**WEB SCRAPING**

## **WORKSHEET – 1(Solutions)**

**Ans 1)** B (Web scrapping)

**Ans 2)** C (Requests)

**Ans 3)** A (Browser based applications)

**Ans 4)** B(Crawler)

**Ans 5)** B(tag.name)

**Ans 6)** A (html.parser)

**Ans 7)** D (to download any content from a webpage)

**Ans 8)** C (the list of all webelements associated with the ‘given xpath’)

**Ans 9)** D (**‘a’** number of pixels vertically)

**Ans 10)** A(<a>), B(<href>)

**Ans 11)**

**Web Crawler**: A web crawler (also known as a web spider or web robot) is a program or automated script which browses the World Wide Web in a methodical, automated manner. This process is called **Web crawling** or **spidering**. Many legitimate sites, in particular, search engines, use spidering as a means of providing up-to-date data. Web crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine, which will index the downloaded pages to provide fast searches. Crawlers can also be used for automating maintenance tasks on a Web site, such as checking links or validating HTML code.

Furthermore, crawlers can be used to gather specific types of information from Web pages, such as harvesting e-mail addresses (usually for spam). Web crawler is also known as a Web Spider, automatic indexer or simply crawler.

**Web Scraper:** Web scraping is a process of extracting data from websites, and the tool which is used to do web scrapping is known as **Web-Scraper.** Scraper sends a GET query to a specific websites and then parses an HTML document based on the received, searches for the data that you have requested within that document and converts it into the specified format.

Web scraping is a technique used for extracting data from websites. It’s based on web crawling because before you can extract any data from a webpage you have to crawl it.

**Ans 12)**

A **robots.txt** file is a text-only format that contains instructions the web crawlers and robots are supposed to follow. A **robots.txt** file tells search engine crawlers which pages or files the crawler can or can't request from your site. This is used mainly to avoid overloading your site with requests; **it is not a mechanism for keeping a web page out of Google.**To keep a web page out of Google, you should use [no index directives](https://developers.google.com/webmasters/control-crawl-index/docs/robots_meta_tag), or password-protect your page. This file is included in the source files of most websites. Robots.txt files are mostly intended for managing the activities of good bots like [web crawlers](https://www.cloudflare.com/learning/bots/what-is-a-web-crawler/), since bad bots aren't likely to follow the instructions.

**Ans 13)**

**Static Web pages:**

* Static Web pages are very simple. It is written in languages such as HTML, JavaScript, CSS, etc.
* For static web pages when a server receives a request for a web page, then the server sends the response to the client without doing any additional process. And these web pages are seen through a web browser.
* In static web pages, Pages will remain the same until someone changes it manually.
* In Static web pages generally no involvement of database for data redecoration.

**Dynamic Web Pages:**

* Dynamic Web Pages are written in languages such as CGI, AJAX, ASP, ASP.NET, etc.
* In dynamic web pages, the Content of pages is different for different visitors. It takes more time to load than the static web page.
* Dynamic web pages are used where the information is changed frequently, for example, stock prices, weather information, etc.
* Dynamic web page database is used for data redecoration.

**Ans 14) Program to check whether the webpage is having title or not**

**Using Selenium:**

from selenium import webdriver

url="https://www.thehindu.com/"

keyword='Data Science'

driver=webdriver.Chrome(executable\_path='chromedriver')

driver.get(url)

title=driver.title

if title!=None:

print("Title of the webpage is:",title)

else:

print("Webpage do not have any title",title)

**Using Beautiful soup:**

from bs4 import BeautifulSoup

from urllib.request import urlopen

soup = BeautifulSoup(urlopen("https://www.google.com"))

title=soup.title.string

if title!=None:

print("Title of the webpage is:",title)

else:

print("Webpage do not have any title",title)

**Ans 15) Python program to access the search bar and search button on images.google.com**

from selenium import webdriver

url='https://images.google.com/'

keyword='Data Science'

# keyword = input("Enter the Keywords: ") # can be used to get value at runtime.

driver=webdriver.Chrome(executable\_path='chromedriver')

driver.get(url)

searchbar=driver.find\_element\_by\_name('q')

searchbar.send\_keys(keyword)

searchbar.send\_keys('\n')