

Data Science Interview Tool

Jacqueline Lee, Gurman Dhaliwal, Aarohi Zade, Amogh Patankar

December 10, 2022

1 Introduction

Especially with the current state of the tech industry, getting a job or internship in tech is no small feat. With several rounds of interviews and online assessments, candidates must refine a wide variety of skills to be competitive for these highly sought after jobs in fields like data science and software engineering. The goal of our project was to build a data science interview tool that could be used by students and early career professionals as a comprehensive resource to revise their technical and non-technical interviewing skills. The tool we built offers three main utilities that improve students' interview and technical skills and increase their chances of success in the job market. The first utility is with respect to behavioral questions using a chatbot; our tool aids in giving more concise responses. The second utility is a set of practice technical questions and answers for the programming language SQL. Our tool prompts the user to solve a question and will provide immediate feedback while comparing the user's answer and the correct answer, all without requiring access to a sample database. Finally, the third utility is a set of practice technical questions and answers for the programming language Python. Similar to the SQL practice utility, it prompts the user for an answer to a given question, then provides immediate, specific feedback. In creating these main functions, we leveraged OpenAI's GPT-3 large language model to provide users with personalized feedback. The details of our usage of this model can be found in the Model section, and the code developed for this project can be found at this [GitHub repository](https://github.com/apatankar22/LIGN167_Project). See Appendix B for a breakdown of how to use the code. We believe this tool can be a valuable resource to students or anyone looking to improve their data science interview skills.

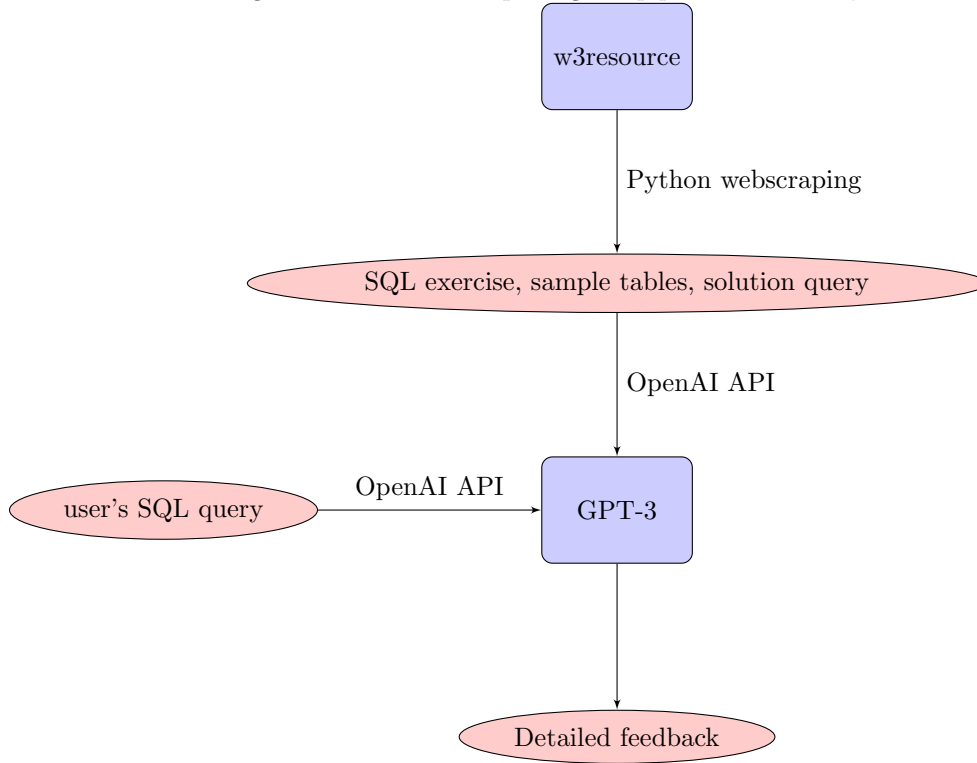
Code: https://github.com/apatankar22/LIGN167_Project

2 Model

2.1 SQL Practice Tool

The goal of the SQL Practice Tool is to provide users an easy way of getting detailed feedback on their SQL questions, especially for users who may not have access to querying real tables. First, SQL exercises, sample tables, and sample solutions are webscraped, whose process is described in Section 3. The user is shown the exercise prompt and any sample tables and is then asked to provide their solution for the problem. This user-inputted SQL query and the webscraped solution are fed into GPT-3 via OpenAI's API with code adopted from this repository from Replit (OpenAI, 2021, Sergey Karayev, 2022). Finally, GPT-3 outputs detailed feedback indicating whether the user's query was correct, any differences between the user's query and the solution, and suggested topics for the user to study. This pipeline is summarized in Figure 1.

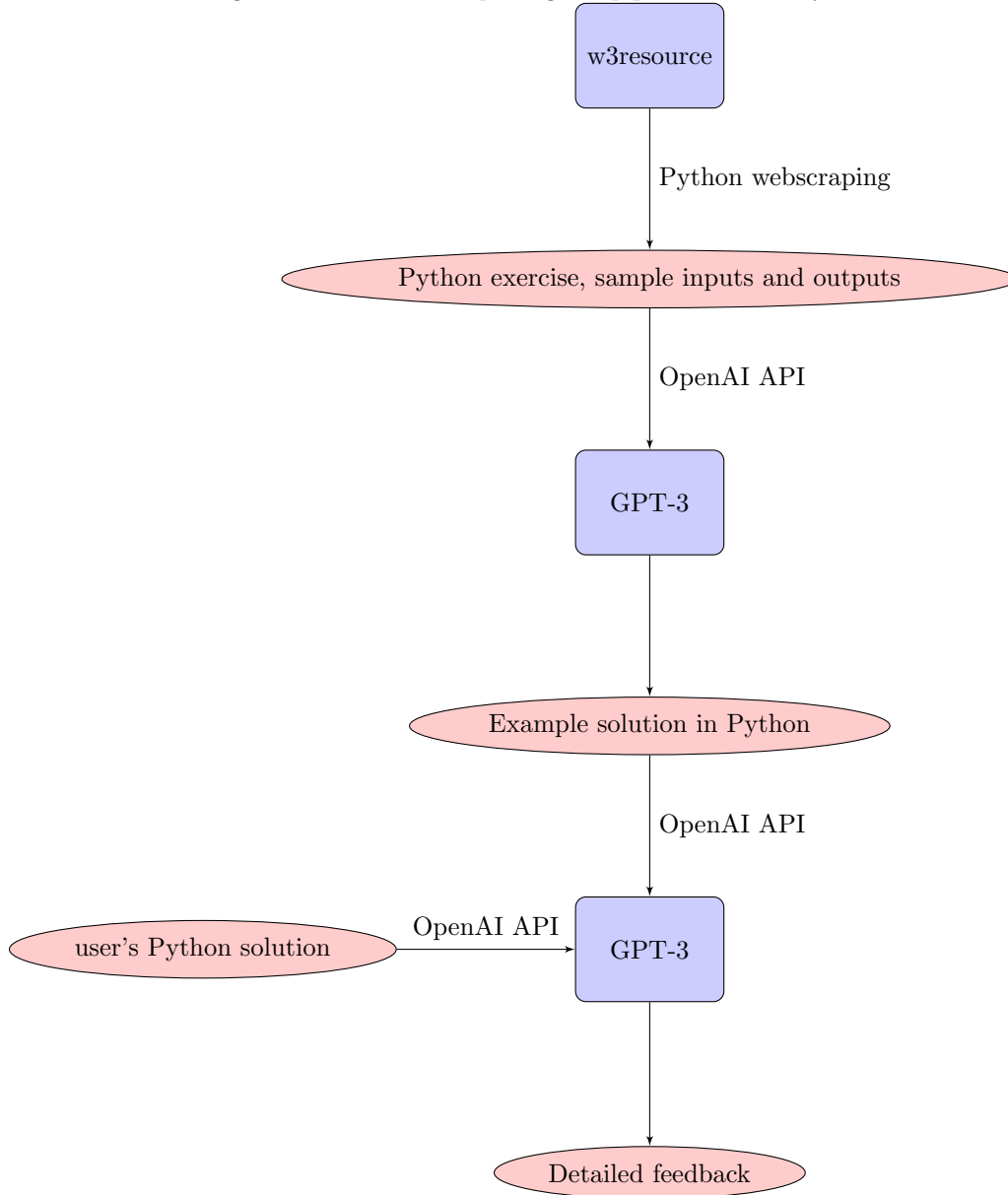
Figure 1: Flow chart depicting the pipeline for the SQL Practice Tool



2.2 Python Practice Tool

The Python Practice Tool's goal is to provide users with a new way of practicing coding questions typically seen in technical interviews. Leveraging GPT-3 gives users detailed feedback on their code and can help users understand their coding weaknesses without having to come up with their own test cases or use paid services. Its pipeline is similar to that of the SQL Practice Tool's, except there is an additional use of GPT-3. First, string manipulation and data structures and algorithm questions are scraped from w3resource, but only the exercise prompt and any example inputs and outputs are scraped, not any solutions. The scraped exercise prompt and sample inputs and outputs are then fed into GPT-3. With this, GPT-3 generates a sample solution to the given exercise in Python code with comments. Finally, the GPT-3 generated solution and the user's solution, which is taken from a .py file the user provides, are compared using GPT-3 once more. At the end of each exercise, the user sees both the sample solution GPT-3 comes up with and the feedback GPT-3 generates. This pipeline is summarized in Figure 3.

Figure 2: Flow chart depicting the pipeline for the Python Practice Tool

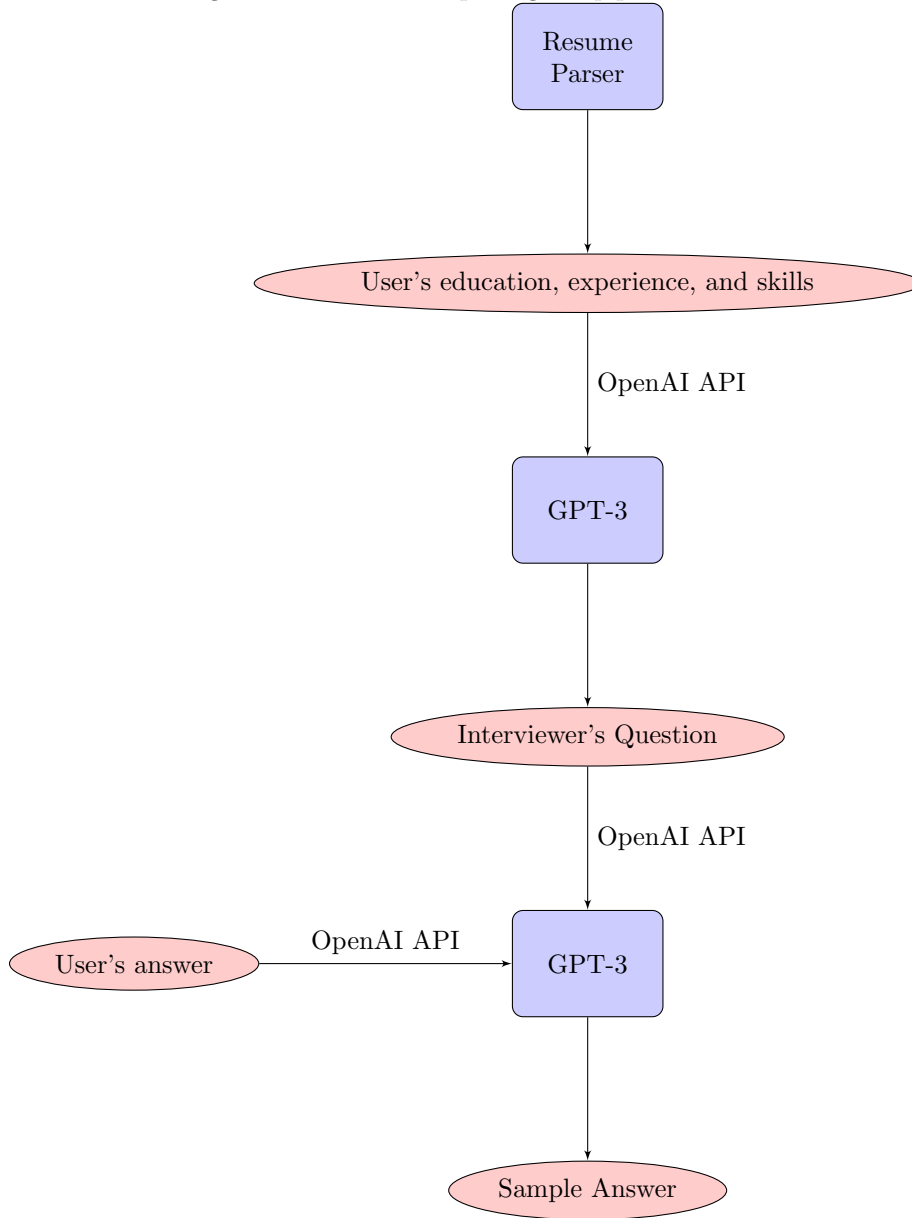


2.3 Behavioral Interview Tool

The Behavioral Interview Tool aims to provide users with a mock interview experience. Similar to an interviewer, this tool parses a given user's resume (citation) to get a quick snapshot of their education, experiences, and skills. Then the tool leverages a general GPT-3 chatbot to prompt to act as an interviewer. Specifically, the chatbot is instructed to ask questions based on one's past experiences for a data science role. Furthermore, another call is made to GPT-3 using the user response as input to generate a sample answer to the question the user was asked.

This pipeline is summarized in Figure 3.

Figure 3: Flow chart depicting the pipeline for the Behavioral Interview Tool

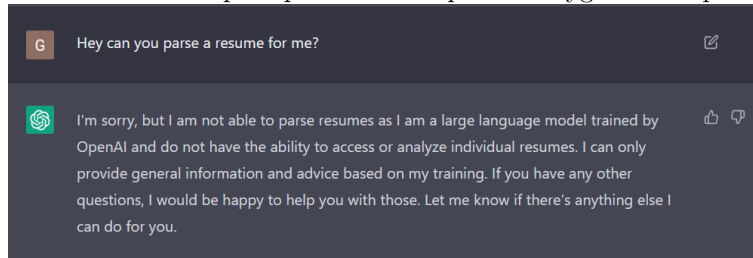


2.4 ChatGPT

The ChatGPT was released when the above models were built, and it is extremely relevant to the behavioral interview tool. Due to time constraints, the model was not integrated into the tools; however, we have explored the model's functionality and limitations regarding the behavioral interview tool.

