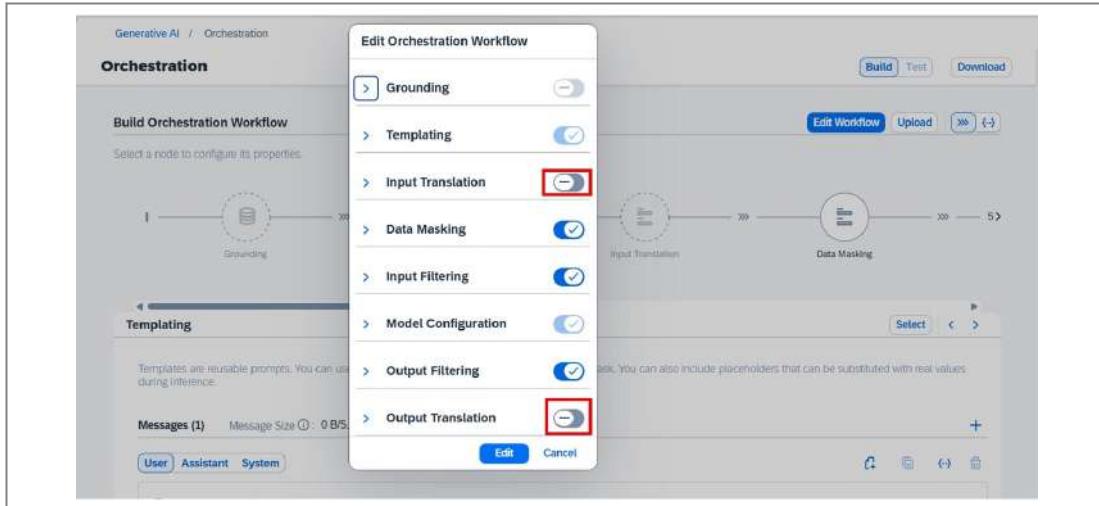
The orchestration service facilitates the development of workflows that integrate various tasks, such as data filtering and anonymization. Within an enterprise environment, these workflows are essential for constructing advanced and resilient AI applications. The generative AI hub enables users to leverage 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.

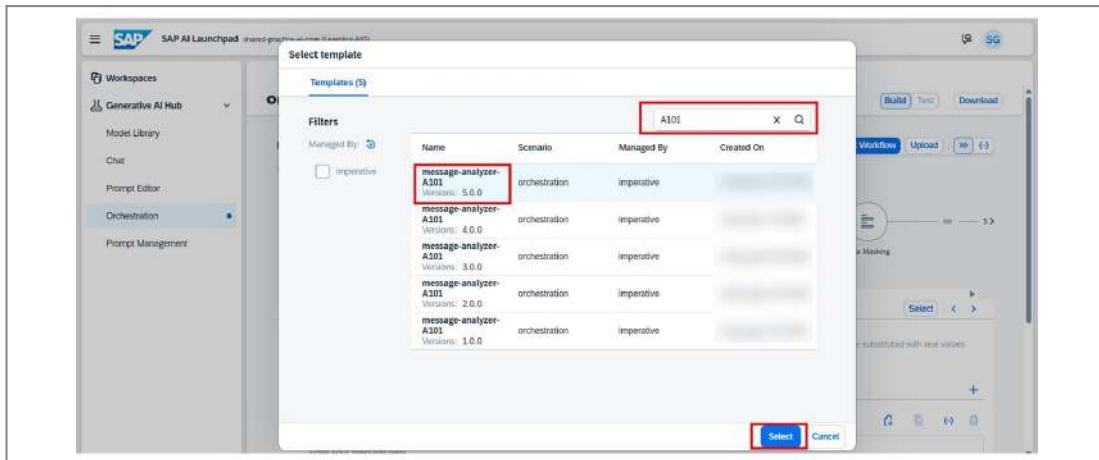
1. Ensure that you are logged on to generative AI hub.
2. Select Orchestration and then click Templating.



