**Ideation**

This structured approach outlines the decision-making process and considerations at each step, ensuring a thoughtful and user-centric development journey for the data analysis interview preparation tool.

1. **Platform Selection: Chatbot vs. Webpage**

**Objective:** Decide on the most suitable platform, considering available resources and optimizing user experience.

**Considerations:** Evaluate resource constraints and user preferences. Opt for a webpage due to the absence of a full-stack developer in the team.

1. **Technology Choice for the webpage**

**Objective:** Select a technology stack that aligns with the team's skillset.

**Decision:** Opt for Streamlit, a user-friendly Python library for creating web applications, enabling the team to build a responsive and interactive webpage without extensive full-stack development expertise.

1. **Webpage Design and Feature Planning**

**Objective:** Envision the layout and functionality of the webpage.

**Considerations:** Focus on user-friendly design. Plan features such as dynamic difficulty levels, interactive navigation, and a comprehensive feedback system to enhance the user experience.

1. **Question Type and Resource Optimization**

**Objective:** Determine the type of questions to be included and establish resource-efficient methods.

**Considerations:** Choose a mix of technical and behavioral questions. Implement time limits to simulate real interview conditions. Optimize resource usage for scalability.

1. **VertexAI vs. OpenAI**

**Objective:** Opt for a fitting platform for training a Language Model (LLM) and generating interview questions.

**Considerations:** Evaluate the pros and cons of VertexAI and OpenAI. Consider factors such as ease of integration, customization capabilities, and the ability to generate diverse and relevant questions, budget, and time, and skills.

1. **Feedback Timing Decision: Real-Time vs. End-of-Session**

**Objective:** Decide when to provide feedback in the interview preparation tool.

**Decision:** Opt for end-of-session feedback to simulate real interview conditions. This approach aligns with industry practices, offering a comprehensive assessment and allowing users to reflect on their overall performance.

**Development Stage**

1. **Training Database Creation**

In the endeavor to create a dataset of data analysis interview questions encompassing both technical and behavioral dimensions for training the LLM, the methodology employed leveraged ChatGPT for question generation.

The process involved instructing ChatGPT to craft questions across varying difficulty levels, specifically categorized as Easy, Medium, and Hard. A representative example of this process would entail directing the model to generate, for instance, 10 Easy-level technical questions tailored for data analysis interviews. The utilization of ChatGPT as a tool for question creation contributes to the systematic development of a diverse and comprehensive dataset, allowing for effective few-shot learning strategies in the context of data analysis interview preparation.

1. **Few-Shot Learning**

check\_database\_exists()

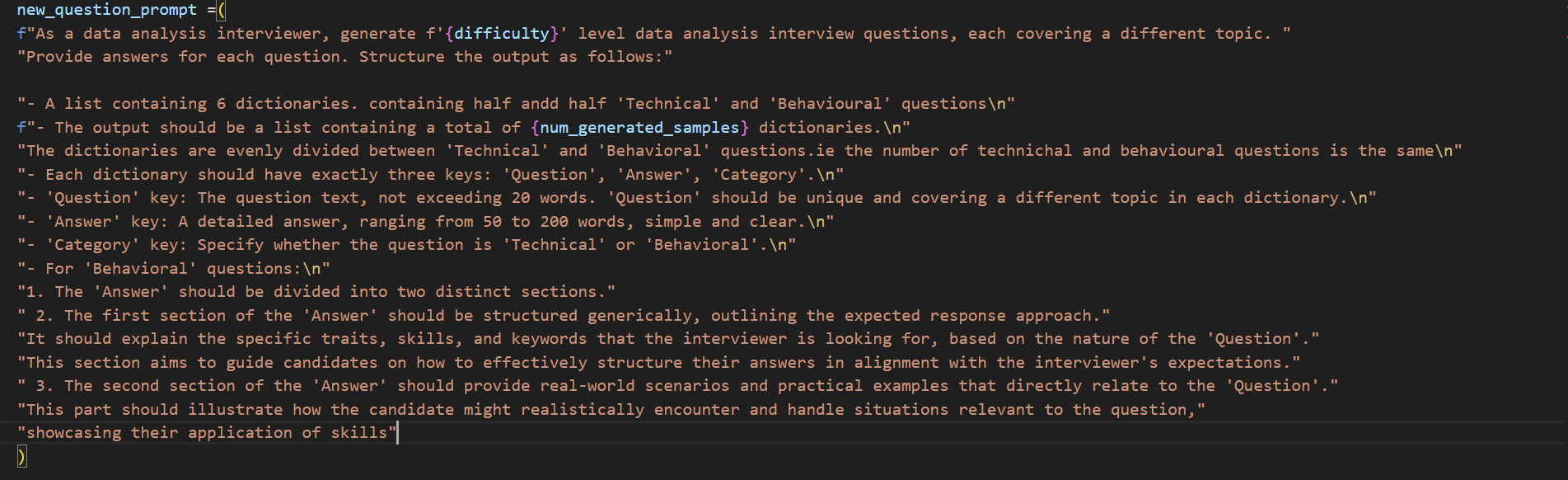
The execution of the aforementioned function involves verifying the presence of files within the 'data' folder, specifically checking for files that represent different difficulty levels [Easy, Medium, Hard], created through the process of few-shot learning.

