Experiment No. 8

Title: Study Experiment on Web Scraping

Batch: B-2 **Roll No.: 16010422234** **Experiment No: 8**

Aim: Study Experiment on Web Scraping

Resources needed: Windows OS

Pre Lab/ Prior Concepts:
Students should have prior knowledge of PHP.

Theory:

What is Web Scraping?
Web Scraping is also termed as Screen Scraping, Web Data Extraction, Web Harvesting etc. Web scraping is an automatic method to obtain large amounts of data from websites. Most of this data is unstructured data in an HTML format which is then converted into structured data in a spreadsheet or a database so that it can be used in various applications. There are many different ways to perform web scraping to obtain data from websites. These include using online services, particular API's or even creating your code for web scraping from scratch. Many large websites, like Google, Twitter, Facebook, StackOverflow, etc. have API's that allow you to access their data in a structured format. This is the best option, but there are other sites that don't allow users to access large amounts of data in a structured form or they are simply not that technologically advanced. In that situation, it's best to use Web Scraping to scrape the website for data.

Web scraping requires two parts, namely the **crawler** and the **scraper**. The crawler is an artificial intelligence algorithm that browses the web to search for the particular data required by following the links across the internet. The scraper, on the other hand, is a specific tool created to extract data from the website. The design of the scraper can vary greatly according to the complexity and scope of the project so that it can quickly and accurately extract the data.

How does a web scraper work?
Web Scrapers can extract all the data on particular sites or the specific data that a user wants. Ideally, it's best if you specify the data you want so that the web scraper only extracts that data quickly. For example, you might want to scrape an Amazon page for the types of juicers available, but you might only want the data about the models of different juicers and not the customer reviews.
So, when a web scraper needs to scrape a site, first the URLs are provided. Then it loads all the HTML code for those sites and a more advanced scraper might even extract all the CSS and Javascript elements as well. Then the scraper obtains the required data from this HTML code and outputs this data in the format specified by the user. Mostly, this is in the form of an Excel spreadsheet or a CSV file, but the data can also be saved in other formats, such as a JSON file.

Activity:

1) List and explain various applications of Web Scraping.
2) List different types of web scraper and explore any available web scraper and prepare a documentation by considering various aspects of working of selected web scraper.

Output (Detailed Documentation):

Ethical Considerations

When performing web scraping, it's essential to consider the ethical and legal implications. Websites have terms of service that specify what is allowed, and scraping certain data may violate these terms or infringe on intellectual property rights. For instance, scraping personal information or proprietary data without permission could be legally questionable. Additionally, scraping should be done responsibly to avoid overloading servers, which could affect the performance of websites for other users. Checking the site's robots.txt file can help determine which pages are permitted for crawling. Whenever possible, it is preferable to use official APIs, as they are designed to provide structured data access in a way that aligns with the website's policies.

Applications of Web Scraping

1) Market Research and Competitor Analysis:
   Collects competitor pricing, product details, and customer reviews, allowing businesses to analyze trends and refine strategies.

2) Price Monitoring:
   Tracks competitor prices in real-time, allowing e-commerce sites to adjust their prices and stay competitive.

3) Job Listings Aggregation:
   Gathers job postings from various sources for centralized job search platforms.

4) Real Estate Listings:
   Compiles property data for buyers or renters, including prices and property features.

5) Lead Generation:
   Collects contact information from online directories for targeted marketing campaigns.

6) Social Media Sentiment Analysis:
   Tracks brand mentions and sentiment to gauge public opinion and brand health.

7) News Aggregation:
   Aggregates news from various sources, allowing real-time access to diverse news content.

8) Academic Research and Analysis:
   Supports research by collecting data from public sources, enabling in-depth analysis.

9) Financial Data Analysis:
   Collects stock prices, trends, and financial reports for informed investment decisions.

10) E-commerce Product Details and Reviews:
   Analyzes customer reviews and product details for better understanding of customer preferences.

11) Healthcare Data Collection:
   Gather data on health trends and treatments, supporting healthcare research.

12) Event Aggregation:
   Compiles event data for users, offering comprehensive event listings.

13) Data for Machine Learning and AI Models:
   Provides datasets required to train machine learning and AI models in the absence of open-source data.

