# Checkpoint 3: Web Scraping for Reviews

## Objective:

Develop a Python script to scrape reviews from the website, enabling the collection of data for fake review detection.

---

## What is Web Scraping?

Web scraping is the process of extracting data from websites using automated scripts. In this task, you will use web scraping tools to collect product reviews. This includes fetching details such as review text, ratings, reviewer name, and date.

---

## Basic Components of Web Scraping:

1. **Website Analysis:**
    - Inspect the website's structure using browser developer tools.
    - Identify the HTML tags and classes corresponding to review content, ratings, reviewer names, and dates.
2. **Libraries and Tools:**
    - Use Python libraries like "requests" for fetching HTML content and "BeautifulSoup" or "lxml" for parsing it.
    - Employ "selenium" if JavaScript rendering is required for dynamic content.
3. **Scrape Reviews:**
    - Extract the review text from the product page.
    - Handle cases where reviews may span multiple lines or have embedded HTML tags, ensuring clean and readable text output.
4. **Save Scraped Data:**
    - Store the extracted reviews in a CSV file (Reviews.csv).

## Tasks:

1. **Set Up Your Environment:**
    - Install required libraries: requests, BeautifulSoup4, pandas, and optionally selenium.
    - Create a Python script (Review_scraper.py) for scraping reviews.
2. **Develop the Web Scraping Script:**
    - Create a Python script named Review_scraper.py.
    - Accept a single product URL as input.
    - Extract all reviews visible on the provided page.
    - Store reviews in a structured format using Pandas DataFrame.
3. **Save Results:**
    - Save the extracted data to a CSV file named Reviews.csv.
4. **Automation & Reusability:**
    - Allow the user to input a product URL to scrape reviews for any product dynamically.
5. **Upload Deliverables to GitHub:**
    - Script file (Review_scraper.py).
    - Extracted data file (Reviews.csv).
    - A README file explaining:
        - The web scraping process.
        - Libraries used.
        - Steps to run the script.

## Deadline:

25th January 2025, 11:59 PM

## Deliverables:

1. **GitHub Repository with:**
    - Python script (**Review_scraper.py**).
    - Scraped data files (**Reviews.csv**).
    - A README file documenting the script and the scraping process.
2. **Checklist for Review:**
    - Ensure the script runs without errors.
    - Validate that the Reviews.csv file contains the required fields.
    - Confirm that all deliverables are uploaded to the GitHub repository titled **“Project_WoC_7.0_Fake_Review_Detection”** in the folder **“checkpoint 3”**.