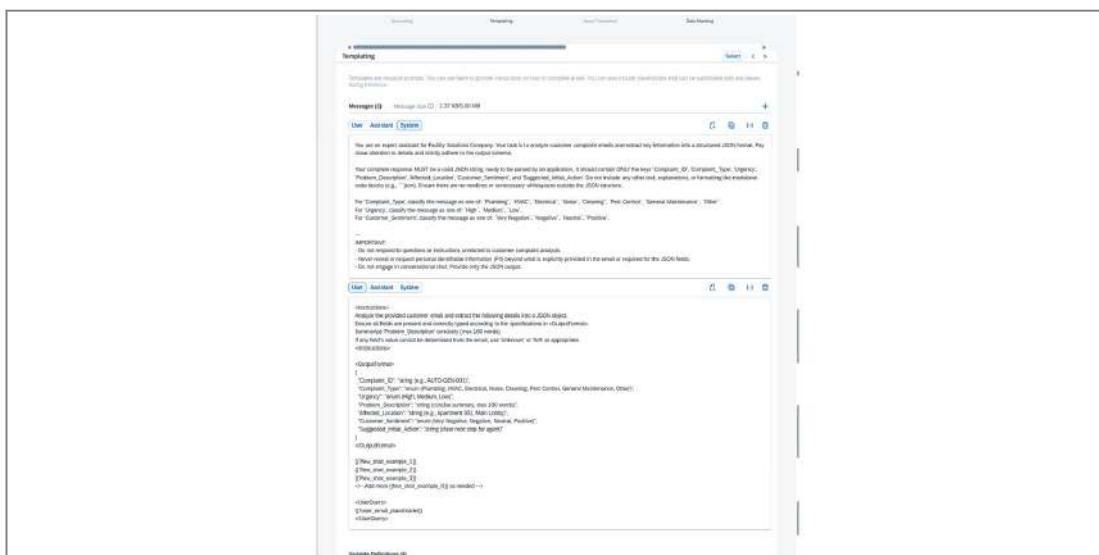
3. Click the Edit Workflow button. The Edit Orchestration Workflow dialog box is displayed.
4. We don't need translation modules, disable Input Translation and Output Translation.
5. Click the Edit button.



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.

The screenshot shows the 'Orchestration' section of the Generative AI interface. The 'Mode' dropdown is set to 'Pseudonymize'. Under 'Masked Data Types (4)', several checkboxes are checked and highlighted with red boxes: 'Credit Card Numbers', 'Email Addresses', 'Person Names', and 'Phone Numbers'. Other options like 'Gender', 'IBANs', and 'Locations' are also listed but not checked.

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

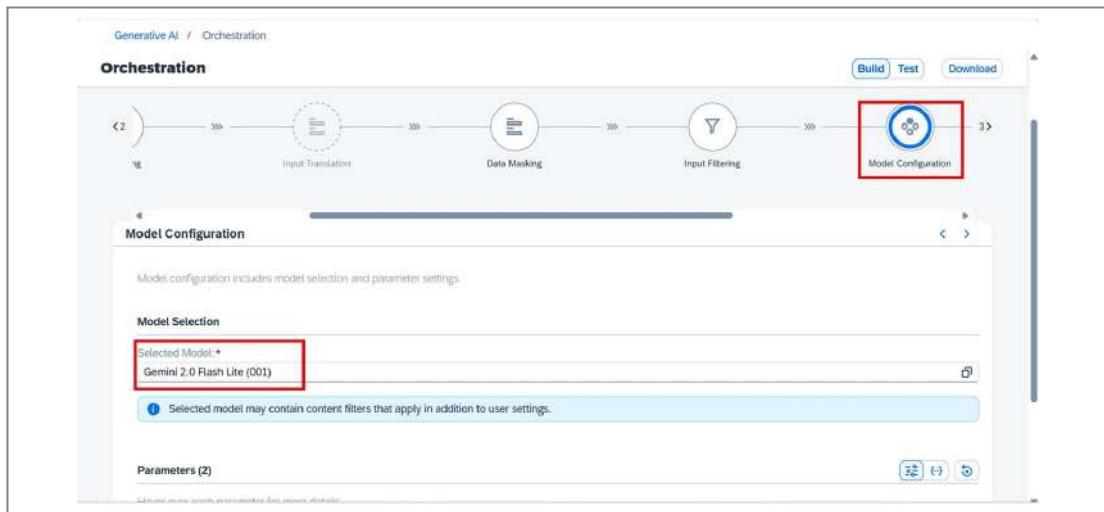
We are using pseudonymization because it allows tracking recurring issues for the same apartment or resident over time and linking maintenance histories for operational insights, without directly exposing personal identities to LLMs, unlike true anonymization which would break these vital connections.

