Chatbot Design: Part 2

The Quiz Bot

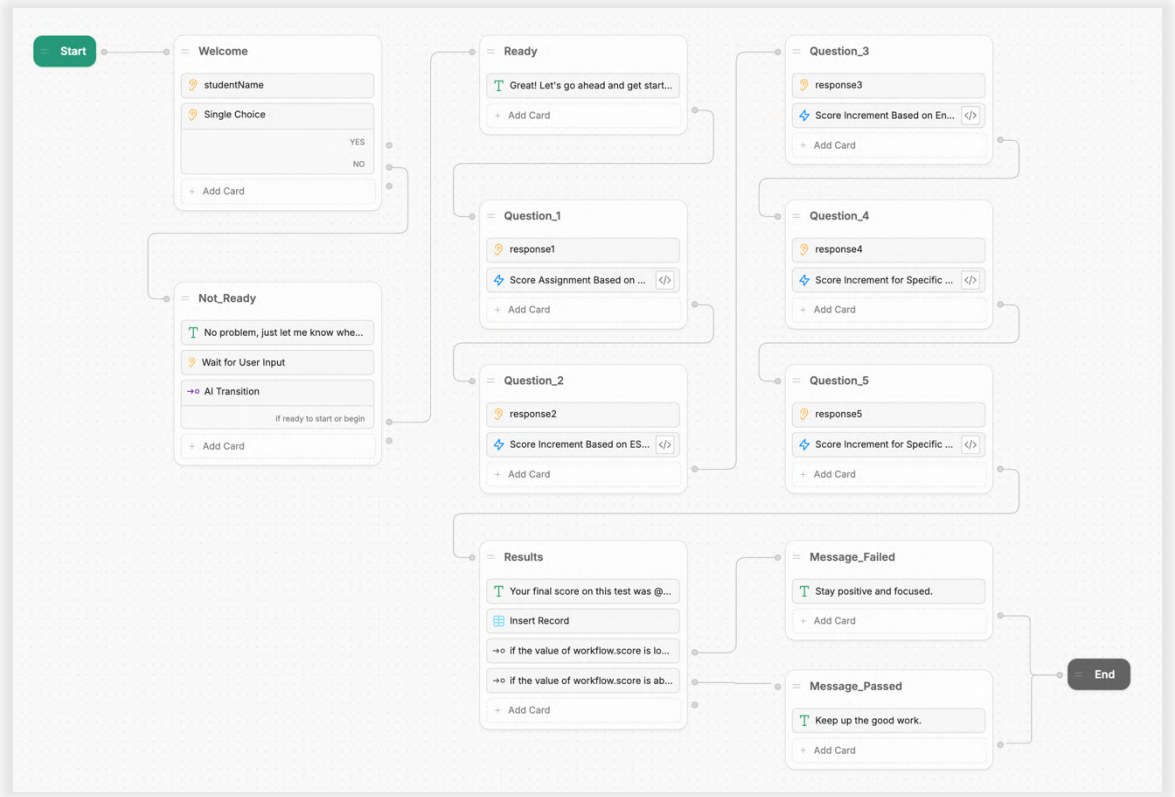
The Quiz Bot is an interactive chatbot designed to guide students through a short multiple-choice quiz in a friendly, conversational style. It begins by greeting the user and capturing their name, then moves on to ask a series of questions—each one evaluated in real-time using logic expressions. At the end, it automatically calculates the score and delivers tailored feedback based on the student's performance.

This bot highlights how chat interfaces can support lightweight assessment in education, making the experience more engaging and immediate for learners.

Key Features

- 📝 Collects student names and responses
- ❓ Delivers up to 5 quiz questions
- 🧠 Uses expressions to track and calculate scores
- 💬 Provides customized feedback based on results
- 🧪 Testable using Botpress Emulator

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



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/ZTwEanOnQ0c.

Meanwhile, our chatbot here focus on routed assessment and evaluation-based feedbacks.

From Support to Assessment 📖 → 📝

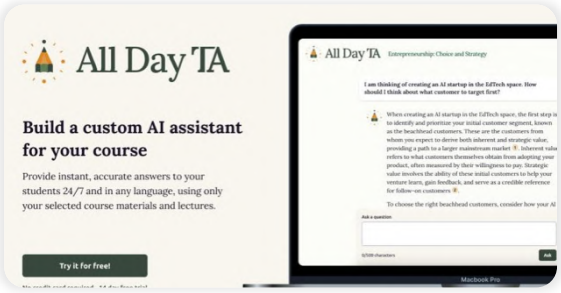
In *Chatbot Design: Part 1*, we explored how Large Language Models (LLMs) enhance chatbots by providing intelligent responses for student inquiries. Now, let's dive deeper into how the same technology can empower educational assessments.

Educational assessment chatbots using LLMs offer numerous advantages:

- **Instant evaluation** of student responses.
- Reduction of teachers' administrative workloads through **automated grading**.
- Enabling personalized learning experiences by identifying students' strengths and weaknesses.

Real-World Examples – LLMs in Action

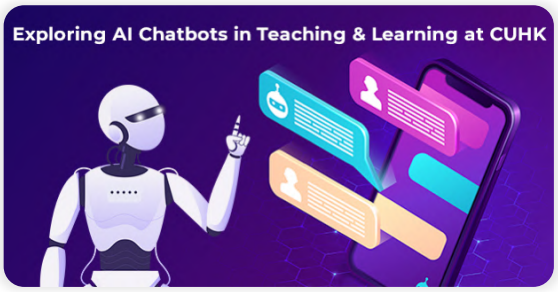
📌 University of Toronto's "All Day TA"



The University of Toronto introduced "All Day TA," an LLM-powered chatbot capable of addressing thousands of student questions instantly and accurately each semester. Beyond answering queries, this AI assistant also supports personalized assessment by instantly evaluating student inputs and guiding them towards resources tailored to their individual learning needs. This approach has dramatically reduced response times and administrative burdens, benefiting both students and educators alike.

Source: Financial Times

📌 CUHK's AI Chatbot for Learning Support



The Chinese University of Hong Kong (CUHK) piloted a chatbot that allows instructors to upload course materials into an integrated knowledge base. Leveraging the power of LLM, this chatbot not only addresses students' academic queries instantly but also enables formative assessment through personalized, real-time feedback on coursework. The adoption has shown significant promise in promoting self-directed learning and reducing instructors' manual tasks.

Source: CUHK Educational Development Team

In our **Quiz Bot**, LLMs help by:

📚 **Understanding Student Inputs**: Effectively capture student names and responses.

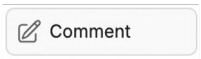
🔍 **Evaluating Answers**: Automatically evaluate quiz responses through logic-based expressions.

🧩 **Customized Feedback**: Offer tailored messages depending on students' performance scores.

🚀 **Scalable Assessment**: Easily deployable for quizzes, homework, or mid-term assessments across multiple classes.

Components

➢ Send Messages / Capture Information

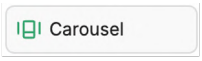


Comment Card

- **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.

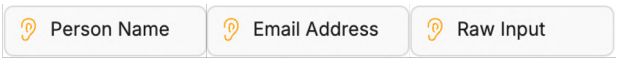
Text/Image/Audio/Video/File/Location

- **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.



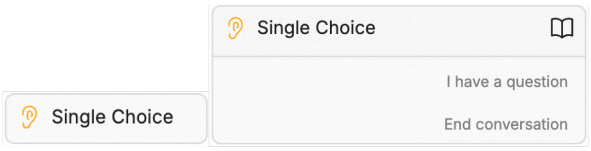
Carousel Card

- **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.



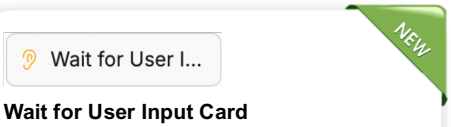
Person Name / Email Address / Raw Input

- **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.



Inputs: Single Choice Card

- **Purpose**: Presents single/multiple-choice options for user navigation.
- **Use Case**: Allows users to select answer, such as “A. Increase productivity,” “B. Climate focused investments,” or “C. Digital Innovation.”



