Web Scraping for Article Extraction

1) Introduction

In todays fast paced digital world, extracting relevant information from website is crucial for data analysis. With the increasing amount of online content, automating the process of extracting key information saves a lot of time and effort.

The project aims to develop a web scaping solution specifically for news websites, focusing on extracting article links and associated content. The project uses Python libraries such as requests, BeautifulSoup and csv to efficiently scrape and process data from Kathmandu Post, filtering out social media links and storing valuable content in structured format.

2) Overview

The project implements a tool that scrapes the homepage of news website, extracts all article links and retrieves essential content from these articles including heading, author, publication date, and the main body of the text. It ensures that only valid articles are considered by excluding social media links and URLs that don't meet the expected criteria. The extracted data is saved in CSV format allowing for easy analysis, easy access and further processing.

3) Aims and Objectives

Aims :

- To automate the process of web scaping for news articles from target website.
- To filter out social media link and focus on extracting high quality content.

Objectives:

- Develop a web scraper to extract all available URLs from the Kathmandu Post homepage.
- Filter out URLs leading to social media or invalid pages.
- Extract article content such as heading, author, publication date and body text.
- Save extracted data in CSV files, allowing for structured data analysis.
- Ensure robust error handling during the scraping process to account for nrtwork issues or missing data.

4) Methodology

The project employs web scraping techniques using Python libraries to access, parse and extract data from news website. The following steps outline the methodology:

i. Website Access: The requests library is used to send HTTP GET requests to the target website's homepage, retrieving the HTML content for further processing.
ii. HTML Parsing: The BeautifulSoup library parses the HTML content and extracts all the anchor tags (<a>) containing URLs

iii. Link Processing: The relative URLs are converted to absolute URLs using urljoin function, and social media links are filtered out using is_social_media_url() function.

iv. Content Extraction: For each valid URL, a separate HTTP request is made, and the article's heading, author, publication date, and body content are extracted using specific HTML tag references.

v. Data Storage: The extracted data is saved in two CSV files:
   - One CSV files stores the extracted links.
   - Another CSV files store the details of articles, including metadata such as heading, author, date and content.

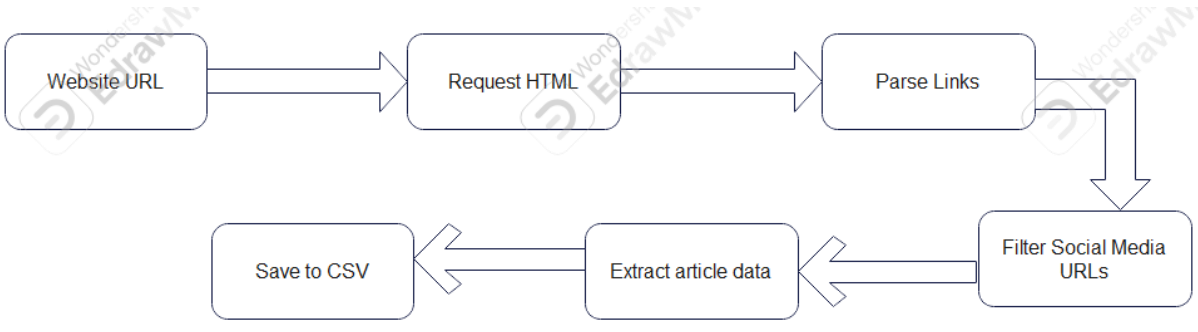
vi. Error Handling: The script includes error handling for HTTP connection issues, invalid URLs and cases where the required data is missing from webpage.

5) Workflow

The web scraping workflow follows these steps:

I. Extract All Links: The extract_all_links() function retrieves all the links available on the homepage.
II. Filter Social Media Links: The is_social_media_url() function filters out any social media URLs such as Twitter, Facebook, or Instagram.
III. Extract Article Content: The extract_specific_content() function is used to extract relevant information (heading, author, publication date, body) from each valid URL.
IV. Save Data to CSV: The links and article content are saved in separate CSV files using the csv module.