Types of Web Scrapers

1) HTML Parsers:
   Parses HTML tags and elements using libraries like BeautifulSoup, suitable for static content.

2) DOM Parsers:
   Accesses HTML elements based on the DOM, useful for handling dynamic content.

3) CSS Selectors:
   Uses CSS selectors to locate specific elements, enabling targeted scraping.

4) API-Based Scrapers:
   Extracts data via APIs, allowing efficient and reliable data collection.

5) Headless Browsers:
   Simulates a browser without a GUI for JavaScript-heavy websites using tools like Puppeteer.

Challenges in Web Scraping

Web scraping can present various challenges, including:

- Dynamic Content: Many modern websites use JavaScript to load content dynamically, making it challenging to scrape using traditional HTML parsers. For such cases, tools like Selenium or headless browsers that can render JavaScript are needed.

- CAPTCHAs and Anti-bot Mechanisms: Websites often have protections like CAPTCHAs or IP blocks to prevent automated access. These measures can make scraping difficult without proper tools and may require more advanced techniques.

- Rate Limits: Sending too many requests in a short period can cause websites to throttle or block requests. Respectful scraping practices, like adding delays between requests, are essential to avoid triggering these rate limits.

By understanding these challenges, users can select the appropriate tools and strategies to overcome them while being mindful of the website's restrictions.

Best Practices

To ensure effective and responsible web scraping, consider the following best practices:

- Respect Robots.txt and Terms of Service: Always check a site’s robots.txt file to see which pages can be crawled. This helps avoid any unintended violations of the site's policies.

- Use Headers to Mimic a Real Browser: Adding headers, such as the User-Agent, helps mimic a browser request, reducing the likelihood of getting blocked.

- Throttle Requests: Avoid sending too many requests in a short time. Adding a delay between requests helps prevent overloading the website’s server.

- Use Proxies When Necessary: For high-volume scraping, rotating IP addresses via proxies can reduce the chances of IP blocks.

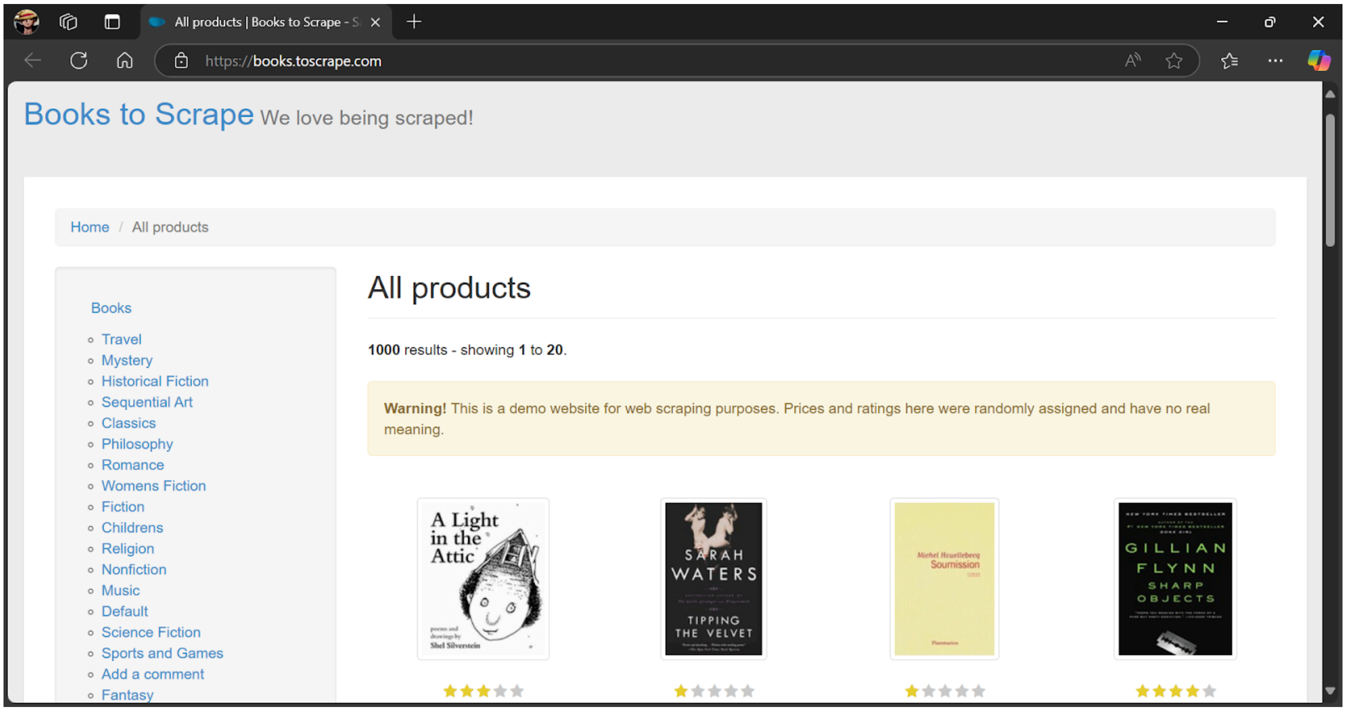
- Handle Errors Gracefully: Implement error handling to manage unexpected issues, like missing data or failed requests.

