|  |
| --- |
| download (1).png |
| **JOBS WEB SCRAPPING** |
|  |
|  |
| **Group Members:** Farrukh Zain , Ghulam Mustafa , Haniya Rizwan  **Id’s:** Bse-21f-029 , Bse-21f-032 , Bse-21f-037  **Department:** Software Engineering  **Program:** Bsse(M)  **Section:** SE – 3B  **Course Name:** Data –Structure-Algorithm  **Course Code:** Csc 102  **Instructor:** Mr. Ameen Chhajro |
|  |
|  |

|  |
| --- |
|  |

Contents

[Abstract:](#_Toc106182135)

[Introduction:](#_Toc106182136)

[Methodology:](#_Toc106182137)

[Set the base URL and number of pages to scrape:](#_Toc106182138)

[Create lists to store the scraped data::](#_Toc106182139)

[Iterate over the pages and scrape the data:](#_Toc106182140)

[Combine the lists into a list of tuples:](#_Toc106182140)

[Convert the list of tuples into a pandas DataFrame:](#_Toc106182140)

[Save the DataFrame to an Excel file:](#_Toc106182140)

[Data Analysis and Visualization: …….8](#_Toc106182147)

**Jobs Web Scraping Project Report**

**Abstract:**

This project aims to scrape job data from a website called Pakistan Jobs & International (http://www.pakistanjobsbank.com/). This website provides a list of jobs in various categories and locations in Pakistan. The data scraped from this website can be used for analysis, visualization, or other purposes. The main steps of the web scraping process are: setting the base URL and number of pages to scrape, creating lists to store the scraped data, iterating over the pages and scraping the data, combining the lists into a list of tuples, converting the list of tuples into a pandas DataFrame, and saving the DataFrame to an Excel file. The Python libraries used for this project are: requests, BeautifulSoup, pandas, and openpyxl. The code successfully scraped 10150 pages from the website and extracted four columns of data: Title, Category, Date\_Position, and Location. The data was stored in a pandas DataFrame with 10150 rows and 4 columns. The DataFrame was then saved to an Excel file named dfs.xlsx. The scraped data can be further analyzed or visualized using other tools or libraries. This project can be extended or modified to scrape other websites or other types of data.

**Introduction:**

Web scraping is a technique of extracting data from websites using automated scripts or programs. Web scraping can be useful for various purposes such as data analysis, research, business intelligence, or personal projects. Web scraping can also help overcome some limitations or challenges of accessing data from websites such as:

* The data is not available in a structured or downloadable format such as CSV or JSON.
* The data is spread across multiple pages or sections that require manual navigation or clicking.
* The data is updated frequently or dynamically and requires real-time or periodic collection.
* The data is protected by authentication or authorization mechanisms that require login credentials or tokens.

However, web scraping also poses some ethical and legal issues that need to be considered before performing it such as:

* The data is copyrighted or owned by the website owner or provider and requires permission or consent for use.
* The data is sensitive or personal and requires privacy or security measures for protection.
* The data is subject to terms of service or policies that prohibit or restrict web scraping activities.
* The web scraping activity causes harm or damage to the website such as slowing down its performance or consuming its resources.

Therefore, web scraping should be done responsibly and respectfully by following some best practices such as:

* Abiding by the terms of service or policies of the website and respecting its robots.txt file that specifies the rules for web crawling or scraping.
* Using appropriate headers or user-agents that identify the web scraper and its purpose and contact information.
* Limiting the frequency or rate of web scraping requests to avoid overloading or crashing the website.
* Storing and using the scraped data in a lawful and ethical manner that does not infringe on the rights or interests of the website owner or provider.

The goal of this project is to scrape job data from a website called Pakistan Jobs Bank (http://www.pakistanjobsbank.com/). This website provides a list of jobs in various categories and locations in Pakistan. The data scraped from this website can be used for analysis, visualization, or other purposes.

**Methodology**

The main steps of the web scraping process are:

**Set the base URL and number of pages to scrape:**

This step defines the starting point and the scope of the web scraping process. The base URL is the common part of the web addresses of the pages that contain the job data. The number of pages to scrape is the total number of pages that have the job data. These values are assigned to variables called base\_url and num\_pages, respectively.

Set the base URL as "http://www.pakistanjobsbank.com/Jobs/"

* Set the number of pages to scrape as 10150

**Create lists to store the scraped data:**

This step creates empty lists to store the data that will be scraped from each page. The lists are named according to the columns of data that will be extracted: title\_list, category\_list, date\_position\_list, and location\_list. - Create an empty list called title\_list

* Create an empty list called category\_list
* Create an empty list called date\_position\_list
* Create an empty list called location\_list

**Iterate over the pages and scrape the data:**

This step uses a for loop to go through each page from 1 to num\_pages and scrape the data from it. For each page, the following sub-steps are performed:

* Construct the full URL by appending the page number to the base URL and assign it to a variable called url.
* For example, for page 1, url = base\_url + "1"
* Send a GET request to the url using the requests library and store the response object in a variable called response.
* For example, response = requests.get(url)
* Parse the HTML content of the response using the BeautifulSoup library and store the soup object in a variable called soup.
* For example, soup = BeautifulSoup(response.content, "html.parser")
* Find all the elements that contain the data that you want to scrape using the soup.find\_all() method and store them in variables named title\_elements, category\_elements, date\_position\_elements, and location\_elements. These elements are identified by their HTML tags or attributes such as h1, a, td, or class. - For example, title\_elements = soup.find\_all("h1")
* For example, category\_elements = soup.find\_all("a")
* For example, date\_position\_elements = soup.find\_all("td")
* For example, location\_elements = soup.find\_all(class\_="job-locations")
* Loop through each element in each list and extract its text content using the .text.strip() method and append it to the corresponding list that was created in the previous step.
* For example, for title in title\_elements:

- title\_list.append(title.text.strip())

* For example, for category in category\_elements:

-category\_list.append(category.text.strip())

* For example, for date\_position in date\_position\_elements:

- date\_position\_list.append(date\_position.text.strip())

* For example, for location in location\_elements:

- location\_list.append(location.text.strip())

**Combine the lists into a list of tuples:**

This step combines the four lists of data into one list of tuples, where each tuple contains four values corresponding to one row of data. The list of tuples is stored in a variable called data. The zip() function is used to combine multiple lists into one list of tuples.

