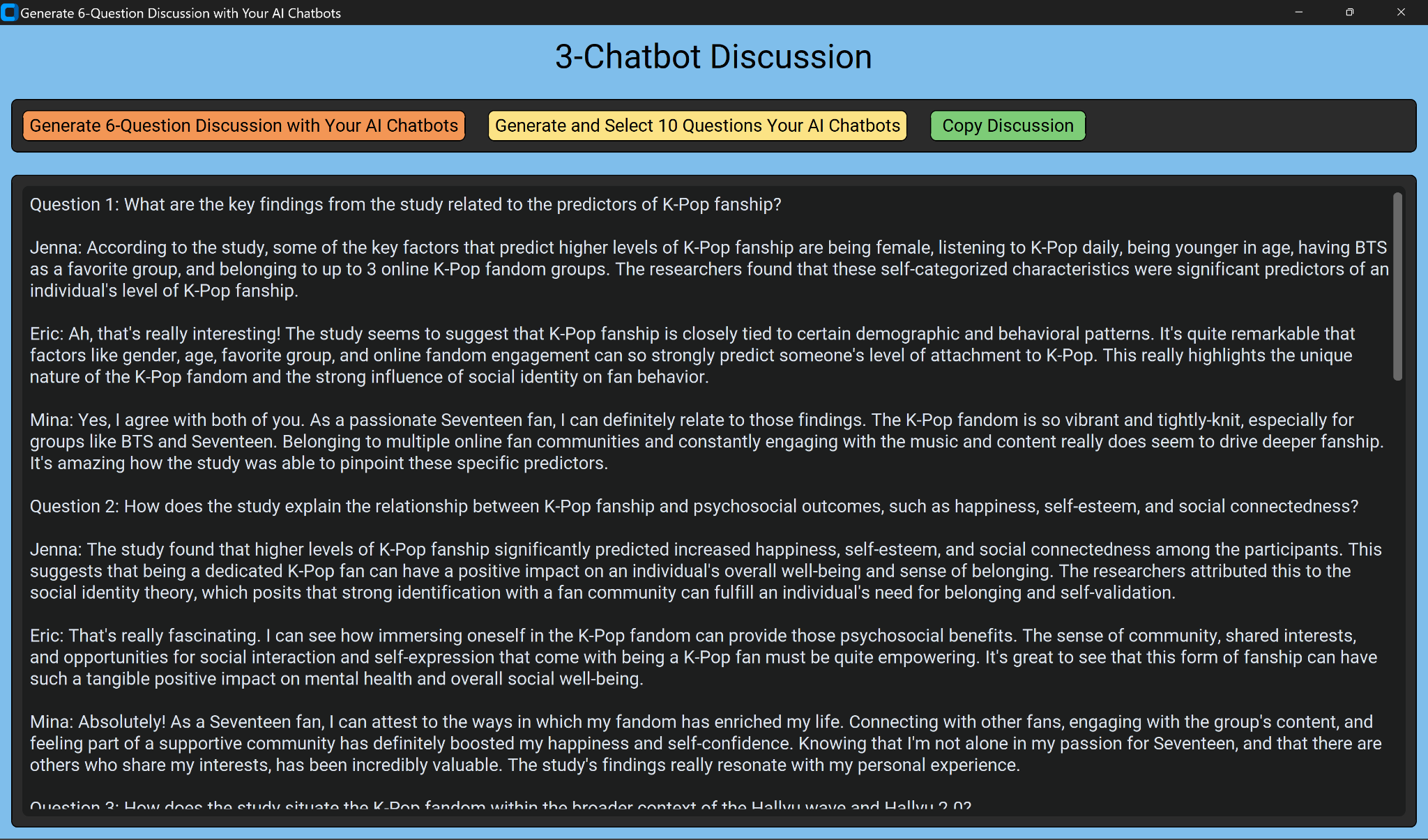
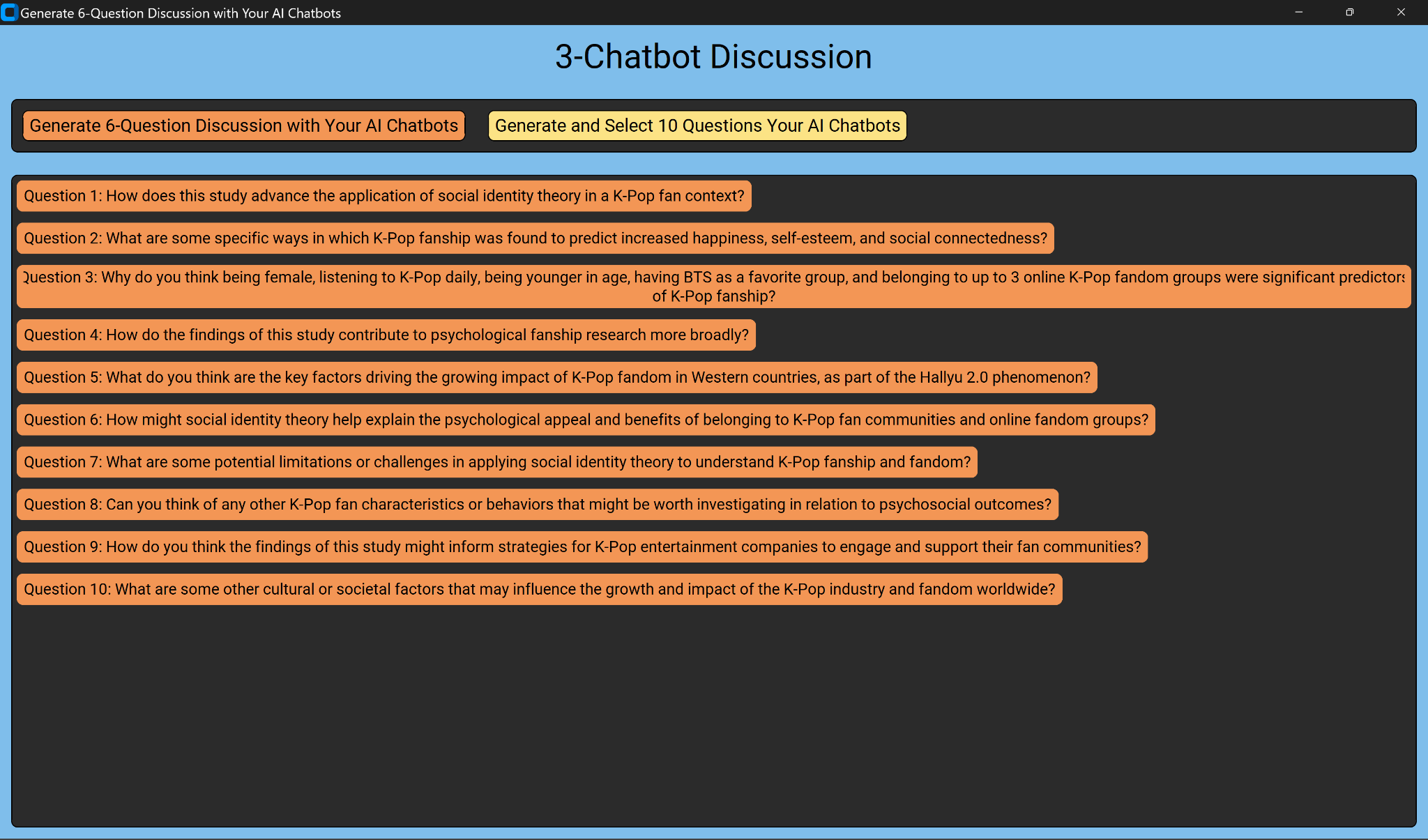
# Role Play Simulation with AI Agents + Creation of Personas

