Project Title:

"Web Scraping Project: Extracting Data from [Website URL] and Saving to CSV"

Name:

Mohsin Raza

Organization:

AKSA_SDS

Date:

23/07/2025

Table of Contents

Table of Figures

Title

WEB SCRAPING PROJECT: EXTRACTING DATA FROM WEBSITE URL AND SAVING TO CSV

Abstract

⇒ Brief overview of the project

⇒ Purpose: Extract data from a website and save it to a CSV file

⇒ Methodology: Use web scraping techniques with Python and relevant libraries

⇒ Outcome: Successfully extracted data stored in a CSV file for further analysis

Introduction

Background and Context

⇒ Background: Importance of web scraping in data extraction

⇒ Problem Statement: Manual data extraction is time-consuming and inefficient

⇒ Solution: Automate data extraction using web scraping techniques

⇒ Objective: Extract specific data from a website and save it to a CSV file

Objectives

⇒ Extract specific data from a website (e.g., product information, reviews, etc.)

⇒ Store the extracted data in a structured format (CSV file)

⇒ Automate the data extraction process using web scraping techniques

Methodology

⇒ Use Python as the programming language

⇒ Utilize relevant libraries (e.g., BeautifulSoup, requests, pandas) for web scraping and data manipulation

⇒ Inspect the website's HTML structure to identify the data to be extracted

⇒ Write a script to send HTTP requests to the website and parse the HTML responses

⇒ Extract the desired data and store it in a CSV file

Tools, Materials, and Technologies Applied

⇒ Python programming language

⇒ BeautifulSoup library for HTML parsing requests library for sending HTTP requests pandas library for data manipulation and CSV export

⇒ CSV (Comma Separated Values) file format for data storage

Results

⇒ Successfully extracted data from the website

⇒ Data stored in a CSV file with the desired structure

⇒ Automation of the data extraction process using web scraping techniques

Analysis

⇒ The extracted data can be used for further analysis, such as data visualization, statistical analysis, or machine learning

⇒ The CSV file can be easily imported into various data analysis tools and software

Code

```
import requests
from bs4 import BeautifulSoup
import csv
import os

data_path = r'D:\INTERNSHIP'

def scrape_url(url):
    try:
        # Send a GET request to the URL
        response = requests.get(url)

        # Check if the request was successful
        if response.status_code == 200:
            # Parse the HTML content using BeautifulSoup
            soup = BeautifulSoup(response.content, 'html.parser')

            # Find all the paragraph and heading texts
            paragraphs = soup.find_all(['p', 'h1', 'h2', 'h3', 'h4', 'h5', 'h6'])

            # Create a list to store the scraped data
            data = []

            # Loop through the paragraphs and extract the text
            for paragraph in paragraphs:
                data.append(paragraph.text.strip())

            return data
        else:
            print("Failed to retrieve the webpage")
            return None
```

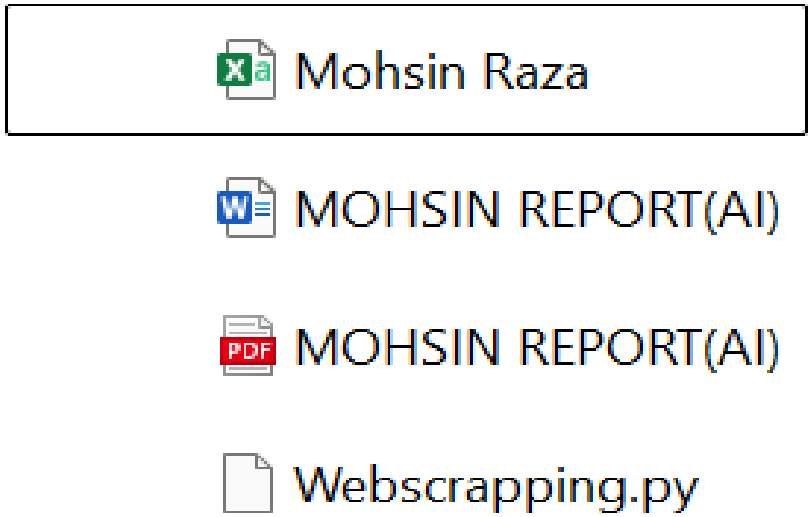
```
    except Exception as e:
        print("An error occurred: ", str(e))
        return None

def save_to_csv(data, filename):
    try:
        # Ensure the data_path directory exists
        os.makedirs(data_path, exist_ok=True)

        # Full path for the csv file
        full_path = os.path.join(data_path, filename)

        # Create a CSV file and write the data to it
        with open(full_path, 'w', newline='', encoding='utf-8') as csvfile:
            writer = csv.writer(csvfile)
            writer.writerow(["Data"])  # header row
            for item in data:
                writer.writerow([item])
        print("Data scraped and saved to", full_path)
    except Exception as e:
        print("An error occurred while saving to CSV:", str(e))

def main():
    url = input("Enter the URL to scrape: ")
    filename = input("Enter the filename to save (default: output.csv): ")
    if filename.strip() == "":
        filename = "output.csv"
    if not filename.endswith(".csv"):
        filename += ".csv"

    data = scrape_url(url)
    if data is not None:
        save_to_csv(data, filename)

if __name__ == "__main__":
    main()
```

Outputs

⇒ A CSV file containing the extracted data

⇒ A Python script that can be used to automate the data extraction process

```
In [1]: runcell(0, 'D:/INTERNSHIP/Webscrapping.py')
Enter the URL to scrape: https://www.cgtn.com
Enter the filename to save (default: output.csv): Mohsin Raza
Data scraped and saved to D:\INTERNSHIP\Mohsin Raza.csv
```



Conclusion

⇒ Web scraping is a powerful technique for extracting data from websites

⇒ The project demonstrates the effectiveness of using Python and relevant libraries for web scraping and data storage in a CSV file

⇒ The extracted data can be used for various purposes, such as data analysis, market research, or business intelligence

	Data
1	Data
2	Home
3	Our Privacy Statement & Cookie Policy
4	By continuing to browse our site you agree to our use of cookies, revised Privacy Policy and Terms of Use. You can change your cookie settings through your browser.
5	Xi: China, EU are forces for multilateralism, openness and cooperation
6	China urges opposition to unilateral tariffs at WTO
7	Live updates: Thai and Cambodian troops clash at disputed border area
8	Russia, Ukraine agree on prisoner exchange, differ on ceasefire
9	WHO says Gaza facing man-made 'mass starvation' as hunger deaths surge
10	BREAKING NEWS
11	Russia rescuers find missing plane in flames: ministry
12	256 killed, 616 injured in torrential rain-related incidents across Pakistan since June 26
13	South Korea's Lee: I am very pleased to have my first telephone conversation with Chancellor Merz today
14	Japanese media: When asked by reporters about his future plans, Prime Minister Ishiba said, "I will continue to do my utmost" in response to the Japan-U.S. tariff negotiations
15	How China Works
16	Thailand, Cambodia exchange border fire
17	Xi: China and EU leaders must show vision and make wise choices
18	Chinese premier: Cooperation should be keynote of China-EU relations

Mohsin Raza

References

Programming and Data Structures

⇒ List of sources used in the project, including:

⇒ Documentation for Python libraries (e.g., BeautifulSoup, requests, pandas, csv, os)

⇒ Web scraping tutorials and guides

⇒ Relevant research papers or articles on web scraping and data extraction

Formatting Tips

- **Font** : Times New Roman, 12pt
- **Spacing** : 1.5 line spacing
- **Margins** : 1-inch margins on all sides
- **Page Numbers:** On the top right corner of each page