



Chatbot Design: Part 1



ESG Course Support Chatbot

This is an intelligent chatbot designed to enhance student engagement and support in ESG (Environmental, Social, and Governance) education. Powered by Botpress, it seamlessly integrates a **dynamic knowledge base**, **LLM-driven responses**, and **personalized user interactions** to provide instant answers, midterm performance feedback, and lecturer details. With **structured workflows**, **interactive carousel cards**, and **data-driven personalization**, this chatbot transforms learning into an intuitive and engaging experience. Whether you're seeking ESG insights or academic guidance, this is your go-to digital companion! 🚀

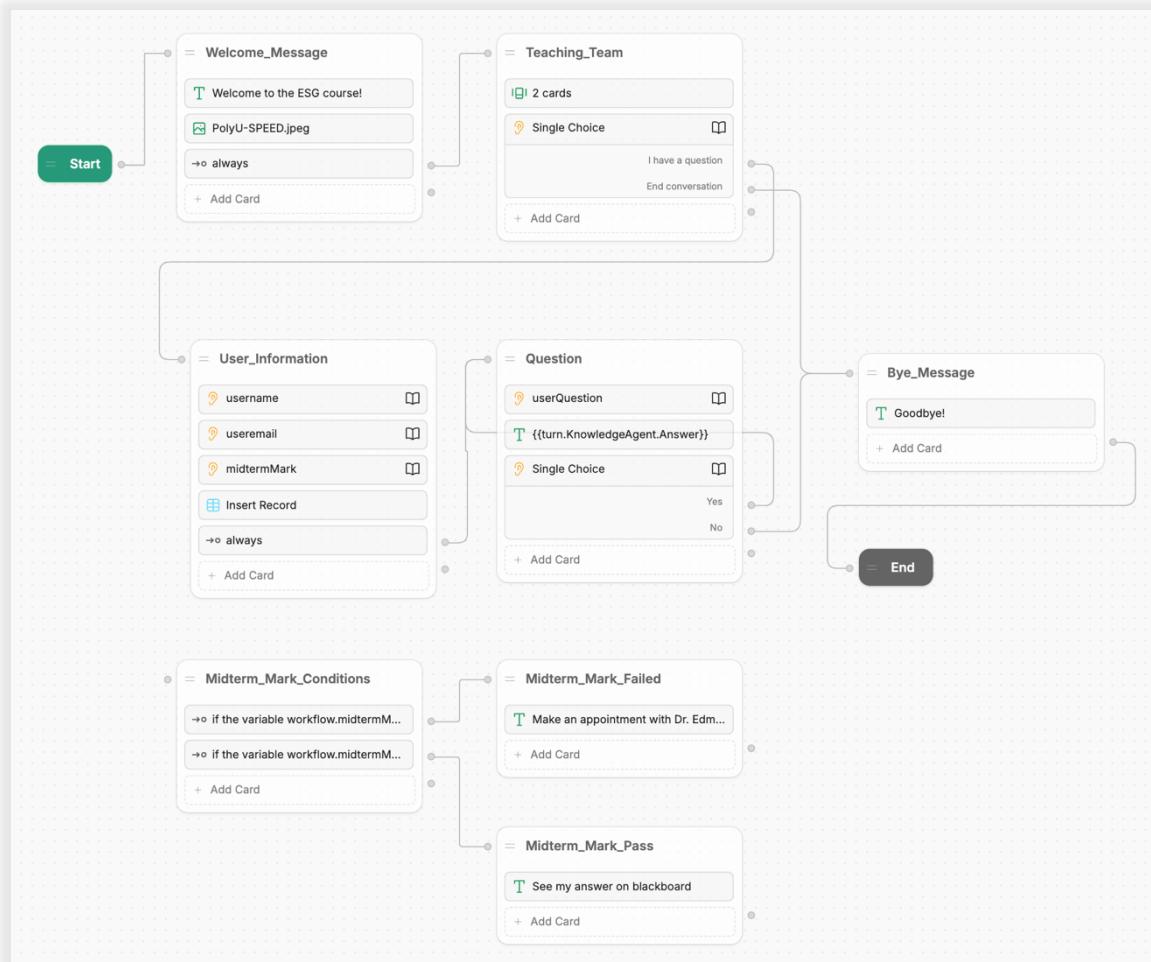
Key Features

- ✓ AI-Powered Knowledge Base – Instantly answers student queries using Botpress's LLM and a structured ESG knowledge base.
- ✓ Interactive Teaching Team Showcase – Displays lecturers' details in a carousel card, allowing students to easily identify their instructors.
- ✓ Personalized Student Records – Captures and stores user details (name, email, midterm marks) in a custom database.
- ✓ Midterm Performance-Based Feedback – Dynamically provides customized messages based on students' midterm scores, guiding them toward the next steps.



💡 Want to learn Botpress?
Check out my notes and tutorials here! 📚🚀
https://linktr.ee/21kaw.botpress_tutorial

Building an LLM-based ESG Course Support Chatbot on Botpress



Chatbot workflow.

The **ESG Course Support Chatbot** is designed to assist students in an ESG (Environmental, Social, and Governance) course by providing **automated responses** to questions, guiding users through structured interactions, and displaying **customized messages** based on midterm performance. The chatbot utilizes an LLM (Large Language Model) and a knowledge base to handle user queries dynamically. It also **collects user information**, such as names, emails, and midterm marks, to personalize responses.

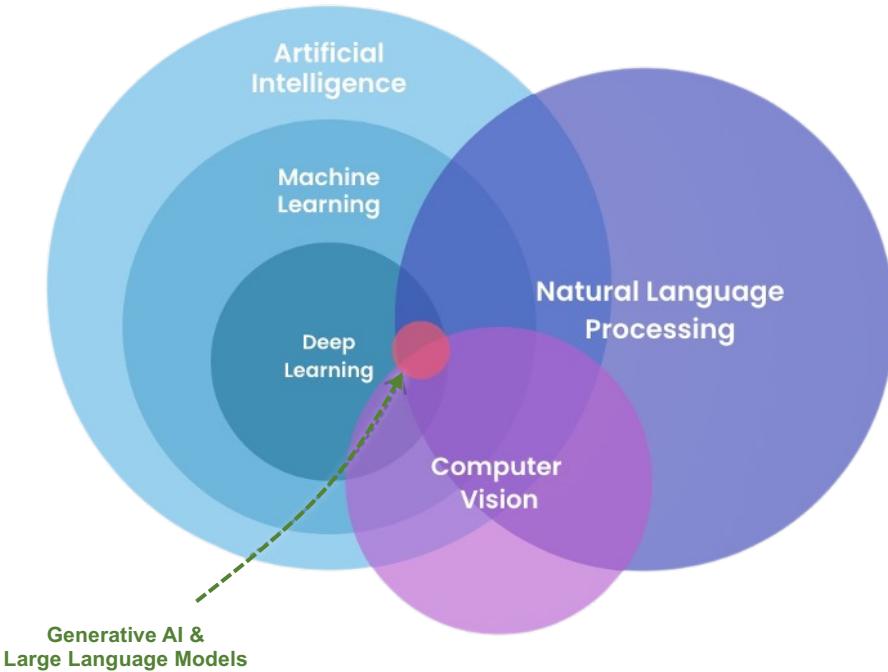
Video Tutorial

For basic operations of Botpress. You may refer to the videos on Botpress's official channel or the video here: <https://youtu.be/v05qGw-wX2Y>.

Meanwhile, our chatbot here focus on delivering course-related information to students and answer routine queries.



What is LLM (Large Language Model)?



Appypie.com, accessed 25 February, 2025,
<https://images.appypie.com/wp-content/uploads/2023/08/24053956/1.jpg>

A **Large Language Model (LLM)** is an advanced AI system designed to understand, process, and generate human-like text. These models are trained on vast amounts of text data, allowing them to respond to queries, complete sentences, summarize information, and even engage in conversations.

