LEARN LOG

MICRO APP CHALLENGE

SUDESH KUMAR S.

1. PROBLEMS ENCOUNTERED

- First and foremost I was not familiar with HTML, CSS. So it was little onerous for me to decipher how the response was extracted by BeautifulSoup by passing “id’s” or “classes” or even “tags” like <a> and <p>.

- While entering the URL as input to my program, I found that without http://, the requests package cannot give http requests to the website given by the user. So I spent loads of time to concatenate the hyper text protocols to the broken urls given by the user. Moreover some websites do not produce the result if we do not specify “www.”. Worked on that and combined both to form a function and then made requests using requests package.

- Spent hell lot of time figuring out a way to get the internal and external links from the homepage. Used urlparse() to get the domain name and then used this to check if a link is internal or external

- When I searched for links in the home page, got some broken links but those where extensions proceeding after the Home Page's URL. Got held up there and analysed all the broken links and came up with a logic to append them behind the Home Page's URL.
- I had little knowledge about meta data in a website. So web scrapped my college site and found plenty of information from the meta data they had included in their site. Finally caught the "Name" and "Content" from the <meta> in the <Head>.
- Found it arduous to get the size of the webpages. Found some websites that return the file size. Compared my results with their results and optimised my code to return the file size in KB

2. PROGRAMMING IDIOMS

- Requests
- bs4 (BeautifulSoup4)
- from urlmate import urlpass (User-defined package)
- pandas
- urllib.parse import urlparse
- Regex
- nltk
- sys
- time
- from IPython.display import clear_output

3. Decision Making

- First understood the basics of HTML and CSS. Had an option to just go with requests package and extact content from the web page and learn from that. But I spent couple of hours by inspecting various web pages and scrapped it from scratch using “id’s” and “tags”

- Secondly, used request.get() method to make http requests to the URL’s. Had an option to go with urllib. But found it a bit difficult to understand and many blogs and medium articles were using requests. So went on with that.

- Had an option to just code all the functions in a single file and just make function calls. But I refused to do that and created a package called “url mate” which I built from scratch so that once my features get bigger and bigger everything will be well organised. And especially while working with VSCode and shifting to Jupiter need not copy much. Just import the package and play with it. Future usability is also assured

- To get the “External” and “Internal” links had an option to use scrap and linkextractor. But did not do that since it was a high level package. Proceeded with urlparse() and came up with the logic to use domain names and use them to check if a url is external or internal.

- To remove duplicates had an option to check if a url is present twice or thrice by iterating through it and remove the repeated ones. I went with set() to make the task easier and efficient enough.

- Had an option to use requests and just get the “content length” to find the size of a page, but sometimes content length was not available. So proceeded with urllib to get the page size in bytes

3. Decision Making

- First understood the basics of HTML and CSS. Had an option to just go with requests package and extact content from the web page and learn from that. But I spent couple of hours by inspecting various web pages and scrapped it from scratch using “id’s” and “tags”

- Secondly, used request.get() method to make http requests to the URL’s. Had an option to go with urllib. But found it a bit difficult to understand and many blogs and medium articles were using requests. So went on with that.

- Had an option to just code all the functions in a single file and just make function calls. But I refused to do that and created a package called “url mate” which I built from scratch so that once my features get bigger and bigger everything will be well organised. And especially while working with VSCode and shifting to Jupiter need not copy much. Just import the package and play with it. Future usability is also assured

4. What I learnt?

- Understood the how the internet works.
- Had a basic understanding of various tags in HTML and what is CSS. How does it help in styling the web page.
- Understood what is TLD (Top Level Domain) and how industries charge for each and every domain.
- Also understood how http:// is automatically changed to https:// when we call the web page.
- Had a complete understanding on what is “Web Scrapping” and how to approach web scrapping with various intriguing libraries in python.
- Also learnt how to create a user - defined module or package in python and use it to improve readability and efficiency of code.

4. Summary of the Experience of the coding challenge

- It was a very good experience to take up this micro app challenge. I was really excited to say how apple introduces its new iPhone, My excitement level was ultra pro max because it has been so long since I worked on a problem statement with some time limit.

- It felt more like a hackathon where I used the “Divide and Conquer” technique to break the problem statement into small parts and completed them one by one thereby making it easy.

- Overall I am really happy because the past 48 hours were really really productive. I learnt so many new things like Scrapping the web and at the same time it was like a test for me to understand where I stand. This challenge helped me brush up all the techniques I learnt in python during my 2nd year of college.

- Very good initiative by Exeter Premedia Services to come up with this challenge. I extend my Thanks to Mr. Pari, Mr. Ravi and Mr. Dorai for helping us out through the entire process of this event.

Thank You!