Manasa Cherukupally

11604712

**In-Class-Exercise-02**

**Introduction to ParseHub:**

ParseHub is a very efficient user-friendly web-scraping tool. This tool has the capability to allow the user to scrape the website without writing the code. This tool is useful for individuals who want to perform research by scraping web pages. It is also useful for businesses that want to gather information to research.

There are both free and paid versions of this tool available. With the paid version there are advanced capabilities and the latest features being offered.

**Some Advantages include:**

* ParseHub is available in the cloud, which means it can be directly accessed without installation.
* It is very user-friendly as selecting the web elements is done by clicking.
* There is the chance of Previewing the scraped data, modifying, and editing.
* There are advanced techniques for selecting web elements.
* Exporting with ParseHub is a very easy task.
* The paid version of ParseHub also offers API integration.

**Steps:**

ParseHub follows a simple set of rules.

**Select-** for selecting a web element.

**Correct-** Confirming selected element.

**Command-** adding command to selection.

**Example Project created using ParseHub:**

1. After Installing ParseHub, create an account with your credentials and log in the ParseHub. Select the new project for creating the new scraping project

A screenshot of a computer

Description automatically generated

1. After selecting a new project. Provide the website URL from where the data needs to be scrapped. Here, I have selected the **Wikipedia website** that consists of **the “List of cities by international visitors.”**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Once the URL is pasted, the website page will be loaded. Next, select the web scraping element from the website which you want to extract. Rename the element meaningfully.

A screenshot of a computer

Description automatically generated

1. Select the second element from the website to select all the similar elements from the webpage.

A screenshot of a computer

Description automatically generated

1. Once all the elements are selected. Press the + button beside the selected element tab. And selected relative select to link the other elements with he selected element.

A screenshot of a computer

Description automatically generated

1. Click the first elements and tag their relative elements on the webpage.

A screenshot of a computer

Description automatically generated

1. Select the second element and tag it to its relative element to tag all the elements with their respective relative elements and the element accordingly

A screenshot of a computer

Description automatically generated

1. Similarly select the element and tag another relative element from the webpage that we want in our result. Name the relative element accordingly

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

9. Continue the Process with all the elements and all its relative elements. Name the relative elements accordingly

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Once the all the elements are scraped. Preview the data scraped from the lower right corner from the window. If everything is correct, then press the “Get Data” button to start scraping the data selected.

A screenshot of a computer

Description automatically generated

1. Press the “Run” button to start the scraping from the website.

A screenshot of a web page

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. When everything is ready. Save the file in the format required. For example I have chosen the CSV/Excel file to store the data scraped.

A screenshot of a computer

Description automatically generated

**Output:**

The sample of the output scraped looks like the one below. It consists of 100 rows. The link to output data <https://github.com/ManasaCherukupally1/Manasa_INFO5731_Spring2023/blob/main/Cherukupally_Manasa_wikipedia_results.csv>

The columns include

1. **City\_Rank**-Rank of city
2. **City\_Name**-Name of City
3. **City\_Country**-Country of city
4. **City\_Arrivals**- Number of arrivals in the city in 2018
5. **City\_income**-Income value of the city
6. **City\_Name\_Url**- Url link for city description
7. **City\_country\_Url**-Url link for description of country of city

A screenshot of a computer

Description automatically generated