Key Features of LLMs

- 🧠 **Deep Understanding of Language** – Can interpret context, detect intent, and generate coherent responses.
- 📚 **Trained on Massive Datasets** – Uses diverse sources, including books, articles, and online content.
- 🔍 **Capable of Answering Complex Queries** – Supports knowledge-based Q&A, content creation, and reasoning.
- 💡 **Continuously Learning & Improving** – Models like GPT-4 evolve with advancements in AI research.

How LLMs Power Chatbots?

In chatbot development, **LLMs enhance user interactions** by:

- Understanding **natural language input** from users.
- Generating **relevant and contextual responses**.
- Summarizing and retrieving information from a **knowledge base**.
- Adapting to **different user intents** for a smoother experience.

LLM in the ESG Course Assistant Chatbot

The **ESG Course Support Chatbot** integrates an LLM to:

- **Answer student queries** about ESG topics using a knowledge base.
- **Provide intelligent responses** beyond simple predefined answers.
- **Enhance user experience** by making interactions more dynamic and human-like.

By leveraging LLM technology, the chatbot becomes more than just a rule-based assistant—it **understands, adapts, and responds intelligently**, making it an invaluable tool for student engagement. 🚀

Components

➤ Sending Messages / Capture Information



Comment Card

- Purpose:** Allows adding internal notes or remarks within the chatbot workflow.
- Use Case:** Used for documentation purposes to help developers or collaborators understand specific parts of the workflow.



Text/Image/Audio/Video/File/Location

- Purpose:** Displays multimedia content, including text, images, audio, video, files, or location data, to enhance user interaction.
- Use Case:** Can be used for delivering course-related images, lecture recordings, downloadable study materials, or location-based information.



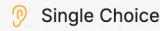
Carousel Card

- Purpose:** Presents multiple cards in a horizontal scrollable format, allowing users to browse and select content.
- Use Case:** Used for displaying multiple course modules, recommended textbooks, or learning resources in an interactive format.



Person Name / Email Address / Raw Input

- Purpose:** Captures user-provided text input, such as a name, email address, or any free-text response.
- Use Case:**
 - Person Name:** Collects a user's name for personalization.
 - Email Address:** Captures the user's email for follow-ups or authentication.
 - Raw Input:** Accepts open-ended text responses for flexible user input.



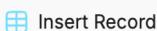
I have a question

End conversation

Inputs: Single Choice Card

- Purpose:** Presents multiple-choice options for user navigation.
- Use Case:** Allows users to select topics, such as "Basic Information," "Learning Outcomes," or "Assessment Methods."

➤ Execute



Execute: Insert Record

- Purpose:** Stores user-provided information into a database or external system.
- Use Case:** Used to log student details, chatbot interactions, or form submissions for tracking and data management.

Guidelines on Botpress Chatbot Design: Part 1

➤ Nodes

= Start

Start Node

- **Purpose:** Kicks off the chatbot conversation and initializes the workflow.
- **Use Case:** Includes an initial greeting, menu introduction, or chatbot instructions to guide users into the conversation flow.

= End

End Node

- **Purpose:** Marks the end of a chatbot interaction, terminating the conversation flow.
- **Use Case:** Used when the chatbot has completed its task, ensuring a natural exit for the user or providing an option to restart the conversation.



Linking the Nodes

- **Purpose:** Connects different nodes to create a logical flow.
- **Use Case:** Ensures smooth transitions between menu options, user inputs, and responses, allowing structured navigation.

➤ Expressions

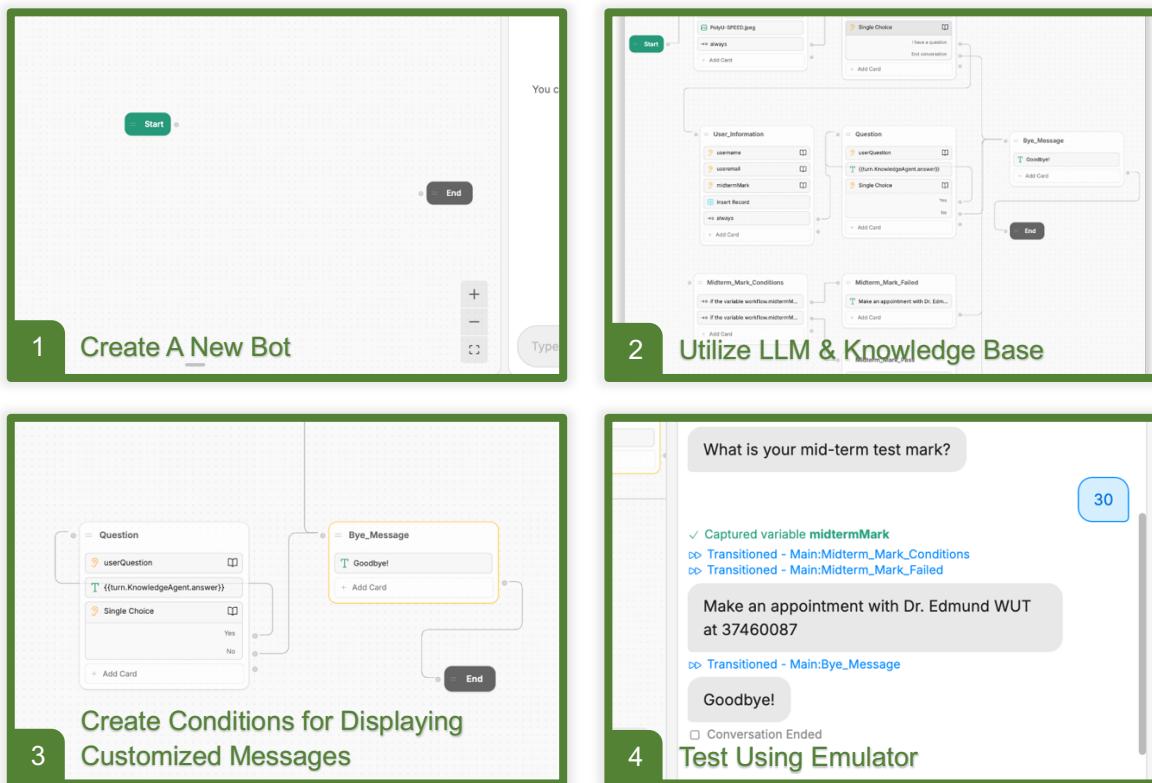
→○ Expression

Expression (Condition)

- **Purpose:** Enables the use of dynamic expressions to control chatbot logic and process data.
- **Use Case:** Used to create conditional responses, personalize user interactions, or automate chatbot decisions based on user input.

Guidelines on Botpress Chatbot Design: Part 1

Quick Look



The workflow consists of four main parts:

1. Setting up the bot by removing unnecessary nodes and retaining the **Start** and **End** nodes.
 - 1-A: Start a New Bot
 - 1-B: Remove Unnecessary Nodes
 - 1-C: Ensure Start and End Nodes Remain
2. Leveraging LLM and a knowledge base to generate intelligent responses to user queries.
 - 2-A: Welcome Message
 - 2-B: Teaching Team Selection
 - 2-C: Create a Knowledge Base and Upload Course Materials
 - 2-D: Capture & Store User Information
 - 2-E: Handling User Questions
3. Implementing conditional logic to display customized messages based on midterm marks.
4. Testing the chatbot using the emulator before deployment.

