

## **A: EXPLANATION OF CHATBOT FUNCTIONALITIES**

In the given scenario, I am a career advisor for the computer science students of a particular university. Because of a dramatic increase in computer science students, I am no longer able to give one-on-one career advice for each student; thus, I have been tasked with developing a chatbot that can suggest a computing-related career choice for each student that interacts with it. This career suggestion is determined by ascertaining a student's strengths, preferences, and personality traits.

The chatbot I developed meets the given criteria. The chatbot begins by stating its purpose and asking whether or not the user would like to continue speaking with the chatbot (after knowing its purpose). If the user wishes to continue, the chatbot then asks the user whether or not they are currently majoring in computer science. If so, the chatbot asks the user whether or not they enjoy certain activities; depending on the user's answers, the chatbot then recommends one of five computing-related careers for the user (and provides the user with a link where they can learn more about said career) <sup>[1] [2] [3] [4] [5]</sup>. If the user does not enjoy any of the specified activities, the chatbot provides the user with a link where they can read about other computing jobs that may better suit their preferences.<sup>[6]</sup>

If the user is *not* currently majoring in computer science, the chatbot clarifies that it can only provide advice regarding computing-related careers, and gives the user the option to either continue or end the session.

Thus, my chatbot performs the functions it was designed for — it helps students ascertain their strengths and preferences, and then uses these strengths and preferences to suggest a computing-related career that would be a good fit for the student. Additionally, my chatbot is able to act as a substitute for me (a university career advisor) when necessary, thus decreasing my workload and allowing for greater efficiency.

## **B: COMPUTING JOB TYPES**

As delineated in the scenario, I have identified five computing-related careers that the chatbot can potentially recommend to the user (based on the strengths and preferences of that user).

Those five careers are listed as follows:

1. Database Administrator<sup>[1]</sup>
2. Computer Forensic Investigator<sup>[2]</sup>

3. Computer Repair Technician<sup>[3]</sup>
4. Back-End Software Developer<sup>[4]</sup>
5. Front-End Software Developer<sup>[5]</sup>

## **C: CHATBOT CODE FILES**

The chatbot's code files have been included in the task submission (separately from this document) as a ZIP file. The name of this ZIP file is "careeradvicebot.zip".

## **D: CHATBOT TRAINING CASES**

The chatbot was trained in various ways so that it could handle certain outlier cases. Because of the resulting AIML-based enhancements, the chatbot was able to properly handle the following cases. (To clarify, these cases play out this way *after* the training process has occurred).

---

### **Case 1:**

Ruby is a visual arts major who is very much undecided regarding the career she wants to pursue after graduating from college. She finds the career advice chatbot on the university website, and because she doesn't realize that the chatbot only recommends computing-related jobs, she begins to chat with it.

"Are you currently pursuing a degree in computer science?" the bot asks.

"No, I'm majoring in something else," Ruby replies.

"I can only offer advice regarding jobs related to computer science," the bot states. "Do you still want to continue?"

'*Why not?*' Ruby thinks to herself. '*I don't even know what field I want to go into after I graduate.*'

"Yes, I still want career advice!" Ruby tells the bot confidently.

The bot proceeds to give Ruby career advice after determining her strengths and interests.

---

## Explanation of Case 1:

The bot is able to continue with its normal functionality after Ruby states that she isn't a computer science major; this is possible due to an enhancement made to the bot using AIML.

When the bot is used as intended (i.e. by a computer science major), the bot asks "Are you currently pursuing a degree in computer science?" and the user would respond by clicking the button that says "Yes, I'm a CS major." Clicking this button causes the chatbot to receive the input "csDegreeYES" (because that is the postback generated by that reply button; see **Code Sample 1**). Upon receiving the input "csDegreeYES", the bot proceeds by beginning to ask the user about their interests. This occurs because the act of the bot asking the first question about the user's interests is triggered when the bot receives the input "csDegreeYES". In more technical terms, a specific <category> tag is triggered by the input pattern "csDegreeYES"; this <category> tag outputs the question "Do you enjoy working with data and/or databases?" as well as two buttons for the user to respond with (see **Code Sample 2**).

