Extraction Of Stock Price By Web Scrapping

Abstract:

This web scraping project aims to extract stock price data for 14 companies from the financial website "https://ticker.finology.in/". The extracted information includes the company name, current, and highest stock price. The implementation involves using Python with the requests, BeautifulSoup, and pandas libraries for data retrieval, HTML parsing, and organization. The project emphasizes ethical web scraping practices, ensuring compliance with the website's terms of service.

Introduction:

In the digital age, access to real-time financial data is crucial for investors and analysts. This project focuses on automating the extraction of stock price information from an online financial portal, providing a convenient and efficient way to monitor the performance of 14 selected companies. By leveraging web scraping techniques, the script retrieves relevant data, enabling users to stay informed about stock prices and make data-driven investment decisions.

Objectives:

The primary objectives of this project include:

- Extracting stock-related data from the website "https://ticker.finology.in/".
- Organizing the data into a structured format, including company names, current stock prices, and highest stock prices.
- Creating a pandas DataFrame for easy data manipulation and analysis.
- Saving the extracted data as a CSV file for further use.

Methodology:

The methodology involves several key steps:

- HTTP Request: Initiating an HTTP GET request to the target URL to obtain the HTML content.
- HTML Parsing: Using BeautifulSoup to parse the HTML and navigate the webpage structure.
- Table Identification: Locating the relevant table containing stock information based on its class attribute.
- Data Extraction: Extracting headers and data from the table, focusing on specific columns.
- DataFrame Creation: Constructing a pandas DataFrame to structure the extracted data.
- Output Handling: Printing the DataFrame and saving it as a CSV file named "Stock_Data.csv."

Results:

The results include a structured DataFrame containing company names, current stock prices, and highest stock prices for the selected companies. The data is successfully saved in a CSV file, providing a user-friendly format for further analysis and reference.

Conclusion:

This web scraping project demonstrates an effective approach to automate the extraction of stock price data from an online financial platform. The use of Python, along with libraries such as BeautifulSoup and pandas, allows for efficient data retrieval and organization. The project meets its objectives by providing a clear and concise representation of stock-related information, enhancing accessibility for investors and analysts.

Challenges and Limitations:

While the project achieves its goals, challenges and limitations are inherent in web scraping activities. Some potential issues include:

- Sensitivity to changes in the HTML structure of the target website.
- Dependency on the website's terms of service and potential legal implications.
- Limited scalability if the number of companies or data points increases.

Future Scope:

Future enhancements and expansions for this project include:

- Implementation of error handling mechanisms to address changes in website structure.
- Integration of data visualization tools for more comprehensive stock analysis.
- Incorporation of additional data points such as trading volume and historical performance.
- Exploration of APIs for more ethical and structured data access.
- In conclusion, this web scraping project serves as a foundational step in automating financial data extraction, with ample opportunities for further development and improvement.