- **Follow the previous set of guideline (Setting-up Course Support Chatbot on Botpress) for detailed steps on initial setup.**

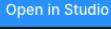


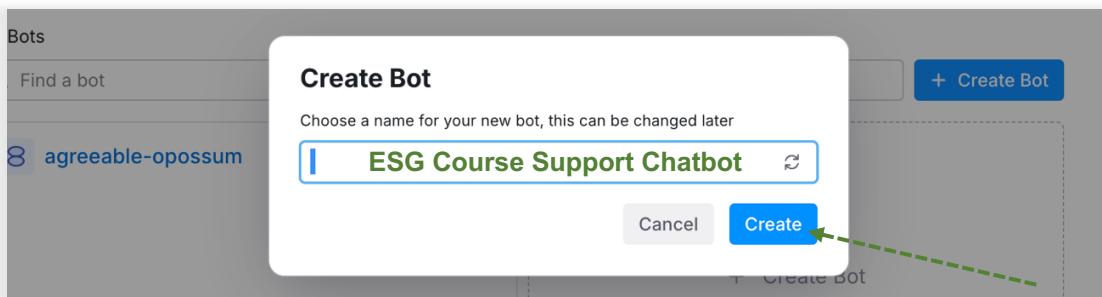
Guidelines on Botpress Chatbot Design: Part 1

Section 1: Create A New Bot

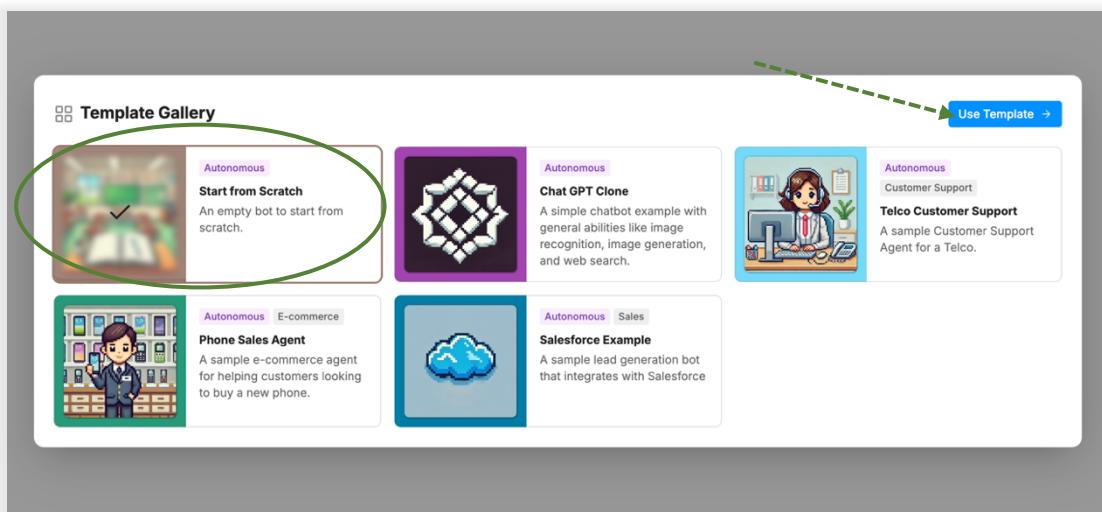
1-A: Start a New Bot

1.1. Open **Botpress**  app.botpress.cloud and create a new bot named **ESG Course Support**

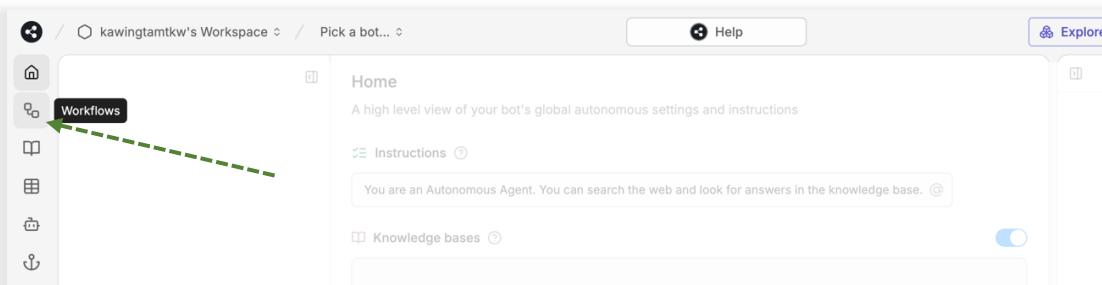
Chatbot. Then, click on  **Open in Studio**.



1.2. From the **Template Gallery**, select **Start from Scratch**. Then click on  **Use Template**.



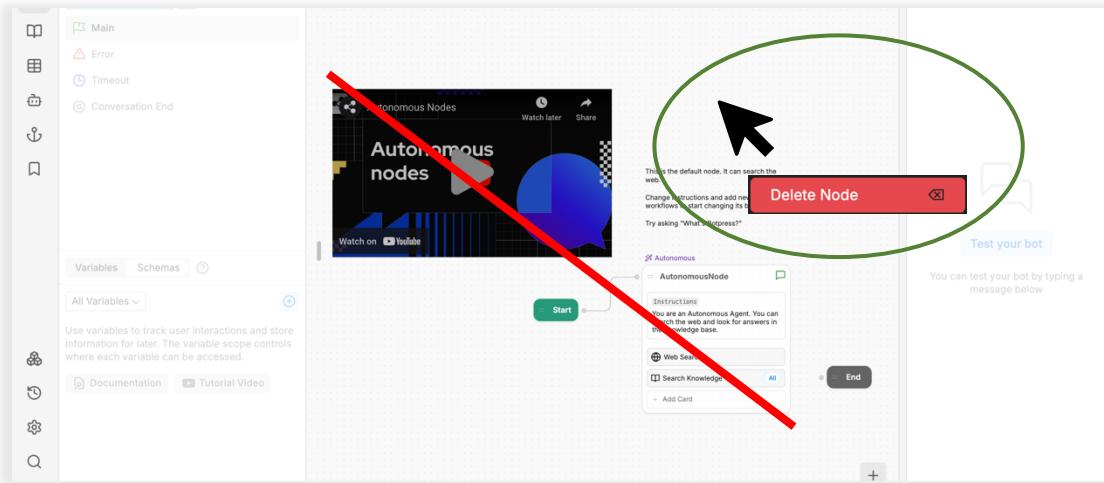
1.3. From the left menu (figure 14), click on  **Workflows**.



Guidelines on Botpress Chatbot Design: Part 1

1-B: Remove Unnecessary Nodes

1.4. After the bot is generated, remove any unnecessary default nodes **Start** and **End** to maintain a clean workspace.



1-C: Ensure Start and End Nodes Remain

1.5. The Start node will be the entry point of the bot, triggering when a user initiates a conversation.

1.6. The End node will terminate the conversation gracefully after responses are provided.



End of Section 1

Great work!

Now, let's move on to integrating the LLM and knowledge base to make your chatbot smarter. 

