CS2209: Assignment – 1

Web Scraping: Extracting Data from Websites using BeautifulSoup

Aim:

To collect the latest news headlines of the day from the Indian Express website and represent the data dynamically using the Flask development framework.

Website Overview:

The Indian Express (https://indianexpress.com) is a leading Indian news website that delivers reliable and up-to-date information across various categories such as Politics, Business, Sports, and Entertainment. It serves as the data source for this assignment, where the top news headlines of the day are extracted.

Tools and Technologies:

1. **Python Libraries**:
    a. **BeautifulSoup**: For web scraping and extracting data from HTML content.
    b. Requests: For fetching HTML content from the Indian Express website.
2. **Frameworks:**
    a. Flask: For building a dynamic web application to render the extracted data.
    b. Jinja2: For dynamically generating HTML content using the scraped data

Project Structure:

```
    cs2209_2301me80_2301ce03/
     ├── data/
         │   ├── latest_news.csv
```

```
│   ├── latest_news.txt
│   ├── top_news.csv
│   ├── top_news.txt
├── News Scrapping Application/
├── main.py
├──Report.pdf
```

- **Parent folder:** The parent folder consists of a python file named main.py which on execution updates the data in the ` data ` directory. The data is saved in both in a CSV field as well as a text file.
- **News Scrapping Application:** This consists of the Flask application which displays the fetched data using Jinja2 templates.

Process Overview:

1. **Data extraction:**
   a. Fetch the HTML content of the Indian Express homepage using the requests library.
   b. Parse and extract top and latest news headlines using BeautifulSoup.
2. **Data Storage:** Save the extracted data in a CSV file
3. **Dynamic Representation:**
   a. Build a flask application to render the fetched data on a webpage.
   b. The latest news headlines are rendered on the /latest-news route, and the top news headlines are rendered on the /top-news route.

c. Use Jinja2 to display the data, ensuring user-friendly interface.

Conclusion:

Web scraping is a versatile and efficient method for extracting data from websites when APIs are unavailable or overly restrictive. Unlike using API keys, which often require registration, authentication, rate limits, and maintenance of access credentials, web scraping directly parses the HTML content of a webpage, bypassing these hurdles. This approach minimizes dependency on external services and provides more flexibility to extract customized data, such as specific sections or elements of a webpage. While ethical considerations and adherence to a website’s terms of service are essential, web scraping can significantly simplify data acquisition for applications that do not have readily available APIs, saving time and effort while enabling greater control over the data extraction process.

Group Members:

1. **Ankit Bhagat – 2301CE03**
2. **Vinay Khedkar – 2301ME80**