By following these best practices, users can conduct web scraping more effectively and ethically.

```php
<?php
// Set the target URL for the book website
$targetUrl = "http://books.toscrape.com/";
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $targetUrl);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

// Execute cURL to fetch HTML content
$htmlContent = curl_exec($ch);
```

```php
if(curl_errno($ch)) {
    echo 'cURL error: ' . curl_error($ch);
    curl_close($ch);
    exit;
}
curl_close($ch);

// Load HTML content into DOMDocument
$dom = new DOMDocument();
libxml_use_internal_errors(true); // Suppress warnings for invalid HTML
$dom->loadHTML($htmlContent);
libxml_clear_errors();

// Use XPath to locate book information
$xpath = new DOMXPath($dom);
$books = $xpath->query("//article[@class='product_pod']");

// Open a CSV file to save the scraped data
$file = fopen("books_data.csv", "w");
fputcsv($file, ['Title', 'Price', 'Availability']); // Header row

// Iterate through each book container and extract details
foreach ($books as $book) {
    // Extract book title
    $titleNode = $xpath->query(".//h3/a", $book)->item(0);
    $title = $titleNode ? $titleNode->getAttribute("title") : "N/A";

    // Extract book price
    $priceNode = $xpath->query(".//p[@class='price_color']", $book)->item(0);
    $price = $priceNode ? $priceNode->textContent : "N/A";

    // Extract availability status
    $availabilityNode = $xpath->query(".//p[contains(@class, 'instock')]",
$book)->item(0);
    $availability = $availabilityNode ? trim($availabilityNode->textContent) :
"N/A";

    // Write data to CSV file
    fputcsv($file, [$title, $price, $availability]);
}

// Close the CSV file
fclose($file);
```

```
echo "Data scraping completed. Check 'books_data.csv' for results.";
?>
```


All products | Books to Scrape - S
https://books.toscrape.com

Books to Scrape We love being scraped!