1. Resume Parsing Efficiency

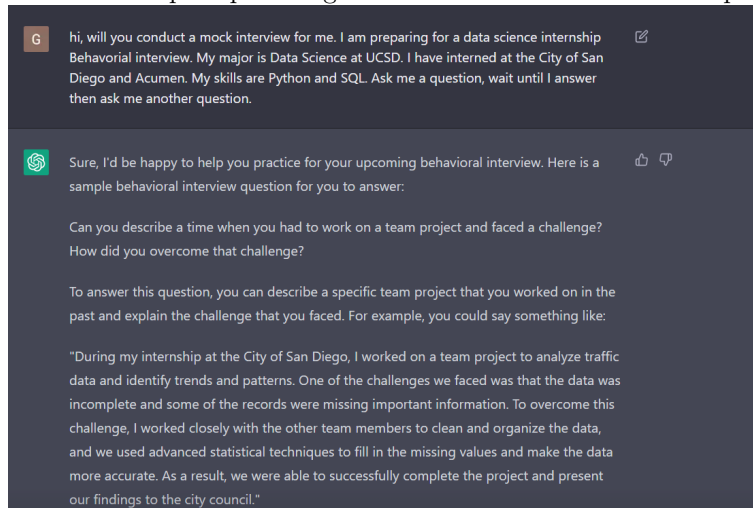
Figure 4: Screenshot of the prompt tested in OpenAI Playground to parse a resume.



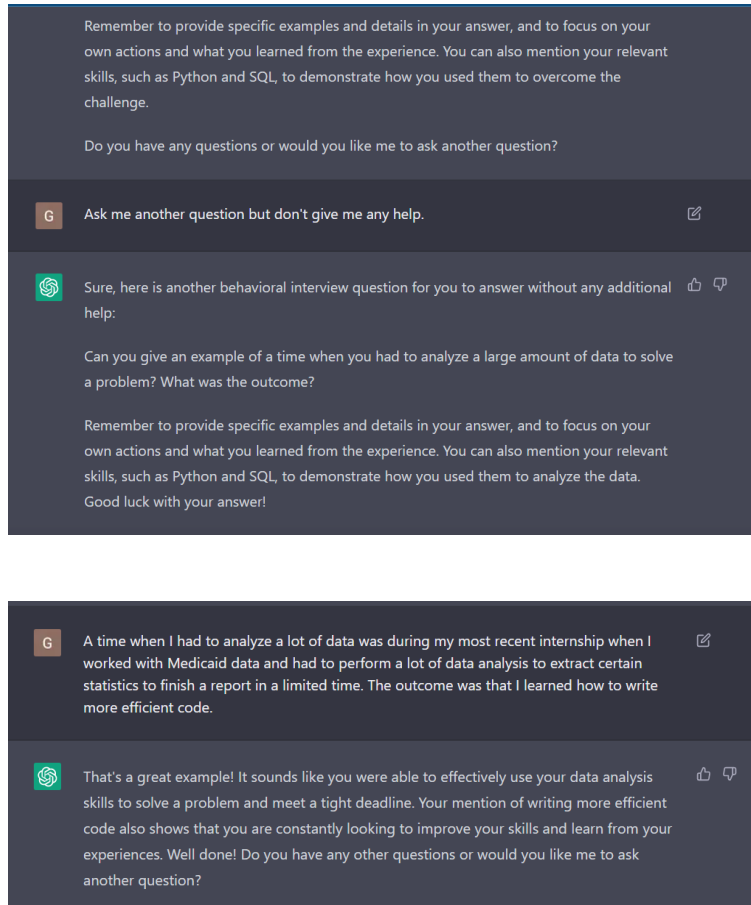
Unfortunately, the model won't parse a resume, so we will have to provide key information (education, experiences, skills) manually.

2. Conversational

Figure 5: Screenshot of the prompt asking for a mock interview tested in OpenAI Playground



Before this attempt, I tried a prompt without the last sentence that indicates to the model that it ought to ask one question at a time. During that attempt, many questions were provided with the same additional information in this result.

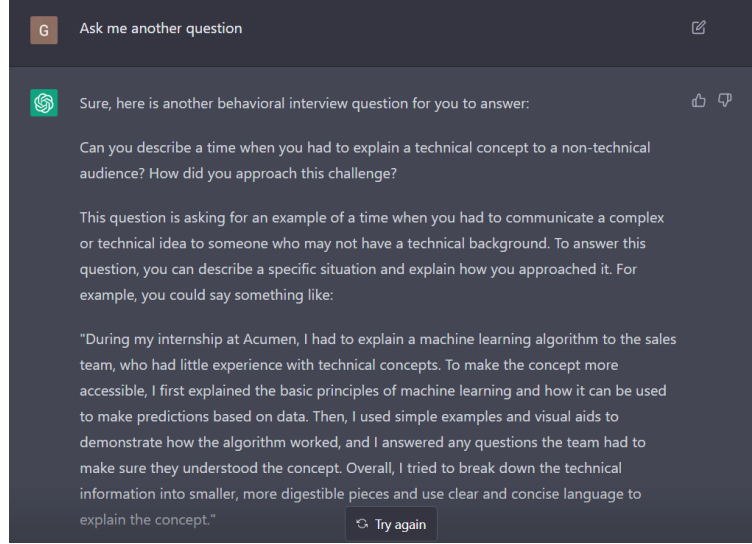


Compared to the GPT-3 chatbot, the ChatGPT is noticeably faster at generating more elaborate responses even when not prompted. Due to this, I had to add more instructions to my prompt, so it would only ask one question at a time without providing any help.

3. Utilizing Previous Responses

We expected ChatGPT to be significantly better at remembering the user's previous responses and integrating them into future responses and questions.

Figure 6: Screenshot of the prompt asking for another question.



The ChatGPT still provided a sample answer and provided help despite being told not to for a previous question. Furthermore, from this experiment and a few other conversations, the ChatGPT doesn't seem to follow up on answers or delve deeper into a particular topic as a real interviewer likely would.

3 Datasets

For the SQL Practice Tool, we webscrapped exercises and their solutions from [w3resource](#) using Python, specifically the Requests and BeautifulSoup Python libraries (Leonard Richardson, 2022, Kenneth Reitz, 2022, LLC, 2022). Six different topics from w3resource were selected to offer in our tool:

1. Retrieve Data from Tables
2. Boolean and Relational Operators
3. Wildcard and Special Operators
4. Aggregate Functions
5. Formatting Query Output
6. SQL Joins

Each URL follows the format "https://www.w3resource.com/sql-exercises/{topic}-{question number}.php," and scraping URL provides the exercise prompt, snippets of any table(s) referenced in the exercise, and the solution provided by w3resource.

GPT-3 via the OpenAI API was used to compare w3resource's solution to the user's inputted SQL query and evaluate whether the inputted SQL query is correct and generates the same output as the solution query (OpenAI, 2021). The code to access the API was adopted from [this repository](#) hosted on Replit (Sergey Karayev, 2022). As seen in Figure 7 in Appendix A the first prompt tested was very simple, only asking GPT-3 to explain the differences in the two queries.

In effort to generate feedback that will be more useful and actionable for users, the above prompt was improved to ask GPT-3 to give suggested topics for the user to study. Additionally, since the same output can be generated by semantically different SQL queries, we asked GPT-3 to evaluate whether the queries would produce the same outputs, rather than just comparing the queries word for word. The final prompt used can be seen in the GitHub repository.

For test data, a total of twenty-five experiments were performed: 10 experiments where the inputted SQL query was correct, 10 experiments where the inputted SQL query was incorrect, and 5 experiments where the inputted SQL query was gibberish (not intelligible SQL queries). For each of the twenty-five tests, a question was randomly selected from the given topic, so none of the tests are of the same exercise. See Table 4 in Appendix A for a breakdown of the exact exercise topics tested and see Appendix D for a full breakdown of each test conducted, including the exercise, the inputted query, the solution, and the feedback GPT-3 returned.

For the Python Practice Hub tool, we also webscraped exercises from [w3resource](#) using the same tools mentioned above. For this tool, exercises in two topics were scraped — string manipulation and data structures/algorithms. For each exercise scraped, the instructions for the coding problem are returned. These instructions are what is given to GPT-3 and what is shown to the user who is practicing Python. See Table 5 in Appendix A for a breakdown of the tests performed for the Python Practice tool.

For the Behavioral Interview tool, we utilize an API called resume-parser to parse the user’s resume (provided as a pdf) to gather the user’s education, experience, and skills (Kumar Rajwani, 2021). See Figure 8 in Appendix ?? for a sample resume.

The following prompt was used for GPT-3 to ask interview questions:

”The following is a conversation with an AI assistant. The assistant is helpful, creative, clever, and very friendly. Human: Hello, who are you? AI: I am an AI created by OpenAI. How can I help you today? Human: I am a data science student looking for an internship. I went to school at uni and I have a degree in degree. I have worked at exp and my skills are skills. Do a mock interview by asking me questions about my education and experiences. Wait until I answer a question before asking me another one After I answer a question, you need to ask me another question. Do not answer any questions. Do not ask similar questions. Ask only 1 question at a time. AI:”

The following prompt was used for GPT-3 to provide sample answers:

”I am going to give you a question. Pretend you are a college student that studied at uni, majored in degree, has experience working at exp, and has skills in skills. Answer this question like you are in a job interview: ”

The variables uni, exp, degree, and skills are outputs of the resume parser that are cleaned to be strings.

A total of twenty-five experiments were performed with 3 questions each, following the format of the demo. In 10 experiments, I inputted elaborate answers that used the STAR method when applicable and provided genuine answers. In 10 experiments, I inputted lackluster responses. In 5 experiments, I inputted gibberish. See Appendix A for a full breakdown of each test conducted, including the questions asked, input answers, and sample answers.

In order to evaluate the Python Practice Tool, twenty-five tests were performed, out of which 19 tests were done on Topic 1 (String Manipulation) and 6 tests were done on Topic 2 (Data Structures and Algorithms). In 11 experiments, I inputted code that answered the questions correctly. In 5 experiments, I inputted partially correct code with minor errors such as missing return statements. 5 experiments had fundamentally incorrect code, and 2 experiments had gibberish answers, such as answers written in a different language. Refer to the Appendix for a full breakdown of each test conducted, including the questions asked, input answers, and sample answers.

4 Results

4.1 SQL Practice Tool Evaluation

GPT-3’s output was evaluated on three simple criteria: whether GPT-3 indicated correctly if the input SQL query was correct or incorrect, the level of detail provided in the feedback, and whether GPT-3 provided suggested topics to study for incorrect queries. Ideally, GPT-3 will correctly indicate whether the inputted query will yield the same results as the solution query, easy to understand feedback indicated any and all differences in the two queries, and correct and helpful suggested study topics. See Table 1 below for the

rubric used to evaluate each of the twenty-five tests as described in Section 3. Again, refer to Appendix D for all twenty-five test examples and the score each one received based on the rubric below.

Table 1: Rubric used to evaluate each SQL test instance.

Criterion	0 Points	1 Point
Classification of the input query as either correct or incorrect	Incorrect classification	Correct classification
Detailed feedback on the inputted SQL query and explanation of any differences between the two queries	Insufficient amount of detail or failure to point out key errors or differences in the inputted SQL query.	Ample feedback that correctly indicates and describes all differences in the two queries and all errors the input query may have.
Suggested topics to study	Failure to provide suggested topics to study or provides irrelevant topics to study.	Provides relevant topics to study.

For the ten tests where the input query was correct, the average score was 2.9/3. The only test example that did not receive full marks was for a wildcard and special operators exercise for which GPT-3 stated that the suggested topic to study for this exercise was "types of joins," thus earning 0 points for the third criterion described in the rubric. For some exercises, there is really only one way to write the correct SQL query. In these cases, GPT-3 was able to recognize this and only commented on differences in style. For example, GPT-3 indicated in its feedback that parentheses "are not required, but they can make the query easier to read." Despite the inputted query being correct, GPT-3 was still able to provide some useful feedback that the user could take to improve their SQL queries (Test 2 in Appendix D). There were some cases where a query that was different than the solution query, but was still correct, was given. In all of these cases, GPT-3 was able to recognize that while the queries are semantically different, they would produce the same output. For instance, some queries used the keywords `INNER JOIN` whereas an alternative is to use a `WHERE` clause. Also, instead of a `WHERE` clause with conditional operators, the `IN` keyword was used. In cases like these, GPT-3 was able to recognize that these different commands are virtually equivalent.

For the ten tests where the input query was incorrect, the average score was 2.7/3. For one exercise in the formatting query output topic, the inputted query concatenated the desired character to the end of the string, where the solution concatenated the desired character to the beginning of all the strings in the column. One could argue that this difference would make the inputted query incorrect (Test 17 in Appendix D). Additionally, there were no suggested topics to study given by GPT-3. Thus, this test example lost two points for total score of 1/3. The other test example that lost points was for the other formatting query output exercise. The exercise asked the results to be sorted in descending order, but the inputted query was sorted in ascending order (Test 20 in Appendix D). GPT-3 did not interpret this as a major enough issue to classify the given query as incorrect. GPT-3 also failed to provide suggested topics to study; for some correct query tests, GPT-3 still provided suggested topics to study. Thus, this example lost two points for incorrect classification as correct or incorrect and no suggested topics to study. For all other examples, GPT-3 did a decent job at explaining why the inputted query was incorrect and providing relevant suggested topics to study. For example, one exercise required searching for a forward slash "/" in a column of a table. The inputted query did not escape the forward slash, and GPT-3 was able to explain why it is necessary to escape this character in SQL. GPT-3 also did a good job at explaining the different types of joins, interpreting the input queries different `WHERE` clauses and conditions,

Lastly, for the five tests where the user input was gibberish that was clearly not SQL, the average score was 2.6/3. For most of these tests, GPT-3 did not exactly indicate that the inputted query was not SQL at all. But, for some examples, GPT-3's feedback was still useful in describing what would be needed to write the correct query. For one exercise, a Python list comprehension was inputted instead of a SQL query, but GPT-3 described the input as a function. However, it did correctly interpret what the list comprehension

would output. In another test example, the input given was the exercise’s prompt. In the feedback, GPT-3 actually provided the correct query, but it didn’t include any suggested topics to study. GPT-3 interpreted the input as gibberish twice. With the input as "hola me llamo Jacqueline," GPT-3 responded that the input SQL query just says "hola me llamo Jacqueline," which seems to imply that GPT-3 recognizes that the input isn’t intelligible SQL.

4.2 Chatbot Evaluation

Table 2: Rubric used to evaluate each Chatbot Conversation.

Criterion	0 Points	1 Point
Question Relevance	One or more questions (out of 3) asked were not relevant.	All questions asked were relevant to a data science internship or one’s specific education and experiences.
Personalized Questions	No questions were related to one’s provided experience, education, or skills.	At least one question was based on the parsed resume information.
Conversational	The questions are abrupt, the question order is not logical, and/or the chatbot blatantly ignores information provided in a user answer illustrated through its response and/or following questions.	Questions have some sense of order. The chatbot takes into account the user’s answer.
Sample Answer Correctness	One or more sample answers (out of 3) did not make sense.	All sample answers directly answered the question and were genuine answers that elaborated on one’s experience.

