1. Introduce yourself, explain your connection to IGN, and tell us why we should pick you to participate in IGN's Code Foo program. Show your enthusiasm and passion for IGN in the form of a YouTube/Vimeo video, written document, audio track, or whatever format you feel most comfortable with.

Hello,

My name is Dimitar Dimitrov and I'm currently in my senior year as a software engineer at San Jose State University. Due to my interest in games, I would occasionally watch IGN's YouTube channel or look at game reviews on their website. Since I am highly invested in programming, while also having a good understanding of what the company has to offer to their consumers, I decided to apply to this program. I am always seeking different opportunities to learn and grow as a programmer, where this position seems like it would be a great learning experience.

I believe I am a strong candidate for the Code Foo program since I have experience in languages such as Java, C/C++, and Python. Besides the listed languages, I have done projects that involved database management and front-end development. Outside of my experience with MongoDB, I also recently started learning SQLite in my free time, since I thought that it might be both interesting and beneficial to learn. I am always in search for new learning opportunities in order to further develop both my soft and hard skills.

Outside of my academic background, I have worked with a company as a freelancer where I was able to collaborate with another programmer in order to read a CSV file and plot the data utilizing a Python script. This tool was used in order to rank how relevant each Amazon web service documentation page was to a search query, where the main goal was to provide customers with a more optimal way of searching for specific services. Additionally, I have worked as a teacher's assistant at San Jose State University for roughly a year. In this position, I was responsible for helping out students and answering any questions that they had about the current lab. Both positions have helped develop my communication, leadership, and problem solving skills. I have also been able to develop good work ethics, where I try to be active and professional at all times.

2. Hisui's New Power Plant

Requested By: Professor Laventon

The Galaxy Team has decided that in order to advance the research on Pokemon and the Pokedex, Jubilife village needs a brand new power plant. Luckily, Professor Laventon has discovered that the Pokemon, Voltorb, is the best candidate to help power the village.

Voltorbs can cleanly and efficiently produce electricity. An average Voltorb is about 0.5m (1'08") tall and weighs 10.4 kg (22.8lbs). However, they are uncommon and are only found in the Sacred Plaza.

Objective: How many Voltorbs will you need to catch to fully power the village. Describe each step in your thought process.

Some details to note:
Voltorb size: 0.5m
weight 10.4kg
Uncommon to find, only found in the Sacred Plaza

The first concern with a project such as this would be the amount of watts required to generate enough power for the entire village. This should factor in each residential power consumption, while also making sure that there is enough energy if some residents go past the set limit. A rough estimate can be created by verifying how much power the average household uses up in a single day and multiplying that value by the amount of houses in the village. Another approach, that would be more time consuming, would involve collecting data on how much power each household uses, then obtaining the summation of the entire data set.

Once there is a rough estimate on the amount of power required for the village, then the next step would be to take into consideration transportation needs. Since each Voltorb weighs 10.4kg (22.8lbs) and has a height of 0.5m (1'08"), then the method of transportation will have to take into account two attributes. These attributes will involve the transportation's available volume (carrying capacity) and the maximum supported weight. Transportation will be an important factor because It will require the Voltorbs to be transported from the Sacred Plaza to the Jubilife village.

Once the transportation is taken care of, catching them would be another factor. From the information provided, they are categorized as uncommon. Assuming that it could take a significant amount of time to collect and transport enough Voltorbs to power an entire village, then the currently transported Voltorbs can be used in conjunction with the old power plant

model. This approach will not provide 100% clean and efficient electricity, but it will guarantee that some of the generated electricity will be clean and efficient. As more Voltorbs are transported and integrated in the new power plant model, the higher the percentage will increase.

For a general equation on how many Voltorbs would be required to power an entire village:

0 = Total_power_required - (Voltorb_Amount * Voltorb_power_output)

Where the total amount of power required subtracted by the total amount of volt orbs multiplied by a Voltorb's power output should result in 0. If the result is 0, then that indicates that all the village's power is generated through Voltorbs. For the amount of Voltorbs required, the equation could look like this:

Voltorb_amount = Total_power_required / Voltorb_power_output

3. Back End:
1.Using this CSV, build a service to pull/read the content of the file, and store it in a SQL database. Explain how you are storing, sanitizing, normalizing, and indexing the data.

When I stored the database, I imported the CSV file using SQLite Studio's desktop application, and then exported the database so I could utilize it in my Python script. The database was indexed based on how it was entered in the CSV file, where each row has the same columns provided in the file and a unique ID value.

I was not sure how to normalize the database, since I am still fairly new to databases (SQL/SQLite), but I did make an attempt at sanitizing the data in my Python script. I created a method called "sanitize_database()" where it reads the CSV file and removes data such as "<p>". Besides that, my sanitize_database() function also strips additional "\n" characters that might be stored in the database. This was all done in a variable called executeCommand, where the command updates the SQLite database with the new updated changes.

2. Build an API service that utilizes your database. Include at least 3 endpoints that allow for clients to request data based on how you decided to store it. In addition, incorporate data/response filtering and/or sorting based on additional parameters supplied by the client utilizing your endpoint(s). Explain your decisions and methods of implementation.

The API service that I created allows for the user to look at all the possible queries in the database, or look for specific queries. If the user wants to see the entire database, but at the same time limit it to specific columns, they can select the "View all queries" option. This allows the

user to enter inputs that would allow them to see specific columns, such as ID, name, description, etc.

The more intricate method that was created was the search for a specific query option. This option allows the user to ask for specific categories, and a term to search for in that category. If the user is interested, they can also sort the output based on ascending or descending order on a specific column. The user can also limit the output to the top N results. This allows users to search for specific categories, and sort those categories based on different columns. For example: the user can search for all the rows that contain the "Horror" genre tag. Next, the user has the ability to categorize the searched data based on a specific column order. For example, the user could ask to sort based on the review score in ascending order. The last option it allows for the user to do is to only view the top N results. For example, if there are many rows that the user is provided with, the user has the ability to see the first top 5 rows.

For my implementation, I tried to make it class based where I had a class called QueryEngine. This class incorporates all the query functionalities, in order to make the code more organized and easier to modify in the future. Besides that, my implementation is based on user input, and for each user input I create a command string that is executed using connection.cursor(). During all the inputs, I tried to prevent the user from being able to manually enter input, and instead I had pre-defined inputs (such as the numbers 1-17). The only exception was when it asks for a term to search for, where the user has the ability to enter any type of string they want. This can potentially result in issues with malicious code injection, but it can be fixed by looking for specific edge cases in the user input (such as brackets/parenthesis/or SQL commands/similar syntax). I decided to allow the user to enter any type of string in order to save time since this is a prototype design and there are many different types of search options that the user can look for. If a system like this was to be published for general user usage, then issues such as this would have to be addressed.