Wait for User Input Card

- **Purpose**: Temporarily pauses the chatbot flow until the user provides additional input.
- **Use Case**: Ideal for scenarios where the user needs time before continuing, such as confirming readiness or responding to an open-ended question.

➢ Execute

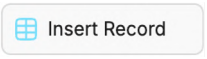
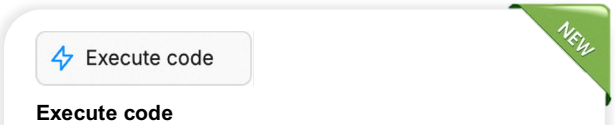


Table: Insert Record

- **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.



Execute code

- **Purpose**: Allows custom JavaScript code execution within the chatbot flow to handle advanced logic or data manipulation.
- **Use Case**: Typically used for advanced scoring calculations, conditional logic, or to manage complex interactions and decisions within the chatbot workflow.

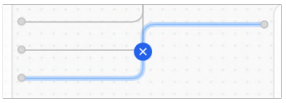
➢ Nodes

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 Node

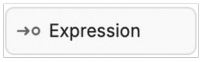
- **Purpose**: Marks the end of a chatbot interaction, terminating the conversation flow.
- **Use Case**: Used when the chatbot has completed its task, ensuring a natural exit for the user or providing an option to restart the conversation.



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.

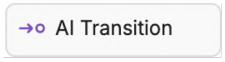
➢ Flow Logic



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.

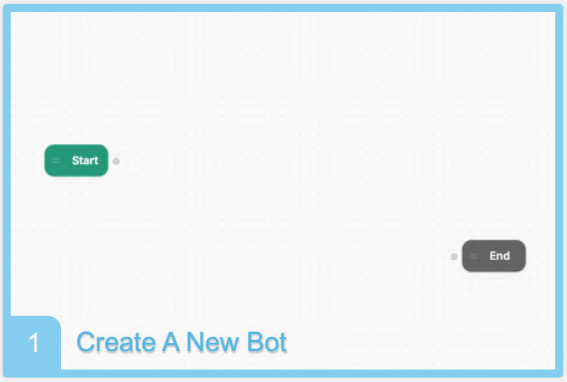
➢ AI

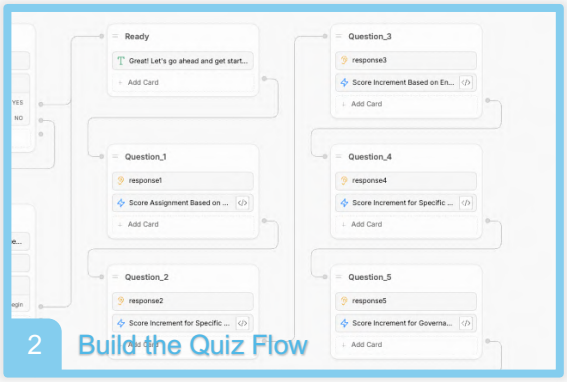


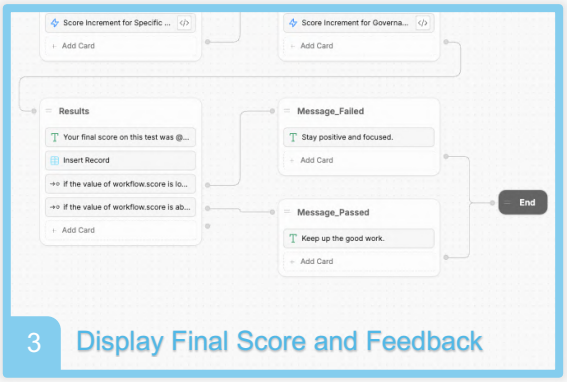
AI Transition

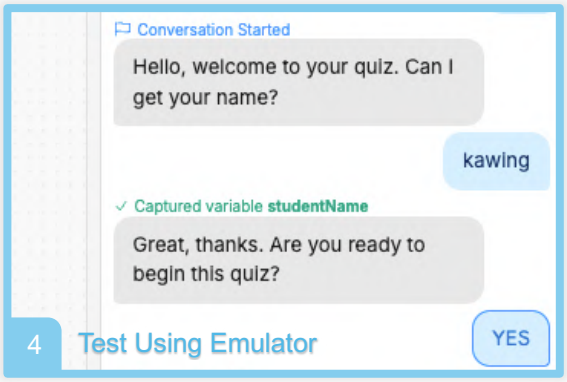
- **Purpose**: Uses AI-based logic to interpret user input and automatically routes to appropriate nodes based on specific keywords or intents.
- **Use Case**: Helps create flexible conversational flows, especially useful in waiting scenarios, to recognize when users express readiness, agreement, or specific commands.

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. **Build the quiz flow.**
   2-A: Welcome and Collect Student Info
   2-B: Wait for Student Ready
   2-C: Present Questions and Update Score
3. **Display final score and feedback.**
4. **Testing the chatbot using the emulator before deployment.**

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

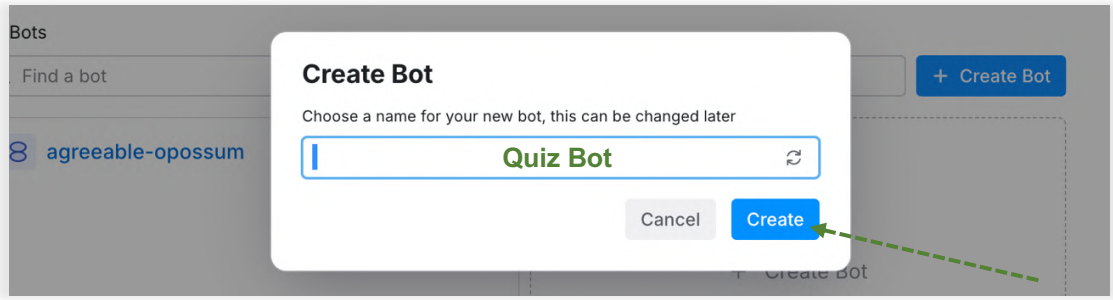
Intro → Section 1

Section 1: Create A New Bot

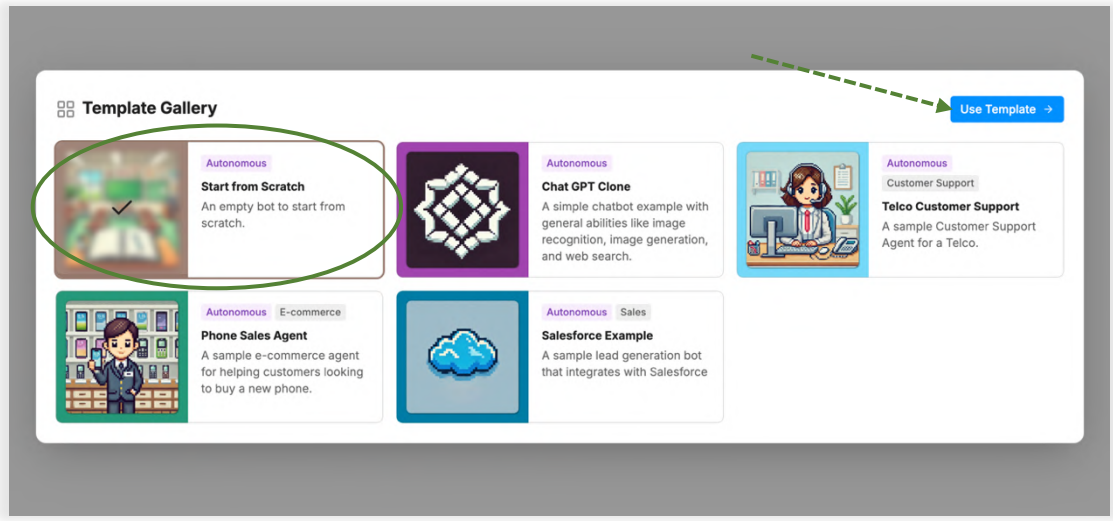
In this section, we'll guide you through creating a new bot project in Botpress Studio. You'll set up a new bot from scratch, and configure it for the quiz, so you can start building the interaction flow.

