WEB SCRAPING – ASSIGNMENT 4

- **Read all the problem statements, notes carefully and scrape the required data using any web scraping tool of your choice.**
- **You have to handle commonly occurring EXCEPTIONS by using exception handling programing. To get information about selenium Exceptions. You may visit following links:**
    1. https://selenium-python.readthedocs.io/api.html
    2. https://www.guru99.com/exception-handling-selenium.html
    3. https://stackoverflow.com/questions/38022658/selenium-python-handling-no-such-elementexception/38023345

1) Scrape the details of most viewed videos on YouTube from Wikipedia. Url = https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos

    You need to find following details:
    A) Rank
    B) Name
    C) Artist
    D) Upload date
    E) Views

Answer:

```
import requests

from bs4 import BeautifulSoup


# Send a GET request to the Wikipedia page

url = "https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos"

response = requests.get(url)


# Create a BeautifulSoup object to parse the HTML content

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


# Find the table containing the most viewed videos

table = soup.find("table", class_="wikitable")


# Iterate over each row in the table (skipping the header row)

rows = table.find_all("tr")[1:]

for row in rows:

    # Extract the data from each cell in the row

    cells = row.find_all("td")

    rank = cells[0].text.strip()
    video_name = cells[1].text.strip()
    artist = cells[2].text.strip()
    upload_date = cells[3].text.strip()
    views = cells[4].text.strip()

    # Print the details of each video
    print("Rank:", rank)
    print("Name:", video_name)
    print("Artist:", artist)
    print("Upload Date:", upload_date)
    print("Views:", views)
    print("--------------------")
```

2. Scrape the details teamIndia'sinternationalfixtures from bcci.tv. Url = https://www.bcci.tv/.

You need to find following details:

A) Match title (I.e. 1stODI)

B) Series

C) Place

D) Date

E) Time

Note: - From bcci.tv home page you have reach to the international fixture page through code.

Answer:

```
import requests
from bs4 import BeautifulSoup

# Send a GET request to the BCCI.tv home page
url = "https://www.bcci.tv/"
response = requests.get(url)

# Create a BeautifulSoup object to parse the HTML content
soup = BeautifulSoup(response.content, "html.parser")
```

```
# Find the link to the international fixtures page

fixtures_link = soup.find("a", string="International Fixtures")["href"]


# Construct the URL for the international fixtures page

fixtures_url = f"https://www.bcci.tv{fixtures_link}"


# Send a GET request to the international fixtures page

fixtures_response = requests.get(fixtures_url)


# Create a new BeautifulSoup object for the fixtures page

fixtures_soup = BeautifulSoup(fixtures_response.content, "html.parser")


# Find the container for the fixtures

fixtures_container = fixtures_soup.find("div", class_="js-list")


# Find all the fixtures

fixtures = fixtures_container.find_all("div", class_="fixture-widget")


# Iterate over each fixture and extract the details

for fixture in fixtures:

    match_title = fixture.find("p", class_="fixture__additional-info").text.strip()

    series = fixture.find("span", class_="u-unskewed-text").text.strip()

    place = fixture.find("p", class_="fixture__additional-info").find_next("span").text.strip()

    date = fixture.find("div", class_="fixture__datetime").find("span",
class_="fixture__date").text.strip()

    time = fixture.find("div", class_="fixture__datetime").find("span",
class_="fixture__time").text.strip()


    # Print the details of each fixture

    print("Match Title:", match_title)

    print("Series:", series)
    print("Place:", place)

    print("Date:", date)

    print("Time:", time)

    print("--------------------")
```

3. Scrape the details of State-wise GDP ofIndia fromstatisticstime.com. Url = http://statisticstimes.com/

You have to find following details:

A) Rank

B) State

C) GSDP(18-19)- at current prices

D) GSDP(19-20)- at current prices

E) Share(18-19)

F) GDP($ billion)

Note: - From statisticstimes home page you have to reach to economy page through code.