If any of these files does not exist within the function, we call the following function:

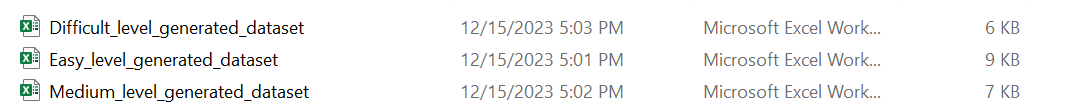
 train\_model\_to\_generate\_data(difficulty\_level,num\_generated\_samples=10)

a process that facilitates few-shot learning based on the training database and the provided prompt.

The utilized model is GPT-3.5-turbo-16k

 **The prompt used to generate new data**

**The generated data will be saved in the 'data' folder with the following structure:**



In broad terms, after determining the difficulty level, the code randomly chooses a question from the associated generated dataset file. It then displays the question to the user and save the user’s answers to a list.

1. **User’s Answer Evaluation**

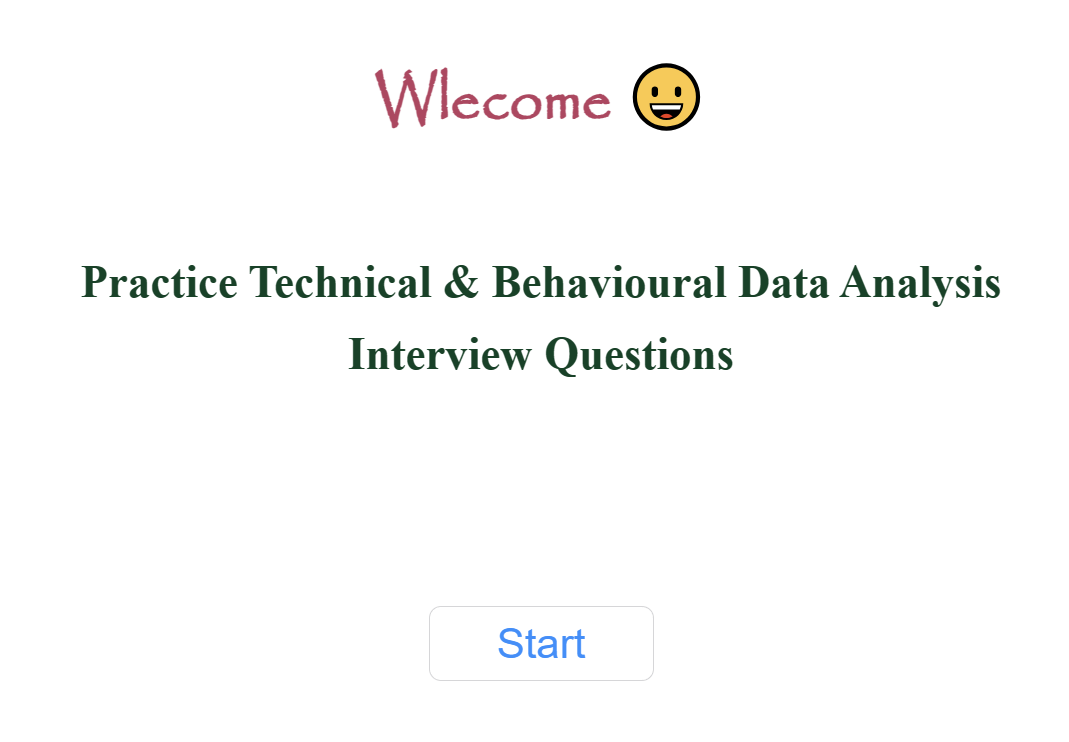
The feedback analysis depended on the cosine similarity score, where the anticipated answer and the user's response were encoded using the sentence transformer **"all-MiniLM-L6-v2"** A threshold of 0.7 was applied for cosine similarity to determine a satisfactory answer. The feedback was communicated to the user through two methods.