1-A: Start a New Bot

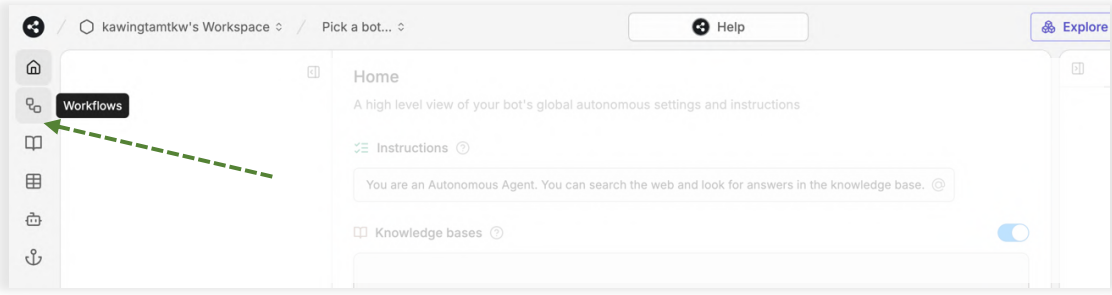
1.1. Open **Botpress** app.botpress.cloud and create a new bot named **Quiz Bot**. Then, click on Open in Studio **Open in Studio**.



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

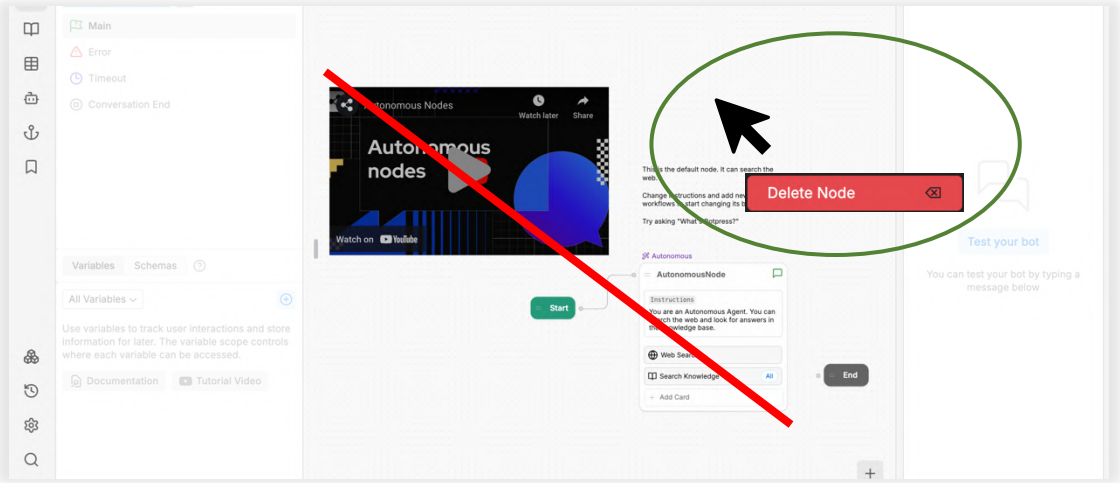


1.3. From the left menu (figure 14), click on **Workflows**.



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 job! You've created the bot and set up the basic flow.

Let's move on to building the quiz functionality! ✏️

1 → **Section 2**

Section 2: Build the Quiz Flow

This section covers how to build the core quiz functionality. You'll create the flow for collecting student information, presenting quiz questions, and updating the score. This will be the foundation of your quiz bot.

2-A: Welcome and Collect Student Info

1. Open your Botpress workflow editor and select the default `Main` flow.
2. Rename the default node to `Welcome` for clarity.
3. Inside `Welcome` Node:

   3.1. Add a Person Name **Person Name** Capture card

   Question to ask the user: "Hello, welcome to your quiz. Can I get your name?"

   Store result in: `workflow.studentName`

   Knowledge Base: Disable Search

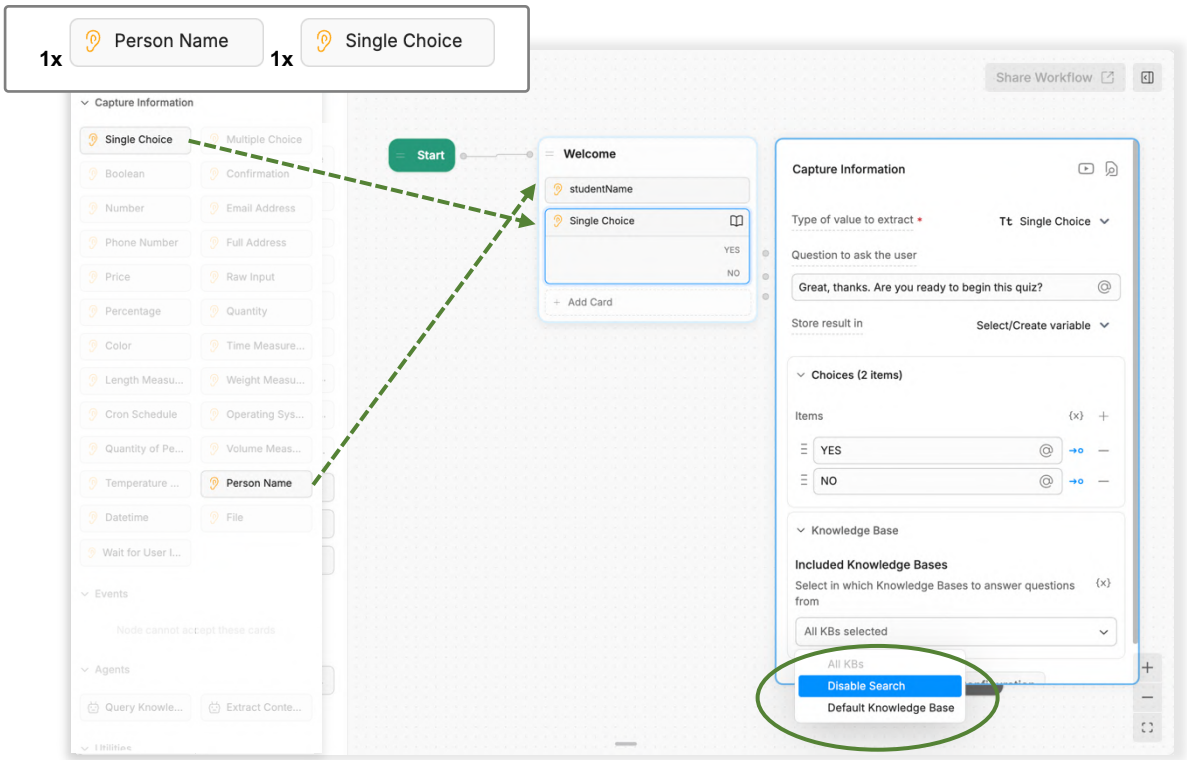
   3.2. Add a Single Choice **Single Choice** card

   Question to ask the user: "Great, thanks. Are you ready to begin this quiz?"

