# Automatic News Scraping and Visualization with Python

Author: Yashdhar Gandhi

## Project Overview

This project aims to scrape and extract news articles from multiple sources using RSS feeds, parse relevant information, and visualize the extracted data using a word cloud and an interactive dashboard.

## Objectives

- Automatically fetch news articles from various sources.
- Extract relevant information such as the **title, author, publish date, and article content**.
- Store the collected data in a structured format (**CSV file**).
- Generate a **word cloud** to highlight frequent terms.
- Build an **interactive dashboard** to visualize trends in the news.

---

## Technologies and Libraries Used

### Web Scraping:

- `newspaper3k`: Extracts full-text news articles.
- `feedparser`: Parses RSS feeds to fetch news links.

### Data Handling & Storage:

- `pandas`: Structures data into a DataFrame and saves it as a CSV file.
- `os`: Handles file and directory operations.

### Visualization:

- `matplotlib`: Generates plots and figures.
- `wordcloud`: Creates a graphical representation of word frequency in the news articles.

### Dashboard & UI:

- `dash`: Creates an interactive web-based dashboard.
- `dash-core-components`: Adds interactive components to the dashboard.
- `dash-html-components`: Structures HTML elements within the dashboard.

---

# Project Workflow

### Step 1: Fetching News Articles

- The script pulls news data from multiple RSS feeds, including **BBC, The New York Times, and The Guardian**.
- Uses `feedparser` to parse the RSS feeds and extract article links.
- `newspaper3k` processes each article, downloading and extracting its **title, author(s), publish date, and full content**.

### Step 2: Storing the Data

- The extracted data is structured into a **pandas DataFrame**.
- It is then **saved as a CSV file** to the user's **Downloads folder** for further analysis.

### Step 3: Generating a Word Cloud

- The collected content is processed to create a **word cloud**.
- The generated **image file is saved** in the Downloads folder.

### Step 4: Interactive Dashboard

- The dashboard includes:
  - **Word Cloud Display**: Showcases frequent terms in the news articles.
  - **News Trends Graph**: Visualizes the number of articles published over time using a bar chart.
- The dashboard is built using `Dash` and can be accessed via a web browser.

---

# Project Files and Directory Structure

```
📂 NewsScraperProject
├── 📁 data
│   ├── articles.csv  # Scraped news data
├── 📁 notebooks
│   ├── news_scraper.ipynb  # Jupyter Notebook
├── 📁 scripts
│   ├── scraper.py  # Python script for scraping and visualization
├── 📁 visuals
│   ├── wordcloud.png  # Generated word cloud image
├── README.md  # Project documentation
```

---

# Future Enhancements

- **Sentiment Analysis**: Analyze the tone of news articles (positive, neutral, negative).

- **More Data Sources**: Extend the scraper to fetch news from additional websites.
- **Scheduled Automation**: Automate daily or hourly news scraping.
- **Advanced Analytics**: Add interactive filters to analyze trends by keywords, author, or source.

---

# Conclusion

This project demonstrates how **web scraping, data processing, and visualization** can be combined to build an automated news analytics tool. The integration of **Dash** allows for real-time interactive exploration of collected news data.