Assignment 3:

Web Scraping and Data Visualization with BeautifulSoup

(10 marks)

**Objective:**

The objective of this assignment is to practice web scraping using the `requests` library and `BeautifulSoup` to extract data from a webpage. The task involves scraping articles from **Hacker News** (https://news.ycombinator.com/), specifically focusing on extracting article titles, their associated points, and other metadata such as rank and URLs. After collecting the data, you will process and analyze it using ‘Pandas’ and visualize the results using ‘Matplotlib’.

**Task:**

1. **Web Scraping with BeautifulSoup:**

Hints:

user\_agents = [

    'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.0.0 Safari/537.36',

    'Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.0.0 Safari/537.36',

    'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36',

    'Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36',

    'Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36',

    'Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.1 Safari/605.1.15',

    'Mozilla/5.0 (Macintosh; Intel Mac OS X 13\_1) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.1 Safari/605.1.15'

]

# User-Agent header to simulate a real browser request

headers = {'User-Agent': random.choice(user\_agents)}

while is\_scraping and current\_page <= max\_pages:

    try:

        # Make the request to fetch the page content

        response = **requests.get(f"https://news.ycombinator.com/?p={current\_page}", headers=headers)**

        html\_content = response.content

- Scrape 10 pages of articles from Hacker News using `BeautifulSoup`. Each page contains multiple articles, including metadata such as:

- Article Rank (e.g., 1st, 2nd, etc.)

- Title of the article

- URL of the article

- Points (the number of upvotes the article has received)

- Ensure you handle any potential errors, such as network issues or missing points, by implementing retry logic.

**2. Data Collection:**

- Extract the following details from each article:

- Rank: The rank of the article (e.g., "1st", "2nd").

- Title: The title of the article.

- URL: The URL linking to the article.

- Points: The points the article has received. Extract only the numeric value (e.g., 25 points should be stored as 25).

**3. Data Storage:**

- Store the scraped data in a Pandas DataFrame. Each article should be represented as a row, and the columns should include:

- Rank

- Title

- URL

- Points

- After collecting the data, save the DataFrame to a **CSV file** named ` Assignment3\_yourname.csv`.

**4. Data Analysis and Sorting:**

- Sort the articles in the DataFrame based on the points column in **ascending order**.

**5. Data Visualization:**

- Histogram: Plot a histogram to show the distribution of points across all scraped articles.

- Bar Plot: Display a bar plot of the top 10 articles with the highest points.

- Pie Chart: Create a pie chart to show the distribution of articles in different points ranges (e.g., 0-10, 11-20, 21-30, etc.).

**Submission:**

1. Jupyter Notebook:

- A Jupyter notebook with all the code for scraping, data processing, and visualization.

- Make sure the code is well-commented to explain the logic and steps in each part.

2. CSV File:

- The ` Assignment3\_yourname.csv ` file containing the scraped and processed data, with articles sorted by points in ascending order.

**Evaluation Criteria:**

1. Correctness of Data Extraction: Were the correct fields (rank, title, URL, points) extracted from the website? Did you handle errors properly? (5 marks)
2. Visualizations: Were all required visualizations created? Were the plots clear and informative? (3 marks)
3. Code Quality: Was the code clean, well-commented, and easy to follow? (2 marks)

Good luck, and happy coding!