* Feedback
* Star rating

**The Design**

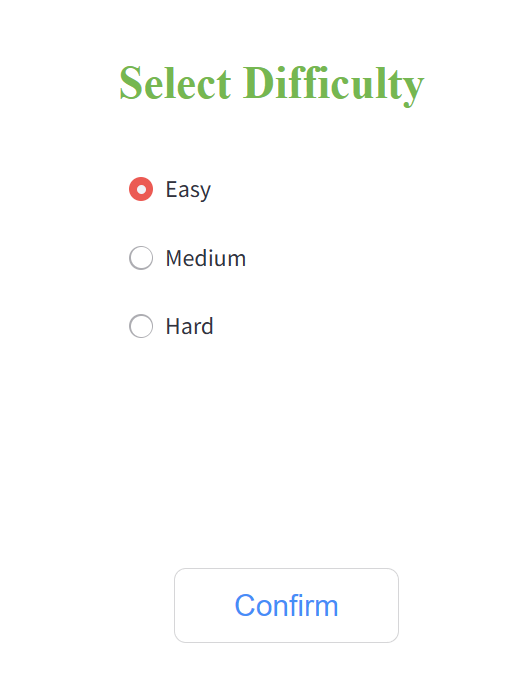
1. **Start Page**

This is the initial landing page that users encounter upon accessing the webpage link. Here, users are prompted to initiate the quiz experience by clicking the 'Start' button. The quiz offers an opportunity to test and expand their knowledge. Join us in this journey of discovery and click 'Start' to begin the quiz challenge.



1. **Selecting Difficulty**

On this page, the user selects the difficulty level for the quiz.