   Choices:

   YES → go to `Question_1`

   NO → go to `Not_Ready`

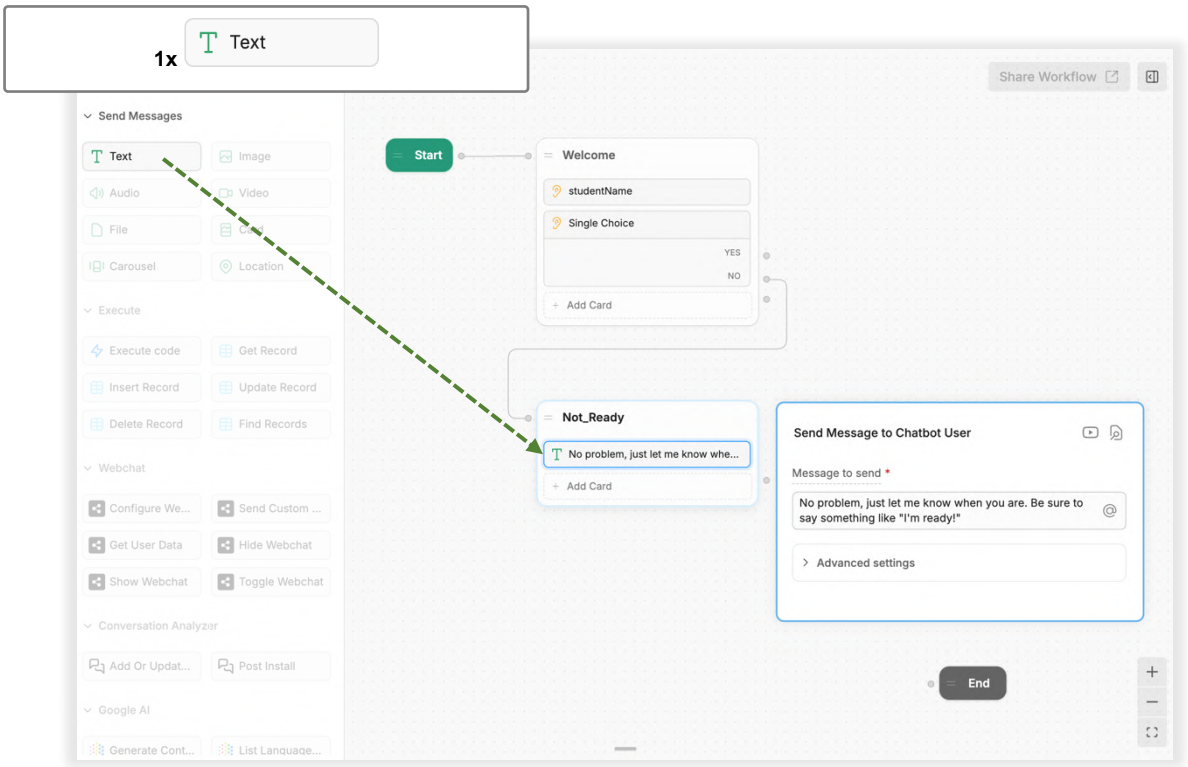


2-B: Wait for Student Ready

1. Create a new node: `Not_Ready`
2. Inside `Not_Ready`:

   2.1. Add a Text **Text** card:

   Message to send: No problem, just let me know when you are. Be sure to say something like "I'm ready!"

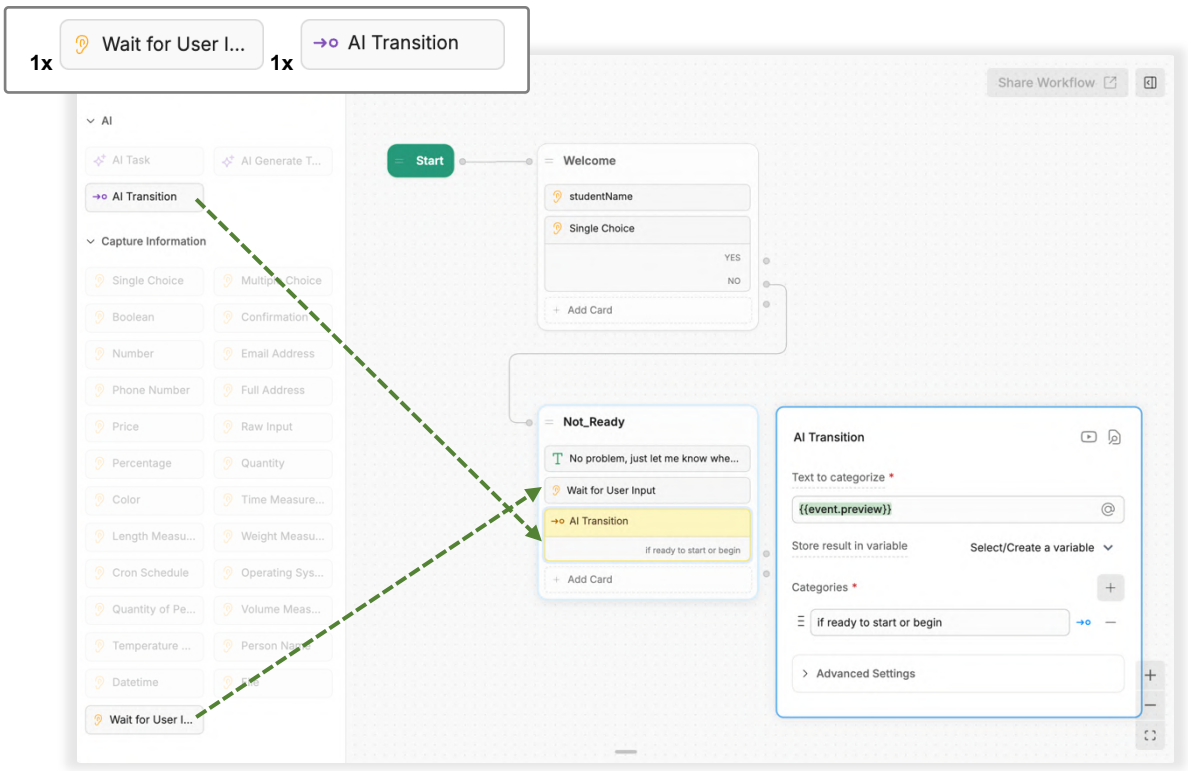


2.2. Add a Wait for User I... **Wait for User Input** card

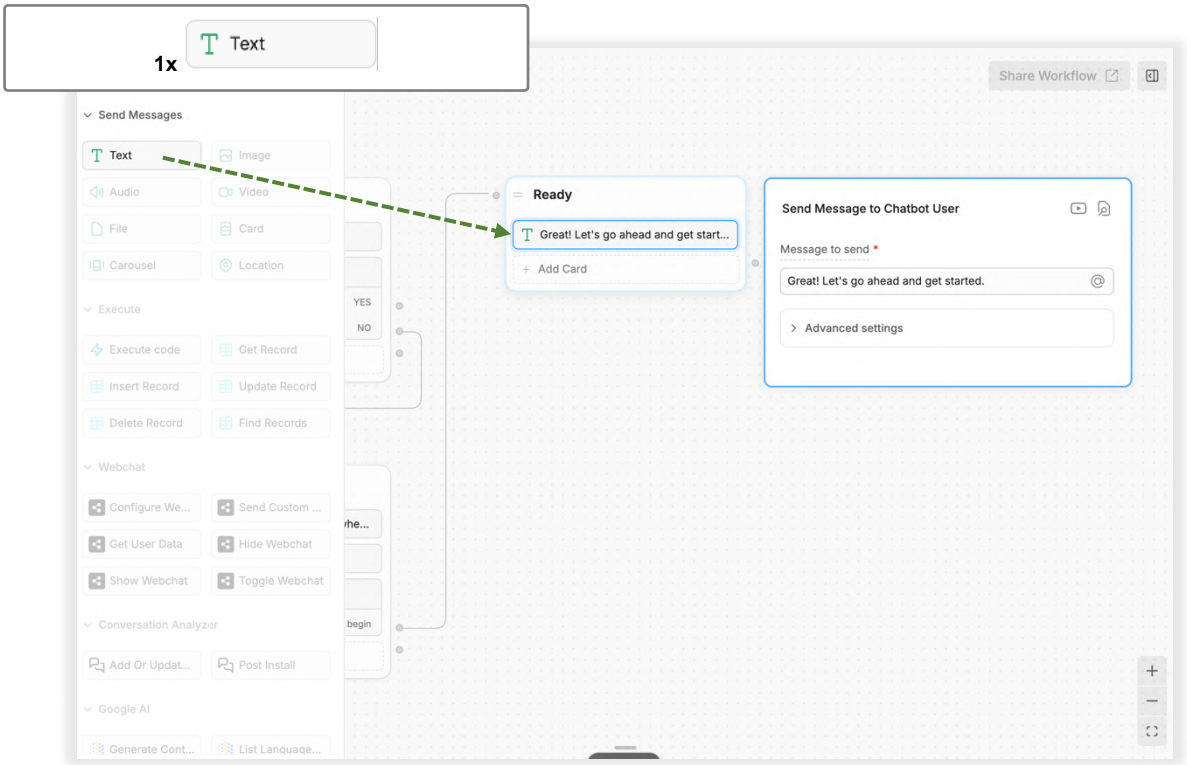
2.3. Add an AI Transition **AI Transition** card

Text to categorize: {{event.preview}}

Categories: if ready to start or begin



3. Create a new node: Ready
4. Inside Ready:
   4.1. Add a **Text** Card:
   Message to send: Great! Let's go ahead and get started.



