Final Submission Write-Up

1. Project Design

**Wolfram: Full Result and Conversional API**

***Project Design***

* Connect to WolframAPI
* Using the Full Result API to search math answers
* Using the Full Result API to search simple questions
* Using the Conversional API to search for a question
* Use Conversional API to search for a follow up question

2. Test Cases

Test Case 1: Simple Math

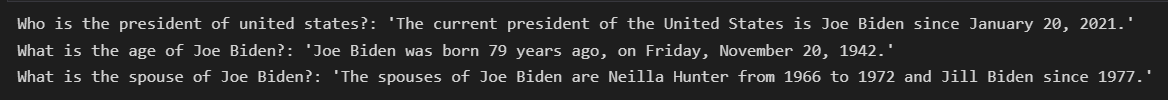


Test Case 2: Simple question

Text

Description automatically generated

Test Case 3: Conversional follow up question



3. Learning Experience

The project 6 is a project that I will save for future research since I am interested in manipulating APIs from different platforms. The assignment was about to analyze get in hands with Full Result and Conversional APIs from Wolfram Alpha website. I learned how to connect the Wolfram API with Python by using the request library, and how to extract data information from JSON files.

In the Full Result API, I needed to pass the question in the  and pass the query\_url in the request.get() in order to get the JSON file.The output is a long list of dictionary that contains data. In order to access it, I go thought each field by using the same method when I access to a list of dictionary. For example, 

I will get whatever is stored in “plaintext” and that will be my data to output it. For the first Full Result API, I use a simple math question “x^2 + 2 = 6” and the result was “x = ± 2.”

For the second enhancement of the Full Result API, I decided to query a simple question: “President of [country],” where country is chosen by the user. The JSON file was more extend than the previous one, so I needed to go through them again by using list of dictionaries. There were 4 kinds of information when querying “the president of [country]”. One the basic result, two basic information, three related information, and four former presidents. The data that was stored in the “plaintext” was not formatted at all, so I convert them into list then into dictionaries to arrange the data. Finally, once I got them arranged in a dictionary, it was easy to print the data out.

In the Conversional API, it was sort of similar to the Full Result API when set up the query and the request.get().json. The only difference that the query url is different from the first part.  This time, it uses “/v1/conversation.jsp” which is for the conversional API. Now, this JSON file is different from the first part as well. The JSON output is shorter, and it has 3 components: result, conversation id, and host. Host and conversation id are passed in the query\_url and will be useful to make another follow-up question.

Text

Description automatically generated

For the first question, the programs prompts what country he has in mind. Once we have the country, we will pass it to the query. In this case, I used “United States” and the result was:



For the second follow-up question, the program used the conversion id and the host to search to know what question was asked in the previous question and make another related to it. So, the second is: “What is the age of Joe Biden?”, and the result is:



For the enhancement of this part, I did another follow-up question, and I was able to do it using the same procedure as the previous ones by taking the conversation id and host. The question is: “What is the spouse of Joe Biden” and the output is:

However, there are times that some president does not have spouse. This leads to output an error in the code, so in order to fix this, I used (try and except) to prompt an error when there is not data

At the end, the program worked successfully and had not complications retrieving data form these JSON files. I am interested to further investigate about APIs in programs and how to use them in different platforms such as Facebook, Google, Amazon, and others. Overall, it was a fun program to code, and I am happy with the outcome of this assignment.

4. Assumptions

* The key API works
* Not run out of requests to pull data when using the APIs