A picture containing text

Description automatically generated

MINI PROJECT REPORT

NAME: PRATHAM JAMWAL

SECTION: CE-CORE

CLASS ROLLNO.: 27

UNIVERSITY ROLLNO.: 2O17352

|  |
| --- |
| Text  Description automatically generated  **CERTIFICATE**  This is to certify that the project report entitled **“WEB SCRAPPER USING PYTHON”** is a Bonafede project work carried out by Pratham Jamwal, roll no- 2017352, in partial fulfillment of the award of the degree of B-tech of Graphic Era Deemed University, Dehradun during the academic year 2021-2022. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated. The project has been approved as it satisfies the academic requirements associated with the degree mentioned.  **Mrs. Garima Sharma, ( CSE DEPT. )** |

|  |
| --- |
| **DECLARATION**  I**,** **Pratham Jamwal** student of B-tech**, Semester 4th,** Department of Computer Science and Engineering**,** **Graphic Era Deemed To Be University, Dehradun,** declare that the technical project work entitled **“WEB SCRAPPER USING PYTHON”** has been carried out by me and submitted in partial fulfillment of the course requirements for the award of degree in B- tech of **Graphic Era Deemed To Be University** during the academic year **2022-2023**.  Date: 13/07/22 **PRATHAM JAMWAL** |

PROBLEM DEFINITION:

The need to pull a large amount of data from websites as quickly as possible is a necessity for companies these days. One cannot manually reach each website and collect the data personally. Thus a standardized method of collecting this data from different websites is important and need of the hour.

OBJECTIVE:

The main objective of my Project is to Scrape or Collect data from local websites present on the web as per the requirement of the client in an organized manner to avoid any inconvenience. The project I created helps the user to have a set of websites following under the URL he/she has added.

TOOLS AND LANGUAGE USED:

TOOLS: PyCharm

LANGUAGE: Python

LIBRARIES USED:

* Tkinter
* sqlite3
* ssl
* urllib.parse
* BeautifulSoup

WHAT IS WEB SCRAPING?

Web scraping is an automated method used to extract large amounts of data from websites. The data on the websites are unstructured. Web scraping helps collect these unstructured data and store it in a structured form. There are different ways to scrape websites such as online Services, APIs or writing your own code. In this article, we’ll see how to implement web scraping with python.

WHY IS WEB SCRAPING USED?

Web scraping is used to collect large information from websites. But why does someone have to collect such large data from websites? To know about this, let’s look at the applications of web scraping:

* **Price Comparison:** Services such as ParseHub use web scraping to collect data from online shopping websites and use it to compare the prices of products.
* **Email address gathering:** Many companies that use email as a medium for marketing, use web scraping to collect email ID and then send bulk emails.
* **Social Media Scraping:** Web scraping is used to collect data from Social Media websites such as Twitter to find out what’s trending.
* **Research and Development:** Web scraping is used to collect a large set of data (Statistics, General Information, Temperature, etc.) from websites, which are analyzed and used to carry out Surveys or for R&D.
* **Job listings:** Details regarding job openings, interviews are collected from different websites and then listed in one place so that it is easily accessible to the user.

METHODS TO SCRAPE DATA:

There are various methods to scrape data which includes:

1. MANUAL (COPY PASTING)
2. AUTOMATED

* HTML Parsing
* DOM Parsing
* Google Sheets
* XPATH

IMPLEMENTATION:

The implementation part is divided into:

1. Finding the URL to be scraped
2. Inspecting the Page
3. Finding the data to be extracted
4. Extracting the data
5. Storing the data in the required format

LIBRARIES USED:

* **TKINTER:**

Tkinter is the standard GUI library for Python. The combination of Python and Tkinter makes it quick and simple to develop GUI apps. The Tk GUI toolkit applications have a strong object-oriented interface provided by Tkinter.

* **SSL:**

SSL helps to establish a secure connection between the client and the server through the secure protocol HTTPS. It is a perfect choice for protecting sensitive information, such as customer contact details and credit card details.

* **BEAUTIFULSOUP:**

Beautiful Soup is a python package which is used for parsing HTML/XML documents.  It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web Scraping.

* **SQLITE3:**

sqlite3 gives our code access to the sqlite3 module. The sqlite3. connect() function returns a Connection object required for interaction.

CONCLUSION:

Now that we have seen several different ways to scrape data from websites and are ready to start working on potentially larger projects, we may ask ourselves whether there are any legal implications of writing a piece of computer code that downloads information from the Internet. The first and most important thing to be careful about when writing a web scraper is that it typically involves querying a website repeatedly and accessing a potentially large number of pages. For each of these pages, a request will be sent to the web server that is hosting the site, and the server will have to process the request and send a response back to the computer that is running our code. Each of these requests will consume resources on the server, during which it will not be doing something else, like for example responding to someone else trying to access the same site.

If we send too many such requests over a short span of time, we can prevent other “normal” users from accessing the site during that time, or even cause the server to run out of resources and crash.

In fact, this is such an efficient way to disrupt a web site that hackers are often doing it on purpose.