2-C: Present Questions and Update Score

To present the first question,

1. Create a new node: `Question_1`
2. Inside `Question_1`:

   2.1. Add a Single Choice **Single Choice** Capture Card:

   Question to ask the user: 1. Which one below is not the key driver of ESG adoption?

   Store result in: **response1** ……..……..

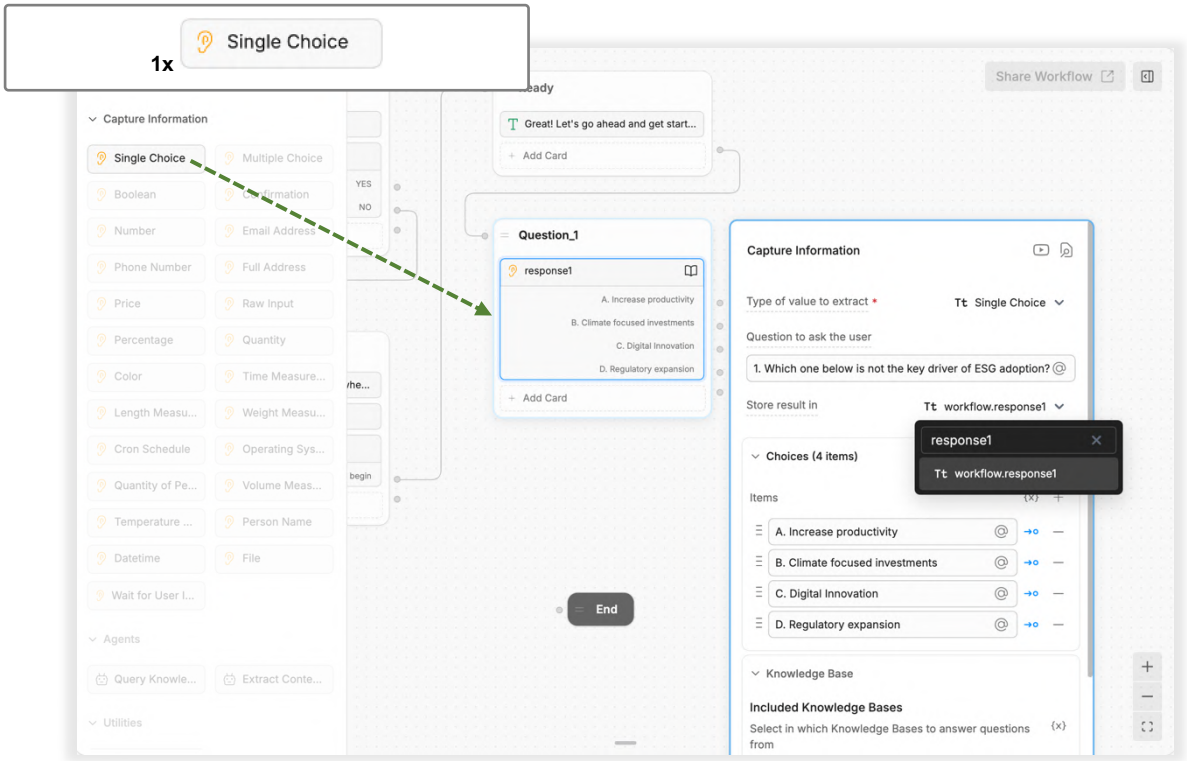
   Choices:

   A. Increase productivity

   B. Climate focused investments

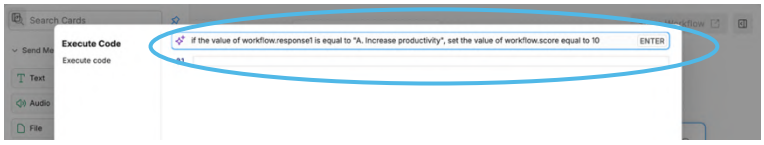
   C. Digital Innovation

   D. Regulatory expansion



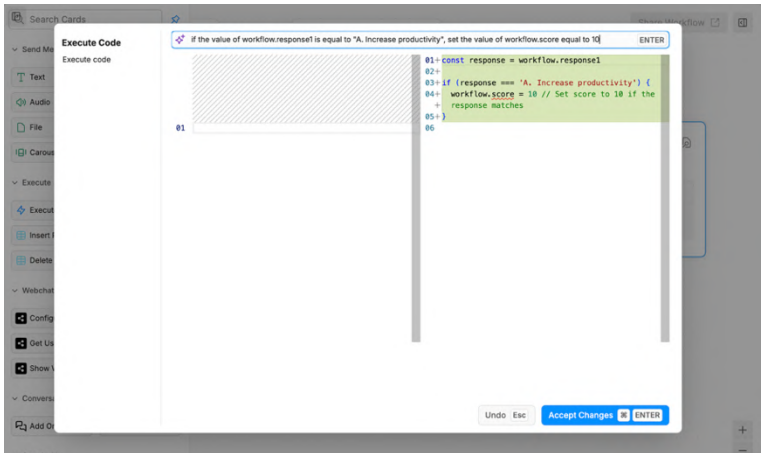
To save the initial score,

2.2. Add a Execute code **Execute code** Card. At the AI editor, type: ***if the value of workflow.response1 is equal to "A. Increase productivity", set the value of workflow.score equal to 10*** and press **Enter**.



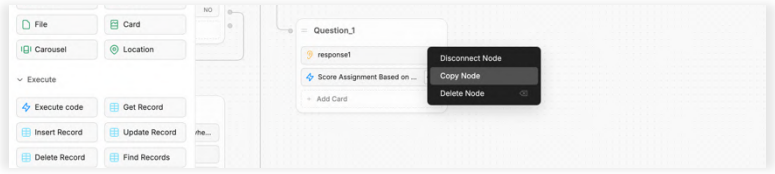
2.3. Wait for the generated code.

2.4. Click Accept Changes ⌘ ENTER **Accept Changes** to use the generated code. Then, click Close Esc **Close** to close the code editor.

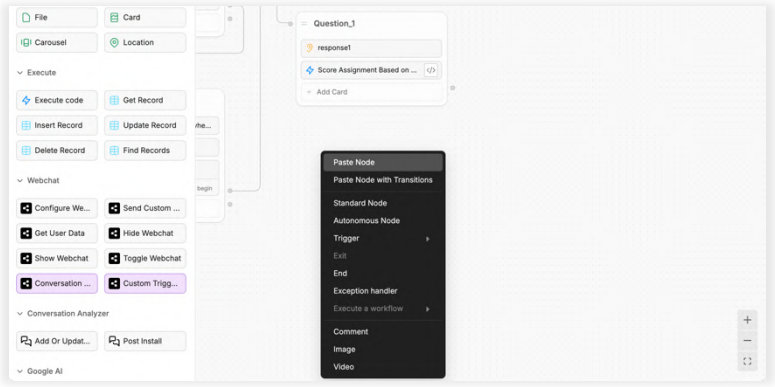


To present the other questions (i.e., `Question_2` to `Question_5`),

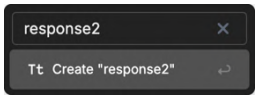
3. Copy the existing node: `Question_1` by hovering on the node. Then **right click** and **select** Copy Node **Copy Node**.