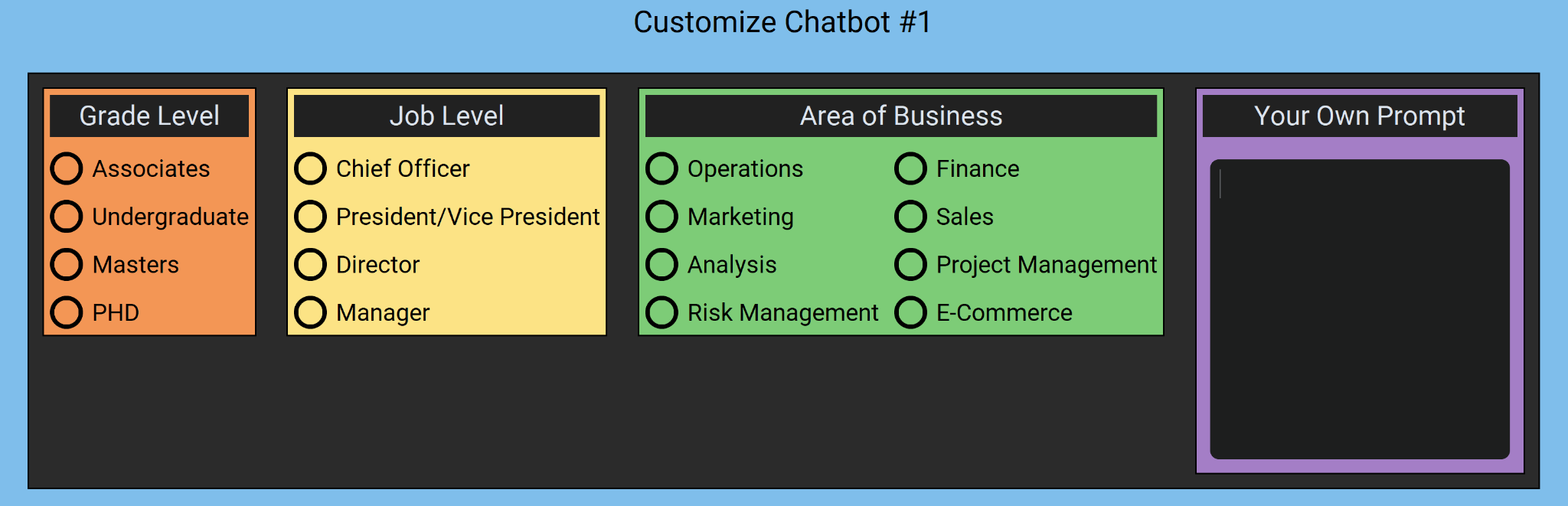
To properly observe the interactions between AI and itself, I created a program that simulates reading-based discussions between AI personas, generates discussion questions, and summarizes PDF files and text as dictated by the user. The user can either generate a 6-question discussion or select one of ten generated questions and generate a discussion based on that question between the AI personas. While Aditya’s program allows the user to communicate directly with the AI, my program does not; instead, the user can simulate discussions between AI personas and study how the AI personas behave with one another. The user can also note the AI’s interpretations of the discussion’s reading and answers to the discussion’s questions that the user did not think of prior. This occurs by having the program compile all the user-selected details of the AI personas, summarize the reading the user wants to generate a discussion on, and send different AI prompts depending on what the user selects; AI prompting will be discussed later on in the paper. As there is a lack of user interaction with the AI, the user holds a more authoritative role rather than a participatory role in this program. This is seen in the user’s ability to customize up to three different AI personas for Claude AI to “host” a discussion between.