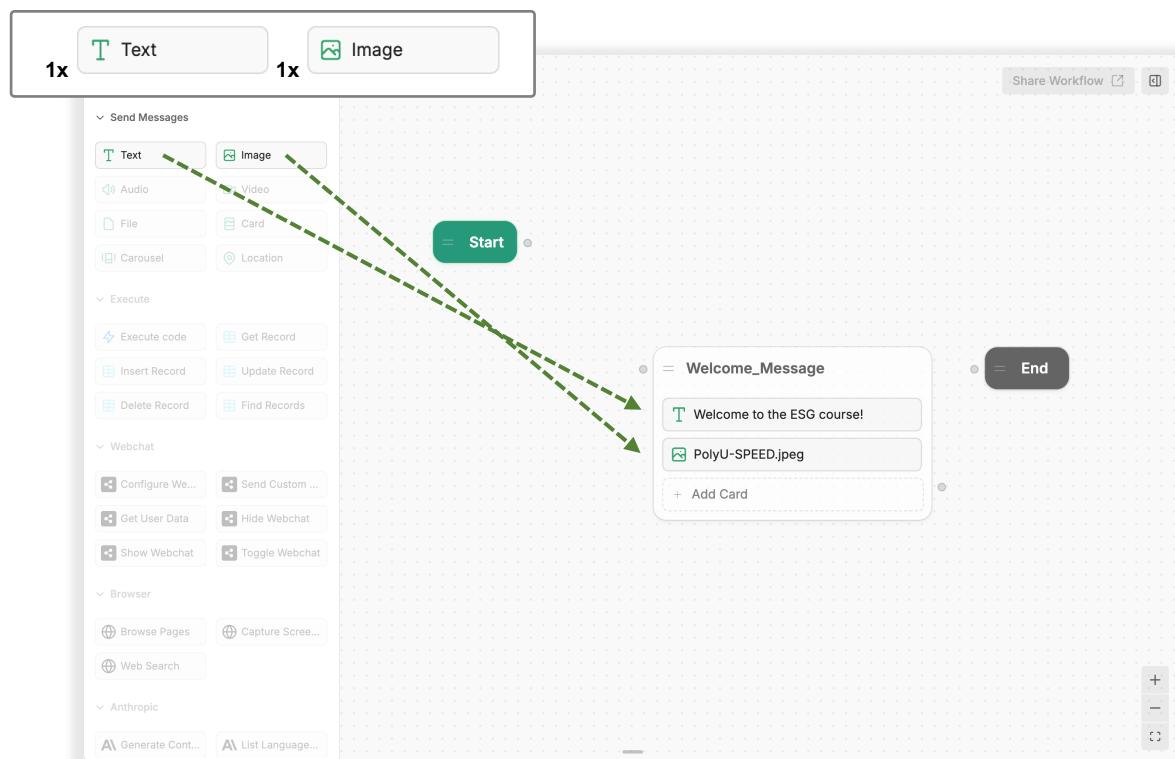


Section 2: Utilize LLM & Knowledge Base

This section handles the upper part of the workflow, where the bot dynamically answers user questions based on an integrated knowledge base.

2-A: Welcome Message

- 2.1. Add a node named **Welcome_Message** with a **Text** card.
- 2.2. Include a text message: "Welcome to the ESG course!"
- 2.3. Optionally, attach an **Image** card related to the course (e.g., *PolyU-SPEED.jpeg* as shown in the workflow.

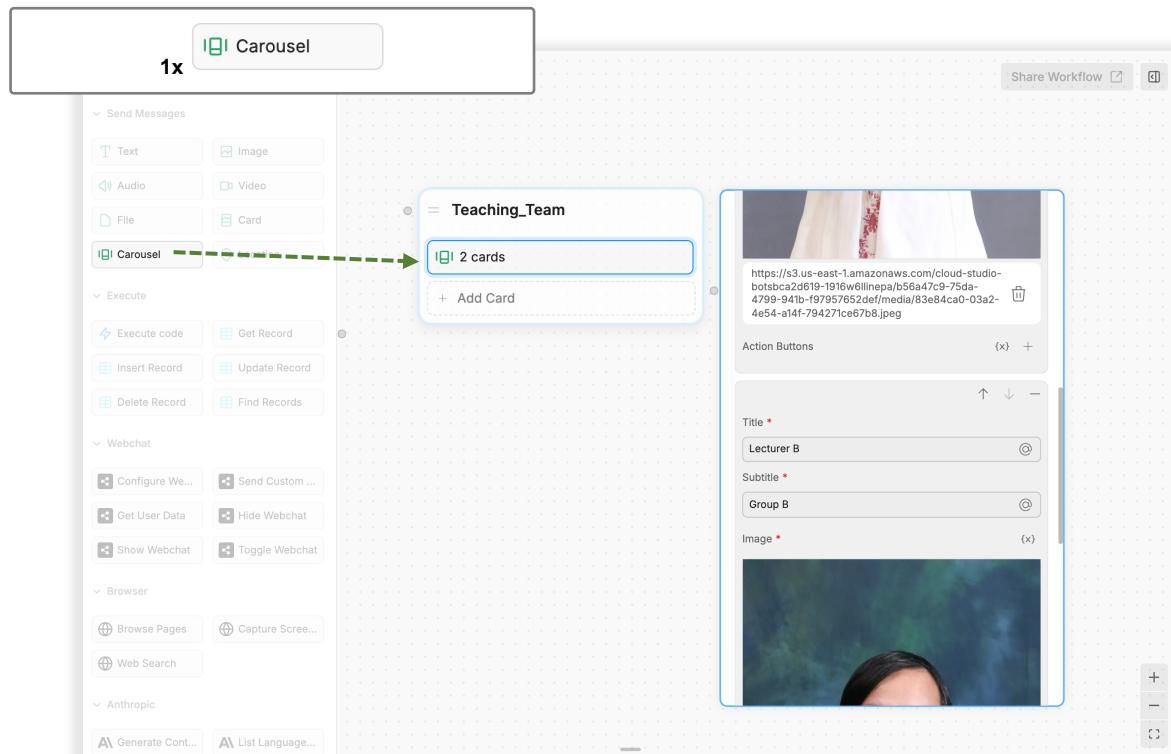


2-B: Teaching Team Selection

2.4. Create a node called **Teaching_Team** with a **Carousel** card. Add carousel item using **Add**

carousel cards . Each carousel item should include:

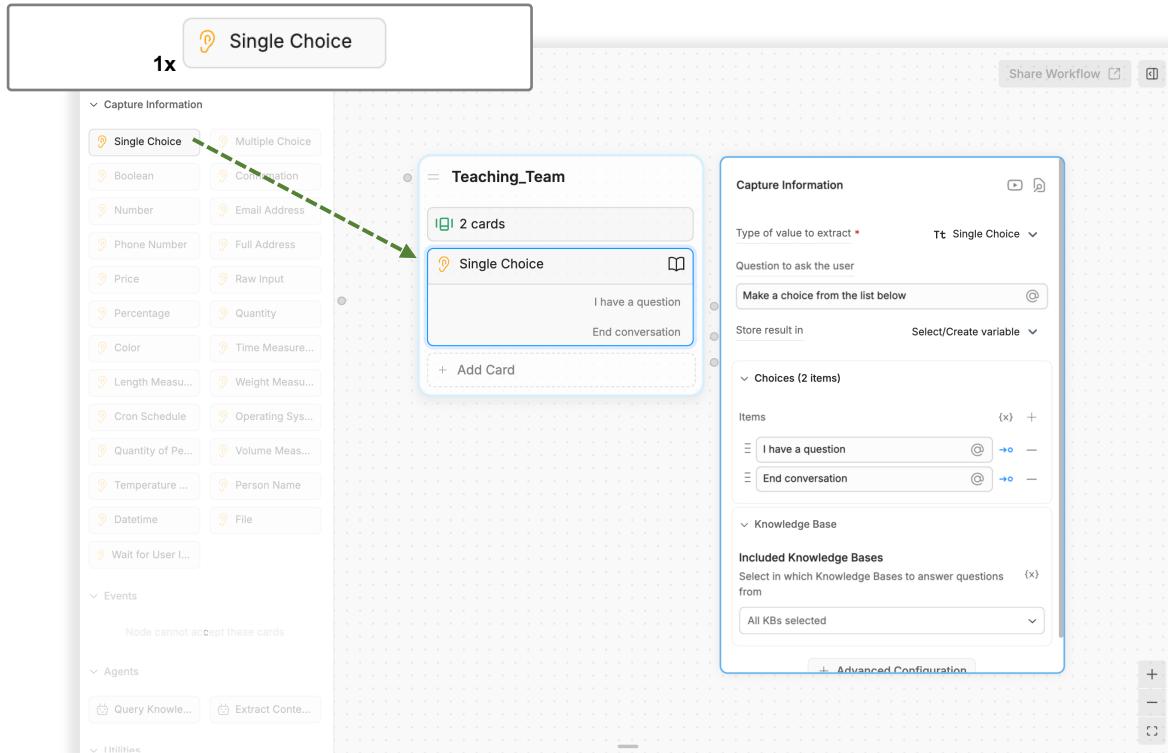
- **Title:** Lecturer's name (e.g., "Lecturer A")
- **Subtitle:** Group information (e.g., "Group A")
- **Image:** Lecturer's portrait