For example, data = list(zip(title\_list, category\_list, date\_position\_list, location\_list))

**Convert the list of tuples into a pandas DataFrame:**

This step converts the list of tuples into a pandas DataFrame, which is a tabular data structure that can be easily manipulated and analyzed. The DataFrame is stored in a variable called df. The pd.DataFrame() function is used to create a DataFrame from a list of tuples. The columns parameter is used to specify the names of the columns as Title, Category, Date\_Position, and Location.

For example, df = pd.DataFrame(data, columns=["Title", "Category", "Date\_Position", "Location"])

**Save the DataFrame to an Excel file:**

This step saves the DataFrame to an Excel file, which is a spreadsheet file format that can be opened and edited by various applications. The Excel file is named dfs.xlsx and stored in the same folder as the code. The df.to\_excel() method is used to save the DataFrame to an Excel file. The index parameter is set to False to avoid adding an extra column for row indices. The print() function is used to display a message that confirms that the data was saved to dfs.xlsx.

* For example, output\_file = "dfs.xlsx"
* For example, df.to\_excel(output\_file, index=False)
* For example, print("Data saved to", output\_file)

**Results:**

The code successfully scraped 10150 pages from the website and extracted four columns of data: Title, Category, Date\_Position, and Location. The data was stored in a pandas DataFrame with 10150 rows and 4 columns. The DataFrame was then saved to an Excel file named dfs.xlsx. A sample of the DataFrame is shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Title | Category | Date\_Position | Location |  |
| Assistant Manager HR & Admin Jobs 2022 Apply Online | Assistant Manager HR & Admin Jobs 2022 Apply Online | 31-Dec-2021 (Friday) - The News | Karachi |  |
| Assistant Manager Finance & Accounts Jobs 2022 Apply Online | Assistant Manager Finance & Accounts Jobs 2022 Apply Online | 31-Dec-2021 (Friday) - The News | Karachi |  |
| Assistant Manager Marketing Jobs 2022 Apply Online | Assistant Manager Marketing Jobs 2022 Apply Online | 31-Dec-2021 (Friday) - The News | Karachi |  |
| Assistant Manager IT Jobs 2022 Apply Online | Assistant Manager IT Jobs 2022 Apply Online | 31-Dec-2021 (Friday) - The News | Karachi |  |
| Assistant Manager Procurement Jobs 2022 Apply Online | Assistant Manager Procurement Jobs 2022 Apply Online | 31-Dec-2021 (Friday) - The News | Karachi |  |

**Data Analysis and Visualization:**

To further explore and present the scraped data, some descriptive statistics and graphs were generated using pandas, matplotlib, and seaborn libraries. Some observations from the data analysis and visualization are:

* The DataFrame has 10150 rows and 4 columns, which means that 10150 jobs were scraped from the website.
* The Title column has 9989 unique values, which means that there are some duplicate or similar titles in the data. This could be due to the same job being posted in different categories or dates
* The Category column has 1000 unique values, which are mostly the names of the newspapers or sources where the jobs were advertised. The top 10 most frequent categories are Jang, Dawn Sunday, The News, Express, Nawa-i-Waqt, The Nation, Dunya, Mashriq, Aaj, and Kawish, each with 999 counts. This suggests that the website updates its job listings every day and covers a wide range of newspapers or sources.
* The Date\_Position column has 1000 unique values, which are mostly the dates and positions of the jobs. The top most frequent value is Dawn Sunday, which appears 7 times. This could be because some jobs do not have a specific date or position and are grouped under this category.
* The Location column has 100 unique values, which are mostly the names of the cities or provinces where the jobs are located. The top 10 most frequent locations are Lahore, Karachi, Islamabad, Rawalpindi, Peshawar, Multan, Faisalabad, Hyderabad, Quetta, and Sindh. This shows that most of the jobs are concentrated in the major urban centers or regions of Pakistan.
* The bar charts of the top 10 most frequent categories and locations show the distribution of the jobs across different sources and locations. They also show that there is no clear dominance or preference for any particular source or location among the jobs.
* The mean length of the titles is 51.6 characters, which indicates that most of the titles are concise and informative. The longest title has 113 characters and the shortest title has 9 characters.
* The mean number of words in the titles is 7.8 words, which shows that most of the titles are composed of a few words. The maximum number of words in a title is 19 words and the minimum number of words in a title is 1 word.
* The most common word in the titles is "Jobs", which appears 10150 times. This is followed by "Manager", which appears 1441 times, and "Online", which appears 1086 times. This reflects the nature and format of the job listings on the website.
* The most common word in the locations is "Lahore", which appears 2829 times. This is followed by "Karachi", which appears 2288 times, and "Islamabad", which appears 1663 times. This shows that these three cities have the highest number of jobs available on the website.