Answer:

```
import requests

from bs4 import BeautifulSoup


# Send a GET request to the statisticstimes.com home page

url = "http://statisticstimes.com/"

response = requests.get(url)


# Create a BeautifulSoup object to parse the HTML content

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


# Find the link to the Economy page

economy_link = soup.find("a", text="Economy")["href"]


# Construct the URL for the Economy page

economy_url = f"http://statisticstimes.com{economy_link}"
```

```
# Send a GET request to the Economy page
economy_response = requests.get(economy_url)

# Create a new BeautifulSoup object for the Economy page
economy_soup = BeautifulSoup(economy_response.content, "html.parser")

# Find the link to the State-wise GDP page
gdp_link = economy_soup.find("a", text="GDP of Indian states")["href"]

# Construct the URL for the State-wise GDP page
gdp_url = f"http://statisticstimes.com{gdp_link}"

# Send a GET request to the State-wise GDP page
gdp_response = requests.get(gdp_url)

# Create a new BeautifulSoup object for the State-wise GDP page
gdp_soup = BeautifulSoup(gdp_response.content, "html.parser")

# Find the table containing the State-wise GDP data
table = gdp_soup.find("table", class_="display dataTable")

# Find all the rows in the table (excluding the header row)
rows = table.find_all("tr")[1:]

# Iterate over each row and extract the details
for row in rows:
   # Extract the data from each cell in the row
   cells = row.find_all("td")
   rank = cells[0].text.strip()
   state = cells[1].text.strip()
   gdp_18_19 = cells[2].text.strip()
```

```
    gdp_19_20 = cells[3].text.strip()
    share_18_19 = cells[4].text.strip()
    gdp_billion = cells[5].text.strip()

    # Print the details of each state
    print("Rank:", rank)
    print("State:", state)
    print("GSDP(18-19) - at current prices:", gdp_18_19)
    print("GSDP(19-20) - at current prices:", gdp_19_20)
    print("Share(18-19):", share_18_19)
    print("GDP($ billion):", gdp_billion)
    print("--------------------")
```

4. Scrape the details of trending repositories on Github.com. Url = https://github.com/ You have to find the following details:

A) Repository title

B) Repository description

C) Contributors count

D) Language used

Note: - From the home page you have to click on the trending option from Explore menu through code.

Answer:
```
import requests
from bs4 import BeautifulSoup

# Send a GET request to the GitHub home page
url = "https://github.com/"
response = requests.get(url)

# Create a BeautifulSoup object to parse the HTML content
soup = BeautifulSoup(response.content, "html.parser")

# Find the link to the Trending page
```

```
explore_menu = soup.find("a", text="Explore")
trending_link = explore_menu.find_next("a", text="Trending")["href"]

# Construct the URL for the Trending page
trending_url = f"https://github.com{trending_link}"

# Send a GET request to the Trending page
trending_response = requests.get(trending_url)

# Create a new BeautifulSoup object for the Trending page
trending_soup = BeautifulSoup(trending_response.content, "html.parser")

# Find the container for the trending repositories
repo_list = trending_soup.find_all("article", class_="Box-row")

# Iterate over each repository and extract the details
for repo in repo_list:
    # Extract the data from each element in the repository container
    repo_title = repo.find("h1", class_="h3 lh-condensed").text.strip()
    repo_desc = repo.find("p", class_="col-9 color-text-secondary my-1 pr-4").text.strip()
    contributors_count = repo.find("a", class_="muted-link d-inline-block mr-3").text.strip()
    language = repo.find("span", itemprop="programmingLanguage").text.strip()

    # Print the details of each repository
    print("Repository Title:", repo_title)
    print("Repository Description:", repo_desc)
    print("Contributors Count:", contributors_count)
    print("Language Used:", language)
    print("--------------------")
```

5. Scrape the details of top 100 songs on billiboard.com. Url = https:/www.billboard.com/ You have to find the following details:

A) Song name

B) Artist name

C) Last week rank

D) Peak rank

E) Weeks on board

Note: - From the home page you have to click on the charts option then hot 100-page link through code.