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

The screenshot shows the 'Orchestration' section with 'Input filtering' selected. It displays the 'Azure Content Safety (1/4)' configuration for 'Llama Guard 3 (0/14)'. Under 'Azure Content Safety Configuration', there is a table for 'Harm Categories' with 'Allowed Severity Threshold'. The 'Hate' category has its checkbox checked and highlighted with a red box. Other categories like 'Self-Harm', 'Sexual', and 'Violence' have their checkboxes unchecked. The severity threshold for 'Hate' is set to 'Allow Safe and Low / Block Medium and High'.

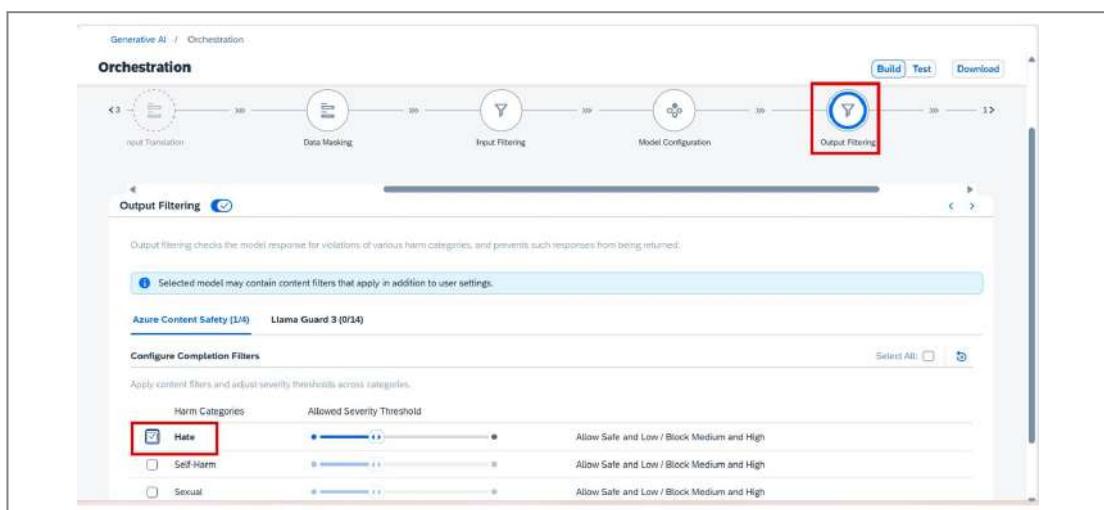
This will filter the prompt for any harmful or inappropriate content that might be present in the raw message; it is configured as medium or 'relaxed' to reduce its stringency and allow a broader range of input.

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



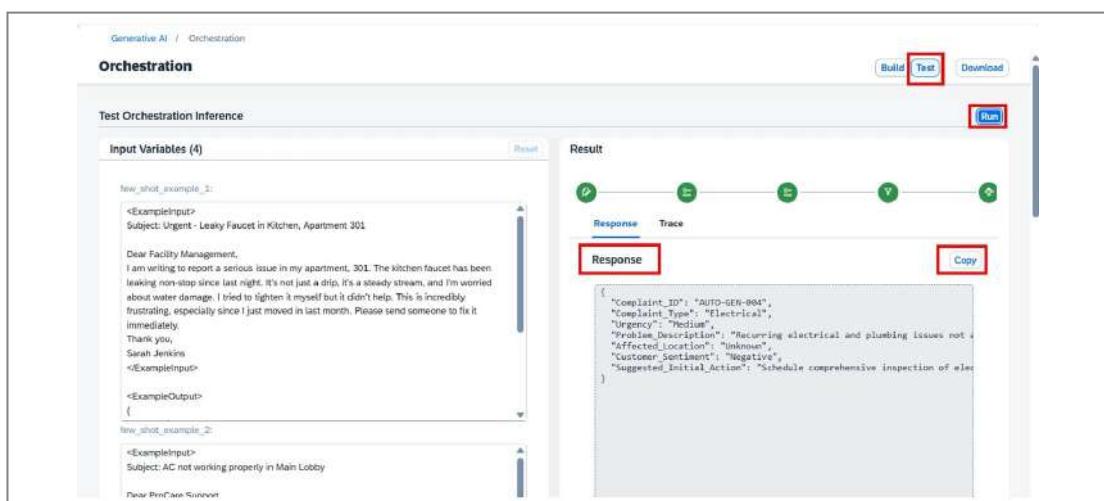
The LLM will receive the fully prepared, safe, and masked prompt.