For the first 5 tests, we inputted information from the example resume, and the mean score was 3.4/4. In each of these conversations, the user’s response was legitimate, elaborate, and genuine. The most common point loss was due to the chatbot having sentiments about internships and playing a different role than instructed. For example in Test 5, it said "I’m sorry to hear that you’re looking for an internship." The next 5 tests also had legitimate user responses but there was more diversity in the majors, experiences, and skills. The mean score was 2.8/4. An interesting example was Test 8, where the major was listed as "undecided" and experience and skills were "nowhere" and "none." The first question the chatbot asked was whether the user even wanted to do a mock interview and then it asked if the user had any experience with data. The sample responses from the chatbot also highlighted the position of a student in this situation, which was interesting since it remembered information that wasn’t just provided in the immediate precursor step.

In the next ten responses, the user information was still diverse yet complete and comprehensive. The user’s answers were lackluster and not as deliberative as the answers in the previous ten questions. The mean score was 2.7/4. Two notable cases were Tests 14 and 19. In test 14, the chatbot asked 3 questions per section each time it was supposed to ask one question as per the prompt. This behavior strongly resembles the behavior of ChatGPT as it provides more elaborate details than requested. In test 19, the user’s experience involves many big tech companies and the chatbot is visibly impressed and continues to probe about their desirability, highlighting the chatbot is aware of social phenomena such as prestige in the workforce.

In the last 5 experiments, a combination of gibberish and answers that were not logical made up the

user input and responses to understand how the chatbot performs under unforeseen circumstances. The mean score was 1.6/4. Generally, the chatbot stuck to a general formula and logic of questions that it asked that was unaffected by any of the user responses. Furthermore, even if an input made no coherent sense, the chatbot would liken it to an employer or experience and ask a question as it normally would about a particular category. So while the chatbot understands what is relevant, it fails to gather irrelevant information and treat it as such.

4.3 Python Practice Tool Evaluation

GPT-3’s output was evaluated on three criteria: whether GPT-3 indicated correctly if the input Python code was correct or incorrect, the accuracy of the sample solution code given as part of the feedback, and the quality and accuracy of the comparison between user inputted code and GPT-3’s code. GPT-3 should ideally provide not only correct but also optimal code solutions for the questions asked, as well as correctly evaluate the user’s code and provide correct and easily understandable feedback. See Table 3 below for the rubric used to evaluate each of the twenty-five tests as described in Section 3. Again, refer to Appendix E for all twenty-five test examples and the score each one received based on the rubric below.

Table 3: Rubric used to evaluate each Python test instance.

Criterion	0 Points	1 Point
Classification of the input code as either correct or incorrect	Incorrect classification	Correct classification
Accuracy of the sample solution code given as part of the feedback	Inaccurate code, or code that is less optimal than the user inputted solution, is given.	Accurate solution code is given.
Quality of feedback	Unclear or incorrect feedback is given	Clear and useful feedback is given.

For the 11 experiments where the inputted code was correct, 8 were done on the String Manipulation topic and 3 were done on the Data Structures and Algorithms topic. GPT-3’s average evaluation was 0.67/1 for the String Manipulation topic and 0.44/1 for the Data Structures and Algorithms topic. GPT-3 performed fairly well when providing feedback on String Manipulation questions, with a strong performance in writing correct sample implementations (a single exception was found when a function was implemented incorrectly in Test 7). However, a few instances were found when GPT-3 was not able to identify correct code, and would provide incorrect feedback (for example, Test 15). Overall, however, GPT-3’s feedback, sample solutions, and analysis of correct inputted code for String Manipulation was pretty strong. GPT-3’s feedback was not as strong for Data Structures and Algorithms questions, however. For quite a few tests, GPT-3 was not able to identify key elements of programs that made them successful. Additionally, GPT-3 would say that an implementation was incorrect when it was in fact correct (such as the Quicksort implementation in Test 9). Additionally, an instance was found where GPT-3 extrapolated from the given prompt and marked a solution as wrong because it did not include an element that was not asked for in the first place (in Test 10). As a whole, though, GPT-3 provided good feedback for questions in both topics.

For the 5 experiments where the inputted code was partially correct, all 5 were done on the String Manipulation topic. GPT-3’s average evaluation was 0.833/1 for the String Manipulation topic. GPT-3’s performance fluctuated wildly, as it would provide helpful feedback to improve some questions, but had poor feedback in other cases. An example of an excellent evaluation is Test 14, in which GPT-3 correctly identified the code as correct, but provided feedback that a return statement should be added to the code. An example of a lackluster evaluation is Test 20, in which GPT-3 identifies the input code as nearly correct, but then gives a entirely different implementation rather than improving the user inputted one. No partially correct solutions were tested on the Data Structures and Algorithms topic.

For the 4 experiments where the inputted code was incorrect, all 4 were done on the String Manipulation topic. GPT-3's average evaluation was 0.917/1 for the String Manipulation topic. GPT-3 was fairly consistent at giving good explanations for why user inputted code was correct, but some points were docked for an instance where GPT-3's suggested implementation also turned out to be incorrect (see Test 7). No incorrect solutions were tested on the Data Structures and Algorithms topic.

For the 4 experiments where the inputted code was correct, 2 were done on the String Manipulation topic and 2 were done on the Data Structures and Algorithms topic. GPT-3's average evaluation was 0.67/1 for both the String Manipulation topic and the Data Structures and Algorithms topic. The two types of Gibberish tests conducted were the implementation of a code solution in Java instead of Python, and the return of a random statement. In Test 1, GPT-3 correctly identified that the inputted code was written in Java. However, in Test 12, GPT-3 failed to recognize that the code was written in the wrong language, and gave an evaluation saying that the user's implementation was almost correct. In Test 3, not only did GPT-3 incorrectly state that the inputted function does not take the string in as a parameter, but it also fails to recognize that the returned value is gibberish. In Test 11, GPT-3 correctly identifies that the input is not an implementation of a sorting result. As a result, it seems as if GPT-3 can be hit or miss with gibberish, where it sometimes fails to understand the magnitude of completely erroneous code.

5 Conclusion

In this project, we developed a tool that provides realistic feedback to data science students regarding their interview responses, both non-technical and technical. In doing so, we utilized an autoregressive large language model, OpenAI's GPT-3, to ensure that the feedback given was tailored to the user's responses. This ensured dynamic, specific feedback, as opposed to generic feedback that one would receive from a different tool.

We were able to draw many conclusions from the tests conducted in this project. Regarding the SQL practice tool, our tool is able to recognize correct solutions that were semantically different than the "ground truth" solution scraped from an online resource. We consider this as a benefit of our tool compared to others available- that it doesn't zero in on a specific answer, but is open to solutions that are still accurate. That being said, we found instances in which our GPT-3 calls wasn't 100% accurate, which is something that can be worked on in the future.

Regarding the behavioral interview tool, the chatbot is able to ask relevant questions based on one's experience, yet it has trouble asking to follow up questions that can be distinguished from one another and taking a user's previous information into account for the next question (this aspect has been improved by ChatGPT though). Further on, a helpful feature would be to evaluate the user's input answer, perhaps by comparing it to the generated sample answer.

Overall, the Python practice tool was able to provide good sample solutions for each of the questions asked, as well as provide useful evaluations of and feedback for user-inputted code. However, instances were found when GPT-3 made completely incorrect evaluations for user code, such as failing to identify correct parameters, return statements, or key elements of algorithm implementation. Additionally, GPT-3 would occasionally fail to flag implementations written in different languages, or identify smaller issues such as indentation errors rather than larger errors. GPT-3 provided better guidance overall for String Manipulation questions, and often had trouble providing useful or correct evaluations for Data Structures and Algorithms questions. In the future, it is necessary to test the Data Structures and Algorithms section more thoroughly, including tests for partially correct and incorrect code. Additionally, further work should be done to improve feedback for these questions by eliminating erroneous feedback and providing the user with better information about how they can fix and optimize their solutions.

References

Kenneth Reitz. (2022, June 29). *Requests* (Version 2.28.1).

Kumar Rajwani. (2021, December 25). *Resume parser* (Version 0.8.4).
Leonard Richardson. (2022, April 8). *Beautiful soup* (Version 4.11.1).
LLC, M. (2022). *W3resource*. Retrieved August 19, 2022, from <https://www.w3resource.com/>
OpenAI. (2021). Openai api.
Sergey Karayev. (2022). *Gpt.py*. <https://replit.com/@SergeyKarayev/gptpy>

Appendix

A Additional Figures

Table 4: Topic Breakdown of Tests Performed for SQL Practice Hub tool

Type of Input	Topic	# of Test Exercises
Correct Query	Retrieve Data from Tables	1
	Boolean and Relational Operators	2
	Wildcard and Special Operators	2
	Aggregate Functions	2
	Formatting Query Output	1
	SQL Joins	2
	Total	10
Incorrect Query	Retrieve Data from Tables	1
	Boolean and Relational Operators	1
	Wildcard and Special Operators	2
	Aggregate Functions	2
	Formatting Query Output	2
	SQL Joins	2
	Total	10
Gibberish	Retrieve Data from Tables	0
	Boolean and Relational Operators	1
	Wildcard and Special Operators	1
	Aggregate Functions	1
	Formatting Query Output	1
	SQL Joins	1
	Total	5
Total Number of Tests Conducted		25

Figure 7: Screenshot of the first prompt tested in OpenAI Playground to compare the solution and a user input query.

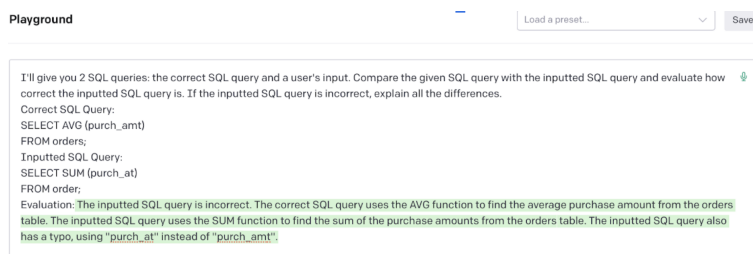


Figure 8: Example Resume Inputted into Behavioral Interview Tool

Elena Turing

eturing_sample@ucsd.edu ♦ (555) 555-5555 ♦ San Diego, CA

WORK EXPERIENCE

Dell Technologies
Junior Data Analyst

- Consulted firm that supports its client's system development, extension, customization and maintenance Needs.
- Handled through a variety of software business applications, databases and scripting methods.
- Maintained security and data integrity of the database Creation of database objects like tables, views, Materialized views, procedures, packages using Oracle tools like PL/SQL, SQL Plus, SQLLoader and Handled Exceptions.

Delta Corporations
Junior Data Analyst

- Prepared analysis, commentary and created strong presentations/dashboards to explain data and findings.
- Supported administrative, development, and change management activities as they relate to the data in Homelinks Salesforce org.

Apple
Data Science Intern

- Deployed inclusive data quality checks to ensure high quality of data, and monitored and solved operational or data issues in the data pipelines.

Pearson and Associates
Market Research Intern

- Assist Analysts in reviewing data and pulling verbatim quotes for client reports.
- Populate data to create client presentations and refine reports for better story-telling.

EDUCATION

University of California, San Diego (UCSD)
Expected Graduation: June 2024
BS in Data Science
3.58 GPA

SKILLS

- **Skills:** Python, Java, C++, SQL, MATLAB, Numpy, Pandas, Matplotlib, Keras, Tensorflow, Scikit-learn, NLP, Computer Vision, Statistics, Multivariable calculus, revenue modeling & forecasting; retail partnerships, sales, & distro; crowdfunding; copywriting; content marketing; brand marketing; DTC; Shopify; supply chain management; Keynote / PPT presentation

Table 5: Topic Breakdown of Tests Performed for Python Practice Hub tool

Type of Input	Topic	# of Test Exercises
Correct Code	String Manipulation	8
	Data Structures and Algorithms	3
	Total	11
Incorrect Code	String Manipulation	4
	Data Structures and Algorithms	1
	Total	5
Partially Correct Code	String Manipulation	5
	Data Structures and Algorithms	0
	Total	5
Gibberish	String Manipulation	2
	Data Structures and Algorithms	2
	Total	4
Total Number of Tests Conducted		25

B Code Breakdown

Link to the [GitHub repository](#).

B.1 SQL Practice Tool

Files needed:

- `sql_session.py`
 - Please enter in your OpenAI API key near the top of the file
- `sql_prompt.txt`
 - This file should be in the same directory as the `sql_session.py` file.

To run a session of the SQL Practice Tool, open a Python interactive session in Terminal or a Jupyter Notebook in the same directory as the above files and run the below code:

```
from sql_session import *
SQLSession('your-username-here')
```

You should see some printed output and will be prompted to enter in some input.

B.2 Python Practice Tool

Files needed:

- `ds_algo_gpt3.py`
 - Please enter in your OpenAI API key near the top of the file
- `python_prompt.txt`
 - This file should be in the same directory as the `ds_algo_gpt3.py` file.

- python_comparison_prompt.txt
 - This file should be in the same directory as the ds_algo_gpt3.py file.

To run a session of the Python Practice Tool, open a Terminal or a Jupyter Notebook in the same directory as the above files and run the below code:

```
from ds_algo_gpt3.py import *
PythonCodingSession( 'your-username-here ' )
```

You should see some printed output and will be prompted to enter in some input.

C

Chat Box Tests

See below for each of the twenty-five tests conducted to evaluate the behavioral interview tool. To diversify the conversations, we provided different sets of input information that included the example of Elena Turing's resume alongside other sample inputs of education, experience, and skills that mirrored the responses users gave during the poster demonstration.

Test 1 (Legitimate Answers)

- Input Information: Elena's Resume (see above)
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: What motivated you to pursue data science?:
- User: I'm very interested in math, and I wanted a career at the intersection of business and technology
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I was motivated to pursue data science because I was curious about how data could be used to solve problems and make predictions. I also liked the challenge of working with large amounts of data and finding patterns.
- Bobby: Tell me about your experience working at Dell Technologies.:
- User: It was a great learning experience. I worked extensively with Oracle tools and learned more about maintaining security and data integrity.
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I had a great experience working at Dell Technologies. I learned much about data science and how to apply it to real-world problems. I also gained valuable experience working with different programming languages and software applications.
- Bobby: What made you decide to pursue data science?:
- User: Once again, I'm interested in technology and business, which seemed like a great way to combine my interests.

- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I became interested in data science because I wanted to find ways to make better decisions based on data. I was also attracted to the challenge of understanding complex data sets and developing models to explain them.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 3/4

Test 2 (Legitimate Answers)

- Input Information: Elena's Resume (see above)
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: Can you tell me about your experience working in the field of data science?
- User: I have had 2 internships and a research position on campus that have helped me apply skills such as visualization and machine learning
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I have experience working in the field of data science at Dell Technologies, Delta Corporations, Apple, Pearson and Associates. I have used Python, Java, C++, SQL, MATLAB, Numpy, Pandas, Matplotlib, Keras, Tensorflow, Scikit-learn, NLP, Computer Vision, Statistics, and Multivariable calculus in my work.
- Bobby: What makes you interested in data science?
- User: I'm really interested in math and computer science as well as understanding how we can apply these to business solutions
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I am interested in data science because it is a field that combines my skills in programming, mathematics, and statistics to solve real-world problems. I enjoy working with data to find patterns and insights that can be used to improve decision-making.
- Bobby: Wow, that's quite a list of skills and experience! Let's start with your education. Tell me about your data science program at UCSD.
- User: It's great! It gives me a lot of experience and flexibility to go deeper into a particular domain
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1

- Bobby: I majored in Data Science at UCSD and I loved it! The program was very well rounded, teaching me everything from programming languages like Python and Java to statistical analysis and machine learning. I feel like I have a strong foundation in the field and am confident in my abilities.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 4/4

Test 3 (Legitimate Answers)

- Input Information: Elena's Resume (see above)
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: How did you get interested in data science?
- User: i really enjoyed math and wanted to stay involved in the business side
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I pursued data science because I wanted to learn how to make decisions using data. I was motivated by the idea of using data to improve outcomes and optimize processes. My previous work experiences have given me a strong foundation in technology and mathematics, which I used to build my skills in data science. I am constantly learning new techniques and methods to improve my abilities in data science.
- Bobby: What is your favorite data science project that you have worked on?
- User: my summer internship at apple, i worked on music
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: My favorite data science project that I have worked on is my final project for my Data Science class. We had to use all of the skills we learned in the class to complete a real-world data science project. We chose to analyze the UCSD Admissions dataset to try and predict which applicants would be admitted to the university. We used a variety of different techniques including feature engineering, data visualization, machine learning, and deep learning. In the end, we were able to create a model that correctly predicted the admissions decision for over 80
- Bobby: How did you like your experience working at Apple?
- User: it was really great. i learned a lot
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I enjoyed working at Apple because it was a very fast-paced and exciting environment. I was able to learn a lot and work with some very talented people.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)

- Score: 4/4

Test 4 (Legitimate Answers)

- Input Information: Elena's Resume (see above)
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: Do you have any experience working with big data?
- User: not really. I worked a bit on during class but I'm pretty new
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I do have experience working with big data. I have worked with Dell Technologies, Delta Corporations, and Apple in the past, and I have used Python, Java, C++, SQL, MATLAB, Numpy, Pandas, Matplotlib, Keras, Tensorflow, Scikit-learn, NLP, Computer Vision, Statistics, and Multivariable calculus to analyze and visualize data.
- Bobby: Can you elaborate on your experience working with Dell Technologies?
- User: I worked with the BigFlower Team for user experience and I learned a lot about the marketing analytics
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I gained a lot of experience working with Dell Technologies. I worked on a variety of projects, including data analysis, data visualization, and machine learning. I also gained experience in programming languages such as Python, Java, and SQL. In addition, I gained experience working with statistical software such as MATLAB and R.
- Bobby: Wow, those are a lot of skills!
What was your favorite class in college?
- User: My favorite class was statistics. I love going through and solving the different probability riddles
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I really enjoyed my Data Science classes the most because I found the subject matter to be really interesting and it was enjoyable to learn more about how to analyze data.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 4/4

Test 5 (Legitimate Answers)

- Input Information: Elena's Resume (see above)
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):

- User: 3
- Bobby: Tell me about your experience working at Dell Technologies.
- User: It was great, I learned a lot about different tools
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: My experience working at Dell Technologies was great. I gained a lot of experience in data science and gained valuable skills in programming languages such as Python, Java, and C++. I also gained experience in SQL, MATLAB, and Numpy, which are all very important for data science. I also gained experience in working with data visualization tools such as Matplotlib and Keras. Overall, working at Dell Technologies was a great experience that helped me gain a lot of experience in data science.
- Bobby: I'm sorry to hear that you're looking for an internship. Do you have any experience in the field of data science?
- User: No, that's why i need an internship
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I actually just graduated with my degree in data science from UCSD! I've had a wide range of experiences working with data at different companies, including Dell Technologies, Delta Corporations, Apple, Pearson and Associates. I'm confident that my skills in Python, Java, C++, SQL, MATLAB, Numpy, Pandas, Matplotlib, Keras, Tensorflow, Scikit-learn, NLP, Computer Vision, Statistics, and Multivariable calculus will be a valuable asset to any company.
- Bobby: Why did you choose data science?
- User: I like math
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I chose data science because it is a rapidly growing field that combines many of the skills I have acquired through my studies and work experiences. I believe that data science will enable me to have a greater impact on the world by helping organizations to make better decisions based on data. Additionally, data science is a field that is constantly evolving, which means that there will always be new challenges to keep me engaged and excited about my work.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 2/4

Test 6 (Legitimate Answers)

- Input: Major: Computer Science. Experience: None Skills: Java
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3

- Bobby: Tell me about your degree in Computer Science.
- User: I chose it because I really like problem solving and I wanted a career that allowed me to apply that in my daily tasks
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I majored in Computer Science at UCSD. I took a variety of courses that taught me programming languages, databases, software engineering, and systems design. I also did a lot of independent research, which gave me experience working on projects by myself and collaborating with other students. I think my degree has prepared me well for a career in software development.
- Bobby: What are your career aspirations?
- User: I hope to be a software engineer working in the healthcare industry
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I would like to work as a software engineer or a software developer.
- Bobby: What is your favorite thing about data science?
- User: I love how versatile and impactful it is
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: Some of my favorite things about data science include the ability to work with large and complex datasets, the challenge of finding patterns and insights in data, and the satisfaction of building models that can make predictions or recommendations.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 4/4

Test 7 (Legitimate Answers)

- Input: Major: Cognitive Science. Experience: Insomnia cookies Cashier Skills: None
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: What made you decide to pursue data science?
- User: I wasn't sure what to do but I liked coding so I went with data science
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1

- Bobby: I decided to pursue data science after graduating from UCSD with a cognitive science degree and working as a cashier at Insomnia Cookies. I developed an interest in data science after taking some courses in college and realized that it was a growing field with a lot of potential. I decided to pursue data science to learn more about how to analyze data and use it to solve problems.
- Bobby: When did you graduate from UCSD?
- User: I haven't actually graduated yet
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I graduated from UCSD in 2016.
- Bobby: Tell me about your experience as a data science student.
- User: It was great, but honestly it just started. I'm actually a freshman
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: Well, as a data science student, I've had the opportunity to learn a lot about the field and gain practical experience through projects and internships. I've also had the chance to work with some amazing people and learn from their experiences.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 1/4

Test 8 (Legitimate Answers)

- Input: Major: Undecided. Experience: Nowhere Skills: None
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: Before we begin, are you sure you want to do a mock interview?
- User: Yes
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: Yes, I am sure. I am ready to do a mock interview.
- Bobby: Do you have any experience working with data?
- User: Not really
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1

- Bobby: I do not have any direct experience working with data, however I am comfortable using computers and I am confident that I could learn how to work with data if given the opportunity. I am also familiar with Microsoft Excel, which I believe would be useful for working with data.
- Bobby: What made you choose to study data science?
- User: I haven't studied it yet actually
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: There are a few reasons why I chose to study data science. Firstly, data science is an immensely powerful tool that can be used to solve complex problems. Secondly, data science is a relatively new field, which means that there is a lot of unexplored territory. Finally, data science is a very interdisciplinary field, which means that I get to learn about a variety of topics.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 4/4

Test 9 (Legitimate Answers)

- Input: Major: Economics. Experience: Bank of America. Skills: None
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: What led you to pursue data science?
- User: I wanted to learn more about the tech and data analysis related to my finance concentration
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: My interest in data science began when I was working at Bank of America. I was intrigued by the way data could be used to improve business outcomes and decided to learn more about it. I pursued a degree in economics because I wanted to understand the underlying principles of data science and how they could be applied to real-world problems. Through my studies, I developed the skillsets necessary to effectively work with data and analytics.
- Bobby: Do you have any experience working with data?
- User: Not yet, but I've seen other people working with it and I'm really interested in learning more
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I have experience working with data from my time at Bank of America. I also have experience working with data from my Economics classes at UCSD.
- Bobby: What do you like most about data science?
- User: I like that it is very versatile and tangible

- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: There are many things that I like about data science. One thing that I find particularly appealing is the ability to analyze data and find trends that can be used to make predictions. I also like the challenge of working with complex datasets and finding ways to make them understandable. Additionally, I enjoy the satisfaction that comes with solving problems and helping organizations to make better decisions.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 3/4

Test 10 (Legitimate Answers)

- Input: Major: Math. Experience: Mathworks. Skills: MATLAB
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: So you're a data science student. What's the most difficult problem you solved?
- User: I am actually not a data science student
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: The most difficult problem I solved was a project for my data science class where we had to analyze a large dataset and find patterns. It was difficult because I had to do a lot of data wrangling and cleaning before I could even start to look for patterns. But in the end, I was able to find some interesting patterns that nobody else in the class found.
- Bobby: What made you interested in data science?
- User: I think its an application of my math major
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I was interested in data science because it is a way to apply my math skills to real world problems. I liked the idea of working with data to find trends and patterns.
- Bobby: Okay, great. Do you have any experience working with data science?
- User: Not really, but I use R for some of my classes
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I do not have direct experience working with data science, however I have taken multiple courses in college relating to data science, and my work experience at Mathworks has given me a strong foundation in the MATLAB programming language which I believe would be beneficial for working with data science.

- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 2/4

Test 11 (Lackluster Answers)

- Input: Major: econ/math/poli sci Experience: city of san diego, the hospital Skills: None
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: Could you tell me more about your experience working at the city of San Diego?
- User: I just made coffee
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I worked as an intern in the city of San Diego's finance department. I was responsible for assisting with the budgeting process and providing analysis on various city projects. I gained a lot of experience working with numbers and Excel, and I also got to see how the government works firsthand. It was a great experience overall.
- Bobby: What was your favorite class in college?
- User: Political Science
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: One of my favorite classes that I took while I was studying at UCSD was my economics class. I really enjoyed learning about how the economy works and how it affects different aspects of our lives. The class was very informative and I feel like I gained a lot of valuable knowledge from it.
- Bobby: Have you ever done an internship before?
- User: yes
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I have not completed an internship before, but I have had various jobs in different industries. I have found that each job has taught me new skills that I can use in my career.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 2/4

Test 12 (Lackluster Answers)

- Input: Major: English. Experience: UCSD. Skills: None
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):

- User: 3
- Bobby: Tell me about your experience working at UCSD.
- User: I am an English tutor at USD
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I have enjoyed my time working at UCSD. I have gained valuable experience in customer service and communication. I have also developed strong organizational skills while working here. I believe that my time at UCSD has prepared me well for a career in customer service and/or communication.
- Bobby: When did you graduate from UCSD?
- User: I'm a current student
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I graduated from UCSD in 2015.
- Bobby: What led you to data science?
- User: I want to try something different
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: There are a few reasons that led me to data science. Firstly, I was interested in finding ways to effectively analyze and interpret large amounts of data. Secondly, I wanted to learn how to use data to make predictions and recommendations. Finally, I was attracted to the interdisciplinary nature of data science, which combines concepts from computer science, statistics, and math.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 3/4

Test 13 (Lackluster Answers)

- Input: Major: Music. Experience: Samsung. Skills: Python.
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: What made you choose to study data science?
- User: I don't study data science actually, but I am interested in learning more about it
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I chose to study data science because it is a growing field with a lot of potential. I also have experience working with Python, which is a popular language for data science.

- Bobby: What is your favorite data science project that you have worked on?
- User: I used NLP to create lyrics during my internship
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: My favorite data science project that I have worked on is my project on predicting the sale price of a home. I used a variety of features to predict the sale price of a home, including the size of the home, the location, and the number of bedrooms and bathrooms. I also utilized a variety of data visualization techniques to help me understand the data and to communicate my findings to others.
- Bobby: What was your favorite part about working at Samsung?
- User: I loved the flexibility and room to try new things
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: My favorite part about working at Samsung was the people. I loved getting to know my colleagues and working together to achieve our goals. I also appreciated the opportunity to learn new things and grow my skills.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 4/4

Test 14 (Lackluster Answers)

- Input: Major: Data Science and Economics. Experience: UCSD Skills: Python
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: What do you think makes data science interesting?
- User: I think there are a lot of different applications and fields to experiment with
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I think data science is interesting because it combines two areas that I am passionate about: economics and computer programming. Data science allows me to use my skills in Python to analyze data and find trends that can be used to make economic decisions.
- Bobby: Questions about your education:
 - What motivated you to pursue a degree in data science and economics? -What did you enjoy most about your studies? -What do you think sets your data science and economics education apart from other programs?
- Questions about your experiences:
 - Can you share a project you worked on during your time at UCSD that you're particularly proud of?
 - What challenges did you face while working on this project, and how did you overcome them? -What skills did you develop during your time at UCSD that you think will be helpful in your future career?

- User: which question do I answer
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: Questions about your skills:-How would you describe your Python skills? -What type of data do you feel most comfortable working with? -What do you think are the most important skills for a data scientist?
- Bobby: What is your favorite data science project that you have done?
- User: I'm so confused
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: My favorite data science project was one that I did during my time at UCSD. I worked on a team to develop a Python tool that helped students choose their courses. The tool used data from student surveys and course evaluations to provide Course recommendations. It was interesting to work with such a large dataset and to develop a tool that had the potential to help a lot of students.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 1/4

Test 15 (Lackluster Answers)

- Input: Major: Computer Science. Experience: Apple. Skills: SQL
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: How long have you been studying data science?
- User: 3 years
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I have been studying data science for about two years now.
- Bobby: What motivated you to pursue data science?
- User: I really liked how fun and interesting it was
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I was motivated to pursue data science because it is a field that combines my interests in computer science and business. I believe that data science is a rapidly growing field with immense potential. Additionally, I have heard from many sources that data scientists are in high demand and are able to command high salaries.

- Bobby: Tell me about your experience working at Apple
- User: Apple was cool. I got a t shirt
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I worked at Apple as a software engineer. I was responsible for developing and maintaining software for the Apple Store. I also created and maintained software for the Apple Online Store. I worked with a team of engineers to create software that would allow customers to purchase and download products from the online store. I also worked with customer service to provide support for customers who had questions or problems with their purchases.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 4/4

Test 16 (Lackluster Answers)

- Input: Major: Data Science. Experience: Nowhere. Skills: Java
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: Where did you hear about the internship?
- User: what internship
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I heard about the internship from a friend who is currently interning at Nowhere.
- Bobby: Where did you hear about the opening?
- User: handshake
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I heard about the opening through a friend who works at Nowhere.
- Bobby: What made you decide to pursue data science?
- User: I really like math
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: There are many reasons why I decided to pursue data science. Firstly, data science is an interdisciplinary field that combines aspects of computer science, statistics, and mathematics, which appealed to my interests. Secondly, data science is a rapidly growing field with many potential applications, which motivated me to learn more about it. Finally, my previous experience working at Nowhere gave me the skills and knowledge necessary to pursue data science, and I am confident that I can make significant contributions to the field.

- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 1/4

Test 17 (Lackluster Answers)

- Input: Major: data science. Experience: cashier at flower shop Skills: None
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: What is your favorite data science project that you have worked on?
- User: I made a spreadsheet with a lot of visualization for my flower shop
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I really enjoyed working on my senior thesis project, which was a predictive modeling project using data from the UC San Diego library system. I was able to use a lot of different techniques that I had learned in my coursework and it was really gratifying to see the model I created actually be accurate in predicting which books people would check out.
- Bobby: When did you graduate from UCSD?
- User: I did not
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I graduated from UCSD in May of 2020.
- Bobby: What made you decide to pursue a degree in data science?
- User: I'm still thinking about it tbh
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: When I was considering what to major in during college, data science was a field that really interested me. I liked the idea of being able to work with data and help organizations make better decisions. Additionally, my experience working at a flower shop showed me that I enjoyed working with numbers and data. Therefore, I decided to pursue a degree in data science so that I could combine my interests in math and science with my desire to help organizations make better decisions.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 3/4

Test 18 (Lackluster Answers)

- Input: Major: data science. Experience: apple, google, facebook, microsoft. Skills: python java sql matlab sas stata r

- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: Do you think your experience with big tech companies will give you an advantage in the job market?
- User: yeah
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: Yes, I believe my experience with big tech companies will give me an advantage in the job market. These companies are leaders in cutting-edge technological advances and data science, so having experience with them on my resume will make me a more attractive candidate to potential employers. Additionally, I have gained valuable skills while working at these companies that I can use in future roles, such as programming languages and software tools.
- Bobby: Have you ever done a mock interview before?
- User: obviously
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: No, I haven't done a mock interview before.
- Bobby: Wow, you have worked at some big name companies! Can you tell me a little more about your experience at each company?
- User: yeah, they each had a unique experience and provided a lot of diverse opportunities
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: Well, I had a great time at each company! I really learned a lot about data science while working at UCSD and I gained valuable experience working with different programming languages and software platforms. At Apple, I worked on a team that helped develop and improve the Siri voice recognition software. I also got to work on some big data projects at Google and Facebook. And at Microsoft, I helped develop a new tool for analyzing data sets. So, overall, I feel like I really gained a lot of experience and knowledge working at each of these companies.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 3/4

Test 19 (Lackluster Answers)

- Input: Major: data science. Experience: lighthouse. Skills: matlab
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3

- Bobby: How did you like your time at UCSD?
- User: it was nice
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I had a great time at UCSD. I really enjoyed the data science program and found the professors to be very engaging. The coursework was challenging but also very rewarding. I also really enjoyed my time working at the lighthouse. It was a great opportunity to gain experience in the field and learn new skills.
- Bobby: What led you to data science?
- User: data science seemed to combine math, programming, and people
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I was always interested in math and science, and I really enjoyed working with computers. Data science seemed like a perfect fit for me, because it combines all of those things. I also really liked the idea of being able to use data to help people make better decisions.
- Bobby: Where did you go to school for your data science degree?
- User: UCSD
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I went to school for my data science degree at UCSD. I majored in data science and have experience working at lighthouse. I have skills in matlab.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 2/4

Test 20 (Lackluster Answers)

- Input: Major: computer science. Experience: data analyst at mcdonalds. Skills: python, sklearn, numpy
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: What made you choose data science as your focus?
- User: I loved programming and statistics
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1

- Bobby: I decided to focus on data science because it is a field that uses scientific methods to extract knowledge and insights from data. I was attracted to the challenge of working with data to find patterns and insights that could help businesses and organizations make better decisions. I also liked the idea of using my computer science skills to develop algorithms and tools to analyze data.
- Bobby: Wow, those are great skills! Can you tell me more about your experience working with data at McDonald's?
- User: yeah, I brought back the mcRib during my final report
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I worked as a data analyst at McDonald's for two years. I was responsible for analyzing customer data and identifying trends. I used various tools to do this, including Python, sklearn, and numpy. I was able to learn a lot about data analysis during my time at McDonald's, and I gained valuable experience working with data.
- Bobby: What inspired you to pursue a career in data science?
- User: I just love being able to work in any field in an analytical capacity
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: In my previous job working as a data analyst at McDonalds, I was constantly inspired by the power of data to drive business decisions. It was clear to me that data science was the future of decision-making, and I wanted to be at the forefront of this exciting field. Additionally, my skills in Python, sklearn, and numpy made me confident that I could succeed as a data scientist.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 4/4

Test 21 (Gibberish)

- Input: Major: asldkfj. Experience: asldkfj. Skills: asldkfj
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: Ok, let's get started. Tell me about your experience working at asldkfj.
- User: lsakdfj
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: asd;lfkj
- VALUE ERROR PROGRAM ENDS
- Score: 2/4 - Performed as expected

Test 22 (Gibberish)

- Input: Major: i don't know. Experience: i have never worked. Skills: None

- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: Can you tell me more about your schooling?
- User: I dropped out of ucsd
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I attended UCSD as an undergraduate student and majored in psychology. I have experience working in a research lab and have skills in statistical analysis and data interpretation.
- Bobby: What made you decide to study data science?
- User: I don't want to, my mom made me
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I was originally interested in data science because I wanted to learn how to code and analyze data. I was also interested in the mathematical aspects of data science, such as statistics and machine learning.
- Bobby: Where did you go to school for your data science degree?
- User: ucsd
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I went to school for my data science degree at the University of California, San Diego.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 1/4

Test 23 (Gibberish)

- Input: Major: BLANK . Experience: BLANK Skills: BLANK
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: Tell me about your experience working with data.
- User: I had a couple of internships where i was a data analyst
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1

- Bobby: I have experience working with data from my time as a college student at UCSD. I majored in economics, so I am very familiar with using data to analyze trends and make predictions. I also have experience working with data from my time working at a financial institution. There, I was responsible for managing customer accounts and handling customer inquiries. This experience has given me a lot of experience working with data and has taught me how to effectively manage and organize data.
- Bobby: Do you have any experience working with data?
- User: yes i just told you i had two internships
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: Yes, I have experience working with data. I have worked with data in my studies as a college student, and I have experience working with data in my job. I have also used data in my work with clients.
- Bobby: What made you interested in data science?
- User: the potential for slightly above minimum wage
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I became interested in data science after taking a few courses in college where I was able to analyze data and see the impact that it could have on businesses and organizations. I also interned at a company that used data science to improve their product offerings and saw first-hand how data science could be used to solve real-world problems. Lastly, I have always been interested in mathematics and statistics, which are necessary skills for data science.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 1/4

Test 24 (Gibberish)

- Input: Major: undecided. Experience: asdlfkj. Skills: a;lsdkfj
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: What was your favorite class in college?
- User: i don't have one aldsfkj
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I really enjoyed my psychology classes. I found them fascinating and they really opened my eyes to a lot of things about myself that I never knew before.
- Bobby: Tell me about your experience working at asdlfkj.
- User: I hated it

- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I worked at asdlfkj for two years, during which time I gained valuable experience in customer service and teamwork. I was able to hone my skills in a;lsdkfj, and I learned how to effectively communicate with customers and resolve conflicts. My experience at asdlfkj was essential in preparing me for my current role as a college student.
- Bobby: Tell me about your background in data science.
- User: it doesn't exist
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I have experience working with data science tools and techniques at asdlfkj. I have also taken courses in data science at UCSD, which has provided me with the skills and knowledge necessary to work effectively with data. In addition, I have strong skills in a;lsdkfj, which enhances my ability to analyze and interpret data.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 3/4

Test 25 (Gibberish)

- Input: Major: lololol. Experience: BLANK Skills: None
- Bobby: Hi, I'm Bobby, the chatbox. I'll be your mock interviewer today for behavioral questions! How many questions do you have time to practice? (I recommend 3 for a quick demo):
- User: 3
- Bobby: Tell me about your experience working at .
- User: it doesn't exist lol
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I enjoyed my time working at . I was able to learn new skills and expand my knowledge in the field. I also gained valuable experience working with a team of professionals.
- Bobby: What is your favorite data science topic?
- User: i don't have one lol
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: Out of all the data science topics I have studied, my favorite would have to be predictive modelling. I enjoy the challenge of taking a dataset and using it to build a model that can accurately predict future outcomes. It is a very powerful tool that can be used in a variety of applications, from financial forecasting to medicine.

- Bobby: Why did you choose to study data science?
- User: i hate data science
- Bobby: Would you like to see a sample answer to the question you were just asked? Enter 1 if yes, 2 if no:
- User: 1
- Bobby: I chose to study data science because I wanted to learn how to analyze and interpret data. I was also interested in learning how to use data to predict future trends.
- Bobby: Well, it was great learning more about you! Good luck with your behavioral interview :)
- Score: 1/4

D

SQL Tests

See below for each of the twenty-five tests conducted to evaluate GPT-3's feedback in comparing two SQL queries. Refer to this list to understand what each topic number refers to.

Test 1 (Topic 1, Correct Query)

- Prompt: Write a query to display a string "This is SQL Exercise, Practice and Solution"
- Inputted Query: `select "This is SQL Exercise, Practice and Solution";`
- Solution: `SELECT "This is SQL Exercise, Practice and Solution";`
- Feedback: The input SQL query is correct. It would produce the same output as the solution SQL query. The only difference is that the input SQL query is lowercase while the solution SQL query is uppercase. In SQL, keywords are not case sensitive, so the query would still run. However, it is generally considered best practice to use uppercase for SQL keywords. Suggested topics to study: SQL case sensitivity
- Score: 3/3

Test 2 (Topic 2 Correct Query)

- Prompt: From the following table, write a SQL query to find the details of those salespeople whose commissions range from 0.10 to 0.12. Return salesman_id, name, city, and commission.
- Inputted Query: `select * from salesman where commission >= 0.10 and commission <= 0.12;`
- Solution:


```
SELECT salesman_id ,name ,city ,commission
FROM salesman WHERE (commission > 0.10
AND commission< 0.12);
```
- Feedback: The input SQL query is correct. It would produce the same output as the solution SQL query. The only difference is that the solution SQL query uses parentheses while the input SQL query does not. Parentheses are not required, but they can make the query easier to read.
- Score: 3/3

Test 3 (Topic 5, Correct Query)

- Prompt: From the following table, write a SQL query to find all the customers. Sort the result-set by customer_id. Return cust_name, city, grade

- Inputted Query: `select cust_name, city, grade from customer order by customer_id;`
- Solution:

```
SELECT cust_name , city , grade
FROM customer
ORDER BY customer_id;
```

- Feedback: The input SQL query is correct. It would produce the same output as the solution SQL query. The only difference is that the input SQL query is lowercase while the solution SQL query is uppercase.
- Score: 3/3

Test 4 (Topic 6, Correct Query)

- Prompt: SQL statement to generate a report with customer name, city, order number, order date, order amount, salesperson name, and commission to determine if any of the existing customers have not placed orders or if they have placed orders through their salesman or by themselves.
- Inputted Query:

```
SELECT c.cust_name , c.city , o.ord_no , o.ord_date ,
o.purch_amt AS "Order_Amount" , s.name, s.commission
FROM customer c LEFT OUTER JOIN orders o ON c.customer_id=o.customer_id
LEFT OUTER JOIN salesman s ON s.salesman_id=o.salesman_id;
```

- Solution:

```
SELECT a.cust_name ,a.city , b.ord_no ,
b.ord_date ,b.purch_amt AS "Order_Amount" ,
c.name,c.commission
FROM customer a
LEFT OUTER JOIN orders b
ON a.customer_id=b.customer_id
LEFT OUTER JOIN salesman c
ON c.salesman_id=b.salesman_id;
```

- Feedback: The input SQL query is correct. It would produce the same output as the solution SQL query. The only difference is that the input SQL query uses table aliases while the solution SQL query does not. Table aliases are used to give a table a temporary name. In this case, it does not matter whether table aliases are used because all the column names are unique. Suggested topics to study: table aliases
- Score: 3/3

Test 5 (Topic 6, Correct Query)

- Prompt: From the following tables write a SQL query to display the first and last names of each employee, as well as the department name and sanction amount.
- Inputted Query:

```
select e.EMP_FNAME, e.EMP_LNAME, d.DPT_NAME,
d.DPT_ALLOTMENT from emp_details e, emp_department d
where d.DPT_CODE = e.EMP_DEPT;
```

- Solution:

```

SELECT emp_details.emp_fname AS "First_Name", emp_lname AS "Last_Name",
       emp_department.dpt_name AS "Department",
       dpt_allotment AS "Amount_Allotted"
FROM emp_details
      INNER JOIN emp_department
      ON emp_details.emp_dept = emp_department.dpt_code;

```

- Feedback: The input SQL query is correct. It would produce the same output as the solution SQL query. The only difference is that the solution SQL query uses an inner join while the input SQL query uses a regular join. An inner join only returns rows where there is a match in both tables while a regular join will return all rows from both tables, even if there is no match. In this case, both queries would produce the same output because all the rows in both tables match on the DPT_CODE column. Suggested topics to study: types of joins
- Score: 3/3

Test 6 (Topic 3, Correct Query)

- Prompt: From the following table, write a SQL query to retrieve the details of the salespeople whose names begin with any letter between 'A' and 'L' (not inclusive). Return salesman_id, name, city, commission.
- Inputted Query: `select * from salesman where name between "A" and "L";`
- Solution:

```

SELECT *
FROM salesman
WHERE name BETWEEN 'A' and 'L';

```

- Feedback: The input SQL query is correct. It would produce the same output as the solution SQL query. The only difference is that the input SQL query uses double quotes while the solution SQL query uses single quotes. In SQL, single quotes are used to denote strings while double quotes are used for column or table names. In this case, both queries would produce the same output because all the rows in both tables match on the city column. Suggested topics to study: types of joins
- Score: 2/3; incorrect suggested topics to study.

Test 7 (Topic 4, Correct Query)

- Prompt: From the following table, write a SQL query to count the number of orders based on the combination of each order date and salesperson. Return order date, salesperson id.
- Inputted Query: `select ord_date, salesman_id, count(*) from orders group by ord_date, salesman_id;`
- Solution:

```

SELECT ord_date ,salesman_id ,COUNT(*)
FROM orders
GROUP BY ord_date ,salesman_id;

```

- Feedback: The input SQL query is correct. It would produce the same output as the solution SQL query. The only difference is that the solution SQL query has extra spaces.
- Score: 3/3; feedback was slightly incorrect because the count function was used correctly, but the feedback said it didn't have parentheses. it also didn't indicate that there are no commas.

