AIM 5001 M9 Assignment (100 Points)

Working with HTML, JSON, Web Scraping, and Web APIs

***** You may work in small groups of no more than three (3) people for this Project *****

Part I (30 points): Working with HTML and JSON

Pick three of your favorite books on one of your favorite subjects. At least one of the books should have more than one author. For each book, include the title, authors, and two or three other attributes that you find interesting. Take the information that you've selected about these three books, and separately create two files, one of which stores the books' information in HTML (using an html table) and the other of which stores the books' information in JSON format (e.g. "books.html" and "books.json"). To help you better understand the different file structures, you should create each of these files "by hand" unless you're already very comfortable with the file formats. Post the two source files to your GitHub repository, and load them you're your Jupyter Notebook from your online repository. Then, write Python code, using your packages of choice, to load the information from each of the two files you've created into separate PANDAS data frames. Are the two data frames identical?

Your deliverable for Part I of this Assignment is the two source files and your Python code. Package your Python code within a Jupyter notebook (along with your code for Parts II and III below).

Part II (40 points): Scraping the Katz School's "Staff" Web Page

For Part II of this Assignment, you will be using your web scraping skills to extract data from a Katz School web page. Specifically, you will be extracting specific content from the Katz School's "Staff" information page.

- From within your Python environment, access the web page containing the Katz School's "Staff" information (https://www.yu.edu/katz/staff) using whatever Python method you feel is most appropriate.
- Create a BeautifulSoup **class** to parse the page you have downloaded.
- Within the downloaded content of the web page, use your Python skills to locate the **div** with **class="text-only"**, and assign the results to a variable named **staff.**
- Create a Pandas dataframe named **staff_info** having columns **office, name, title, email,** and **phone**. Each column should be capable of storing character strings.
- Within the HTML content stored within your **staff** variable, locate and extract each staff member's name, title, email address (if provided), phone (if provided) and the name of the office to which they are assigned and save these items to your **staff_info** dataframe. Note that if either an email address or phone number is not provided for a staff member, your Python code should place the text string "**N/A**" within the corresponding dataframe column for that staff member. When finished you should have one dataframe row for each staff member listed on the Katz School's "Staff" web page. Note that

it is up to you to determine how to most effectively extract the individual data items for each staff member from the block of HTML data contained within your **staff** variable: You may use whatever HTML extraction and/or string processing methods (e.g., Python's string manipulation functions; regular expressions, etc.) you believe are most appropriate.

For example, for staff member Aaron Ross the valid data values for each of the items you are required to extract from the HTML page are as follows:

Office: Office of the Dean

Name: Aaron Ross

Title: Director of Strategic Initiatives and Deputy to the Dean

Email: aaron.ross2@yu.edu

Phone: 646-592-4148

By contrast, for staff member Paul Russo, the valid data values for each of the items you are required to extract from the HTML page are:

Office: Office of the Dean

Name: Paul Russo

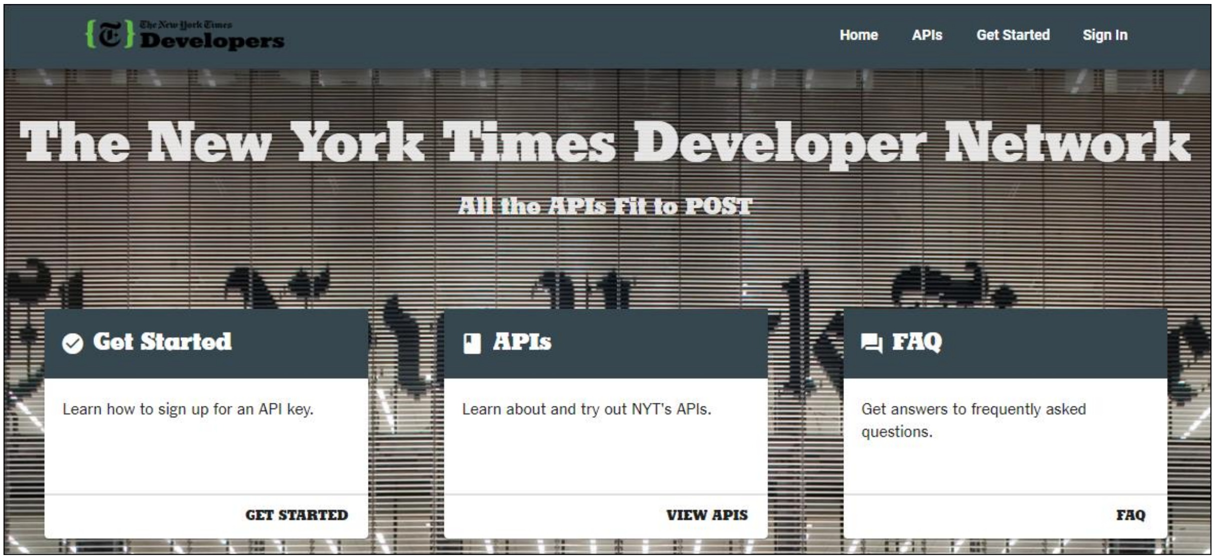
Title: Vice Provost and Dean

Email: N/A

Phone: N/A.

When finished, display the content of your dataframe within your Jupyter Notebook. Do your results appear to accurately represent the content of the web page from which you extracted the data?

Part III (30 points): Working with Web API's



The New York Times Developer Network

All the APIs Fit to POST

Get Started

Learn how to sign up for an API key.

APIs

Learn about and try out NYT's APIs.

FAQ

Get answers to frequently asked questions.

The New York Times web site provides a rich set of APIs, as described here: https://developer.nytimes.com

You'll need to start by signing up for an API key.

Your task is to then choose one of the New York Times APIs and construct an interface in Python to read JSON data accessible via the API and transform that data into a Pandas data frame that is suitable for use in data analysis work. Once you've captured the data you are interested in, do some basic analysis of your choosing using the content of your dataframe and provide a written narrative within formatted Markdown cells explaining the results of your analysis.

Your Jupyter Notebook deliverable should be similar to that of a publication-quality / professional caliber document and should include clearly labeled graphics, high-quality formatting, clearly defined section and sub-section headers, and be free of spelling and grammar errors. Furthermore, your Pythion code should include succinct explanatory comments.

Save all of your work for this project within **a single Jupyter Notebook** and upload / submit it within the provided M9 Assignment Canvas submission portal. Be sure to save your Notebook using the following nomenclature : **first initial_last name_M9_Assn**" (e.g., J_Smith_M9_Assn). ***Small groups should identity all group members at the start of the Jupyter Notebook and each team member should submit their own copy of the team's work within Canvas.***