ASSIGNMENT-1 WEB SCRAPING

1) Write a python program to display all the header tags from wikipedia.org and make data frame.
Solution):

```
from urllib.request import urlopen

from bs4 import BeautifulSoup

html = urlopen('https://en.wikipedia.org/wiki/Main_Page')

bs = BeautifulSoup(html, "html.parser")

titles = bs.find_all(['h1', 'h2','h3','h4','h5','h6'])

print('List all the header tags :', *titles, sep='\n\n')
```

2) Write a python program to display IMDB's Top rated 50 movies' data (i.e. name, rating, year of release) and make data frame.
Solution):

```
import requests

from bs4 import BeautifulSoup

import pandas as pd


# Fetch the HTML content from the URL

url = "https://www.imdb.com/chart/top"

response = requests.get(url)

html_content = response.content


# Parse the HTML content using BeautifulSoup

soup = BeautifulSoup(html_content, "html.parser")


# Find all the movie details using their CSS selectors

movie_titles = soup.select(".titleColumn a")

movie_ratings = soup.select(".imdbRating strong")

movie_years = soup.select(".titleColumn span.secondaryInfo")


# Create a list of dictionaries to store the movie details

movie_list = []
```

```
for i in range(50):
    movie_dict = {}
    movie_dict["Name"] = movie_titles[i].text
    movie_dict["Rating"] = movie_ratings[i].text
    movie_dict["Year"] = movie_years[i].text.strip("()")
    movie_list.append(movie_dict)

# Create a pandas dataframe from the list of dictionaries
movie_df = pd.DataFrame(movie_list)

# Display the dataframe
print(movie_df)
```

3) Write a python program to display IMDB's Top rated 50 Indian movies' data (i.e. name, rating, year of release) and make data frame.

Solution):

```
        import requests
from bs4 import BeautifulSoup
import pandas as pd

# Fetch the HTML content from the URL
url = "https://www.imdb.com/india/top-rated-indian-movies"
response = requests.get(url)
html_content = response.content

# Parse the HTML content using BeautifulSoup
soup = BeautifulSoup(html_content, "html.parser")

# Find all the movie details using their CSS selectors
movie_titles = soup.select(".titleColumn a")
movie_ratings = soup.select(".ratingColumn strong")
movie_years = soup.select(".secondaryInfo")
```

```
# Create a list of dictionaries to store the movie details
movie_list = []
for i in range(50):
    movie_dict = {}
    movie_dict["Name"] = movie_titles[i].text
    movie_dict["Rating"] = movie_ratings[i].text
    movie_dict["Year"] = movie_years[i].text.strip("()")
    movie_list.append(movie_dict)

# Create a pandas dataframe from the list of dictionaries
movie_df = pd.DataFrame(movie_list)

# Display the dataframe
print(movie_df)
```

4) Write s python program to display list of respected former presidents of India(i.e. Name , Term ofoffice) from https://presidentofindia.nic.in/former-presidents.htm and make data frame.
Solution):

```
        import requests
from bs4 import BeautifulSoup
import pandas as pd

# Fetch the HTML content from the URL
url = "https://presidentofindia.nic.in/former-presidents.htm"
response = requests.get(url)
html_content = response.content

# Parse the HTML content using BeautifulSoup
soup = BeautifulSoup(html_content, "html.parser")

# Find all the former presidents using their CSS selectors
presidents = soup.select(".table-responsive tbody tr")
```

```
# Create a list of dictionaries to store the president details
president_list = []
for president in presidents:
    president_dict = {}
    president_dict["Name"] = president.select("td")[0].text.strip()
    president_dict["Term of Office"] = president.select("td")[1].text.strip()
    president_list.append(president_dict)

# Create a pandas dataframe from the list of dictionaries
president_df = pd.DataFrame(president_list)

# Display the dataframe
print(president_df)
```

5) Write a python program to scrape cricket rankings from icc-cricket.com. You have to scrape and make data framea) Top 10 ODI teams in men's cricket along with the records for matches, points and rating. b) Top 10 ODI Batsmen along with the records of their team andrating. c) Top 10 ODI bowlers along with the records of their team andrating
Solution):

```
        import requests
from bs4 import BeautifulSoup
import pandas as pd

# Fetch the HTML content from the URL
url = "https://www.icc-cricket.com/rankings/mens/team-rankings/odi"
response = requests.get(url)
html_content = response.content

# Parse the HTML content using BeautifulSoup
soup = BeautifulSoup(html_content, "html.parser")

# Find all the table rows for the teams
```

```
teams = soup.select(".table-body tbody tr")


# Create a list of dictionaries to store the team details
team_list = []
for team in teams[:10]:
    team_dict = {}
    team_dict["Team"] = team.select("td")[1].text.strip()
    team_dict["Matches"] = team.select("td")[2].text.strip()
    team_dict["Points"] = team.select("td")[3].text.strip()
    team_dict["Rating"] = team.select("td")[4].text.strip()
    team_list.append(team_dict)


# Create a pandas dataframe from the list of dictionaries
team_df = pd.DataFrame(team_list)


# Display the dataframe
print(team_df)
```