*Photo of a generated 6-question discussion between three different AI personas customized by the user based on a research paper studying K-Pop fanship.*

**

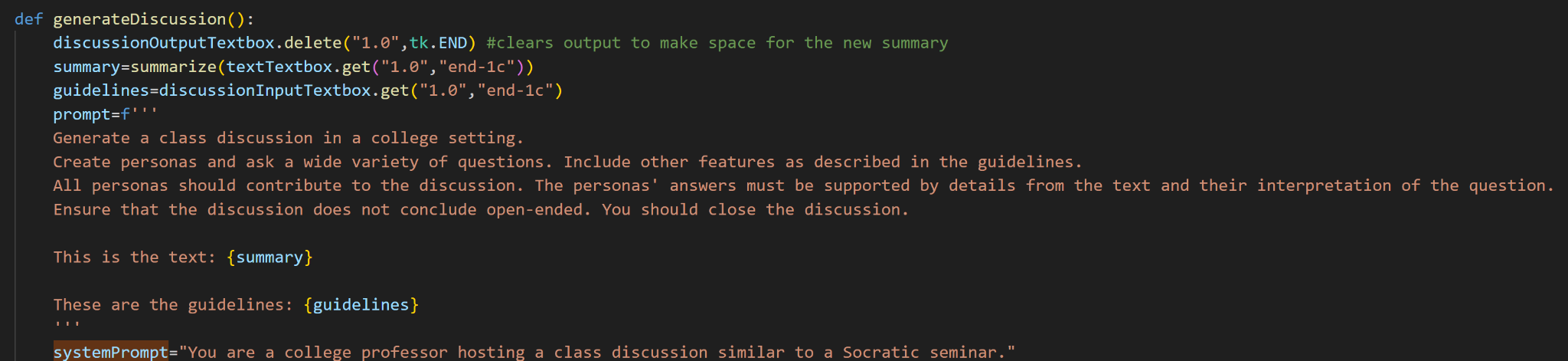
*Photo of ten generated questions for a discussion based on a research paper studying K-Pop fanship. The user can click on one of these questions to generate a discussion based on that question between the AI personas.*

