A1 – Scraping Introduction

Name: Sai Disha. D

Roll no. 231057026

1. What is Web Scraping? Explain with an example.

- **Web scraping** is a technique used to extract data from websites. It involves automatically retrieving information from web pages and then structuring and storing that data for analysis or other purposes. This is typically done by writing a program that simulates the process a human would use to browse a website and manually copy data.
- All kinds of data can be scraped from the internet. Common data types organizations collect include images, videos, text, product information, customer sentiments and reviews etc. One should keep in mind the legal rules and what type of information can be scrapped.
- **Example:** Imagine you want to gather information about product prices from an e-commerce website. Instead of manually copying each price, you can create a web scraping script that navigates to the product pages, extracts the prices, and stores them in a structured format like a spreadsheet or a database.
- This allows you to quickly collect and update prices without the need for manual copying and pasting. However, it's important to note that while web scraping can be powerful, it should be done ethically and in accordance with website terms of use.

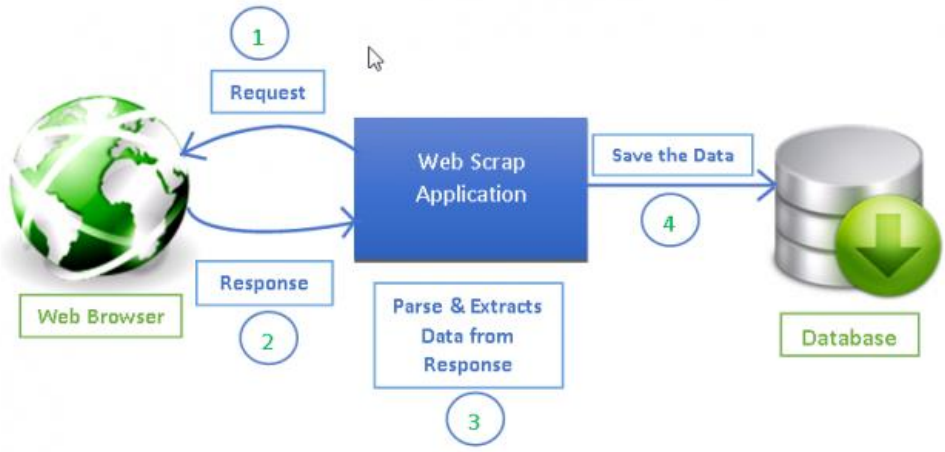
2. What are the challenges in performing web scraping?

- **Honeypots:** Some websites use honeypots, deceptive content that appears legitimate, to trap scraping bots. When a bot triggers a honeypot, it gets stuck in an endless loop of requests, failing to extract useful data.
- **Website Structure Changes:** Websites frequently update their layout and structure, altering the HTML and JavaScript code that holds the data. Web scrapers must be adaptable to these changes to avoid scraping incomplete or incorrect information.
- **Rate Limiting and IP Blocking**: Websites can detect and block excessive or rapid scraping activity to prevent server overload. This can lead to IP bans or throttling.
- **Data Quality and Cleaning**: Scraped data often requires cleaning and validation due to inconsistencies, missing values, or formatting issues on the website.
- **Captcha and Bot Detection**: Websites may implement CAPTCHA challenges or bot detection mechanisms to hinder automated scraping.
- **Authentication and Cookies**: Some websites require user authentication or session cookies to access certain data, making scraping more complex.

3. Describe the basic architecture for performing web scraping.

The basic architecture for performing web scraping involves these steps:

- **Request:** A program sends an HTTP request to the target website's server, indicating the specific webpage to access.
- **Retrieve:** The server responds with the requested webpage's HTML content, containing the data to be scrapped.
- **Parse:** The HTML content is parsed to extract the desired data using tools like libraries or frameworks (e.g., Beautiful Soup or Scrapy).
- **Process:** Extracted data is processed, cleaned, and structured into a usable format, such as a spreadsheet or database.
- **Storage:** The processed data is stored in a local file, database, or cloud storage for further analysis or use.



4. Describe various techniques of web scraping.

Several techniques are used for web scraping:

- **Static HTML Parsing:** Involves parsing the HTML source code of a webpage to extract desired data using libraries like Beautiful Soup. Best for simple sites without dynamic content.
- **Dynamic HTML Parsing:** Uses tools like Selenium to automate interaction with web pages, including executing JavaScript. Useful for sites with dynamic content loaded after page load.
- **API Scraping:** Accesses data through a website's API (Application Programming Interface), which provides structured data directly, usually in JSON or XML format.
- **Proxy Rotation:** Rotates IP addresses through proxy servers to avoid IP-based blocking and access restrictions, allowing more successful scraping.
- **User-Agent Rotation:** Changes the User-Agent header in HTTP requests to mimic different browsers or devices, preventing detection as a scraper.
- **Web Scraping Frameworks:** Frameworks like Scrapy offer pre-built tools for managing requests, parsing, and storing data efficiently.
- **Headless Browsers:** Utilizes browser automation in a headless mode, allowing JavaScript rendering and interaction, often done using tools like Puppeteer.
- **Crawling vs. Targeted Scraping:** Crawling involves systematically navigating and scraping multiple pages, while targeted scraping focuses on specific data on selected pages.
- **Regular Expressions:** A powerful text pattern matching technique used for data extraction when the data follows a consistent pattern.
- **Machine Learning for Data Extraction:** Using machine learning algorithms to identify and extract specific data points from unstructured content.

Each technique has its strengths and limitations, and the choice depends on the target website's structure, complexity, and the desired data extraction goals.