**ASSIGNMENT 1**

1. What is Web Scraping, explain with an example.

Web scraping is a crucial technology that is used to produce structured data from unstructured material that is already present on the internet. Data that was created by scaping was organised and stored in a central database for analysis. Some of the common methods for data scraping include traditional copy-and-paste, text grapping and regular expression matching, HTTP programming, HTML parsing, DOM parsing, web scraping software, vertical aggregation platforms, semantic annotation recognising, and computer vision web-page analysers.

Web harvesting, web data scraping, web scraping, and screen scraping are other terms for web scraping. Data mining includes web scraping.

The fundamental and crucial goal of the online scraping procedure is to extract data from various, unstructured websites and convert it into a format that can be understood, such as spreadsheets, databases, or comma-separated values (CSV) files. Web scraping may be used to collect information such as item price, stock pricing, various reports, market pricing, and product data.

Example:

Consider a scenario in which we are eager to learn more about product prices from an online store we'll name "Online Mart." In order to compare pricing between several online retailers, you want to gather the product name, price, and availability.

1. **Choose the Target Website:** In this instance, you will select the Online Mart website from which you wish to scrape product information.
2. **Use the developer tools** in your web browser to look into the web page's HTML structure. This will assist you in locating the HTML components that have the information you wish to extract. For instance, you might discover that product names are kept in **<h2>** tags, prices are kept in **<span>** tags with a particular class, and availability is kept in another element.
3. **Select a Scraping Tool or Library:** A variety of programming languages and libraries are available to assist in web scraping, including Python's Beautiful Soup, Scrapy, and libraries like Puppeteer if you're using JavaScript. These tools will be used to automate the retrieval and HTML content parsing procedure.
4. **Write the Scraping Code:** For the purposes of this example, let's suppose that you are using Python and Beautiful Soup. Using a package manager like pip, you would first install Beautiful Soup before creating a script to retrieve and parse the HTML from the website.
5. What are the challenges in performing web scraping?

Due to the growing demand for big data, web scraping has become a trendy topic. As their businesses grow, more and more people are hungry for data from various websites. However, as online scraping procedures increase, a number of issues will emerge, such as blocking techniques, which may make it more difficult for users to access data. Let's take a closer look at these difficulties.

1. **Use of bots:** First determine whether the target website permits scraping. You can approach the website owner for permission to scrape their site if you discover that it isn't authorized via their robots.txt by describing your reasons and goals for doing so. It is preferable to choose a different website with comparable content if the owner continues to disagree.
2. **Complicated and changeable web page structures:** Most websites use HTML (Hypertext Markup Language). Web page structures vary greatly because web page designers can use their own standards to construct the pages. You must create a separate scraper for each website if you need to scrape more than one website.
3. **IP Blocking:** Web scrapers are frequently prevented from accessing website data by IP blocking. The majority of the time, this occurs when a website notices a large volume of requests coming from the same IP address. To stop the scraping operation, the website would either completely block the IP or restrict its access.
4. **Captcha:** To distinguish people from scraping software, CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) frequently uses visuals or logical puzzles that are simple for people to answer but difficult for scraping software.
5. **Honeypot Traps:** The website owner sets up a honeypot as a trap to catch scrapers. Links that are invisible to humans but visible to scrapers can serve as the traps. When a scraper stumbles into the trap, the website can utilise the data it gathers (such as the scraper's IP address) to stop it.
6. **Slow/unstable load speed:** When a website receives too many access requests, it may react slowly or perhaps stop working altogether. When people use the site, they just need to reload the page and wait for the website to recover, so it is not an issue. However, scraping could be interrupted if the scraper is unable to handle the situation.
7. **Real-time data scraping:** When it comes to pricing comparison, inventory tracking, etc., real-time data scraping is crucial. The data can alter in a split second and might result in significant monetary gains for a company. The scraper must continuously scan the websites and collect data from them. Even so, there is still some delay since it takes time to process the request and send the data. A significant problem is also posed by gathering a lot of data in real-time.
8. **Login requirements:** You might need to log in before accessing certain protected information. Most websites recognise you as the same user who just checked in by your browser automatically appending the cookie value to any subsequent requests you make once you've submitted your login information. Therefore, be sure that cookies have been provided with the requests when scraping websites that require logging in.
9. **Dynamic content:** AJAX is frequently used on websites to refresh dynamic online content. Examples include limitless scrolling, slow loading of graphics, and using AJAX requests to provide more information when a button is clicked. On these kinds of websites, consumers can easily access additional info, but scrapers cannot.
10. Describe the basic architecture for performing web scraping.

As we all know, scraping is a method for extracting data from web pages using script functions. Hypertext Markup Language (HTML), and more recently XHTML, which is based on XML, are the two languages used to create web pages. The Document Object Model (DOM) tree, which is used to describe web pages, was created with the purpose of defining the text display format for web browsers.



A web scraping job appears to be a manual copy and paste task. The distinction in this instance is that a virtual computer agent performing this task does so in an orderly and automatic manner. An agent is actually carrying out the identical action that a human would typically take when engaging with a website when it follows each link on a web page. This agent can browse across numerous different web pages, click links (using HTTP GET requests), and submit forms (using HTTP POST). A human would have to think, take the mouse, point at the link, and then click it, but a machine would carry out manual activities at the pace of a computer instruction.

When a user must click on multiple links before reaching the desired page, the benefit is now apparent. However, it should come as no surprise that using a script speed for requests also has a drawback. Due to the high volume of requests made in a short amount of time when using web scraping without a policy for restricting requests, the requested server may discover that someone is attempting a Denial-of-Service attack.

1. Describe various techniques of web scraping.
2. **Classical copy and paste:** The most effective and practical web-scraping technology is human manual scrutiny and copy and paste. But when a user needs to analyse and store several datasets, this strategy tends to be difficult to execute or to produce mistakes.
3. **Hypertext Transfer Protocol (HTTP) Programming:** This approach allows users to retrieve data from both static and dynamic web sites. By sending HTTP queries to the distant web server using socket programming, data can be retrieved.
4. **Hyper Text Markup Language (HTML) Parsing:** To read HTML pages, obtain, and modify page content, semi-structured data query languages can be used. Examples are XQuery and the Hyper Text Query Language (HTQL).
5. **Document Object Model (DOM)Parsing:** Programs can access the dynamic content produced by client-side scripts by embedding a full-featured web browser, such as the Internet Explorer or Mozilla browser control. These browser settings also break down web pages into a DOM tree, which allows programs to get specific pages in chunks.
6. **Web Scraping Software:** There are numerous software tools available now that can be utilized to tailor web-scraping solutions. This software may try to recognize a page's data structure automatically or offer a recording interface that eliminates the need for manually writing web scraping code, as well as some scripting functions for extracting and transforming content and database interfaces for storing the scraped information in local databases.
7. **Computer vision web-page analysers:** There are initiatives employing computer vision and machine learning to recognize and extract information from web sites by visually evaluating pages like a human might.