4. On an empty space of the workflow, **right click** and select Paste Node **Paste Node**.

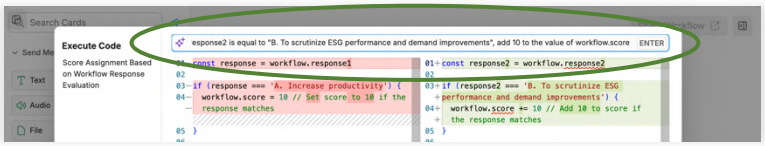


5. **Rename** the copied node to `Question_2`.
6. Inside `Question_2`:
   6.1. Modify the Single Choice **Single Choice** Capture Card:
   Question to ask the user: 2. Social networking sites make it possible for
   Store result in: **response2** ……..……..
   Choices:



   A. Companies to better manage their employee
   B. To scrutinize ESG performance and demand improvements
   C. Companies to keep network information secure
   D. Social gatherings to be monitored

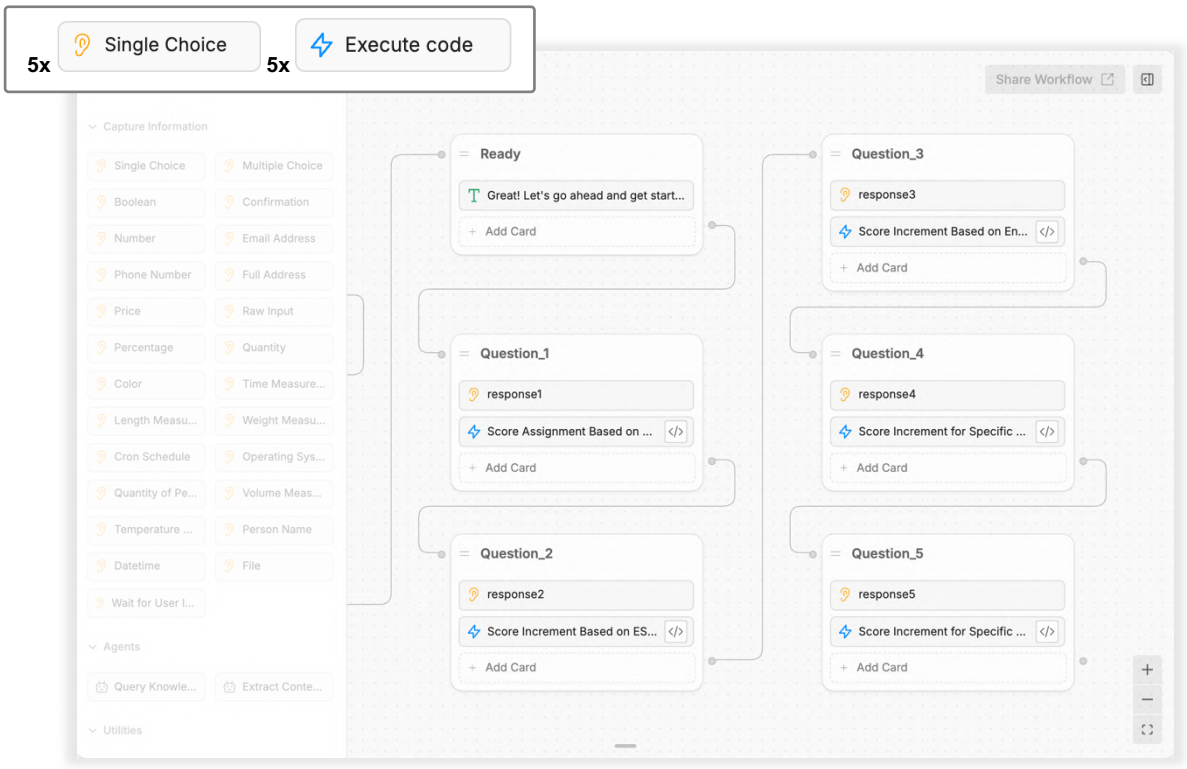
   6.2. Modify the Execute code **Execute code** Card. At the AI editor, type: ***if the value of workflow.response2 is equal to "B. To scrutinize ESG performance and demand improvements", add 10 to the value of workflow.score*** and press **Enter**.



   6.3. Wait for the generated code.

   6.4. Click Accept Changes ⌘ ENTER **Accept Changes** to use the generated code. Then, click Close Esc **Close** to close the code editor.

7. To create nodes for the remaining questions (i.e., `Question_3` to `Question_5`), repeat the above steps 3 to 6. After completion, you should see the workflow like this:



End of Section 2

Awesome! You've set up the quiz questions and score tracking.

Next, we'll display the final score and give feedback! 🎖️

2 → **Section 3**

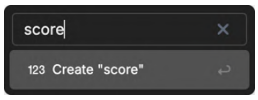
Section 3: Display Final Score and Feedback

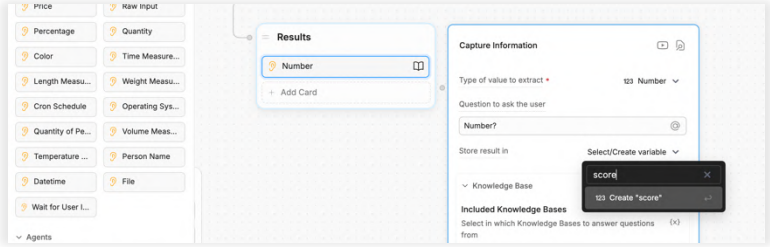
In this section, you will display the final quiz score and provide feedback based on the student's performance. We'll also guide you through setting up conditions to route students to specific responses based on their score.

3-A: Display Final Score

To declare the variable `workflow.score`,

1. Create a new node: `Results`
2. Inside `Results`:
   2.1. Add a Number **Number** capture card:
   Store result in: **score** ……

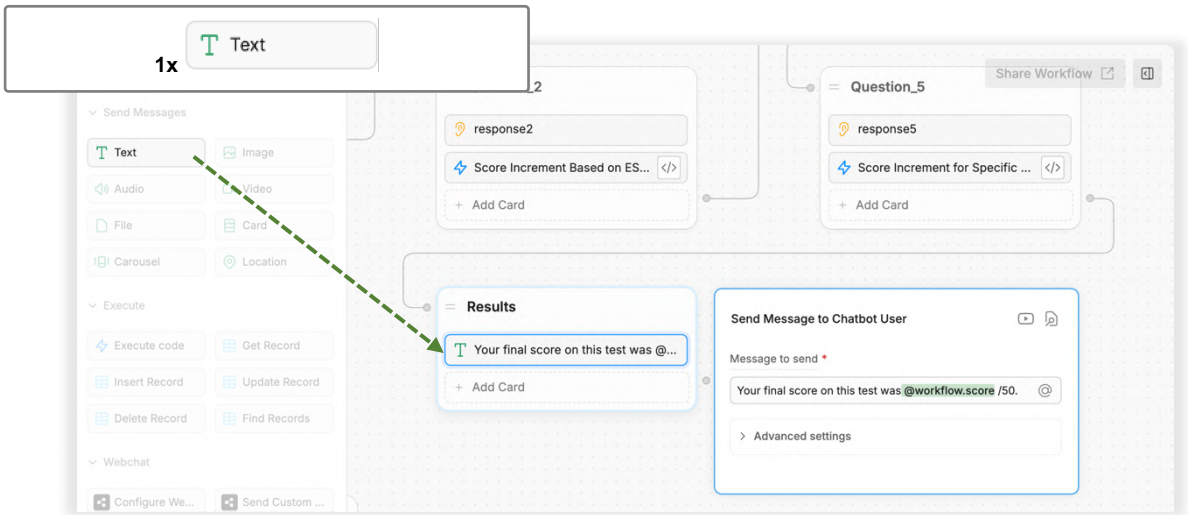




3. **Remove** the score Number capture card.