Guidelines on Botpress Chatbot Design: Part 1

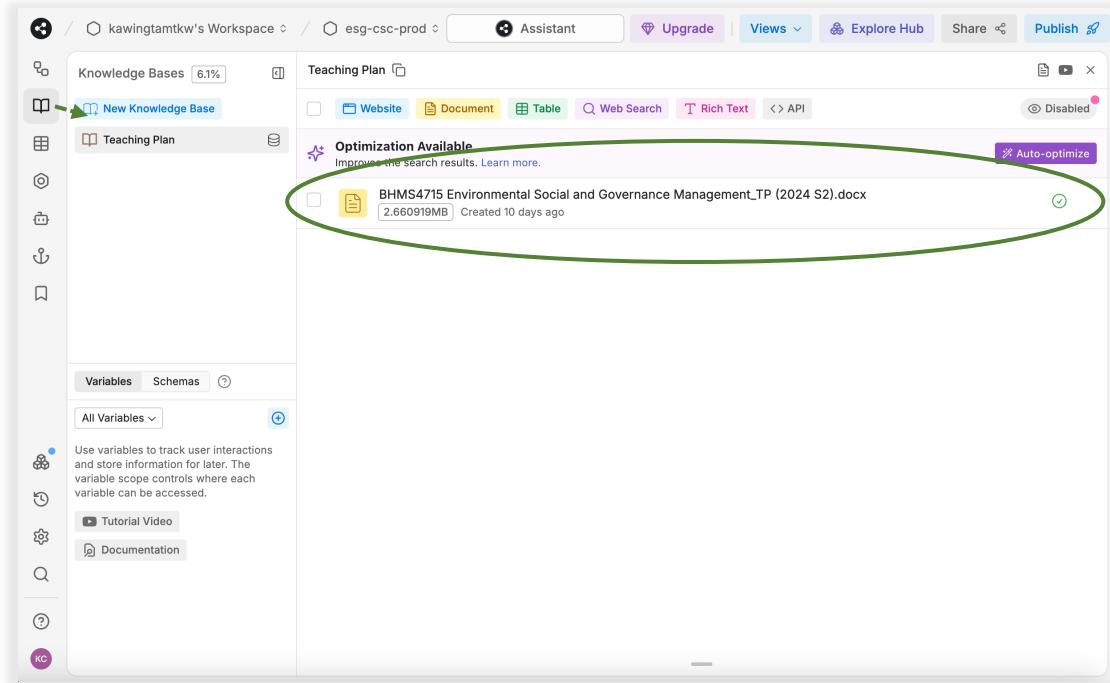
2.5. After the carousel, add a **Single Choice Card** with two options:

- Question to ask the user: **Make a choice from the list below**
- Choice 1: "**I have a question**" (leading to LLM-powered responses)
- Choice 2: "**End conversation**" (leading to the **Bye_Message** node)



2-C: Create a Knowledge Base and Upload Course Materials

To allow the chatbot to reference course materials, create a Knowledge Base (KB) and upload the teaching plan.



2.6. Go to the  **Knowledge Bases** section in Botpress.

2.7. Click  **New Knowledge Base** and name it **Teaching Plan**.

2.8. Upload the document "**BHMS4715 Environmental Social and Governance Management_TP (2024 S2).docx**" to the Knowledge Base.

- 💡 If the knowledge base is unavailable on runtime. Go to the  **Agents** section in Botpress, and ensure the  **Knowledge Agent** is **enabled** for chatbot responses.
- 💡 The chatbot will use the teaching plan document to answer course-related queries dynamically. This enables more accurate and context-aware responses.

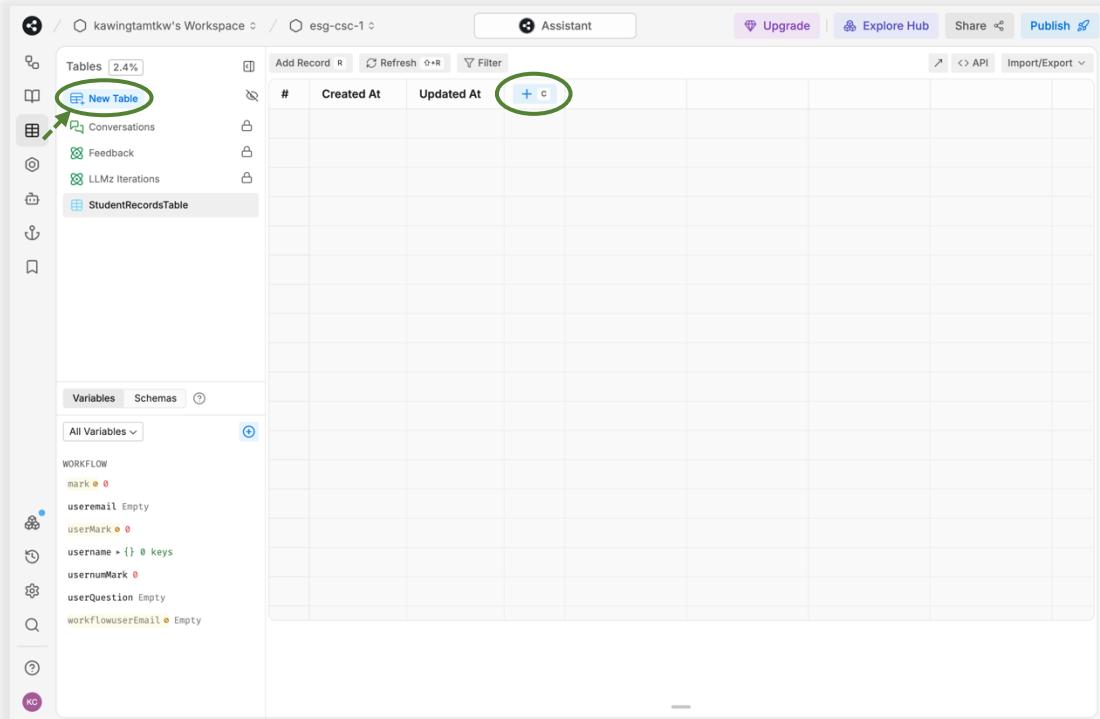
2-D: Capture & Store User Information

Before creating the User_Information node, users need to set up a StudentRecordsTable to store student details. Follow these steps:

2.9. Navigate to the  Tables section in Botpress.

2.10. Click  New Table and name it **StudentRecordsTable**.  Add these columns:

-  **Name** (Type: Object, Nullable)
-  **Email** (Type: String, Nullable)
-  **Marks** (Type: Number, Nullable)



