THIS IS AN INDIVIDUAL TAKE-HOME EXAM. YOU ARE REQUIRED TO SOLVE IT ON YOUR OWN. YOU ARE <u>NOT ALLOWED TO COMMUNICATE ABOUT IT WITH ANYONE ELSE</u>. YOU ARE ALLOWED TO USE <u>YOUR OWN CLASS NOTES</u> AND GOOGLE / BING. <u>THE UC DAVIS HONOR CODE IS IN EFFECT</u>.

Individual Take Home Project

Due: Feb. 4, 2021 11:59<u>am</u>

PART 1   30 points   (we already did 90% of the work in DDR class 4)

Preparation:

Read and understand the FlightAware code we developed in class. If you haven't paid full attention in DDR class 4 (very unlikely), the last 45 minutes of may be very interesting to review; it will make part 1 a breeze.

Use your browser to create a login on Planespotters.net . Please choose a username and password that you are willing to share with us as we ask you to share your code.

Verify that your login works: start an incognito session, navigating to https://www.planespotters.net/user/login , log in, go to your profile page https://www.planespotters.net/member/profile.

Task:

Write code in Java or Python (full source code not markup) that logs into Planespotters.net and verifies that it is logged in my navigating to https://www.planespotters.net/member/profile and checking that your username is displayed on the page. (The task at hand requires at least 2 page requests. I have solved it with 3 page requests but have not tried if less page requests work as well.) Specifically, your code should do the following:

  (1) Access https://www.planespotters.net/user/login using a standard GET request. Read and store the cookies received in the response. In addition, parse the response and read (and store to a string variable) the value of the hidden input field that will (most likely) be required in the login process.
  (2) Make a post request using the cookies from (1) as well as all required name-value-pairs (including your username and passwords).
  (3) Get the cookies from the response of the post request and add them to your cookies from (1).
  (4) Verifies that you are logged in by accessing the profile page https://www.planespotters.net/member/profile with the saved cookies.
  (5) Then, print out the following at the end:

     (a) The entire Jsoup/BeautifulSoup document of your profile page.

     (b) All (combined) cookies from (3).

     (c) A boolean value to show your username is contained in the document in part (6)(a).

Tips:

- Please don't forget to set the full user agent (e.g., "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.96 Safari/537.36 Edg/88.0.705.56") in all of your requests.

- Please make sure that you don't forget to set all hidden input-tag values when trying to log in (there might be one called "csrf" that may be important to set).

Please submit two files: your source code "[yourname]_part1_source.[py OR java]" (source code not markup) as well as a copy of your screen print "[yourname]_part1_screen.txt".

PART 2 70 points   (new but similar to our last assignment)

In this part we go back in time and visit the online store of the most important bookstore in the United States before Amazon came around: https://www.barnesandnoble.com

We will access the Barnes & Noble Top 100: Book Bestsellers and record the old price, new price, and book description for each result.

(1) Do the following IN YOUR BROWSER, no programming required. Please write down your answers in a txt (concise written answers please).

Use your browsers development tools. Open the network tab and analyze the network for the following:

a) go to https://www.barnesandnoble.com and navigate to "Books" > "B&N Top 100". Have a look at the URL.

b) navigate to page 2 of "B&N Top 100". Did the URL change? What type of request is B&N using, GET or POST?

c) which URL variable represents the page number? Is there a variable that represents the number of items per page?

d) try to modify these numbers to view the first page with 40 items of "B&N Top 100".

e) inspecting the HTML source code of (e), how can we access each book in the list of B&N's top 40? How can we access each book's product page URL (e.g., https://www.barnesandnoble.com/w/the-last-thing-he-told-me-laura-dave/1137937512?ean=9781501171345)?

(2) Let's program!

a) Use the URL identified above and write code that loads the first page with 40 items per page of "B&N Top 100".

b) Take your code in (a) and create a list of each book's product page URL. This list should be of length 40.

c) Write a loop that downloads each product page of the top 40 books in "B&N Top 100". I.e., save each of these pages to your computer using a meaningful filename (e.g., "bn_top100_01.htm"). IMPORTANT: Each page request needs to be followed by at least a 5 second pause! Remember, you want your program to mimic your behavior as a human and help you make good purchasing decisions.

d) Write a separate piece of code that loops through the pages you downloaded in (c), opens and parses them into a Python or Java xxxxsoup-object. Next, access the "Overview" section of the page and print the first 100 characters of the overview text to screen.

Please submit three files: your answers "[yourname]_part2_answers.txt", source code "[yourname]_part2_source.[py OR java]" (source code not markup) as well as a copy of your screen print "[yourname]_part2_screen.txt".

Late work policy: The deadline is strict. Projects submitted after 11:59am will be considered late.