To display the final score to the user,

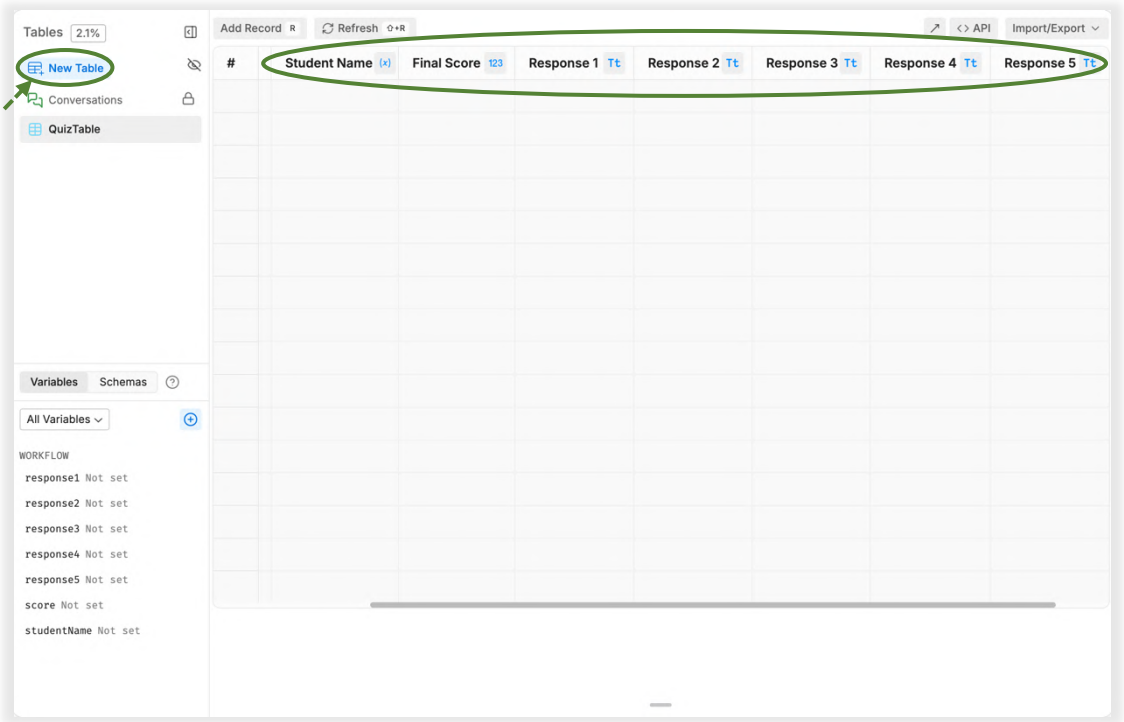
4. Inside `Results`:
   4.1. Add a Text **Text** card:
   Message to send: Your final score on this test was **@workflow.score** /50.



3-B: Setup QuizTable for Storing Student Test Results

Before storing student's quiz result, we need to set up a QuizTable. Follow these steps:

5. Navigate to the **Tables** section in Botpress.
6. Click **New Table** and name it **QuizTable**. **Add** 7 columns:
   - (x) Object **Student Name** (Type: Object, Nullable)
   - 123 Number **Final Score** (Type: Number, Nullable)
   - Tt String **Response 1, Response 2, Response 3, Response 4, Response 5** (Type: String, Nullable)