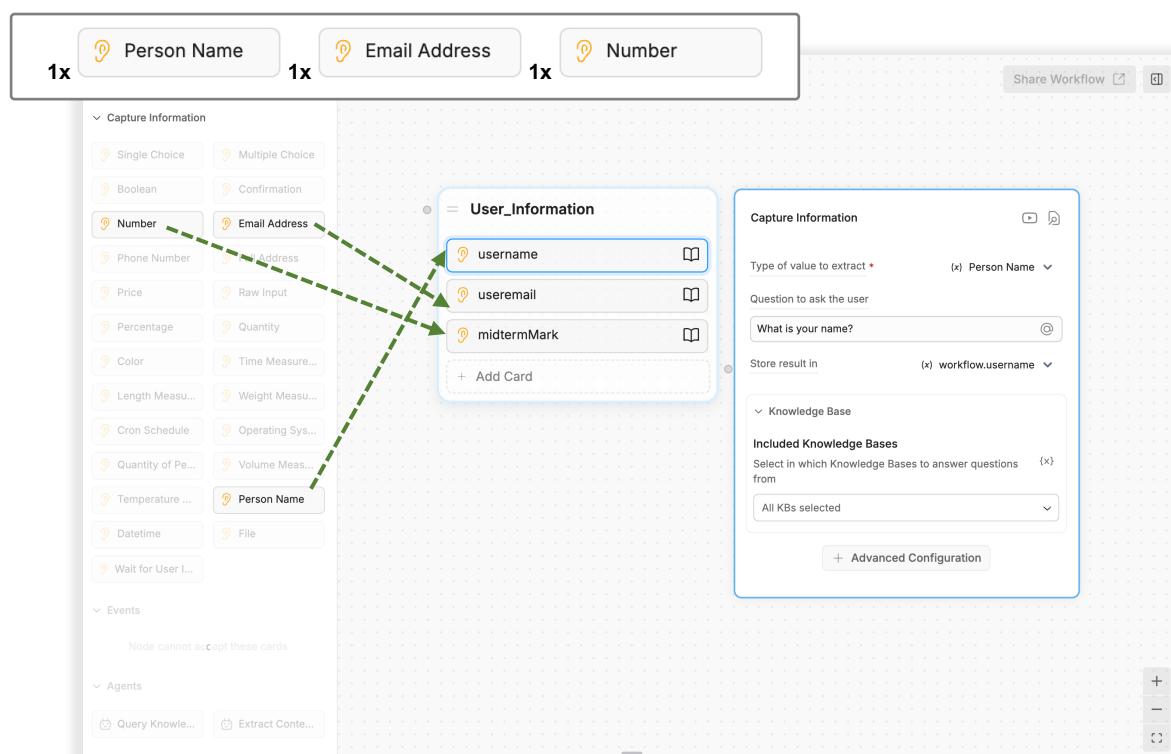
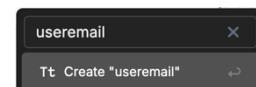
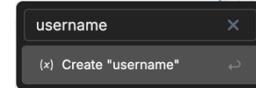
The screenshot shows the Botpress interface with the 'Tables' section open. On the left, there's a sidebar with 'Tables' (2.4%), 'Conversations', 'Feedback', 'LLMz Iterations', and 'StudentRecordsTable' (selected). The main area shows a table with columns '#', 'Created At', and 'Updated At'. At the top right of the table area is a blue button with a '+' and a 'c' icon. The bottom left of the interface shows a 'Variables' section with a list of variables under 'WORKFLOW' and 'userQuestion'.

Guidelines on Botpress Chatbot Design: Part 1

To capture user inputs,

2.11. Then, back to  Workflows. Create a node called **User_Information** to collect:

-  **Username (username)**
 - Question to ask the user: **What is your name?**
 - Store result in: **username**
-  **Email (useremail)**
 - Question to ask the user: **Email?**
 - Store result in: **useremail**
-  **Midterm Mark (midtermMark)**
 - Question to ask the user: **What is your mid-term test mark?**
 - Store result in: **midtermMark**



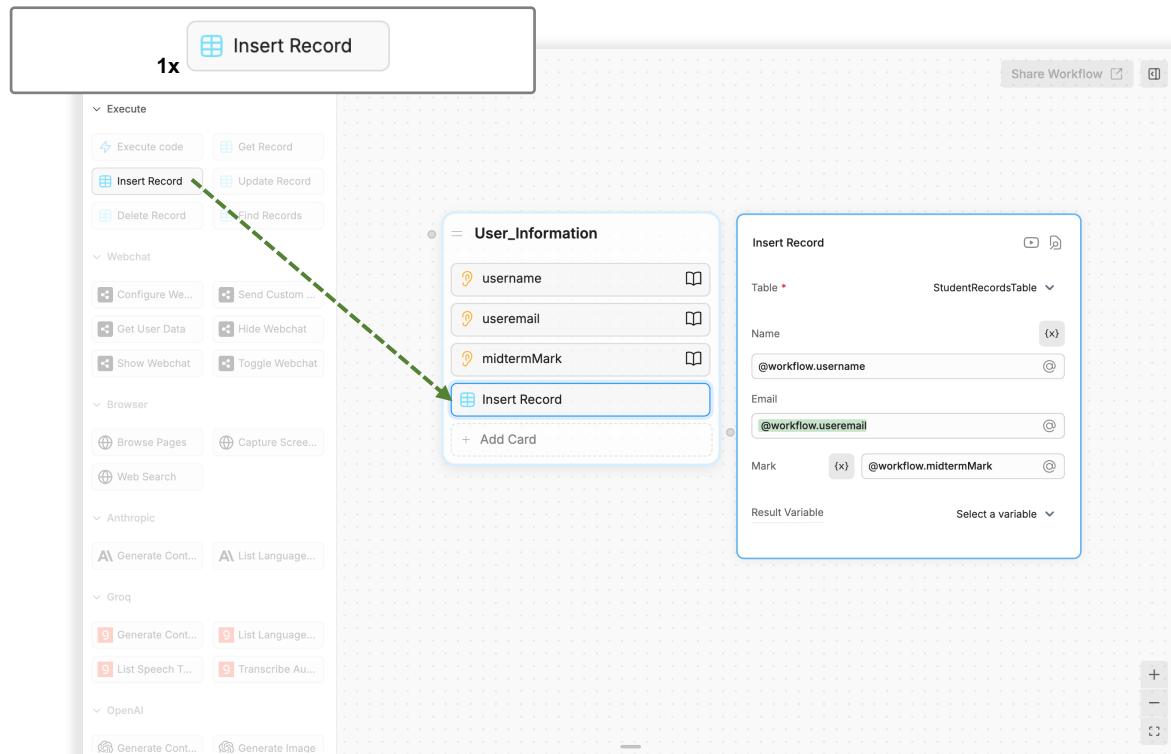
Guidelines on Botpress Chatbot Design: Part 1

To store user inputs to StudentRecordsTable,

2.12. Add **Insert Record** card to the **User_Information** node.

2.13. Select **StudentRecordsTable** for Table. Then, click  to insert variables to corresponding fields:

- Name: **@workflow.username**
- Email: **@workflow.useremail**
- Mark: **@workflow.midtermMark**



Guidelines on Botpress Chatbot Design: Part 1

To filter student records in StudentRecordsTable,

The screenshot shows the Botpress workspace interface. On the left, there's a sidebar with icons for Tables, Conversations, Feedback, and LLMz Iterations. The 'Tables' section is expanded, showing 'StudentRecordsTable' and other tables like 'New Table', 'Conversations', 'Feedback', and 'LLMz Iterations'. The main area shows a table with columns 'Name' and 'Mark'. A green oval highlights the 'Filter' button in the top right of the table header. A black dropdown menu is open, showing filter options like 'is equal to', 'is not equal to', 'less than', etc. A green arrow points from the 'is equal to' option in the dropdown to the 'is equal to' option in the dropdown menu.

2.14. Open the StudentRecordsTable

- Navigate to **Tables** in Botpress.
- Click on **StudentRecordsTable** to access the stored records.

2.15. Apply a Filter

- Click the **Filter** button at the top of the table.
- A filter panel will appear, allowing you to specify conditions.

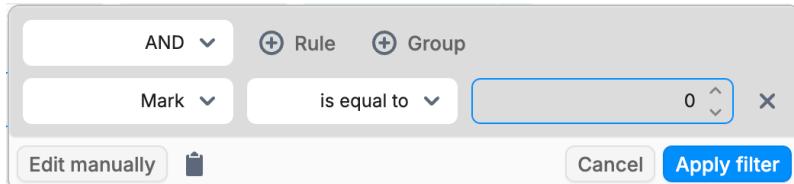
Guidelines on Botpress Chatbot Design: Part 1

2.16. Set Filtering Conditions

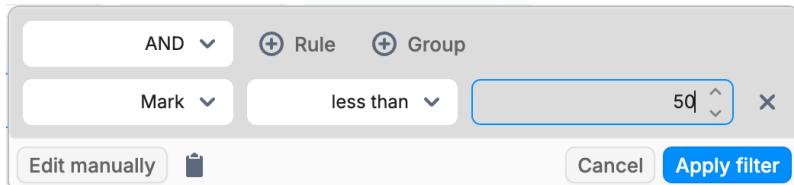
- In the **Filter Panel**, you can define rules for filtering midterm marks (**Mark** variable) using various conditions:

Condition	Usage Example	Description
is equal to	Mark = 70	Shows students who scored exactly 70.
is not equal to	Mark \neq 50	Displays students who did not score 50.
less than	Mark < 50	Filters students who scored below 50.
less than or equal to	Mark \leq 60	Lists students with marks 60 or lower.
greater than	Mark > 80	Displays students scoring above 80.
greater than or equal to	Mark \geq 75	Shows students with marks 75 or higher.
in	Mark in (50, 60, 70)	Filters students with specific marks.
not in	Mark not in (30, 40)	Excludes students with selected marks.