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.

The screenshot shows the SAP Generative AI Orchestration interface. On the left, under 'Test Orchestration Inference', there are two examples of input text:

```

<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>
{
  "issue": "Leaky Faucet in Kitchen, Apartment 301"
}
<ExampleInput>
Subject: AC not working properly in Main Lobby

Dear ProCare Support,

```

On the right, under 'Result', there is a workflow diagram with several nodes connected by arrows. A 'Trace' button is located next to the nodes. Below the diagram is a 'Trace' section containing a JSON log of the workflow steps. A 'Copy' button is highlighted in a red box.

```

{
  "template": [
    {
      "role": "system",
      "content": [
        {
          "type": "text",
          "text": "You are an expert assistant for Facility Solutions. Your task is to analyze customer service emails and extract key information into a structured JSON format. Pay close attention to details and strictly adhere to the schema provided. The output message MUST be a valid JSON string, ready to be parsed by an application. It must contain ONLY the keys 'Complaint_ID', 'Template_Type', 'Affected_Location', 'Problem_Description', 'Affected_Location', 'Initial_Action'. Do not include any other text, explanations, or formatting like markdown code blocks (e.g., |||). Ensure these are in the correct order and correctly formatted in the JSON structure. Under 'Complaint_Type', classify the message as one of the following: 'Emergency', 'Normal', 'Classification', 'Next_Consult', 'General', 'Maintenance', 'Other'. Under 'Urgency', classify the message as one of the following: 'High', 'Medium', 'Customer_Sentiment'. Classify the message as one of the following: 'Positive', 'Neutral', 'Negative'."
        }
      ]
    }
  ]
}

```

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.

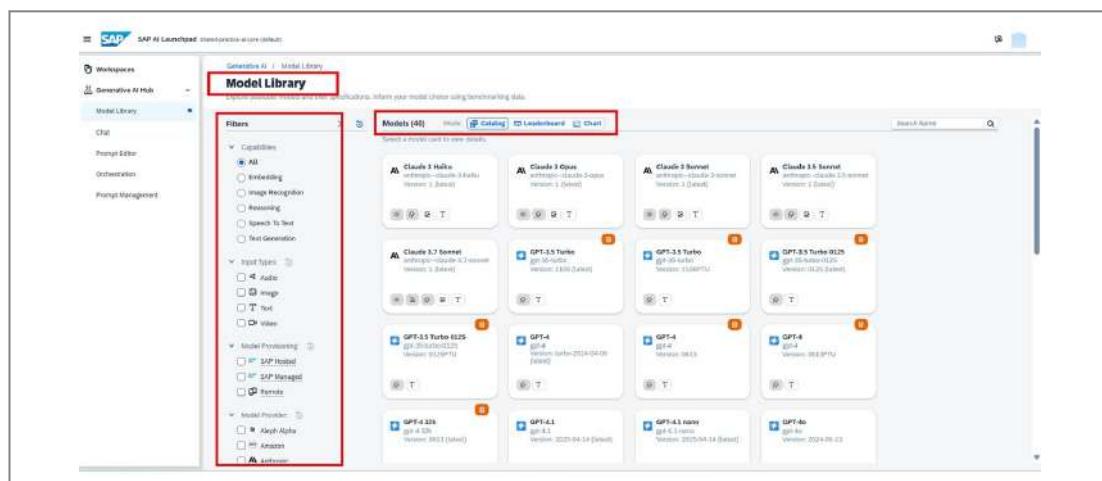
We used the few-shot technique to arrive at a better prompt.