To save student quiz result to QuizTable,

7. Inside `Results`:

   7.1. Add an Insert Record **Insert Record** execute card:

   Table: Quiz Table

   Response 1: @workflow.response1

   Response 2: @workflow.response2

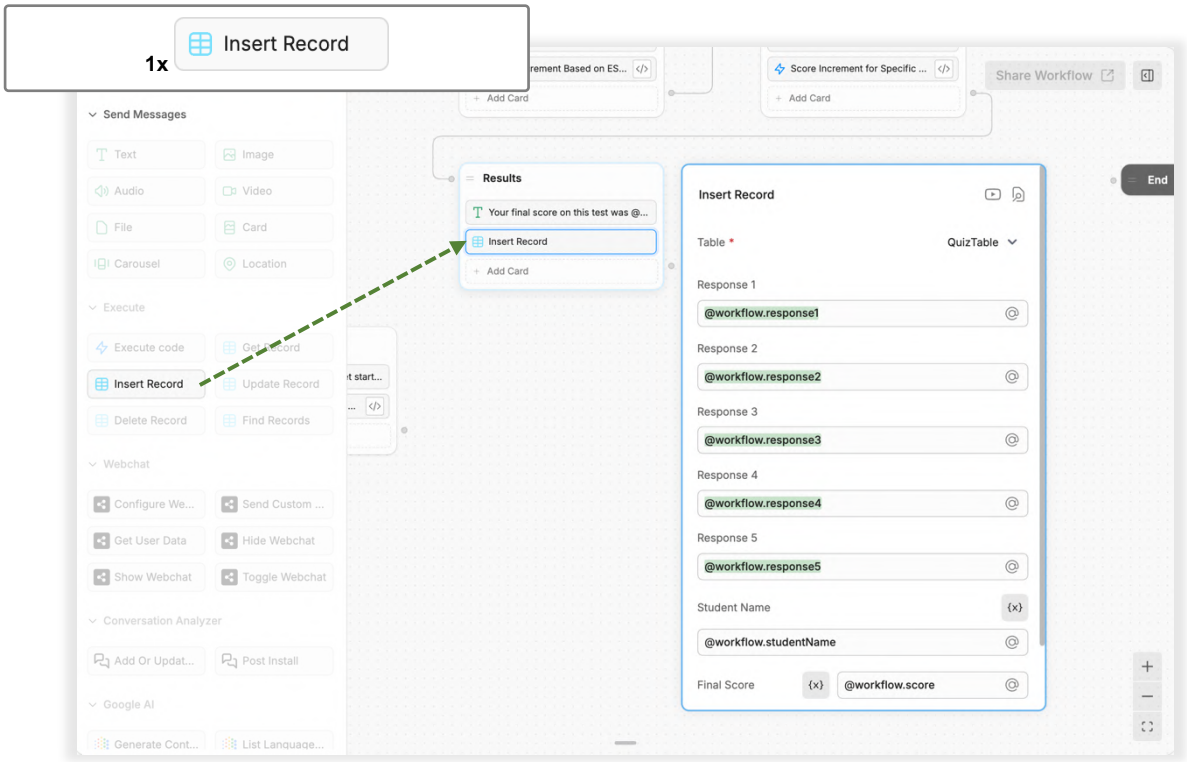
   Response 3: @workflow.response3

   Response 4: @workflow.response4

   Response 5: @workflow.response5

   Student Name: @workflow.studentName

   Final Score: @workflow.score



3-C: Create Conditions for Displaying Customized Messages

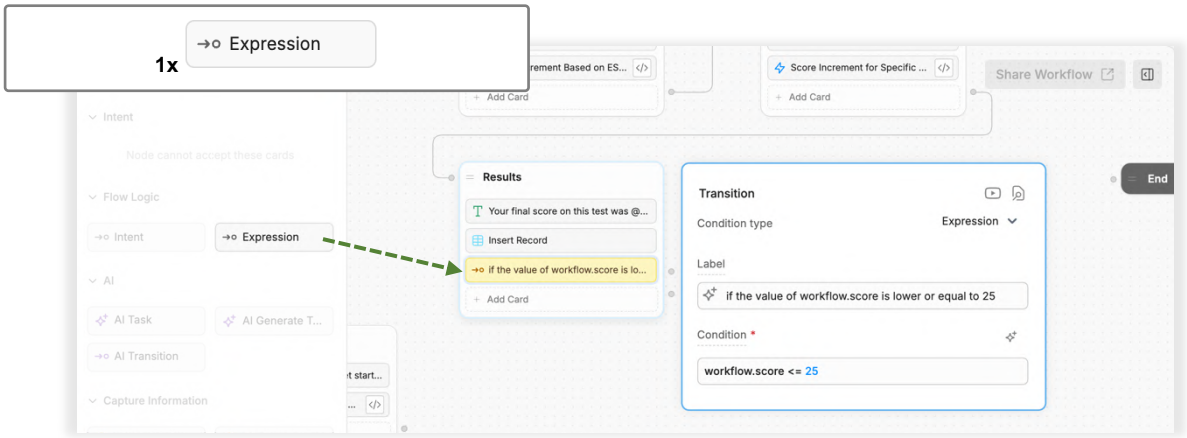
To setup the conditions,

8. Inside `Results`:

8.1. Add an Expression **Expression** flow logic card:

Label: if the value of workflow.score is lower or equal to 25

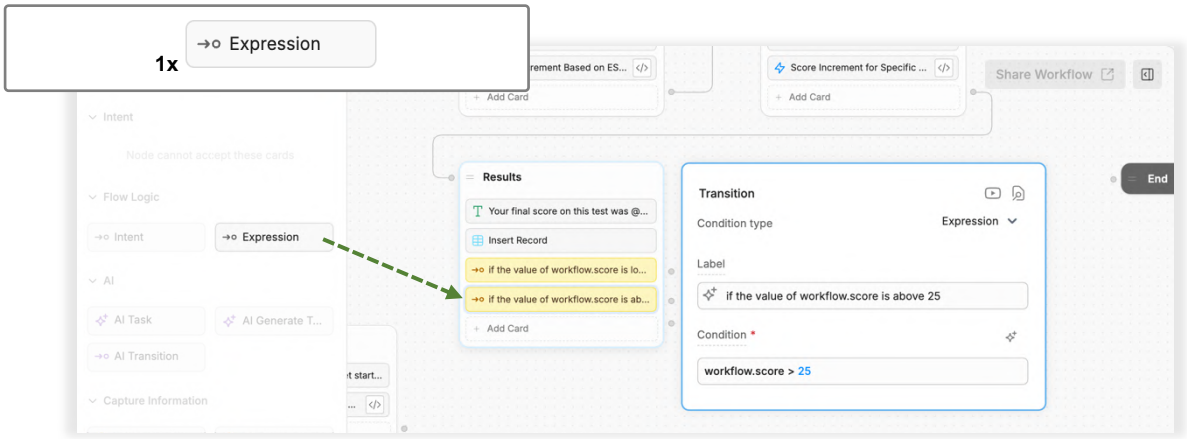
Condition: workflow.score <= 25



8.2. Add an Expression **Expression** flow logic card:

Label: if the value of workflow.score is above 25

Condition: workflow.score > 25

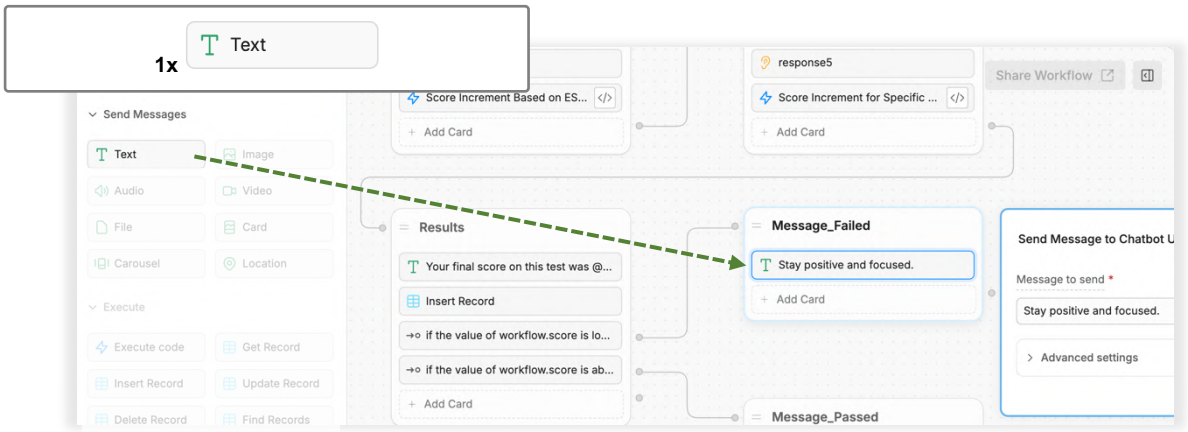


3-D: Return Different Message Based on Mark Value

9. Create a new node: `Message_Failed`
10. Inside `Message_Failed`:

    10.1. Add a Text **Text** card:

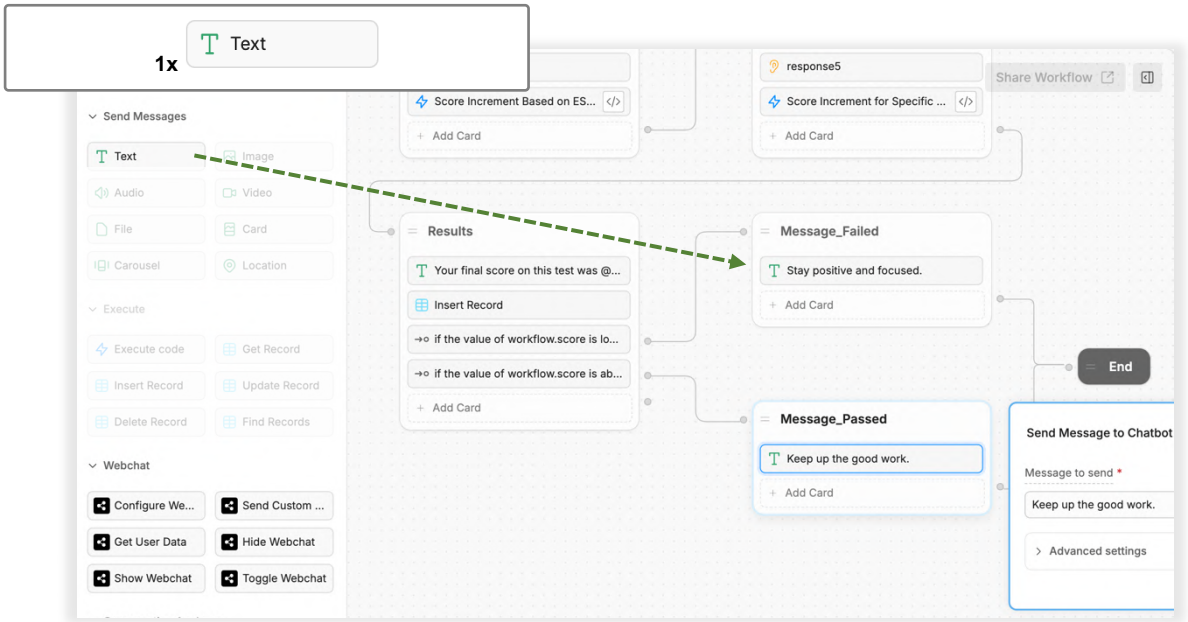
    Message to send: Stay positive and focused.



11. Create a new node: `Message_Passed`
12. Inside `Message_Passed`:

    12.1. Add a Text **Text** card:

    Message to send: Keep up the good work.



End of Section 3

Nice work! Your bot now calculates and shows the score with personalized feedback. 🔍

3 → Section 4

Section 4: Test Using Emulator

This section is all about testing your bot using the Botpress Emulator. You'll test the quiz flow, ensure the logic works as expected, and confirm that the conversation ends correctly. This step is essential to ensure everything is functioning properly before moving forward.

4-A: Run the Emulator

1. Open Botpress and click on the **Test Emulator** button.
2. Start the conversation in the emulator and observe how the flow works from start to finish, checking for any errors or unexpected behavior.

4-B: Validate the Quiz Flow

1. Test different scenarios by entering correct and incorrect answers to verify how the bot calculates the score and displays the final result.

4-C: Check Score Display and Feedback

1. Ensure the final score is shown correctly and that the feedback message displays according to the student's performance (pass or fail).

4-D: Ensure Proper Conversation End

1. After the final feedback, confirm the bot transitions to the End Node and closes the conversation cleanly.

End of Section 4

Great work! Your bot is now fully functional and ready to assess students.

Continue testing and fine-tuning it! 🎉

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