Test 8 (Topic 2, Correct Query)

- Prompt: From the following table, write a SQL query to find the details of all employees whose last name is 'Dosni' or 'Mardy'. Return emp_idno, emp_fname, emp_lname, and emp_dept.

- Inputted Query: `select * from emp_details where emp_lname in ('Dosni', 'Mardy');`

- Solution:

```
SELECT *
FROM emp_details
WHERE emp_lname = 'Dosni' OR emp_lname='Mardy';
```

- Feedback: The input SQL query is correct. It would produce the same output as the solution SQL query. The only difference is that the input SQL query uses the IN operator while the solution SQL query uses the OR operator. The IN operator is used to check if a value is in a list of values while the OR operator is used to check if either of two conditions are true. In this case, both queries would produce the same output because the desired output is all rows where the emp_lname is either Dosni or Mardy. Suggested topics to study: IN operator, OR operator

- Score: 3/3

Test 9 (Topic 3, Correct Query)

- Prompt: From the following table, write a SQL query to find the details of those salespeople who come from the 'Paris' City or 'Rome' City. Return salesman_id, name, city, commission.

- Inputted Query: `select * from salesman where city = 'Paris' or city = 'Rome';`

- Solution:

```
SELECT *
FROM salesman
WHERE city = 'Paris' OR city = 'Rome';
```

- Feedback: The input SQL query is correct. It would produce the same output as the solution SQL query. The only difference is that the input SQL query uses single quotes while the solution SQL query uses double quotes. In SQL, single and double quotes are interchangeable.

- Score: 3/3

Test 10 (Topic 4, Correct Query)

- Prompt: From the following table, write a SQL query to find the minimum purchase amount.

- Inputted Query: `select min(purch_amt) from orders`

- Solution:

```
SELECT MIN(purch_amt)
FROM orders;
```

- Feedback: The input SQL query is correct. It would produce the same output as the solution SQL query. The only difference is that the input SQL query does not have a new line after the SELECT statement.

- Score: 3/3

Test 11 (Topic 1, Incorrect Query)

- Prompt: From the following table, write a SQL query that combines the winners in Physics, 1970 and in Economics, 1971. Return year, subject, winner, country, and category.

- Inputted Query:

```
select * from nobel_win
where (SUBJECT = 'Physics' and YEAR = 1970)
or (SUBJECT = 'Economics' and YEAR = 1971);
```

- Solution:

```
SELECT * FROM nobel_win
WHERE (subject = 'Physics' AND year=1970)
UNION (SELECT * FROM nobel_win
WHERE (subject = 'Economics' AND year=1971));
```

- Feedback: The input SQL query is incorrect with a fatal error. The solution SQL query uses the UNION operator to combine the results of two different SQL queries. The input SQL query uses the OR operator which would return all rows that match either conditions. The OR operator would return duplicate rows if a row matched both conditions while the UNION operator would only return unique rows. In this case, the solution SQL query is looking for two specific years and subjects. The input SQL query would return rows for other years and subjects as well. Suggested topics to study: UNION vs OR operator

- Score: 3/3

Test 12 (Topic 2, Incorrect Query)

- Prompt: Write a SQL query that displays order number, purchase amount, and the achieved and unachieved percentage (%) for those orders that exceed 50% of the target value of 6000.

- Inputted Query:

```
select ord_no , purch_amt , purch_amt / 60000 as "percentage"
from orders where (purch_amt / 6000) > .5;
```

- Solution:

```
SELECT ord_no , purch_amt , (100*purch_amt)/6000 AS "Achieved_%" ,
(100*(6000-purch_amt)/6000) AS "Unachieved_%"
FROM orders WHERE (100*purch_amt)/6000>50;
```

- Feedback: The input SQL query is incorrect with fatal and minor errors. The correct SQL query calculates the percentage of the total goal that was achieved and the percentage of the total goal that was unachieved. The input SQL query only calculates the percentage of the total goal that was achieved. The minor error is that the input SQL query uses 60000 instead of 6000 to calculate the percentage. Suggested topics to study: SQL mathematical operators, SQL aliases

- Score: 3/3

Test 13 (Topic 3, Incorrect Query)

- Prompt: From the following table, write a SQL query to identify those rows where col1 does not contain the forward slash character (/). Return col1.

- Inputted Query:

```
select * from testtable where col1 is not like "%/%";
```

- Solution:

```
SELECT * FROM testtable WHERE col1 NOT LIKE '%//%' ESCAPE '/' ;
```

- Feedback: The input SQL query is incorrect with a fatal error. The input SQL query is missing the ESCAPE '/' clause. The ESCAPE '/' clause tells the SQL engine to treat the '/' character as a regular character and not as an escape character. Without the ESCAPE '/' clause, the SQL engine will treat the '/' character as an escape character. As a result, the query will not return the desired output. Suggested topics to study: LIKE operator, escape characters
- Score: 3/3

Test 14 (Topic 3, Incorrect Query)

- Prompt: From the following table, write a SQL query to select orders between 500 and 4000 (begin and end values are included). Exclude orders amount 948.50 and 1983.43. Return ord_no, purch_amt, ord_date, customer_id, and salesman_id.
- Inputted Query:

```
select * from orders
where (purch_amt >= 500 or purch_amt >= 4000)
and purch_amt not in (948.5 , 1983.43);
```

- Solution:

```
SELECT * FROM orders
WHERE (purch_amt BETWEEN 500 AND 4000)
AND NOT purch_amt IN(948.50 ,1983.43);
```

- Feedback: The input SQL query is incorrect with fatal and minor errors. The correct SQL query uses the BETWEEN operator to find all purchase amounts that are between 500 and 4000. The input SQL query instead uses the >= operator which would return all purchase amounts that are greater than or equal to 500 or 4000. This would return different results than the solution SQL query. The input SQL query also uses the OR operator instead of the AND operator. The AND operator would return all rows where the purchase amount is greater than or equal to 500 AND less than or equal to 4000. The OR operator would return all rows where the purchase amount is greater than or equal to 500 OR greater than or equal to 4000. This would return different results than the solution SQL query. The minor error is with a typo in one of the values being checked. The input SQL query has 948.5 while the solution SQL query has 948.50. Suggested topics to study: -The BETWEEN operator -The AND and OR operators
- Score: 3/3

Test 15 (Topic 4, Incorrect Query)

- Prompt: From the following table, write a SQL query to find the maximum order (purchase) amount in the range 2000 - 6000 (Begin and end values are included.) by combination of each customer and order date. Return customer id, order date and maximum purchase amount.
- Inputted Query:

```
select customer_id , ord_date , and max(purch_amt)
from orders where (purch_amt between 2000 and 6000);
```

- Solution:

```
SELECT customer_id ,ord_date ,MAX(purch_amt)
FROM orders
GROUP BY customer_id ,ord_date
HAVING MAX(purch_amt) BETWEEN 2000 AND 6000;
```

- Feedback: The input SQL query is incorrect with fatal and minor errors. The correct SQL query uses the GROUP BY and HAVING clauses to find the maximum purchase amount for each customer ID and order date that falls between 2000 and 6000. The input SQL query instead uses a WHERE clause. The minor error is with the use of the keyword "and". The input SQL query should not have used "and" between the ord_date and max(purch_amt) columns. Suggested topics to study: GROUP BY and HAVING clauses
- Score: 3/3

Test 16 (Topic 4, Incorrect Query)

- Prompt: From the following table, write a SQL query to find the highest purchase amount ordered by each customer on a particular date. Return, order date and highest purchase amount.

- Inputted Query:

```
select ord_no , purch_amt , purch_amt / 60000 as "percentage"
from orders where (purch_amt / 6000) > .5;
```

- Solution:

```
select ord_no , purch_amt , purch_amt / 60000 as "percentage"
from orders where (purch_amt / 6000) > .5;
```

- Feedback: The input SQL query is incorrect. The solution SQL query groups by customer_id and ord_date while the input SQL query only groups by ord_date. This means that the solution SQL query will return one row for each custome_id and ord_date combination while the input SQL query will return one row for each ord_date. The solution SQL query will also return the customer_id column while the input SQL query will not. Suggested topics to study: GROUP BY
- Score: 3/3

Test 17 (Topic 5, Incorrect Query)

- Prompt: From the following table, write a SQL query to select all the salespeople. Return salesman_id, name, city, commission with the percent sign (%).

- Inputted Query:

```
select ord_no , purch_amt , purch_amt / 60000 as "percentage"
from orders where (purch_amt / 6000) > .5;
```

- Solution:

```
select ord_no , purch_amt , purch_amt / 60000 as "percentage"
from orders where (purch_amt / 6000) > .5;
```

- Feedback: The input SQL query is correct. It would produce the same output as the solution SQL query. The only difference is that the solution SQL query uses the '%' character while the input SQL query uses the concat function to concatenate the "%" character to the end of the commission column.
- Score: 2/3; I would argue that the percent sign in the wrong place would make the SQL query incorrect.

Test 18 (Topic 6, Incorrect Query)

- Prompt: Write a SQL statement to generate a list in ascending order of salespersons who work either for one or more customers or have not yet joined any of the customers.

- Inputted Query:

```

select s.salesman_id
from salesman s left outer join customer c on s.salesman_id = c.salesman_id
group by s.salesman_id
order by s.salesman_id;

```

- Solution:

```

SELECT a.cust_name,a.city,a.grade, b.name AS "Salesman", b.city
FROM customer a RIGHT OUTER JOIN salesman b ON b.salesman_id=a.salesman_id
ORDER BY b.salesman_id;

```

- Feedback: The input SQL query is incorrect with fatal errors. The solution SQL query is a right outer join while the input SQL query is a left outer join. A right outer join returns all rows from the right table, even if there is no match in the left table. A left outer join returns all rows from the left table, even if there is no match in the right table. In this case, the queries would produce different outputs. Suggested topics to study: types of joins
- Score: 3/3

Test 19 (Topic 6, Incorrect Query)

- Prompt: Write a SQL statement to generate a report with the customer name, city, order no. order date, purchase amount for only those customers on the list who must have a grade and placed one or more orders or which order(s) have been placed by the customer who neither is on the list nor has a grade.
- Inputted Query:

```

select c.cust_name, c.city, o.ord_no, o.ord_date, o.purch_amt
from customer c right outer join order o on c.customer_id = o.customer_id
where c.grade is not null or (c.grade is null and o.ord_no is null);

```

- Solution:

```

SELECT a.cust_name,a.city, b.ord_no, b.ord_date,b.purch_amt AS "Order_Amount"
FROM customer a FULL OUTER JOIN orders b ON a.customer_id=b.customer_id
WHERE a.grade IS NOT NULL;

```

- Feedback: The input SQL query is incorrect with fatal and minor errors. The correct SQL query uses a FULL OUTER JOIN while the input SQL query uses a RIGHT OUTER JOIN. A FULL OUTER JOIN returns all rows from both tables, even if there is no match. A RIGHT OUTER JOIN only returns all rows from the right table. The input SQL query also has the table aliases in the wrong order. The solution SQL query has the table alias "a" for the customer table and the table alias "b" for the orders table. The input SQL query has the table alias "c" for the customer table and the table alias "o" for the orders table. The input SQL query also has an extra set of parentheses that are not needed. Suggested topics to study: types of joins, table aliases
- Score: 3/3

Test 20 (Topic 5, Incorrect Query)

- Prompt: From the following table, write a SQL query to find all the orders. Sort the result-set in descending order by ord_date. Return all fields.
- Inputted Query:

```

select ord_no, purch_amt, purch_amt / 60000 as "percentage"
from orders where (purch_amt / 6000) > .5;

```

- Solution:

```
select ord_no , purch_amt , purch_amt / 60000 as "percentage"
from orders where (purch_amt / 6000) > .5;
```

- Feedback: The input SQL query is correct. It would produce the same output as the solution SQL query. The only difference is that the solution SQL query includes the keyword "DESC" after the ord_date column. This keyword indicates that the results should be in reverse order, with the most recent date appearing first.
- Score: 2/3; omission of desc should make the inputted query incorrect.

Test 21 (Topic 6, Gibberish — Not SQL)

- Prompt: SQL statement to generate a report with customer name, city, order number, order date, order amount, salesperson name, and commission to determine if any of the existing customers have not placed orders or if they have placed orders through their salesman or by themselves.
- Inputted Query:

```
select ord_no , purch_amt , purch_amt / 60000 as "percentage"
from orders where (purch_amt / 6000) > .5;
```

- Solution:

```
SELECT a.cust_name ,a.city , b.ord_no ,
b.ord_date ,b.purch_amt AS "Order_Amount" ,
c.name,c.commission
FROM customer a
LEFT OUTER JOIN orders b
ON a.customer_id=b.customer_id
LEFT OUTER JOIN salesman c
ON c.salesman_id=b.salesman_id ;
```

- Feedback: The input SQL query is incorrect with fatal errors. The correct SQL query uses the LEFT OUTER JOIN to join the customer and orders table. The inputted SQL query instead uses the [print(i) for i in range(1,10)] function which will print the numbers 1-9. Suggested topics to study: types of joins
- Score: 2/3; feedback doesn't indicate that what i entered is clearly not SQL.

Test 22 (Topic 5, Gibberish — Not SQL)

- Prompt: From the following table, write a SQL query that calculates the maximum purchase amount generated by each salesperson for each order date. Sort the result-set by salesperson id and order date in ascending order. Return salesperson id, order date and maximum purchase amount.
- Inputted Query: "GPT-3 is dumb"
- Solution:

```
SELECT salesman_id ,ord_date ,MAX(purch_amt)
FROM orders
GROUP BY salesman_id ,ord_date
ORDER BY salesman_id ,ord_date ;
```

- Feedback: The input SQL query is incorrect with fatal errors. The correct SQL query uses the MAX function to find the maximum purchase amount for each salesman and order date. The inputted SQL query instead just says "GPT-3 is dumb". In addition, the input SQL query is missing the GROUP BY and ORDER BY clauses. Suggested topics to study: GROUP BY and ORDER BY clauses

- Score: 3/3

Test 23 (Topic 4, Gibberish — Not SQL)

- Prompt: From the following table, write a SQL query to count the number of customers. Return number of customers.
- Inputted Query: "from the following table, write a SQL query to count the number of customers. return the number of customers."
- Solution: `SELECT COUNT(*) FROM customer;`
- Feedback: The input SQL query is incorrect with a fatal error. The input SQL query is missing the SELECT and FROM keywords. The input SQL query should be: `SELECT COUNT(*) FROM customer;`
- Score: 2/3; didn't give suggest topics, but came up with the correct query either based on solution query or the inputted query.