When the bot is used by a person who is *not* majoring in computer science, the interaction goes as follows: the bot asks "Are you currently pursuing a degree in computer science?" and the user responds by clicking the button that says "No, I'm majoring in something else." The bot then states, "I can only offer advice regarding jobs related to computer science. Do you still want to continue?" If the user clicks on the button that says "Yes, I still want career advice!", then the chatbot receives the input "csDegreeYES" (because that is the postback generated by that specific button; see **Code Sample 3**). As previously stated, a specific <category> tag is triggered by the input pattern "csDegreeYES"; again, this <category> tag outputs the question "Do you enjoy working with data and/or databases?" as well as two buttons for the user to respond with (see **Code Sample 2**).

Thus, because of the implementation of enhancements using AIML, the chatbot is able to handle both scenarios appropriately (I have personally verified this by running Case 1 through the chatbot). In essence, the chatbot has the capability to consolidate two separate pathways back into the same pathway using AIML. This can be accomplished because the AIML (specifically, the <pattern> tags) allows the code to behave similarly to the way a "goto" statement would behave in a different programming language.

## Code Sample 1:

```

<category>
  <pattern>
    continueTalkingYES
  </pattern>

  <template>
    Are you currently pursuing a degree in computer science?

    <button>
      <text>Yes, I'm a CS major.</text>
      <postback>csDegreeYES</postback>
    </button>

    <button>
      <text>No, I'm majoring in something else.</text>
      <postback>csDegreeNO</postback>
    </button>
  </template>
</category>

```

## Code Sample 2:

```

<category>
  <pattern>csDegreeYES</pattern>

  <template>
    Do you enjoy working with data and/or databases?

    <button>
      <text>Yes, I love working with data!</text>
      <postback>likesDatabasesYES</postback>
    </button>

    <button>
      <text>No, I'm not a fan.</text>
      <postback>likesDatabasesNO</postback>
    </button>
  </template>
</category>

```

## Code Sample 3:

```
<category>

  <pattern>csDegreeNO</pattern>

  <template>
    I can only offer advice regarding jobs related to computer science.<br/><br/>
    Do you still want to continue?

    <button>
      <text>Yes, I still want career advice!</text>
      <postback>csDegreeYES</postback>
    </button>

    <button>
      <text>No, I'm not interested.</text>
      <postback>notCSmajorAndContinueNO</postback>
    </button>

  </template>
</category>
```

---

## Case 2:

Adam is a student who is unfamiliar with the most common forms of greeting. Instead of saying “hi” or “hello”, he uses colloquial greetings such as “wassup”, “what’s poppin”, and “what’s shakin”.

Adam comes across the career advice chatbot on the university website. Upon starting up the chatbot, he types in “wassup bro??”

The chatbot understands the unorthodox greeting, and responds by introducing itself to Adam and asking him whether or not he would like to continue with the chat session.

Adam proceeds to engage with the chatbot; he comes away with a tailored career suggestion.

---

## Explanation of Case 2:

Because of the enhancements made to the chatbot using AIML, the chatbot is able to understand Adam’s unconventional greeting and proceed with its normal functionality. This is achieved using `<srai>` tags, which allow certain input patterns to trigger the same response as

other input patterns. In this case, many different types of greetings (including more rare, unconventional greetings) are programmed into the chatbot; these sections of code use <srail> tags to indicate that if these input patterns are detected, the chatbot should respond in the same manner as it would when a user enters the common greeting "hello". (See **Code Sample 4** to view a nonexhaustive code snippet containing the discussed enhancements.)

Another enhancement made to the chatbot using AIML involves the caret character ("^"). The caret character is used at the beginning and end of each greeting input pattern in the AIML; this means that the specified greeting word can be anywhere within a sentence, and the chatbot will still recognize that the person is providing a greeting. (See **Code Sample 4** to view a nonexhaustive code snippet containing the discussed enhancements.)

