Mini Project

Web Scraping using Python

SUBMITTED BY

Aditya Ramanathan N (20PD02)

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE
COURSE

Python Programming Laboratory

**DEPARTMENT OF
APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCES**

PSG COLLEGE OF TECHNOLOGY

COIMBATORE

Letter of Transmittal

Aditya Ramanathan N

20PD02

MSc Data Science

PSG College of Technology

Coimbatore – 641 004

3 April 2021

Dr. Aneetha A.S

Department of Applied Mathematics and Computational Sciences

PSG College of Technology

Coimbatore – 641 004

Respected Mam,

       The following project "**Web Scraping Using Python**" is submitted in partial fulfilment of the requirements of the course "**Python Programming Lab**".
The purpose of this project report is to present a summary of my project developed using Python language.

Yours sincerely,

Aditya Ramanathan N

Acknowledgment

Presentation, inspiration and motivation have always played a key role in the success of any venture.

I express my sincere thanks to **Dr. R.Nadarajan**, Head of the Department.

I pay my deep sense of gratitude to **Dr. Aneetha A.S**, who encouraged me to the highest peak and to provide me with the opportunity to prepare the project. I am immensely obliged to my friends for their elevating inspiration, encouraging guidance and kind supervision in the completion of my project.

Last, but not the least, many thanks to the **Lab Admin**, who gave the permission to use all required equipment and the necessary materials to complete the specific tasks related to my report. I have to appreciate the guidance given by other faculties to complete my report

Abstract

Web scraping is a term used to describe the use of a program or algorithm to extract and process large amounts of data from the web. Web scraping is used to collect large information from websites. The data on the websites are unstructured. Web scraping helps collect these unstructured data and store it in a structured form.

The Internet hosts perhaps the greatest source of information—and misinformation—on the planet. Many disciplines, such as data science, business intelligence, and investigative reporting, can benefit enormously from collecting and analyzing data from websites.

In this project, we have decided to scrape data from an e-commerce website. The main motive of scraping an e-commerce website is Product Comparison. Using their HTML tags, products can be compared for their prices, specifications, ratings and reviews. Since this data is mainly unstructured, using a web scraping tool will be very helpful to get this data in an organised manner. This makes our job much simpler and easier!

Some python libraries that will be helpful for implementing this project are:

- **Requests**: This critical library is needed to actually get the data from the web server onto your machine, and it contains some additional features like caching too.
  The requests module allows you to send HTTP requests using Python. The HTTP request returns a Response Object with all the response data (content, encoding, status, etc).

- **Beautiful Soup**: Beautiful Soup is a library that makes it easy to scrape information from web pages. It sits atop an HTML or XML parser, providing Pythonic idioms for iterating, searching, and modifying the parse tree.

- **Selenium**: The selenium package is used to automate web browser interaction from Python. Selenium is a powerful tool for controlling web browsers through programs and performing browser automation. It is functional for all browsers, works on all major OS.

- **Urllib**: Urllib module is the URL handling module for python. It is used to fetch URLs (Uniform Resource Locators). It uses the urlopen function and is able to fetch URLs using a variety of different protocols.

Web Scraping is also an important skill for any data scientist to have in their toolbox. This project will help develop our skills in data extraction and analysis. Python is also the right language to implement this project due to its syntax simplicity, inbuilt modules and wide range of applications.

****

References

https://realpython.com/beautiful-soup-web-scraper-python/

https://realpython.com/python-web-scraping-practical-introduction/

https://www.geeksforgeeks.org/implementing-web-scraping-python-beautiful-soup/

https://opensource.com/article/20/5/web-scraping-python

https://www.datacamp.com/community/tutorials/web-scraping-using-python