Test 24 (Topic 3, Gibberish — Not SQL)

- Prompt: From the following table, write a SQL query to find the details of the customers whose names end with the letter 'n'. Return customer_id, cust_name, city, grade, salesman_id.
- Inputted Query: "select like '%' from order by group by;"
- Solution:


```
SELECT *
FROM customer
WHERE cust_name LIKE '%n';
```
- Feedback: The input SQL query is incorrect with fatal errors. The query should select all columns from the customer table where the customer name is like '%n'. The input SQL query instead selects from the order table and uses the group by clause without any aggregate functions. The group by clause is used in conjunction with aggregate functions to group together rows with the same values. Suggested topics to study: SQL clauses, aggregate functions
- Score: 3/3; recognized SQL keywords

Test 25 (Topic 2, Gibberish — Not SQL)

- Prompt: From the following table, write a SQL query to find customers who are from the city of New York or have a grade of over 100. Return customer_id, cust_name, city, grade, and salesman_id.
- Inputted Query: "hola me llamo jacqueline"
- Solution:


```
SELECT *
FROM customer
WHERE city = 'New_York' OR grade > 100;
```
- Feedback: The input SQL query is incorrect with fatal errors. The query should select all columns from the customer table where the city is New York or the grade is greater than 100. Instead, the input SQL query just says "hola me llamo jacqueline". Suggested topics to study: SQL syntax
- Score: 3/3; recognizes that what was entered is gibberish!

E Python Tests

Test 1 (Topic 1, Gibberish – Not Python)

- Prompt: Write a Python program to remove repeated consecutive characters and replace with the single letters and print new updated string.

- Inputted Code:

```
• import java.util.Scanner;
  public class Main {
    public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter a string:");
      String str = scanner.nextLine();
      String updatedStr = "";
      char previousChar = '_';
      for (int i = 0; i < str.length(); i++) {
        char currentChar = str.charAt(i);
        if (currentChar != previousChar) {
          updatedStr += currentChar;
        }
        previousChar = currentChar;
      }
      return updatedStr
    }
  }
```

- Solution:

```
def remove_consecutive_duplicates(s):
    # initialize output string
    output = ""
    # keep track of previous character
    prev_char = None
    # iterate through string
    for c in s:
        # if current character is not the same as the previous character
        if c != prev_char:
            # add character to output string
            output += c
        # set previous character to current character
        prev_char = c
    # return output string
    return output
```

- Feedback: The inputted function is written in Java instead of Python.

- Score: 3/3

Test 2 (Topic 1, partially correct code)

- Prompt: Write a Python program to find the first repeated character in a given string.

- Inputted Code:

```
def deletechar(input, char):
    for x in input:
        if x == char:
            return input.replace(char, "")
```

- Solution:

```
def findrepeat(str):
    letterlist = []
    for x in str:
        if x not in letterlist:
            letterlist.append(x)
        else:
            return str.index(x)
```

- Feedback: The inputted function correctly adds each letter to a list if it is not already in the list. However, it only returns the first letter that is repeated, instead of the first letter that is repeated in the string. This can be fixed by returning the index of the first letter that is repeated in the string, instead of the letter itself.
- Score: 2/3; The feedback makes it unclear why the index of the first letter should be returned rather than the letter itself.

Test 3 (Topic 1, Gibberish code)

- Prompt: Write a Python program to check whether a given string contains a capital letter, a lower case letter, a number and a minimum length.
- Inputted Code:

```
def window(str):
    return "Hello_world!_The_sun_is_shining!_O_what_a_joy_to_be_scientist_with_a_beating_heart"
```

- Solution:

```
def check(string):
    capital = 0
    lower = 0
    number = 0
    for i in string:
        if i.isupper():
            capital += 1
        elif i.islower():
            lower += 1
        elif i.isnumeric():
            number += 1
    if capital > 0 and lower > 0 and number > 0 and len(string) >= 8:
        return "Valid_string."
    else:
        return "Invalid_string."
```

- Feedback: The inputted function does not take in the string input as a parameter, so it is not possible to compare the inputted function with the solution function.

- Score: 1/3; Not only does GPT3 incorrectly state that the inputted function does not take the string in as a parameter, but it also fails to recognize that the returned value is gibberish.

Test 4 (Topic 1, correct code)

- Prompt: Write a Python program to change a given string to a new string where the first and last chars have been exchanged.

- Inputted Code:

```
def firstlastswap(str):
    first = str[0]
    last = str[-1]
    str1 = str.replace(str[0], last)
    str2 = str1.replace(str[-1], first)
    return str2
```

- Solution:

```
def firstlastswap(str):
    first = str[0]
    last = str[-1]
    str = str.replace(str[0], last)
    str = str.replace(str[-1], first)
    return str
```

- Feedback: The inputted function correctly swaps the first and last characters of the string, but it does so by creating 2 new strings, str1 and str2. This is unnecessary, as it is possible to swap the first and last characters of the original string, str.

- Score: 3/3

Test 5 (Topic 1, incorrect code)

- Prompt: Write a Python program to count number of substrings from a given string of lowercase alphabets with exactly k distinct (given) characters.

- Inputted Code:

```
def substrings(str):
    import math
    perms = math.factorial(len(str))
    return perms
```

- Solution:

```
def count_k_dist(str1, k):
    str_len = len(str1)

    result = 0
    ctr = [0] * 27
    for i in range(0, str_len):
        dist_ctr = 0
        ctr = [0] * 27
        for j in range(i, str_len):
            if (ctr[ord(str1[j]) - 97] == 0):
                dist_ctr += 1
            ctr[ord(str1[j]) - 97] += 1
```

```

    if (dist_ctr == k):
        result += 1
    if (dist_ctr > k):
        break
    return result

```

- Feedback: This function does not yield the same output as the solution function. The solution function is counting the number of substrings with exactly k distinct characters, while the inputted function is finding the number of all possible substrings. To fix this, the inputted function would need to take in the string and k as parameters, and then count the number of substrings with exactly k distinct characters.
- Score: 1/3; This was good feedback because it explained what my code accomplished and why it was different than what was expected. However, the code solution is not very readable, and seems to have arbitrary numbers.

Test 6 (Topic 1, Partially Correct Code)

- Prompt: Write a Python program to extract numbers from a given string.
- Inputted Code:

```

def isnum(sentence):
    returnlist = []
    for substring in sentence:
        if substring.isdigit():
            returnlist.append(substring)
    return returnlist

```

- Solution:

```

def extract_num(string):
    numbers = []

    # loop over the string
    for i in range(len(string)):
        # check if the character is a digit
        if string[i].isdigit():
            numbers.append(int(string[i]))

```

- Feedback: The inputted function is close, but it is missing a few key elements. It should iterate over the characters in the string instead of the substrings, and it should convert the digit characters to integers before adding them to the list.
- Score: 3/3

Test 7 (Topic 1, Incorrect Code)

- Prompt: Write a Python program to split a string on the last occurrence of the delimiter.
- Inputted Code:

```

def delimiter(input, char):
    for x in reversed(input):
        if x == char:
            split = input.split(char)
    return split

```

- Solution:

```
def delimiter(input, char):
    for x in reversed(input):
        if x == char:
            split = input.rsplit(char, 1)
            return split
```

- Feedback: The inputted function needs to be changed so that it only splits the string by the last instance of the delimiter, and also includes the delimiter in the split. This can be done by using the string method 'rsplit()' instead of 'split()'.
- Score: 2/3; GPT-3 identifies the error in the inputted code, but does not correctly fix the error in its solution code.

Test 8 (Topic 2, Correct Code)

- Prompt: Write a Python program to sort a list of elements using Cycle sort.
- Inputted Code:

```
def cycleSort(array):
    writes = 0

    # Loop through the array to find cycles to rotate.
    for cycleStart in range(0, len(array) - 1):
        item = array[cycleStart]

        # Find where to put the item.
        pos = cycleStart
        for i in range(cycleStart + 1, len(array)):
            if array[i] < item:
                pos += 1

        # If the item is already there, this is not a cycle.
        if pos == cycleStart:
            continue

        # Otherwise, put the item there or right after any duplicates.
        while item == array[pos]:
            pos += 1
        array[pos], item = item, array[pos]
        writes += 1

        # Rotate the rest of the cycle.
        while pos != cycleStart:

            # Find where to put the item.
            pos = cycleStart
            for i in range(cycleStart + 1, len(array)):
                if array[i] < item:
                    pos += 1

            # Put the item there or right after any duplicates.
            while item == array[pos]:
                pos += 1
            array[pos], item = item, array[pos]
```

```

        writes += 1

    return writes

#implementation of cycle sort from
#https://www.geeksforgeeks.org/cycle-sort/

```

- Solution:

```

def cycleSort(array):
    writes = 0

    # Loop through the array to find cycles to rotate.
    for cycleStart in range(0, len(array) - 1):
        item = array[cycleStart]

        # Find where to put the item.
        pos = cycleStart
        for i in range(cycleStart + 1, len(array)):
            if array[i] < item:
                pos += 1

        # If the item is already there, this is not a cycle.
        if pos == cycleStart:
            continue

        # Otherwise, put the item there or right after any duplicates.
        while item == array[pos]:
            pos += 1
        array[pos], item = item, array[pos]
        writes += 1

        # Rotate the rest of the cycle.
        while pos != cycleStart:

            # Find where to put the item.
            pos = cycleStart
            for i

```

- Feedback: The inputted function is missing indentations, which would result in an error. All lines after the for loop should be indented once.
- Score: 1/3; The feedback is incorrect, and incorrectly states that the code is incorrect. In addition, it does not provide feedback on the implementation of cycle sort.

Test 9 (Topic 2, Correct Code)

- Prompt: Write a Python program to sort a list of elements using the quick sort algorithm.
- Inputted Code:

```

    # Function to find the partition position
def partition(array, low, high):
    # choose the rightmost element as pivot
    pivot = array[high]

```

```

# pointer for greater element
i = low - 1

# traverse through all elements
# compare each element with pivot
for j in range(low, high):
    if array[j] <= pivot:
        # If element smaller than pivot is found
        # swap it with the greater element pointed by i
        i = i + 1

    # Swapping element at i with element at j
    (array[i], array[j]) = (array[j], array[i])

# Swap the pivot element with the greater element specified by i
(array[i + 1], array[high]) = (array[high], array[i + 1])

# Return the position from where partition is done
return i + 1

# function to perform quicksort
def quickSort(array, low, high):
    if low < high:
        # Find pivot element such that
        # element smaller than pivot are on the left
        # element greater than pivot are on the right
        pi = partition(array, low, high)

        # Recursive call on the left of pivot
        quickSort(array, low, pi - 1)

        # Recursive call on the right of pivot
        quickSort(array, pi + 1, high)
#implementation of quick sort from
#https://www.geeksforgeeks.org/quick-sort/

```

- Solution:

```

def quickSort(array, low, high):
    if low < high:
        # Find pivot element such that
            # element smaller than pivot are on the left
            # element greater than pivot are on the right
            pi = partition(array, low, high)
        # Recursive call on the left of pivot
            quickSort(array, low, pi - 1)
        # Recursive call on the right of pivot
            quickSort(array, pi + 1, high)

```

- Feedback: The inputted function correctly finds the partition position, but it does not correctly sort the elements before and after the partition. This can be fixed by adding a recursive call to quicksort on the left and right side of the partition.
- Score: 1/3; The inputted code does implement a recursive call to quicksort on the left and right side of the partition, which GPT is not able to identify.

Test 10 (Topic 2, Correct Code)

- Prompt: Write a Python program for sequential search.
- Inputted Code:

```
def sequentialSearch(arr, val):  
    for x in arr:  
        if x == val:  
            return arr.index(x)  
            return (True, arr.index(x))  
    return False
```

- Solution:

```
def sequentialSearch(arr, val):  
    for x in arr:  
        if x == val:  
            return (True, arr.index(x))  
    else:  
        return (False, -1)
```

- Feedback: The inputted function correctly searches through the list for the given value, but it only returns True or the index of the value if the value is found. If the value is not found, the function returns False instead of (False, -1). This can be fixed by adding an else statement that returns (False, -1) if the value is not found in the list.
- Score: 2/3; GPT appears to extrapolate from the prompt, as it is never stated that the program should return (False, -1) instead of False if the value is not found in the list. However, it correctly provides feedback about the rest of the requirements.

Test 11 (Topic 1, Gibberish)

- Prompt: Write a Python program to sort unsorted numbers using non-parallelized implementation of odd-even transposition sort.
- Inputted Code:

```
def timsort(arr):  
    question = "Write a Python program to sort unsorted numbers using Timsort."  
    return question
```

- Solution:

```
def oddEvenSort(arr):  
    isSorted = False  
    while isSorted == False:  
        isSorted = True  
        temp = 0  
        for i in range(1, len(arr)-1, 2):  
            if arr[i] > arr[i+1]:  
                arr[i], arr[i+1] = arr[i+1], arr[i]  
                isSorted = False  
  
        for i in range(0, len(arr)-1, 2):  
            if arr[i] > arr[i+1]:  
                arr[i], arr[i+1] = arr[i+1], arr[i]  
                isSorted = False  
    return arr
```

- Feedback: The inputted function is not a sorting function, and therefore will not yield the same output as the solution function.
- Score: 3/3; Although GPT-3 does not explicitly recognize the returned value as gibberish, it correctly identifies that it is not a sorting function, and provides a correct implementation of odd-even transposition sort.

Test 12 (Topic 2, Gibberish Code)

- Prompt: Write a Python program to sort a list of elements using Radix sort.
- Inputted Code:

```
#Radix sort Java implementation

import java.io.*;
import java.util.*;

class Radix {

    // A utility function to get maximum value in arr[]
    static int getMax(int arr[], int n)
    {
        int mx = arr[0];
        for (int i = 1; i < n; i++)
            if (arr[i] > mx)
                mx = arr[i];
        return mx;
    }

    // A function to do counting sort of arr[] according to
    // the digit represented by exp.
    static void countSort(int arr[], int n, int exp)
    {
        int output[] = new int[n]; // output array
        int i;
        int count[] = new int[10];
        Arrays.fill(count, 0);

        // Store count of occurrences in count[]
        for (i = 0; i < n; i++)
            count[(arr[i] / exp) % 10]++;

        // Change count[i] so that count[i] now contains
        // actual position of this digit in output[]
        for (i = 1; i < 10; i++)
            count[i] += count[i - 1];

        // Build the output array
        for (i = n - 1; i >= 0; i--) {
            output[count[(arr[i] / exp) % 10] - 1] = arr[i];
            count[(arr[i] / exp) % 10]--;
        }

        // Copy the output array to arr[], so that arr[] now
        // contains sorted numbers according to current
```

```

        // digit
        for (i = 0; i < n; i++)
            arr[i] = output[i];
    }

    // The main function to that sorts arr[] of
    // size n using Radix Sort
    static void radixsort(int arr[], int n)
    {
        // Find the maximum number to know number of digits
        int m = getMax(arr, n);

        // Do counting sort for every digit. Note that
        // instead of passing digit number, exp is passed.
        // exp is 10^i where i is current digit number
        for (int exp = 1; m / exp > 0; exp *= 10)
            countSort(arr, n, exp);
    }

    // A utility function to print an array
    static void print(int arr[], int n)
    {
        for (int i = 0; i < n; i++)
            System.out.print(arr[i] + " ");
    }

    // Main driver method
    public static void main(String[] args)
    {
        int arr[] = { 170, 45, 75, 90, 802, 24, 2, 66 };
        int n = arr.length;

        // Function Call
        radixsort(arr, n);
        print(arr, n);
    }
}
//implementation of Radix sort from
//https://www.geeksforgeeks.org/radix-sort/

```

- Solution:

```

def countingSort(array, place):
    size = len(array)
    output = [0] * size
    count = [0] * 10

    for i in range(0, size):
        index = array[i] // place
        count[index % 10] += 1

    for i in range(1, 10):
        count[i] += count[i - 1]

    i = size - 1

```



```

while i >= 0:
    index = array[i] // place
    output[count[index % 10] - 1] = array[i]
    count[index % 10] -= 1
    i -= 1

for i in range(0, size):
    array[i] = output[i]

def radixSort(array):
    max_element = max(array)
    place = 1
    while max_element // place > 0:
        countingSort(array, place)
        place *= 10

data = [121, 432, 564, 23, 1, 45, 788]
radixSort(data)
print(data)

```

- Feedback: The inputted function is close, but it is missing a few key elements. The function is missing a parameter for the place value, which is needed in order to keep track of which digit is being sorted at each stage. The function also needs to include a line that initializes an output array, which will be used to store the sorted values.
- Score: 1/3; GPT-3 does not identify the inputted code as a Java implementation rather than a Python implementation.