Home / All products

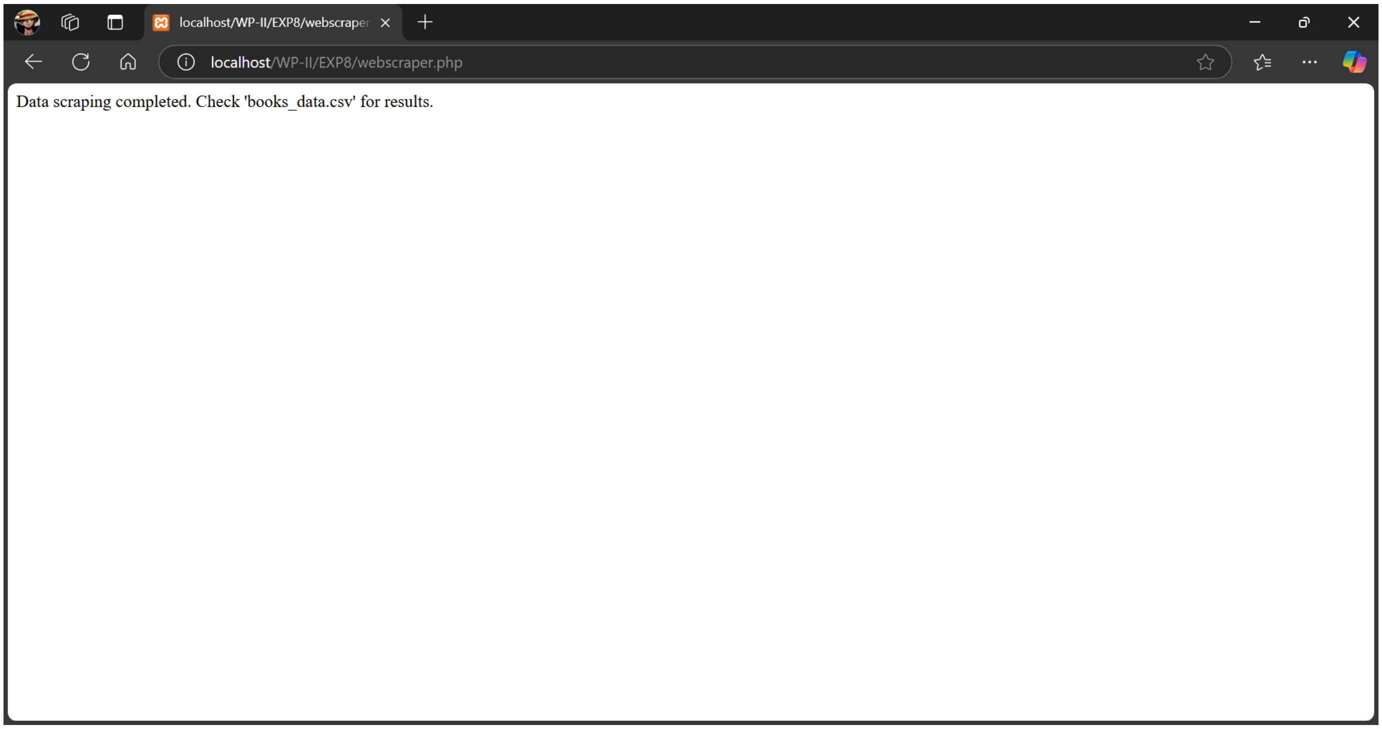
Books

- Travel
- Mystery
- Historical Fiction
- Sequential Art
- Classics
- Philosophy
- Romance
- Womens Fiction
- Fiction
- Childrens
- Religion
- Nonfiction
- Music
- Default
- Science Fiction
- Sports and Games
- Add a comment
- Fantasy