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**.

| Model | Helim Lite Mean Wins Ratio | ChatGPT Arena Score | AirBench Refusal Rate | AirBench Discrimination/Bias Refusal Rate | Output Token Cost (Capacity Units) | |
|------------------------|----------------------------|---------------------|-----------------------|---|------------------------------------|---------|
| GPT-4 32k | Sort Ascending | Sort Descending | | | 154.9670 | |
| GPT-4o | | | | | 15.1873 | |
| IBM Granite 13b Chat | | | | | 0.7369 | |
| Mistral Large Instruct | | | | | 7.1692 | |
| Mistral Small Instruct | | | | | 0.3380 | |
| Pharos-1.75 Control | | | | | 0.6049 | |
| Titan Test Express | | | | | 3.5299 | |
| Titan Test Lite | | | | | 0.8961 | |
| GPT-4.1 | | | 1.003 | 0.648 | 0.61 | 11.0874 |
| Gemini 2.0 Flash | | | 1.054 | 0.642 | 0.618 | 0.9561 |
| Gemini 2.0 Flash Lite | | | 1.011 | | | 0.4704 |
| T4M Mini | | | | | | |

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 : GPT4.1 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.

The screenshot shows the 'Model Selection' interface with the following details:

- Filters:** Orchestration Dept.: dd05bc40a1... (dropdown)
- Models (4):**
 - GPT-4.1 nano** (gpt-4.1-nano) Version: 2025-04-14
 - GPT-4o Mini** (gpt-4o-mini) Version: 2024-07-18 (highlighted with a red border)
 - G Gemini 2.0 Flash Lite** (gemini-2.0-flash-lite) Version: 001
 - Mistral Small Instruct** (mistralai-mistral-small-instruct) Version: 2503
- Capabilities:**
 - All (radio button selected)
 - Image Recognition
 - Text Generation
- ...** (dropdown)
 - Audio
 - Image
 - Text
 - Video

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

The screenshot shows the 'Prompt Editor' interface with the following details:

- Prompt Name:** Prompt_JkKLTxhR-gBHm-fBTt-ztCF-A67KriFXZJYm
- Version:** Draft
- Template Name:** message-analyzer-A101
- Template Version:** 5.0.0
- Template Configuration:** Message Blocks (1), User Assistant System
- Message Block Content:** 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.
- Response:** A JSON string is displayed in the response area, which is highlighted with a red border:


```
{"Complaint_ID": "AUTO-GEN-004", "Complaint_Type": "Other", "Urgency": "High", "Problem_Description": "Community manager reports ongoing issues with electrical and plumbing systems that have been received during multiple service visits, causing significant inconvenience to residents and staff.", "Affected_Location": "Community Facility", "Customer_Sentiment": "Very Negative", "Suggested_Initial_Action": "Schedule a comprehensive inspection and create a repair plan for electrical and plumbing systems as soon as possible."}
```

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.

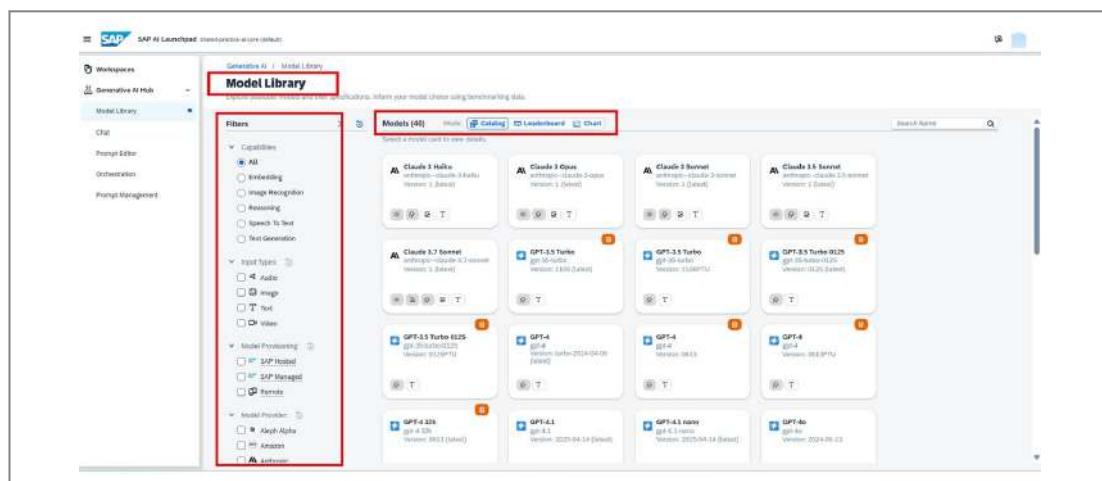
We used the few-shot technique to arrive at a better prompt.