- Example: Filtering student(s) with midterm mark of zero (0):



- Example: Filtering student(s) with midterm mark lower than 50:



2.17. Execute the Filter

- After selecting the filter condition, enter the value (e.g., 50, 75, etc.).
- Click **Apply filter** **Apply Filter** to update the table view.

2.18. Reset the Filter

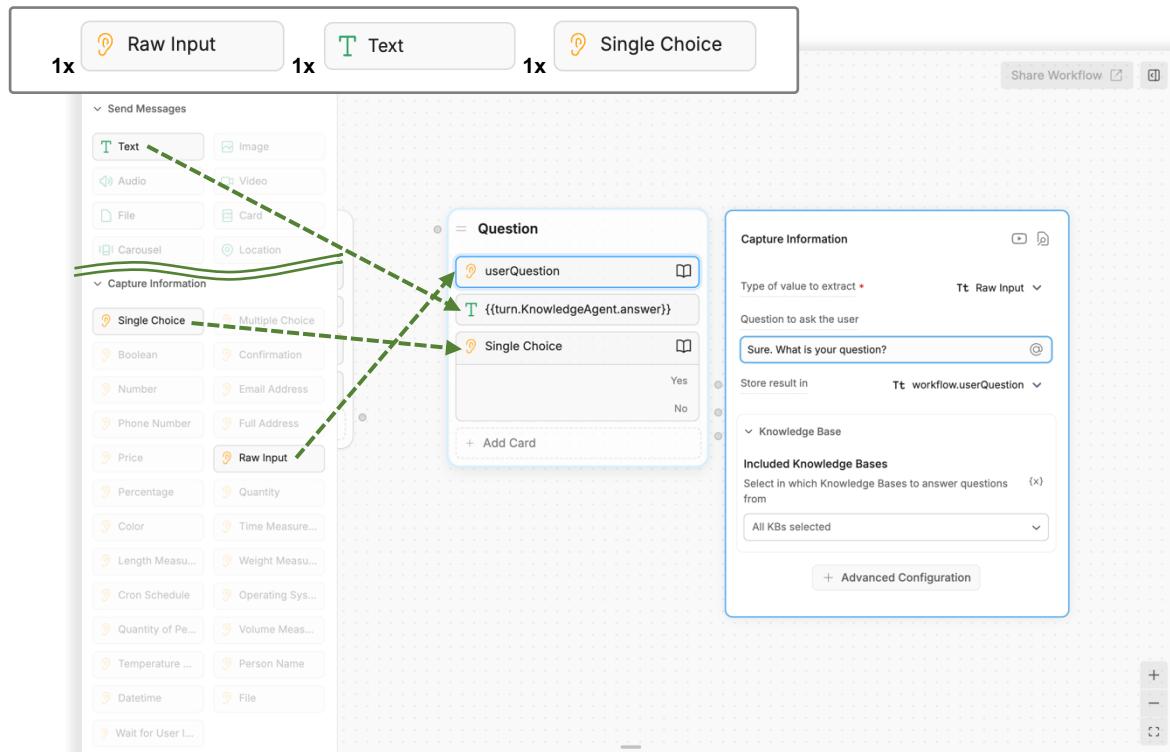
- To view all records again,  remove the filter condition(s).

Guidelines on Botpress Chatbot Design: Part 1

2-E: Handling User Questions

2.19. Go to the  **Workflow** section in Botpress.
2.20. Create a node named **Question**. Add the following cards:

-  **Raw Input Card**(userQuestion)
 - Question to ask the user: **Sure. What is your question?**
 - Store result in: **userQuestion**
-  **Text** Card with message to send: `{ {turn.KnowledgeAgent.answer} }`
-  **Single Choice Card**
 - Question to ask the user: **Do you have any other question?**
 - Choice 1: **"Yes"** (continues the conversation)
 - Choice 2: **"No"** (leads to **Bye_Message**)



End of Section 2

Awesome! Your chatbot can now handle user queries dynamically using LLM and a knowledge base.

Next, let's set up conditions to display customized messages based on midterm performance. 



Section 3: Create Conditions for Displaying Customized Messages

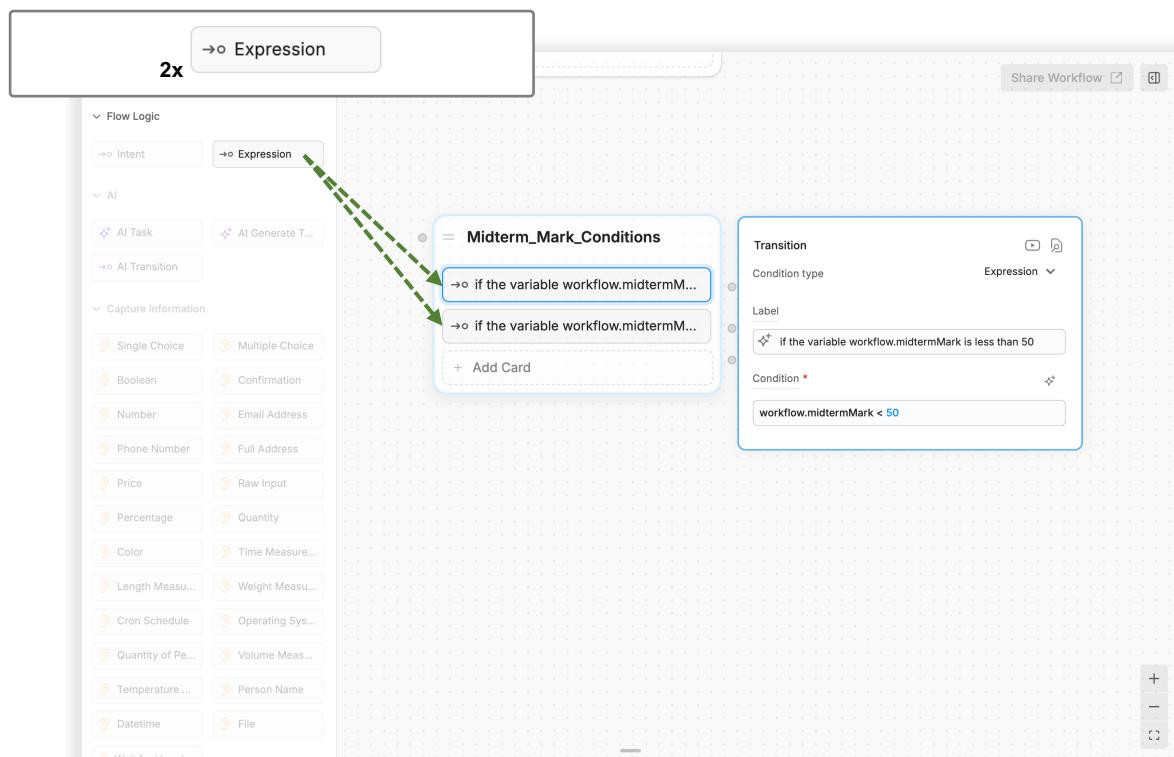
This section focuses on the **midterm performance condition checks**, displayed in the lower part of the workflow.