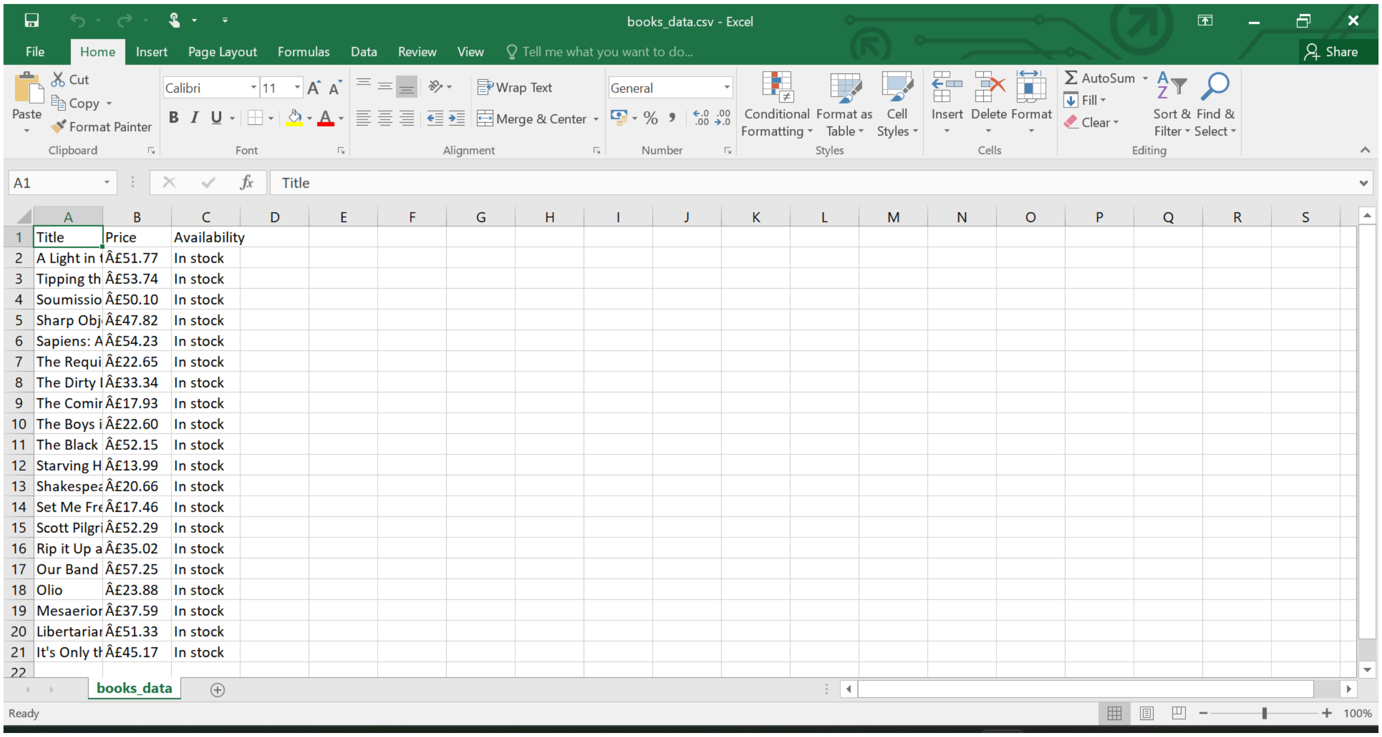
All products

1000 results - showing **1** to **20**.

Warning! This is a demo website for web scraping purposes. Prices and ratings here were randomly assigned and have no real meaning.


localhost/WP-II/EXP8/webscraper
localhost/WP-II/EXP8/webscraper.php

Data scraping completed. Check 'books_data.csv' for results.



Title	Price	Availability
A Light in t	Â£51.77	In stock
Tipping th	Â£53.74	In stock
Soumissio	Â£50.10	In stock
Sharp Obj	Â£47.82	In stock
Sapiens: A	Â£54.23	In stock
The Requi	Â£22.65	In stock
The Dirty I	Â£33.34	In stock
The Comir	Â£17.93	In stock
The Boys i	Â£22.60	In stock
The Black	Â£52.15	In stock
Starving H	Â£13.99	In stock
Shakespea	Â£20.66	In stock
Set Me Fre	Â£17.46	In stock
Scott Pilgri	Â£52.29	In stock
Rip it Up a	Â£35.02	In stock
Our Band	Â£57.25	In stock
Olio	Â£23.88	In stock
Mesaerior	Â£37.59	In stock
Libertariar	Â£51.33	In stock
It's Only tl	Â£45.17	In stock

Further Experimentation Ideas

- REST APIs: Use APIs (e.g., Twitter, Reddit) for structured data.
- Pagination: Scrape data across multiple pages.
- JavaScript Sites: Use Selenium or Playwright for dynamic content.
- Data Storage: Save data in JSON or databases like SQLite.

Outcome: Demonstrate the use advanced features such as REST API, email handling, localization and internationalization in PHP

Conclusion:

Web scraping is a powerful technique for extracting data from websites, allowing users to gather structured information from HTML pages and convert it into usable formats like CSV or JSON. This experiment demonstrated the fundamentals of web scraping, including the role of crawlers and scrapers. By understanding the ethical considerations and legality associated with web scraping, and by learning to use the appropriate tools and techniques, students can apply web scraping effectively for various real-world applications. This experiment also emphasized the need for caution when scraping data and encouraged using official APIs when available, as they offer more reliable and legal access to structured data from websites.

Grade: AA / AB / BB / BC / CC / CD / DD

Signature of faculty in-charge with date

References:

1) Thomson PHP and MySQL Web Development Addison-Wesley Professional , 5th Edition2016.
2) www.geeksforgeeks.org/what-is-web-scraping-and-how-to-use-it
3) towardsdatascience.com/web-scraping-basics-82f8b5acd45c