**LEARN LOG**

**Problems I encountered and how I solved them -**

* While extracting links from the home page, a few of the internal links did not have the home page part. (Example : /about , /careers etc).

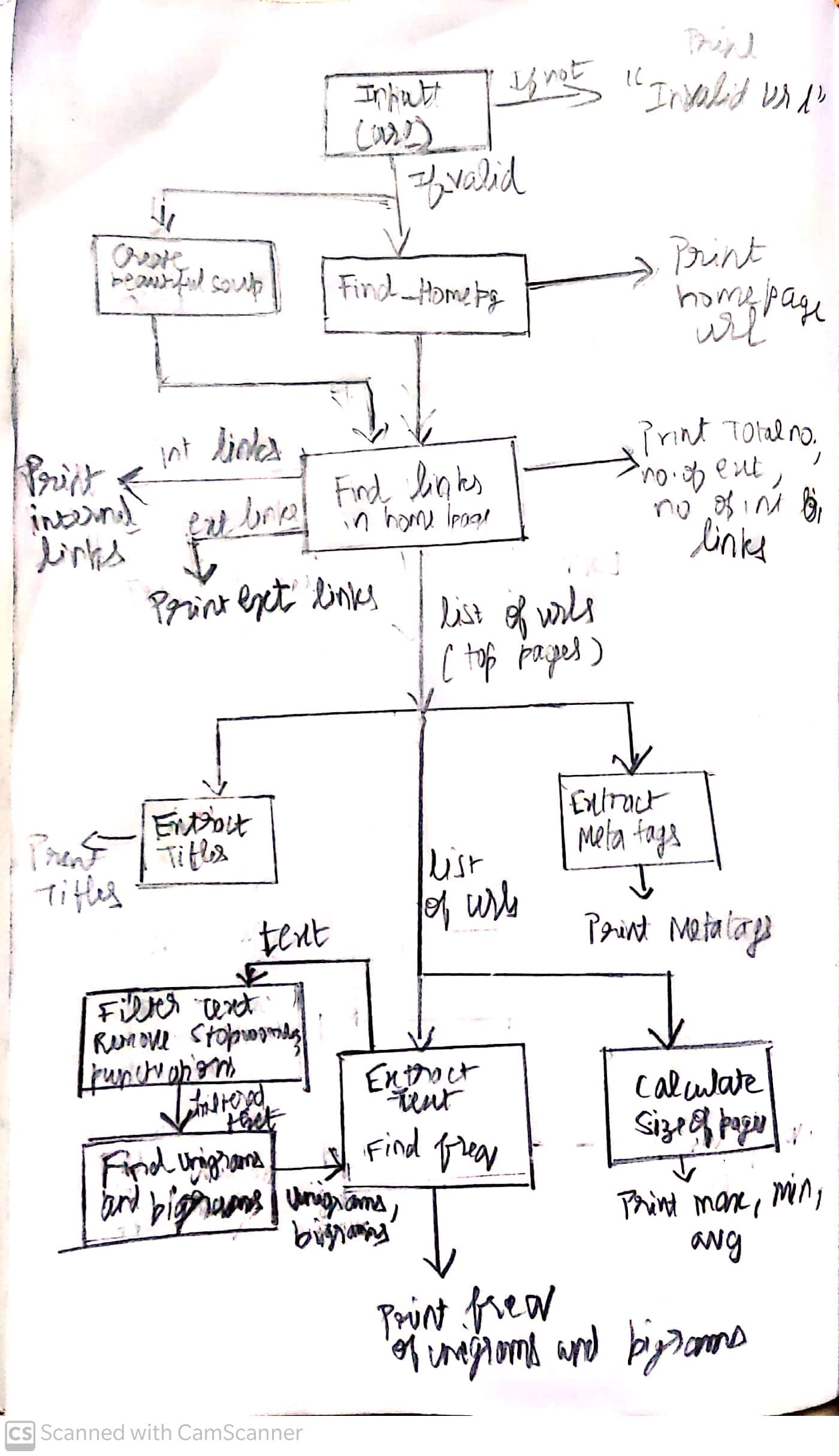
I solved it by identifying them and appending the home page url as prefix. (Example - <https://www.homepage.com/about>)

* A few links were invalid. I did a google search and found a module called validators and filtered out the invalid links.
* The extracted text contained a lot of unwanted spaces and lines. Used .stripped\_strings and concatenated the resulting list of strings, thereby removing extra lines and spaces.
* While extracting the name and content of meta tags, I used a dictionary to store name and content as key value pairs. I realised a few meta tags had the same name but different content. I used lists inside a list instead of storing multiple values for a single key in the dictionary.
* The print function was not able to print a few characters which had a different encoding format. I did a google search and learned how to change the encoding format.
* Counter method worked well for unigrams but did not work for bigrams. Saw the output for every different line by running the module in jupyter notebook and did small changes to make it work.
* A few urls were not responding even though they were valid. Used try and except to ignore unresponsive urls.
* A few characters were undefined. A few images were corrupt. Used try and except in such cases.

**Why did I make a certain decision and what were my options?**

* I had the option of choosing between lxml and html.parser . I chose lxml because it is faster.
* I chose to convert the entire text to lower case because identifying unigrams and bigrams becomes easier and more sensible. Example- Exeter and EXETER would be considered the same if it is converted to lower case, which makes more sense.
* I used for loops instead of while loops because no separate iterator or counter is required for a for loop in python.
* I used regular expressions to remove punctuations instead of defining a set of special characters and using a loop. Regex is just one line of code .
* I decided to use numpy arrays to store sizes of the web pages instead of normal lists. I made this decision because calculating the average size of the page requires looping in case of lists, whereas vectorization is possible in case of numpy arrays which saves time.

**BLOCK DIAGRAM -**

****

**What did I learn?**

* I learnt about HTML tags, CSS and how they are used to build a website.
* I learnt about web scraping and how this technique can be used to collect huge amounts of data.
* I learnt a lot of new packages related to fetching and manipulating urls and text processing.
* I learnt that a strong sense of logic and knowing “how and what to google” can make anyone a good programmer.

**SUMMARY –**

The coding challenge was overall an amazing experience. I must say I enjoyed it thoroughly. After a lot of boring days of online classes, I was deeply involved and interested in building something. It was a different and challenging experience for me. I have built AI models and built robots using microcontrollers in the previous hackathons which I participated. This was a different experience which involved just coding which is my favourite part of any project. This coding challenge helped me get out of my comfort zone and it motivated me to learn and build something new, thereby spending my time productively. Seeing the output being printed on the screen after putting in a lot of effort is really satisfying.