6)Block Diagram



6) Challenges Encountered

Some of the issues encountered during the development of web scraper are:

- ❖ Network Timeouts : Sometimes, the request to website would timeout due to network issues which was overcomed by error handling mechanisms.

- ❖ Missing Data : Some pages contained incomplete information like missing author, where the script checked for missing data and skipped saving incomplete records.

- ❖ Invalid URLs : Some links on the page were not properly formatted, so the scripts converted relative URLs to absolute URLs to ensure they are correctly handled.

7) Results

The web scraped was able to ectract and store a substantial number of links from the target website, filtering out irrelevant links and saving the article details in structured format. This allows for analysis of web content such as sentiment analysis ir trend analysis over time.

Output:

	Heading	Author	Publication_date	Source	Content	Link	Category
2	Heading	Author	Publication_date	Source	Content	Link	Category
3	Heading not Found	Author not Found	Date not found	Kathmandu-Post	Content not found	https://kathmandupost.c	national
4	Heading	Author	Publication_date	Source	Content	Link	Category
5	Heading not Found	Author not Found	Date not found	Kathmandu-Post	Content not found	https://kathmandupost.c	politics
6	Heading	Author	Publication_date	Source	Content	Link	Category
7	Heading not Found	Author not Found	Date not found	Kathmandu-Post	Content not found	https://kathmandupost.c	valley
8	Heading	Author	Publication_date	Source	Content	Link	Category
9	Heading not Found	Author not Found	Date not found	Kathmandu-Post	Content not found	https://kathmandupost.c	opinion
10	Heading	Author	Publication_date	Source	Content	Link	Category
11	Heading not Found	Author not Found	Date not found	Kathmandu-Post	Content not found	https://kathmandupost.c	money
12	Heading	Author	Publication_date	Source	Content	Link	Category
13	Heading not Found	Author not Found	Date not found	Kathmandu-Post	Content not found	https://kathmandupost.c	sports
14	Heading	Author	Publication_date	Source	Content	Link	Category
15	Heading not Found	Author not Found	Date not found	Kathmandu-Post	Content not found	https://kathmandupost.c	art-culture
16	Heading	Author	Publication_date	Source	Content	Link	Category
17	Heading not Found	Author not Found	Date not found	Kathmandu-Post	Content not found	https://kathmandupost.c	weekender
18	Heading	Author	Publication_date	Source	Content	Link	Category
19	Heading not Found	Author not Found	Date not found	Kathmandu-Post	Content not found	https://kathmandupost.c	Category not found
20	Heading	Author	Publication_date	Source	Content	Link	Category
21	Heading not Found	Author not Found	Date not found	Kathmandu-Post	Content not found	https://kathmandupost.c	national
22	Heading	Author	Publication_date	Source	Content	Link	Category
23	Heading not Found	Author not Found	Date not found	Kathmandu-Post	Content not found	https://kathmandupost.c	national
24	Heading	Author	Publication_date	Source	Content	Link	Category
25	Heading not Found	Author not Found	Date not found	Kathmandu-Post	Content not found	https://kathmandupost.c	national
26	Heading	Author	Publication_date	Source	Content	Link	Category
27	Heading not Found	Author not Found	Date not found	Kathmandu-Post	Content not found	https://kathmandupost.c	national
28	Heading	Author	Publication_date	Source	Content	Link	Category
29	Heading not Found	Author not Found	Date not found	Kathmandu-Post	Content not found	https://kathmandupost.c	national
30	Heading	Author	Publication_date	Source	Content	Link	Category
31	Heading not Found	Author not Found	Date not found	Kathmandu-Post	Content not found	https://kathmandupost.c	national
32	Heading	Author	Publication_date	Source	Content	Link	Category

8) Conclusion

The project demonstrates the power of web scraping to automate the extraction of relevant data from a news website. By filtering out irrelevant links and focusing on structured content, the tool can be used for media monitoring, research and data analysis purpose.