Answer:
```
import requests

from bs4 import BeautifulSoup


# Send a GET request to the Billboard.com home page

url = "https://www.billboard.com/"

response = requests.get(url)


# Create a BeautifulSoup object to parse the HTML content

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


# Find the link to the Hot 100 page

charts_link = soup.find("a", text="Charts")["href"]

hot_100_link = soup.find("a", text="Hot 100")["href"]


# Construct the URL for the Hot 100 page

hot_100_url = f"https://www.billboard.com{charts_link}{hot_100_link}"


# Send a GET request to the Hot 100 page

hot_100_response = requests.get(hot_100_url)


# Create a new BeautifulSoup object for the Hot 100 page

hot_100_soup = BeautifulSoup(hot_100_response.content, "html.parser")
```

```
# Find the container for the top 100 songs
song_list = hot_100_soup.find_all("li", class_="chart-list__element")


# Iterate over each song and extract the details
for song in song_list:
    # Extract the data from each element in the song container
    song_name = song.find("span", class_="chart-element__information__song text--truncate color-
-primary").text.strip()
    artist_name = song.find("span", class_="chart-element__information__artist text--truncate
color--secondary").text.strip()
    last_week_rank = song.find("span", class_="chart-element__meta text--center color--secondary
text--last").text.strip()
    peak_rank = song.find("span", class_="chart-element__meta text--center color--secondary text--
peak").text.strip()
    weeks_on_board = song.find("span", class_="chart-element__meta text--center color--
secondary text--week").text.strip()


    # Print the details of each song
    print("Song Name:", song_name)
    print("Artist Name:", artist_name)
    print("Last Week Rank:", last_week_rank)
    print("Peak Rank:", peak_rank)
    print("Weeks on Board:", weeks_on_board)
    print("--------------------")
```

6. Scrape the details of Highest sellingnovels. Url = https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-books-all-time-fifty-shades-greycompare You have to find the following details:

A) Book name

B) Author name

C) Volumes sold

D) Publisher

E) Genre

Answer: import requests

```
from bs4 import BeautifulSoup


# Send a GET request to the webpage

url = "https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-books-all-time-
fifty-shades-greycompare"

response = requests.get(url)


# Create a BeautifulSoup object to parse the HTML content

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


# Find the table containing the highest selling novels data

table = soup.find("table", class_="in-article sortable")


# Find all the rows in the table (excluding the header row)

rows = table.find_all("tr")[1:]


# Iterate over each row and extract the details

for row in rows:

    # Extract the data from each cell in the row

    cells = row.find_all("td")

    book_name = cells[0].text.strip()

    author_name = cells[1].text.strip()

    volumes_sold = cells[2].text.strip()

    publisher = cells[3].text.strip()

    genre = cells[4].text.strip()


    # Print the details of each novel

    print("Book Name:", book_name)

    print("Author Name:", author_name)

    print("Volumes Sold:", volumes_sold)

    print("Publisher:", publisher)
    print("Genre:", genre)
    print("---------------------")
```

7. Scrape the details most watched tv series of all time from imdb.com. Url = https://www.imdb.com/list/ls095964455/

You have to find the following details:

A) Name

B) Year span

C) Genre

D) Run time

E) Ratings

F) Votes

Answer:
```
import requests
from bs4 import BeautifulSoup

# Send a GET request to the IMDb webpage
url = "https://www.imdb.com/list/ls095964455/"
response = requests.get(url)

# Create a BeautifulSoup object to parse the HTML content
soup = BeautifulSoup(response.content, "html.parser")

# Find the container for the TV series
series_list = soup.find_all("div", class_="lister-item mode-detail")

# Iterate over each TV series and extract the details
for series in series_list:
    # Extract the data from each element in the TV series container
    name = series.find("h3", class_="lister-item-header").a.text.strip()
    year_span = series.find("span", class_="lister-item-year text-muted unbold").text.strip()
    genre = series.find("span", class_="genre").text.strip()
    runtime = series.find("span", class_="runtime").text.strip()
    ratings = series.find("span", class_="ipl-rating-star__rating").text.strip()
    votes = series.find("span", attrs={"name": "nv"}).text.strip()

    # Print the details of each TV series
    print("Name:", name)
    print("Year Span:", year_span)
    print("Genre:", genre)
    print("Run Time:", runtime)
    print("Ratings:", ratings)
    print("Votes:", votes)
    print("---------------------")
```

8. Details of Datasetsfrom UCI machine learning repositories. Url = https://archive.ics.uci.edu/ You have to find the following details:

A) Dataset name

B) Data type

C) Task

D) Attribute type

E) No of instances

F) No of attribute

G) Year

Note: - from the home page you have to go to the ShowAllDataset page through code.