Test 13 (Topic 1, Incorrect Code)

- Prompt: Write a Python program to find the index of a given string at which a given substring starts. If the substring is not found in the given string return 'Not found'.

- Inputted Code:

```

def idx_substring(sub, word):
    word_len = len(sub)
    for i in range(len(word)-len(sub)):
        if word[i:i + word_len] == sub:
            return i
    return -1

```

- Solution:

```

def idx_substring(sub, word):
    word_len = len(sub)
    for i in range(len(word)-len(sub)):
        if word[i:i + word_len] == sub:
            return i
    return "Not found"

```

- Feedback: The inputted function iterates through the word to find the substring, but it doesn't return the index of the first instance of the substring if it's not found.
- Score: 3/3

Test 14 (Topic 1, Partially Correct Code)

- Prompt: Write a Python program to make two given strings (lower case, may or may not be of the same length) anagrams removing any characters from any of the strings.

- Inputted Code:

```
def idx_substring(sub, word):
    word_len = len(sub)
    for i in range(len(word)-len(sub)):
        if word[i:i + word_len] == sub:
            return i
    return -1
```

- Solution:

```
def anagram(s1, s2):
    # the sorted strings are anagram if they are equal
    if(sorted(s1)== sorted(s2)):
        print("The two strings are anagram of each other")
    else:
        print("The two strings are not anagram of each other")
```

- Feedback: The inputted function is almost correct, but it is missing a return statement at the end of the function. The function should be:

```
def anagram(str1, str2):
    def mapper(w):
        temp = dict()
        for i in w:
            if i not in temp:
                temp[i] = 1
            else:
                temp[i] += 1
        return temp
    str1_map1 = mapper(str1)
    str2_map2 = mapper(str2)
    sol = 0
    for i in str2_map2.keys():
        if i not in str1_map1:
            sol += str2_map2[i]
        else:
            sol += max(0, str2_map2[i]-str1_map1[i])
    for i in str1_map1.keys():
        if i not in str2_map2:
            sol += str1_map1[i]
        else:
            sol += max(0, str1_map1[i]-str2_map2[i])
    return sol
```

- Score: 3/3

Test 15 (Topic 1, Correct Code)

- Prompt: Write a Python program to remove all consecutive duplicates of a given string.

- Inputted Code:

```
def remove_dupes(x):
    if len(x) < 2:
        return x
    if x[0] != x[1]:
        return x[0] + remove_dupes(x[1:])
    return remove_dupes(x[1:])
```

- Solution:

```
def remove_consecutive_duplicates(s):
    # initialize an empty string
    new_string = ""

    # iterate through the string
    for i in range(len(s)):
        # if the current character is different than the next character
        if i == len(s) - 1 or s[i] != s[i+1]:
            # append the current character to the new string
            new_string += s[i]

    return new_string
```

- Feedback: The inputted function correctly checks for duplicates, but it only removes one instance of each duplicate instead of all instances of each duplicate. This can be fixed by adding an else statement that concatenates the first character of the string with the result of the function called on the rest of the string. The inputted function should be:

```
def remove_dupes(x):
    if len(x) < 2:
        return x

    if x[0] != x[1]:
        return x[0] + remove_dupes(x[1:])

    else:
        return remove_dupes(x[1:])
```

- Score: 1/3; Submitted code already had x[0] included; GPT-3 didn't recognize it and told us to add it to the solution code.

Test 16 (Topic 1, Correct Code)

- Prompt: Write a Python function to create the HTML string with tags around the word(s).
- Inputted Code:

```
def add_tags(x, y):
    return "<" + x + ">" + y + "</" + x + ">"
```

- Solution:

```
def add_tags(tag, word):
    return "<%s>%s</%s>" % (tag, word, tag)
```

- Feedback: The inputted function correctly returns a string with the opening and closing tags, but uses string concatenation instead of string interpolation to insert the values of the variables 'x' and 'y'. This can be fixed by using string interpolation, as shown in the solution function.

- Score: 3/3; Gave an alternative, more efficient way even though the method submitted was running perfectly fine.

Test 17 (Topic 1, Partially Correct Code)

- Prompt: Write a Python program to count repeated characters in a string.
- Inputted Query:

```
def repeated_chars(x):
    sol = dict()
    for i in x:
        if i in sol.keys():
            sol[i] += 1
        else:
            sol[i] = 1
    return sol
```

- Solution:

```
def count_repeated_characters(str):
    # initialize count to 0
    count = 0

    # create an empty dictionary
    hash = {}

    # traverse the string and store
    # count of each character
    for i in range(0, len(str)):
        if (str[i] in hash):
            hash[str[i]] += 1
        else:
            hash[str[i]] = 1

    # traverse the dictionary and
    # print all repeated characters
    for key in hash:
        if (hash[key] > 1):
            print(key, hash[key])
```

- Feedback: The inputted function correctly creates a dictionary with the characters in the string as keys and the number of times they appear as values. However, it does not print anything from the dictionary, so the function does not provide the same output as the solution function. To fix this, the inputted function should print the characters and the number of times they appear, like this:

```
def repeated_chars(x):
    sol = dict()
    for i in x:
        if i in sol.keys():
            sol[i] += 1

        else:
            sol[i] = 1

    for i, j in sol.items():
        if j > 1:
            print(i, str(j))
```

- Score: 3/3

Test 18 (Topic 1, Incorrect Code)

- Prompt: Write a Python program to get the last part of a string before a specified character.
- Inputted Code:

```
def breaker(i, j):
    return i[:i.index(j)]
```

- Solution:

```
def get_last_part(url, part):
    # your code here
    return
```

- Feedback: The inputted function does not work for the second test case, where the part to be removed is not at the end of the url. This can be fixed by using the rindex function instead of index, which returns the last index of the part instead of the first index. The inputted function should be:

```
def breaker(i, j):
    return i[:i.rindex(j)]
```

- Score: Score: 3/3; submitted solution doesn't use rindex, model correctly tells us to use it.

Test 19 (Topic 1, Partially Correct Code)

- Prompt: Write a Python program to find the first repeated character of a given string where the index of first occurrence is smallest.
- Inputted Code:

```
def first_repeated_char_smallest_distance(str1):
    temp = {}
    for i in str1:
        if i in temp:
            return str1.index(i);
        else:
            temp[i] = 0
    return 'None'
```

- Solution:

```
def first_repeated_char(str1):
    h = {}
    for ch in str1:
        if ch in h:
            return ch
        else:
            h[ch] = 0
    return
```

- Feedback: The inputted function is almost correct, but it is returning the index of the first repeated character instead of just the character. This can be fixed by removing ", str1.index(i)" from the return statement. The inputted function should be:

```
def first_repeated_char_smallest_distance(str1):
    temp = {}
    for i in str1:
        if i in temp:
            return i
        else:
            temp[i] = 0
    return 'None'
```

- Score: 3/3; submitted solution returns index, and model correctly tells us to return character.

Test 20 (Topic 1, Partially Correct Code)

- Prompt: Write a Python program to find the index of a given string at which a given substring starts. If the substring is not found in the given string return 'Not found'.
- Inputted Code:

```
def first_repeated_char_smallest_distance(str1):
    def first_repeated_char(str1):
        h = {}
        for ch in str1:
            if ch in h:
                return ch
            else:
                h[ch] = 0
        return
```

- Solution:

```
def first_repeated_char_smallest_distance(str1):
    temp = {}
    for i in str1:
        if i in temp:
            return i
        else:
            temp[i] = 0
    return 'None'
```

- Feedback: The inputted function is close, but it iterates through each letter of the string instead of each word. It also doesn't account for the possibility that the substring might be found multiple times in the string. This can be fixed by splitting the string into a list of words, and then iterating through that list. The inputted function should be:

```
def find_index(my_str, sub_str):
    words = my_str.split()
    for i in range(len(words)):
        if words[i] == sub_str:
            return i
    return "Not found"
```

- Score: Score: 2/3; model suggests a different way of implementing function altogether.

Test 21 (Topic 1, Correct Code)

- Prompt: Write a Python program that returns a string sorted alphabetically by the first character of a given string of words. (Q21.py)

- Inputted Code:

```
def Question_21(text):
    return ' '.join(sorted(text.split(), key=lambda c: c[0]))
```

- Solution:

```
def first_char_sort(words):
    # create a list of words sorted by first character
    first_char_sorted = sorted(words, key=lambda word: word[0])
    # join the list of words into a string
    return ' '.join(first_char_sorted)
```

- Feedback: The inputted function correctly splits the string into a list of words and sorts the list by the first character of each word. It then puts it back together- it is correct.
- Score: 3/3

Test 22 (Topic 1, Correct Code)

- Prompt: Write a Python program to find the maximum occurring character in a given string.
- Inputted Code

```
def max_freq(test_str):
    res = Counter(test_str)
    res = max(res.values())
    return res
```

def max_freq(test_str): res = Counter(test_str) res = max(res.values()) return res Solution :

Create an empty list to store all characters and their count in `str ASCII_SIZE = 256`
`ctr = [0]*ASCII_SIZE`
 For each character in str, increment the count in list for i in str: `ctr[ord(i)]+=1`
 return the maximum occurring character `max_char = "max_count = 0`
 for i in str: if `max_count < ctr[ord(i)] : max_char = i max_count = ctr[ord(i)]`
 return `max_char`

- Feedback: The inputted function correctly uses the Counter function to count the occurrences of each character in the string, and then finds the character with the maximum value.

Score: 2/3; Model recognizes other, shorter solution but doesn't tell us that we're missing an import statement (from collections import Counter)

Test 23 (Topic 1, Correct Code)

- Prompt: Write a Python program that accepts a comma separated sequence of words as input and prints the unique words in sorted form (alphanumerically).
- Inputted Code:

```
def sequencer(items):
    words = [word for word in items.split(",")]
    print(" , ".join(sorted(list(set(words)))))
```

- Solution:

```
def unique_words(words):
    # your code here
    unique_words = []
    for word in words:
        if word not in unique_words:
            unique_words.append(word)
    return sorted(unique_words)
```

- Feedback: The inputted function correctly splits the string into a list of words, but it then uses the set function to create a set of unique words, which does not preserve the order of the words in the original string. This can be fixed by using the list function instead of the set function. The inputted function should be:

```
def sequencer(items):
    words = [word for word in items.split(",")]
    print(",".join(sorted(list(words))))
```

- Score: 2/3; modified function outputted by model is identical to submitted solution.

Test 24 (Topic 1, Correct Code)

- Prompt: Write a Python program to calculate the sum of two numbers given as strings. Return the result in the same string representation.

- Inputted Code:

```
def test(x, y):
    if (x==" " or y==" "):
        return False
    m = max(len(x), len(y))
    c = 0
    result = ''
    for x, y in zip(x.rjust(m, '0')[::-1], y.rjust(m, '0')[::-1]):
        s = int(x) + int(y) + c
        c = 1 if s > 9 else 0
        result += str(s)[-1]
    return result[::-1]
```

- Solution:

```
def sum_strings(a, b):

    # initialize sum as an empty string
    sum = []

    # calculate length of both string and
    # do not consider common parts of length
    a_len = len(a)
    b_len = len(b)
    if a_len > b_len:
        for i in range(0, b_len):
            sum.append((int(a[i]) + int(b[i])))
        for i in range(b_len, a_len):
            sum.append(int(a[i]))

    elif a_len < b_len:
```



```

        for i in range(0, a_len):
            sum.append((int(a[i]) + int(b[i])))
        for i in range(a_len, b_len):
            sum.append(int(b[i]))

    else:
        for i in range(0, a_len):
            sum.append((int(a[i]) + int(b[i])))

    # iterate through the sum array
    # and convert element to string and
    # store it in string
    carry = 0
    for i in range(len(sum)):
        sum[i] += carry
        carry = sum[i] // 10
        sum[i] = str(sum[i] % 10)

    # join the elements of sum list
    # and return it as required sum
    return "".join(sum)

```

- Feedback: The inputted function correctly calculates the sum of the two strings.
- Score: 3/3

Test 25 (Topic 1, Correct Code)

- Prompt: Write a Python program to count uppercase, lowercase, special character and numeric values in a given string.
- Inputted Code:

```

def counter(x):
    uc, lc, nc, sc = 0, 0, 0, 0
    for i in range(len(str)):
        if str[i] >= 'A' and str[i] <= 'Z': uc += 1
        elif str[i] >= 'a' and str[i] <= 'z': lc += 1
        elif str[i] >= '0' and str[i] <= '9': nc += 1
        else: sc += 1
    return uc, lc, nc, sc

```

- Solution:

```

def string_operation(str1):

    upper, lower, number, special = 0, 0, 0, 0

    for i in range(len(str1)):

        if str1[i] >= 'A' and str1[i] <= 'Z':
            upper += 1

        elif str1[i] >= 'a' and str1[i] <= 'z':
            lower += 1

```

```

        elif str1[i] >= '0' and str1[i] <= '9':
            number += 1

    else:
        special += 1

    print('Upper_case_characters==%s' %upper)
    print('Lower_case_characters==%s' %lower)
    print('Number_of_digits==%s' %number)
    print('Special_characters==%s' %special)

```

- Feedback: The function written prints the counts of the uppercase, lowercase, special character, and numeric values in a given string.
- Score: 3/3