3-A: Define Midterm Mark Conditions

3.1. Create a node named **Midterm_Mark_Conditions** with two  **Expression cards**.

3.2. Add the following conditions respectively:

1. **Condition card 1:** `workflow.midtermMark < 50`
2. **Condition card 2:** `workflow.midtermMark >= 50`



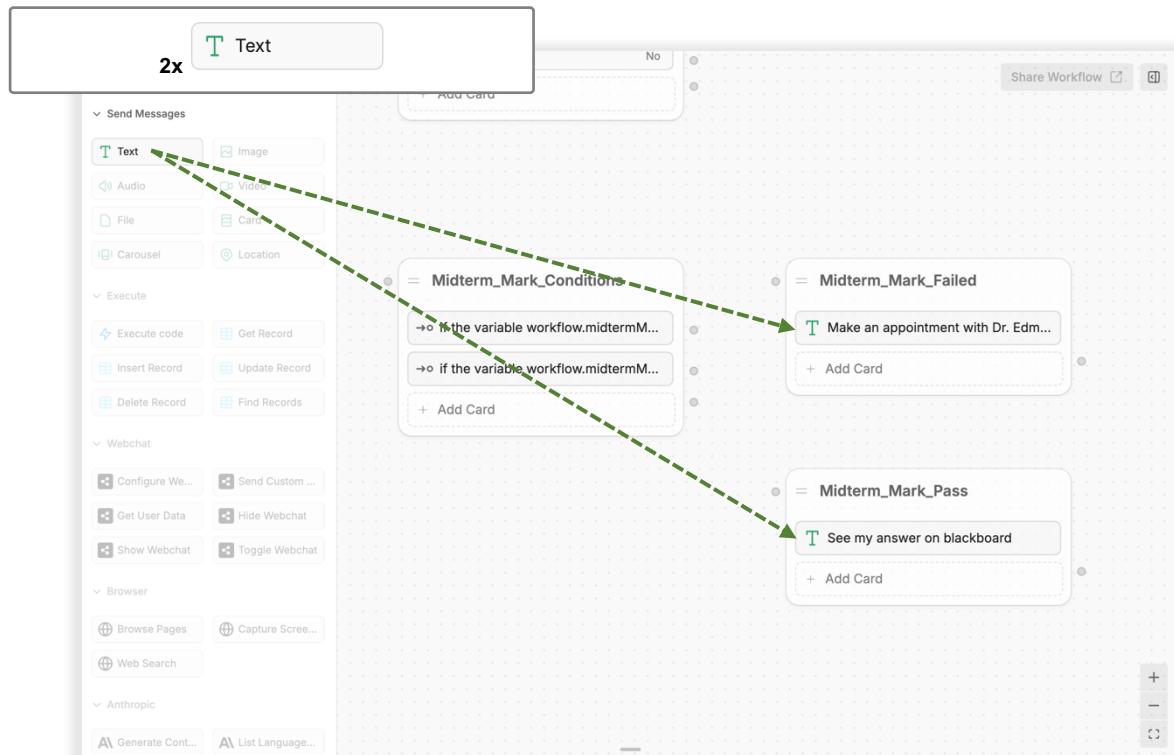
3-B: Return Different Message Based on Mark Value

To show different message to display by conditions,

3.3. Create a node named **Midterm_Mark_Failed** with a  **Text** card.

3.4. Add the following display messages respectively:

1. **Midterm_Mark_Failed** node: “**Make an appointment with Dr. Edmund for further consultation.**”
2. **Midterm_Mark_Pass** node: “**See my answer on Blackboard.**”

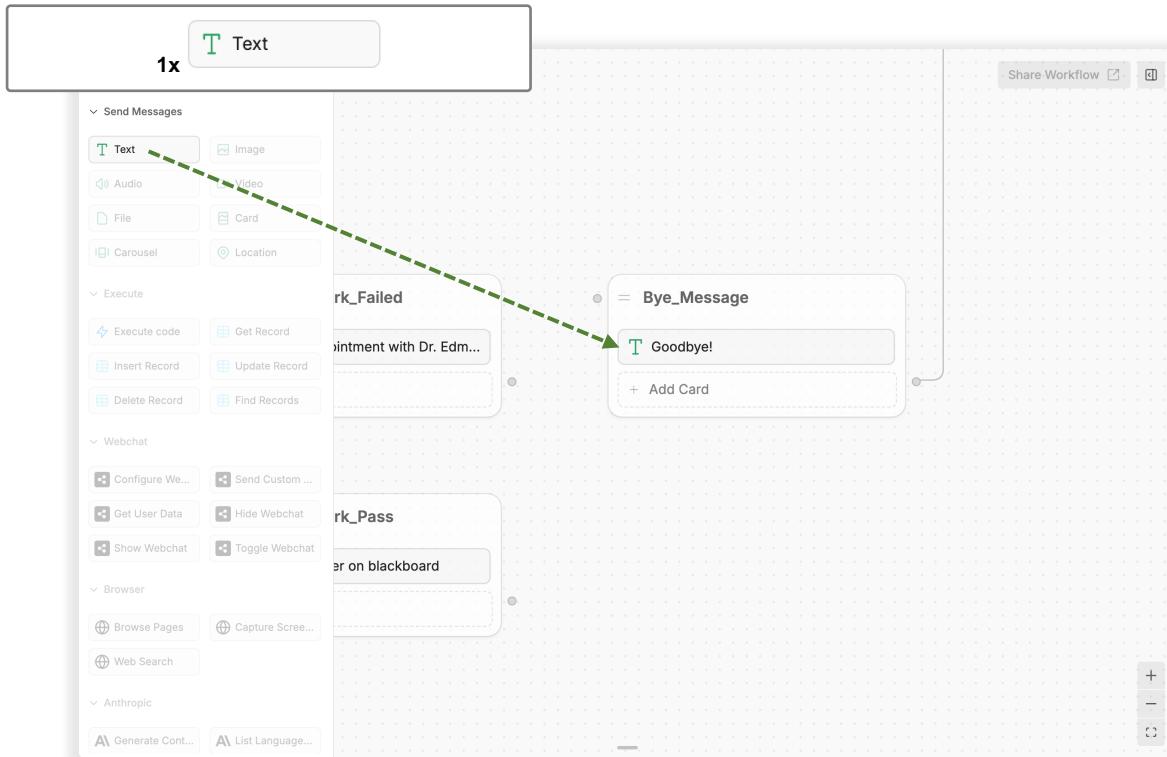


Guidelines on Botpress Chatbot Design: Part 1

3-C: End the Conversation Gracefully

3.5. After processing responses, direct the user to the **Bye_Message** node.

3.6. In **Bye_Message**, set a Text node with message: “Goodbye!” and connect it to the **End** node.



End of Section 3

Well done! Your chatbot can now personalize responses based on user input.

Finally, let's test everything in the emulator to ensure a smooth user experience.



Section 4: Test Using Emulator

4-A: Run the Emulator

- 4.1. Click on the **Test Emulator** within Botpress.
- 4.2. Start a conversation and verify the flow.

4-B: Check Knowledge Base Integration

- 4.3. Ask a sample question in the **Question** node.
- 4.4. Verify that the response is generated correctly from the knowledge base.

4-C: Validate Conditional Logic

- 4.5. Enter different **midterm marks** to test if the bot correctly routes users to **Midterm_Mark_Failed** or **Midterm_Mark_Pass** nodes.

4-D: Ensure Proper Conversation End

- 4.6. Select various conversation options to confirm smooth transitions to the **End** node.

End of Section 4

Great job! Your chatbot is now fully functional and ready to assist students.

Keep refining and enhancing it for an even better experience! 🎉



Improving the Student Learning Experience by Helping Teachers Develop and Utilise Chatbots
Project No: QESS/04/2023

Documentation by Ka Wing TAM and Team QESS Chatbot

HOTLINE (852) 3746-0065

EMAIL qesschatbots@cpce-polyu.edu.hk



Facebook – QESS Chatbots
<https://www.facebook.com/profile.php?id=61558478435727>



Instagram
@qesschatbots

Any opinions, findings, conclusions or recommendations expressed in this material (or by members of the project team) do not reflect the views of the Government of the Hong Kong Special Administrative Region, the Education Bureau, any member in the Committee on Self-financing Post-secondary Education (CSPE) and its Sub-committee on Support Measures, and the Secretariat of the CSPE and its Sub-committee on Support Measures.