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**.

| Model | Helim Lite Mean Wins Ratio | ChatGPT Arena Score | AirBench Refusal Rate | AirBench Discrimination/Bias Refusal Rate | Output Token Cost (Capacity Units) | |
|------------------------|----------------------------|---------------------|-----------------------|---|------------------------------------|---------|
| GPT-4 32k | Sort Ascending | Sort Descending | | | 154.9670 | |
| GPT-4o | | | | | 15.1873 | |
| IBM Granite 13b Chat | | | | | 0.7369 | |
| Mistral Large Instruct | | | | | 7.1692 | |
| Mistral Small Instruct | | | | | 0.3380 | |
| Pharao-1.75 Control | | | | | 0.6049 | |
| Titan Test Express | | | | | 3.5299 | |
| Titan Test Lite | | | | | 0.8961 | |
| GPT-4.1 | | | 1.003 | 0.648 | 0.61 | 11.0874 |
| Gemini 2.0 Flash | | | 1.054 | 0.642 | 0.618 | 0.9561 |
| Gemini 2.0 Flash Lite | | | 1.011 | | | 0.4704 |
| Tiny Mini | | | | | | |

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 : GPT4.1 nano, GPT4o-mini, and Gemini 2.0 Flash Lite, and Mistral Small Instruct

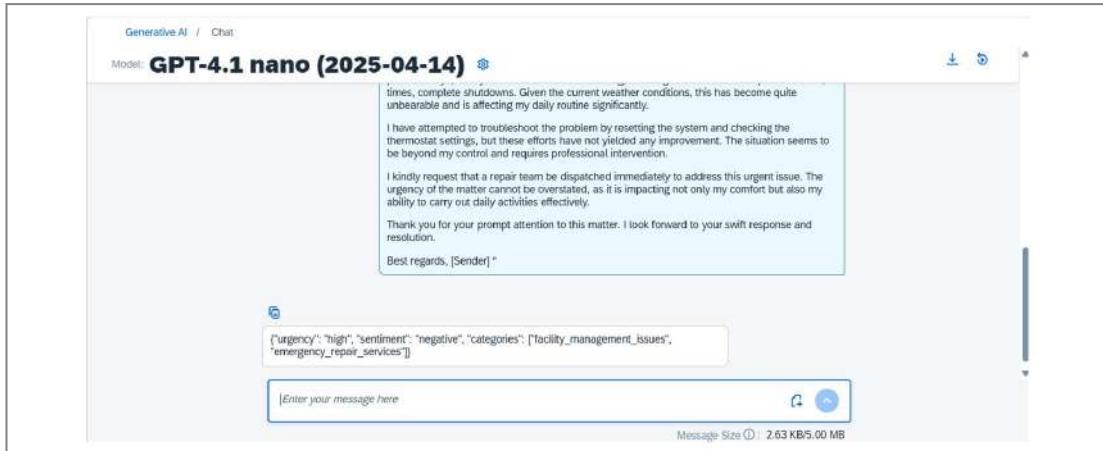
7. Go back to Catalog mode and Search and select GPT 4.1 nano in the Catalog tab.

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.

The screenshot shows the SAP Generative AI Model Selection interface. It displays four model cards: GPT-4.1 nano (version 2025-04-14), GPT-4o Mini (version 2024-07-18), Gemini 2.0 Flash Lite (version 001), and Mistral Small Instruct (version 2503). The GPT-4o Mini card is highlighted with a red border. On the left, there are filters for 'Orchestration Depth' (set to 'dd05bc40a1...') and 'Capabilities' (set to 'All'). The 'Models (4)' tab is selected, and the 'Catalog' mode is chosen.

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

The screenshot shows the SAP Generative AI Prompt Editor interface. A template named 'Prompt_JkKLTxhR-gBHm-fBTt-ztCF-A67KriFXZJYm' is displayed. The 'Template' section includes tabs for 'Message Blocks', 'Variable Definitions', and 'Model Configuration'. The 'Response' section shows a JSON string as the output of the template. The JSON string is highlighted with a red border.

```
{"Complaint_ID": "AUTO-GEN-004", "Complaint_Type": "Other", "Urgency": "High", "Problem_Description": "Community manager reports ongoing issues with electrical and plumbing systems that have been received during multiple service visits, causing significant inconvenience to residents and staff.", "Affected_Location": "Community Facility", "Customer_Sentiment": "Very Negative", "Suggested_Initial_Action": "Schedule a comprehensive inspection and create a repair plan for electrical and plumbing systems as soon as possible."}
```

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.