There are a number of options for the user to customize an AI persona. First, the user can select what level of education the AI persona has: associate degree, undergraduate degree, master’s degree, and doctor of philosophy (PHD). Then, the user can choose the persona’s job level: chief officer, president/vice president, director, and manager. Next, the user can select a field of business for the persona: operations, finance, marketing, sales, analysis, project management, risk management, and e-commerce. Lastly, if there are any additional details the user would like to include, they can type such in a textbox named “Your Own Prompt.” The user can repeat this process two more times. While I had originally allowed the user to select multiple of one category (ie. they can allow the AI to have both associate and master's degrees), I changed it so the user can select only one option in each category with the assistance of Aditya. 

*Photo of the customizing menu for one AI persona*

In order for the AI personas to reflect the customizable features the user included, string values are associated with their respective option. When the user selects an option, a variable for that category is set to the option’s string value. That variable is then included in another variable that represents the overall AI persona. The variable’s value is updated whenever the user selects an option to accurately record the user’s customizations. For example, the user decides that their AI persona will have an associate degree. That option has a string value: “associate.” That string value will be assigned to variable “grade1”, which represents the education level of the first AI persona. When the user is done customizing, the program will include “grade1” in a variable named“chatbot1.” “chatbot1” holds all the details about the first AI persona, which are its education level, job level, area of business it works in, and any more additional details. If the user types in their own details, those details are transcribed as their own string value and also set to a variable that represents any additional details for the AI persona; this variable is then included to the variable that holds all the details about an AI persona. If the user does not select any option, an empty string value will be assigned to the respective variables.

# Prompt Engineering + Improvement Case Methodology

AI prompting is the ability to provide certain specifications as to how the AI should generate its outputs. Using Claude’s API, AI prompting is as simple as describing how one wishes to describe the AI’s behavior. In my program, I assigned those descriptions to the “prompt” argument when calling the “SubmitPrompt” function in the program. To further enhance this and refine the AI’s outputs, one can indicate the AI’s specific role to the “systemPrompt” and set that to the “system” variable in the “SubmitPrompt” function. In other words, the “prompt” holds the AI’s tasks while the “systemPrompt” holds the AI’s role. A simplified example of this is asking the AI to generate a class discussion for a college setting in its “prompt” while indicating that it is a college professor in its “systemPrompt.” 

*Photo of an example of AI prompt engineering in “prompt” and “systemPrompt.”*

Giving the AI specific and concise prompts gives it a better idea as to how the AI should provide its output. When conducting AI prompt engineering, it is also helpful to use positive language rather than negative. For example, to tell the AI that it should produce answers within three paragraphs, it is better to say, “Maintain answers to be within three paragraphs,” rather than, “Answers cannot be longer than three paragraphs.” When refining my AI prompts, I noticed that formatting the prompt in multiple lines and grouping relevant details on the same line works better than putting everything in one line. Although this format takes up more space than a one-line prompt would, it helps with the more precise details. However, there are also some notable limitations and challenges in regards to AI prompting for this program.

A limitation of AI prompt engineering is that, while it is extremely important for the user to be specific, they cannot include too many details in the prompt. The longer the prompt is, the more likely the AI can miss important details and restrictions. Originally, the user was able to generate ten-question discussions instead of six-question discussions. However, the results of this were inconsistent. Every other generated discussion would either have less than ten questions or cut-off an AI persona’s answer due to the AI’s output overflowing. This occurred even when I explicitly prompted the AI to always conclude the discussion with ten questions without cutting off any AI’s answers. When experimenting how many questions the AI could handle generating a discussion of, Aditya and I concluded that five to six questions proved to be successful every time. As a result, I limited the AI to generating only six-question discussions while providing the user an alternative: they could choose to have the AI create ten questions and generate a discussion between the AI personas on the question they selected. Both options ensure appropriately-generated discussions each time.