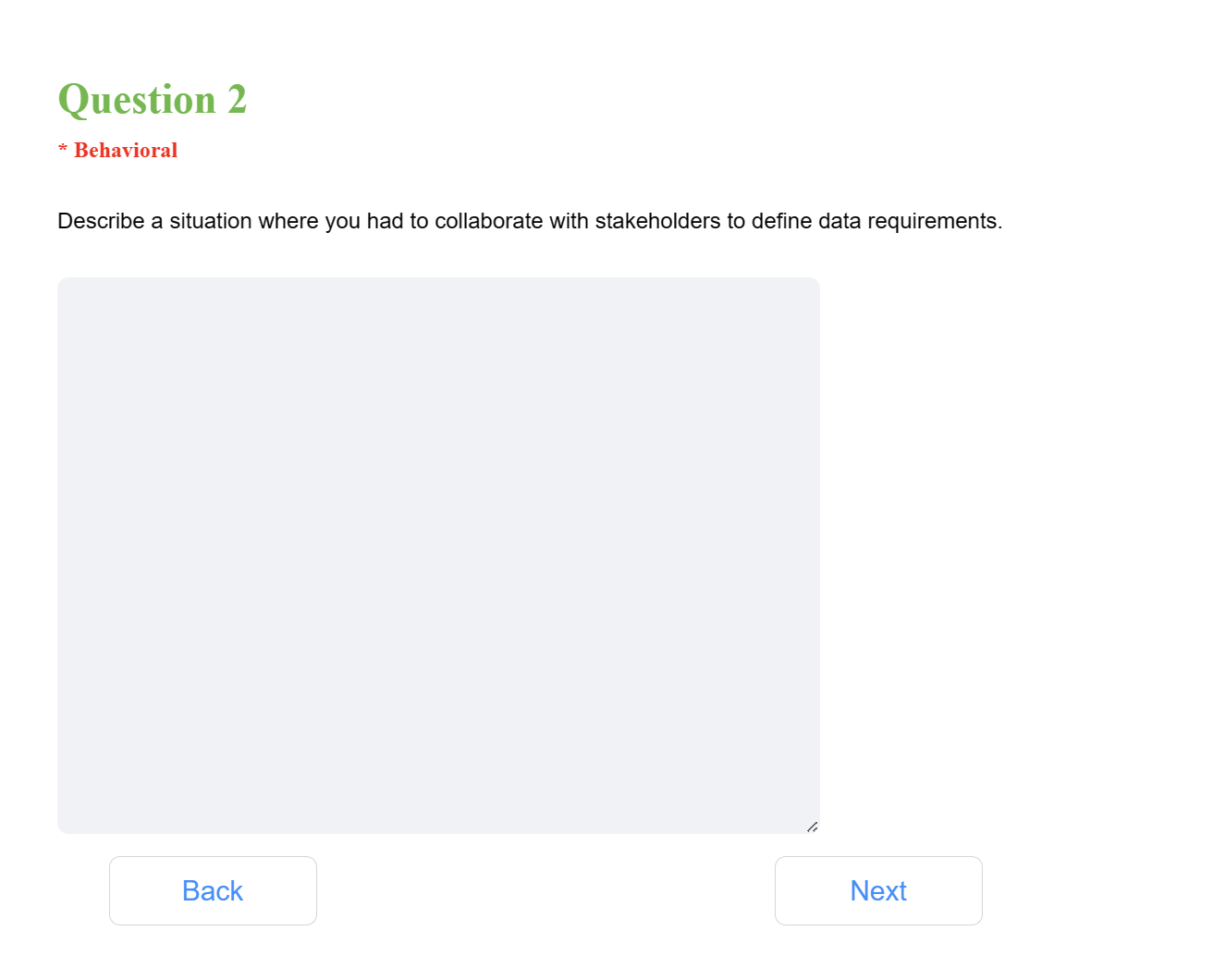


1. **Questions pages**

Here, a text area is provided for users to input their answers. Throughout these pages, both 'Back' and 'Next' buttons are available, affording users the flexibility to modify or review their responses. On the final question page, a 'Submit' button will indicate to the user that this is the last question in the quiz.

Each page is meticulously designed to display the question number along with its corresponding type, categorized as either behavioral or technical.

**Examples of questions:**

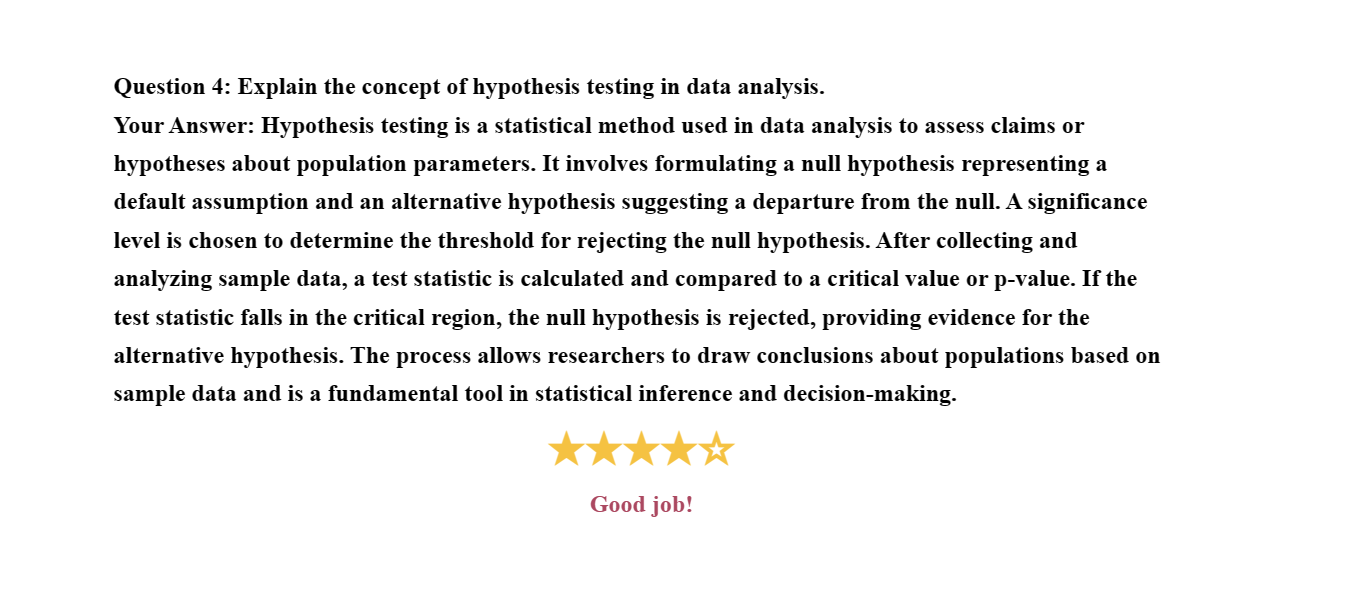


A screenshot of a question

Description automatically generated

1. **Feedback page**

In the feedback page, all questions along with the user's answers will be displayed, accompanied by a star rating and verbal feedback.

**CExample of a good answer:**

**Example of an insufficient answer:**



**Product Testing**

1. **Bugs Rectification**

a. Enable user answer editing functionality upon pressing the back button.

b. Prevent answer reset when utilizing the back button.

c. Optimize answer storage within a list of constant dimensions, eschewing the use of the append method. This ensures seamless updates at the same index when initiated by the user.

1. **UI/UX Enhancement**

a. Revise the color scheme of the initial page to enhance text visibility for a more user-friendly experience.

b. Incorporate a categorization feature indicating the question type (Technical/Behavioral) beneath each respective question for improved user understanding.

c. Letting users go back to check or change their previous answers.