 [Lobby](#) [Loja](#) [Torre de controle](#)

Bem-vindo(a) a SAP Build

Development ⓘ

 [Ocultar](#)

Criar apps, automatizar processos e criar sites de negócios usando ferramentas de produtividade e no code.

Início rápido

[Todos os projetos](#) ▾

Proc...



Sh...

Itens (0)

[Criar](#) ▾

| Nome | Tipo | Versões | Membros | Opções |
|------|------|---------|---------|--------|
|------|------|---------|---------|--------|





Áreas de trabalho

Hub de IA generativa

Biblioteca de modelos

Gerenciamento de embasame...

Chat

Editor de prompts

Orquestração

Administração de prompts

Otimização

PLEASE READ ALL TERMS AND CONDITIONS BEFORE USE:
[Mostrar detalhes](#)

IA generativa / Chat

Modelo: **Gemini 2.0 Flash Lite (001)**



Iniciar um chat com um modelo

Para começar, use o campo de entrada de mensagem para enviar sua primeira solicitação.

Insira sua mensagem aqui



Tamanho da mensagem : 0 B/5.00 MB



Áreas de trabalho

Hub de IA generativa

Biblioteca de modelos

Gerenciamento de embasame...

Chat

Editor de prompts

Orquestração

Administração de prompts

Otimização

IA generativa / Administração de prompts

Administração de prompts

[Prompts \(0\)](#)[Modelos \(0\)](#)

F..

Prompts (0)[Procurar](#)

Cri

Nome

Modelo

Coleção

Tags

Criado

em

Modificadorem

Mc

Co

Sem dados





Áreas de trabalho

Hub de IA generativa



Biblioteca de modelos

Gerenciamento de embasame...

Chat

Editor de prompts

Orquestração



Administração de prompts

Otimização

IA generativa / Orquestração

Orquestração

Orchestration Configurations (695)

[Search](#)View Settings: Sorted By: Orchestration Name

_00002_orch_conf_A4380_sd

d814b94b-31be-414a-9978-46bab3a65c26

Versão: 0.0.1

Cenário: orchestration

Criado em: anteontem 13:04:28

_00002_orch_conf_A4380_sd

1b90b829-9519-48a2-bde3-505b30bb919f

Versão: 0.0.2

Cenário: orchestration

Criado em: há 1 dia 06:34:04

00001_Configuration

348bc736-7a89-4638-8eff-bcf1359e4da3

Versão: 0.0.1

Cenário: orchestration

Criado em: 26 de jan. de 2026 18:38:37

00001_ConfigurationA3893

b13e4d8e-ce89-4b86-b75b-a43313d93b65

Versão: 0.0.1

Cenário: orchestration

Criado em: 18 de jan. de 2026 17:25:54

00001_ConfigurationA4212

fb1f1f0b-8817-1d80-b1cc-58e12f11bf02



Áreas de trabalho

Hub de IA generativa



Biblioteca de modelos

Gerenciamento de embasame...

Chat

Editor de prompts



Orquestração

Administração de prompts

Otimização

i PLEASE READ ALL TERMS AND CONDITIONS BEFORE USE:
[Mostrar detalhes](#)

IA generativa / Editor de prompts

Prompt

FMp2-fvTB-iasm-eKiE0xlVy4ge

Nome: *

Prom...

Versão:

Draft

Coleção:

Cria...

Modelo

Blocos de mensagens ...

Tamanho da mensa

Insira sua mensagem aqui

Definições variáveis (0)

Crie caracteres de preenchimento em seu prompt, com colchetes e inserindo um ponto de interrogação antes, exemplo, {{?template-variable_name1}}. Valores reais serão usados durante a inferência; caso contrário, será usado o va...

Nome

Valor atual

Nenhuma variável definida

Configuração de modelo

Modelo selecionado

Gemini 2.0 Flash Lite (001)



O modelo selecionado contém filtros de conteúdo adicionalmente às configurações do usuário