Due to these AIML-based enhancements, the chatbot is able to handle Case 2 appropriately. I have personally verified this by running Case 2 through the chatbot.

#### **Code Sample 4:**

```

<category>
  <pattern>^ HELLO ^</pattern>

  <template>
    Hey there, I'm Charlie the chatbot! I can recommend careers to you
    based on your interests and personality.<br/><br/>

    Would you like to continue?

    <button>
      <text>Yeah, let's get started!</text>
      <postback>continueTalkingYES</postback>
    </button>

    <button>
      <text>No, I'm not interested.</text>
      <postback>continueTalkingNO</postback>
    </button>
  </template>
</category>

<category>
  <pattern>^ HEY ^</pattern>
  <template><srai>^ HELLO ^</srai></template>
</category>

<category>
  <pattern>^ HI ^</pattern>
  <template><srai>^ HELLO ^</srai></template>
</category>

<category>
  <pattern>^ HAI ^</pattern>
  <template><srai>^ HELLO ^</srai></template>
</category>

<category>
  <pattern>^ HEYO ^</pattern>
  <template><srai>^ HELLO ^</srai></template>
</category>

<category>
  <pattern>^ HEYA ^</pattern>
  <template><srai>^ HELLO ^</srai></template>
</category>

<category>
  <pattern>^ WHAT IS UP ^</pattern>
  <template><srai>^ HELLO ^</srai></template>
</category>

<category>
  <pattern>^ WHATS UP ^</pattern>
  <template><srai>^ HELLO ^</srai></template>
</category>

<category>
  <pattern>^ WHAT UP ^</pattern>
  <template><srai>^ HELLO ^</srai></template>
</category>

<category>
  <pattern>^ WASSUP ^</pattern>
  <template><srai>^ HELLO ^</srai></template>

```

---

## **E: CHATBOT INSTALLATION MANUAL**

The following instructions describe how to access/install the chatbot:

1. If you do not have an account on [pandorabots.com](https://pandorabots.com), please create an account before proceeding.
2. Now, please log into your account on [pandorabots.com](https://pandorabots.com).
3. Next, use the following link to access the Pandorabots Bot Directory:  
<https://home.pandorabots.com/dash/bot-directory>
4. After that, type in “CharlieTheChatbot” (without the quotation marks) into the directory’s search bar (which is in the upper right corner of the webpage).
5. My chatbot should be the only search result. Please click on that search result.
6. Next, you will notice a yellow circular icon appear in the bottom right corner of your page. Please click on that icon. This is where you will be conversing with my chatbot.
7. Finally, type in a greeting such as “hey there”, “hello”, “how’s it going”, or “wassup homie”, and press the “Enter” key (on your keyboard) to submit your greeting to the chatbot. This will start up the chatbot and begin a conversation.

## **F: CHATBOT DEVELOPMENT ENVIRONMENT**

### **Strengths:**

- One strength of the chatbot development environment (Pandorabots) is its use of AIML, an easy-to-use markup language reminiscent of HTML. AIML’s simplicity makes creating a chatbot more accessible for a wider range of people.  
  
In addition, AIML allows pattern recognition of user input, which allows the developer to make the chatbot’s responses directly dependent on the input that the user provides. This gives the chatbot a semblance of intelligence, also known as “artificial intelligence”.
- Another strength of the chatbot development environment (Pandorabots) is that account creation is very simple and involves no cost. Again, this allows chatbot creation to be



accessible to a wider array of people — anyone from absolute beginners to expert developers can easily create a chatbot.

### **Weaknesses:**

- One weakness of the chatbot development environment (Pandorabots) is that it does not allow files to be renamed. This is detrimental to the development process because if renaming file(s) becomes necessary, inefficient workaround methods must be used in place of actually renaming of the files.

The easiest workaround method involves creating an entirely new file with the desired name, pasting everything from the old file into the new file, and then deleting the old file. This may seem to be a simple task if the developer wishes to change the name of only one or two files; however, files quickly add up when creating a chatbot, and the described workaround becomes very cumbersome and inefficient when the developer is faced with the task of “renaming” tens or hundreds of files.