6) Write a python program to scrape cricket rankings from icc-cricket.com. You have to scrape and make data framea) Top 10 ODI teams in women's cricket along with the records for matches, points and rating. b) Top 10 women's ODI Batting players along with the records of their team and rating. c) Top 10 women's ODI all-rounder along with the records of their team and rating.
Solution):

```
        import requests
from bs4 import BeautifulSoup
import pandas as pd


# Fetch the HTML content from the URL
url = "https://www.icc-cricket.com/rankings/womens/team-rankings/odi"
response = requests.get(url)
html_content = response.content


# Parse the HTML content using BeautifulSoup
```

```
soup = BeautifulSoup(html_content, "html.parser")

# Find all the table rows for the teams
teams = soup.select(".table-body tbody tr")

# Create a list of dictionaries to store the team details
team_list = []
for team in teams[:10]:
    team_dict = {}
    team_dict["Team"] = team.select("td")[1].text.strip()
    team_dict["Matches"] = team.select("td")[2].text.strip()
    team_dict["Points"] = team.select("td")[3].text.strip()
    team_dict["Rating"] = team.select("td")[4].text.strip()
    team_list.append(team_dict)

# Create a pandas dataframe from the list of dictionaries
team_df = pd.DataFrame(team_list)

# Display the dataframe
print(team_df)
```

7) Write a python program to scrape mentioned news details from https://www.cnbc.com/world/?region=world and make data framei) Headline ii) Time iii) News Link
Solution):

```
        import requests
from bs4 import BeautifulSoup
import pandas as pd

# Fetch the HTML content from the URL
url = "https://www.cnbc.com/world/?region=world"
response = requests.get(url)
html_content = response.content
```

```
# Parse the HTML content using BeautifulSoup
soup = BeautifulSoup(html_content, "html.parser")

# Find all the news articles on the page
articles = soup.select(".Card-titleContainer")

# Create a list of dictionaries to store the news details
news_list = []
for article in articles:
    news_dict = {}
    news_dict["Headline"] = article.select_one("a").text.strip()
    news_dict["Time"] = article.select_one(".Card-time").text.strip()
    news_dict["News Link"] = article.select_one("a")["href"]
    news_list.append(news_dict)

# Create a pandas dataframe from the list of dictionaries
news_df = pd.DataFrame(news_list)

# Display the dataframe
print(news_df)
```

8) Write a python program to scrape the details of most downloaded articles from AI in last 90 days.https://www.journals.elsevier.com/artificial-intelligence/most-downloaded-articles Scrape below mentioned details and make data framei) Paper Title ii) Authors iii) Published Date iv) Paper URL

Solution):

```
import requests
from bs4 import BeautifulSoup
import pandas as pd

url = 'https://www.journals.elsevier.com/artificial-intelligence/most-downloaded-articles'
response = requests.get(url)
```

```
soup = BeautifulSoup(response.content, 'html.parser')

articles = soup.find_all('div', class_='pod-listing js-pod-clickable')


paper_titles = []

authors = []

published_dates = []

paper_urls = []


for article in articles:

    title = article.find('h4', class_='title').text.strip()

    paper_titles.append(title)


    author_list = []

    authors_html = article.find_all('span', class_='text-sm text-gray-600')

    for author in authors_html:

        author_list.append(author.text.strip())

    authors.append(', '.join(author_list))


    published_dates.append(article.find('span', class_='pod-listing__published-date').text.strip())


    paper_urls.append('https://www.journals.elsevier.com' + article.find('a')['href'])


data = {'Paper Title': paper_titles, 'Authors': authors, 'Published Date': published_dates, 'Paper URL':
paper_urls}


df = pd.DataFrame(data)

print(df)
```

9) Write a python program to scrape mentioned details from dineout.co.in and make data framei) Restaurant name ii) Cuisine iii) Location iv) Ratings v) Image URL
Solution):

```
        import requests

from bs4 import BeautifulSoup
```

```
import pandas as pd

url = 'https://www.dineout.co.in/delhi-restaurants'

response = requests.get(url)

soup = BeautifulSoup(response.content, 'html.parser')

restaurant_names = []
cuisines = []
locations = []
ratings = []
image_urls = []

for item in soup.select('.restnt-card'):
    restaurant_names.append(item.select('.restnt-card__title > h2')[0].get_text())
    cuisines.append(item.select('.restnt-card__cusine > span')[0].get_text())
    locations.append(item.select('.restnt-card__address > p')[0].get_text())
    ratings.append(item.select('.restnt-card__ratings > span')[0].get_text())
    image_urls.append(item.select('.restnt-card__img > img')[0]['src'])

df = pd.DataFrame({'Restaurant Name': restaurant_names,
                   'Cuisine': cuisines,
                   'Location': locations,
                   'Ratings': ratings,
                   'Image URL': image_urls})

print(df)
```