The assignment was to create a chatbot using a publicly hosted server. I accomplished this by using a Virtual Machine through Google Cloud Platform to run a Flask server, which then uses Google's Gemini to respond. The server has a chat route, which allows it to be interacted with through the browser or accessed as an API call through an endpoint request. Through the alternative API endpoint, I created a Discord bot that uses this call to talk to the chatbot directly and is accessible once imported into a server.

For the overall data, I wanted it to be golf-related. Specifically, I had just watched the 2025 Masters, which Rory McIlroy won. I thought it would be interesting to go through the data. Unfortunately, lots of golf datasets weren't Masters related and/or were hidden behind a paywall. I was going to try to scrape The Masters homepage, but luckily found someone else who had already done that exact thing, so I just used their dataset. Unfortunately, it doesn't include any 2025 information, but it still contains 2024 for topics such as winners, win counts, and scores for each day across the 4 day tournament. For my API dataset, I was told a call to an LLM like Gemini was sufficient, likely because it holds lots of data and still requires processing a request and response, which is the core concept of an API, and still shows mastery.

For my ETL, I had to begin with EDA to see what needed to be cleaned up. Across the three Master's datasets, only one had columns with NaN values, and it wasn't an integral column, so I dropped it. For superfluous columns, there was only one date of birth. With these datasets finished, I loaded them as JSON to be passed to Gemini so it could understand them. While Gemini knows lots of old information, it lacks new and current information, which means it is very well complemented by the new Master's data that it does not know about.

In the process of creating this project, I encountered many challenges. To begin with, I had never interacted with an LLM through an endpoint, so I had to look at Gemini's documentation to figure out how to pass it information as a request and how to handle its responses. Beyond just interacting, I had to deal with the idea of prompt engineering and chat persistence. Depending on the prompt, Gemini would ignore some questions while answering others, or repeatedly say its name, or exhibit some other illogical behavior. For chat persistence, I had to add an ordering because otherwise it would sometimes answer out of order or repeat its answers. Though minimal, these were still roadblocks I discovered and had to get around. The next big challenge was understanding Flask. I have never used Flask outside of this class, so I had to look at its documentation to make sure I could handle all the requirements of the assignment. Similarly, I had never created a Discord bot, so that was a learning curve as well, but as also well documented online. All-in-all, the main difficulty was just learning and applying these new skills, which wasn't too problematic overall but required some time.

Through these challenges, I was able to build new skills and refine old ones. One key learning is how API's work. Through making my own, I gained a strong understanding of how they work and how to interact with them. Another was an understanding of Discord bots and LLM interactions. I thought both to be far out of range of my skill set, but through this project, I was able to learn that wasn't the case. I was able to fully integrate and understand them, which was very rewarding. Lastly was the classic "time management". If I had more time, I definitely would have made the frontend more robust. This was not the goal of the project, and as such, it got the least amount of time devoted to it, but is nonetheless an important aspect.

Project-based learning is always a great way to build new skills. This project is no exception– it covers a wide range of topics such as API's, LLMs, bots, and frontend UI with HTML and CSS. Through repeating known skills and learning new ones, I can confidently say I was able to become a more refined and skillful data scientist.