- Another weakness of the chatbot development environment (Pandorabots) is that, at least for the free version of the development environment, it is not possible to program the bot to output a message directly upon startup. Instead, the user must be the one to initiate the conversation — they must submit a message containing the proper keyword(s) in order for the chatbot to respond appropriately and begin with its normal functionality.

This is bad form, because in practice, a random user of the chatbot is not going to know the proper keyword(s) to use in order to start up the bot. Even if the proper keyword(s) are provided to the user (outside of the chatbot), this makes for an awkward and peculiar implementation.

Additionally, it is customary for a chatbot to begin the conversation by introducing itself and stating its purpose; however, the development environment in this case (the free version of Pandorabots) does not allow this functionality.

## **G: MONITORING AND MAINTAINING THE CHATBOT**

Once the chatbot is published onto the university’s website, there will be a feature that records every chatbot interaction. These records will be combed through each month, and data from these records will be used to improve the chatbot (by updating the AIML comprising the chatbot). For example, if there are certain greeting phrases that get used often by users but aren’t understood by the chatbot, then the chatbot’s AIML will be updated so that the chatbot *can* understand these commonly-used greeting phrases.

This will improve the user experience because, after the changes are made, the chatbot will be able to better and more frequently understand what users are trying to say. This will lead to less frustration and overall better satisfaction rates among the chatbot's users.

Additionally, there will be a survey for users of the chatbot. After concluding their chatbot conversation, each user will be prompted to complete a survey. The survey will consist of a rating system (e.g. "rate the chatbot out of five stars") and will prompt the user for suggestions on how to improve the chatbot.

The answers to the survey will be reviewed at the same time that the chatbot interaction records are reviewed (in other words, the answers to the survey will also be reviewed monthly). Based on the feedback from the users, certain features may be added to the chatbot (based on how reasonable the feedback is and how common similar feedback is). For example, if multiple users are offended by a particular phrase that the chatbot uses, then this phrase will be replaced with something less offensive.

This will improve the user experience because the suggestions for improvement come directly from the target audience of the chatbot. If multiple people provide similar suggestions for improvement, then there is a good chance that future users would also benefit from the suggested update. Thus, implementing reasonable suggestions from the user base would lead to higher user satisfaction rates.

## **H: PANOPTO RECORDING**

The Panopto video that I recorded demonstrates a walkthrough of the chatbot's capabilities. After recording the video, I saved it to the folder entitled "Intro to Artificial Intelligence NIP1 | C951 (student creators) [assignments]".

If necessary, please follow this link in order to view the video:

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=ee01b0c3-4ba9-4d8c-b4c0-ae8c00299768>

Additionally, the Panopto video will also be included separately in my task submission.

## **I: SOURCES**

- [1] Codecademy Team. "What Does a Database Administrator Do?" *Codecademy*, Codecademy News, 1 Nov. 2021, <https://www.codecademy.com/resources/blog/what-does-a-database-administrator-do/>.
- [2] Coursera. "Computer Forensic Investigator: 2022 Career Guide." *Coursera*, 29 Apr. 2022, <https://www.coursera.org/articles/computer-forensic-investigator>.
- [3] "Computer Repair Technician." *Field Engineer*, <https://www.fieldengineer.com/skills/computer-repair-technician>.
- [4] Codecademy Team. "What Does a Back-End Developer Do?" *Codecademy*, Codecademy News, 12 Nov. 2020, <https://www.codecademy.com/resources/blog/what-does-a-back-end-developer-do/>.
- [5] Coursera. "What Does a Front-End Developer Do?" *Coursera*, 29 Apr. 2022, <https://www.coursera.org/articles/front-end-developer>.
- [6] "Majoring in Computer Science? 100 Careers to Consider." *Chegg CareerMatch*, 14 Sept. 2018, <https://www.careermatch.com/job-prep/career-insights/articles/majoring-in-computer-science-100-careers-to-consider/>.