Answer:
```
import requests
from bs4 import BeautifulSoup

# Send a GET request to the UCI Machine Learning Repository home page
url = "https://archive.ics.uci.edu/"
response = requests.get(url)

# Create a BeautifulSoup object to parse the HTML content
soup = BeautifulSoup(response.content, "html.parser")
```

```
# Find the link to the Show All Dataset page
show_all_link = soup.find("a", text="Show All Dataset")["href"]

# Construct the URL for the Show All Dataset page
show_all_url = f"{url}{show_all_link}"

# Send a GET request to the Show All Dataset page
show_all_response = requests.get(show_all_url)

# Create a new BeautifulSoup object for the Show All Dataset page
show_all_soup = BeautifulSoup(show_all_response.content, "html.parser")

# Find the table containing the dataset details
table = show_all_soup.find("table", class_="table")

# Find all the rows in the table (excluding the header row)
rows = table.find_all("tr")[1:]

# Iterate over each row and extract the details
for row in rows:
    # Extract the data from each cell in the row
    cells = row.find_all("td")
    dataset_name = cells[0].text.strip()
    data_type = cells[1].text.strip()
    task = cells[2].text.strip()
    attribute_type = cells[3].text.strip()
    no_of_instances = cells[4].text.strip()
    no_of_attributes = cells[5].text.strip()
    year = cells[6].text.strip()

    # Print the details of each dataset
    print("Dataset Name:", dataset_name)
    print("Data Type:", data_type)
    print("Task:", task)
    print("Attribute Type:", attribute_type)
    print("No of Instances:", no_of_instances)
    print("No of Attributes:", no_of_attributes)
    print("Year:", year)
    print("--------------------")
```

9. Scrape the details of Data science recruiters Url = https://www.naukri.com/hr-recruiters-consultants

You have to find the following details:

A) Name

B) Designation

C)Company

D)Skills they hire for

E) Location

Note: - From naukri.com homepage click on the recruiters option and the on the search pane type Data science and click on search. All this should be done through code

Answer:

```
import requests
from bs4 import BeautifulSoup

# Send a GET request to the Naukri.com homepage
url = "https://www.naukri.com/"
response = requests.get(url)

# Create a BeautifulSoup object to parse the HTML content
soup = BeautifulSoup(response.content, "html.parser")

# Find the link to the Recruiters page
recruiters_link = soup.find("a", text="Recruiters")["href"]

# Construct the URL for the Recruiters page
```

```
recruiters_url = f"{url}{recruiters_link}"

# Send a GET request to the Recruiters page
recruiters_response = requests.get(recruiters_url)

# Create a new BeautifulSoup object for the Recruiters page
recruiters_soup = BeautifulSoup(recruiters_response.content, "html.parser")

# Find the search form and set the search query to "Data science"
search_form = recruiters_soup.find("form", id="frmSrh")
search_form["action"] = "https://www.naukri.com/hr-recruiters-consultants"
search_input = search_form.find("input", id="sugInp")
search_input["value"] = "Data science"

# Submit the search form
search_response = requests.post(search_form["action"], data=search_form.form_values())

# Create a new BeautifulSoup object for the search results page
search_soup = BeautifulSoup(search_response.content, "html.parser")

# Find the container for the recruiter details
recruiter_list = search_soup.find_all("div", class_="recSec fl")

# Iterate over each recruiter and extract the details
for recruiter in recruiter_list:
    # Extract the data from each element in the recruiter container
    name = recruiter.find("span", class_="fl").text.strip()
    designation = recruiter.find("span", class_="designation").text.strip()
    company = recruiter.find("a", class_="fl").text.strip()
    skills = recruiter.find("div", class_="hireSec highlightable").text.strip()
    location = recruiter.find("small", class_="highlightable").text.strip()
```

```
# Print the details of each recruiter

print("Name:", name)

print("Designation:", designation)

print("Company:", company)

print("Skills they hire for:", skills)

print("Location:", location)

print("--------------------")
```