Assignment 1

DA5402

EE23S010

Module1:

This module dynamically scrapes the Google News homepage without hardcoding URLs. The implementation ensures that parameters such as the base URL are configurable via a configuration file

Configuration Management Section:

1. A JSON configuration file (config.json) is used to store dynamic parameters such as the base URL and the section of interest (e.g., "Top stories").
2. The script reads and writes to this configuration file for adaptability.

Web Scraping Section:

3. The script employs the requests library to fetch the webpage content.
4. The BeautifulSoup library is used to parse the HTML structure of the page and extract relevant information.

Module 2:

This module dynamically identifies and extracts the “Top Stories” section's URL, avoiding hardcoded text or links.

1. The script dynamically searches for headings (h2, h3, h4).
2. It retrieves the hyperlink associated with the "Top Stories" section without hardcoding its label.

Module 3:

This module extracts each article's headline, publication date, and thumbnail image in the Top Stories section. The script accounts for lazy loading, ensuring images are properly extracted.

1. The script sends an HTTP request to the extracted Top Stories URL.

2. It identifies relevant <article> elements and extracts their associated images and links.

3. It correctly formats publication times for consistency.

Module 4:

This module stores extracted headlines, metadata, and images in a PostgreSQL database. It scrapes Google News, downloads images, and inserts structured data into relational tables

1. A JSON configuration file (db_config.json) stores the database server information.

2. It scrapes Google News, downloads images, and inserts structured data into relational tables

   Key function: save_to_database()

3. Connects to PostgreSQL and creates tables if absent.
4. Downloads and stores images locally.
5. Insert image and article metadata.
6. Commits transactions with error handling.

   Key function: save_to_existing_database()

7. Check the latest image index for unique naming.
8. Downloads and stores new images.
9. Insert metadata (scrape timestamp, article URL, publish date, image URL, and images) into respective tables.

Module 5:

This module is responsible for checking data duplication in the data base.

Key Function: is_duplicate()

1. Prevents duplicate entries based on fuzzy headline matching.

2. Uses fuzzywuzzy to compare new headlines with existing ones.

3. Flags a headline as duplicate if similarity exceeds 85%.

4. Before inserting new data, is_duplicate() checks for existing headlines.

5. If a duplicate is found, the entry is skipped; otherwise, it is stored in the database.

Module 6:

This module ensures the smooth orchestration of all the modules with periodic execution.

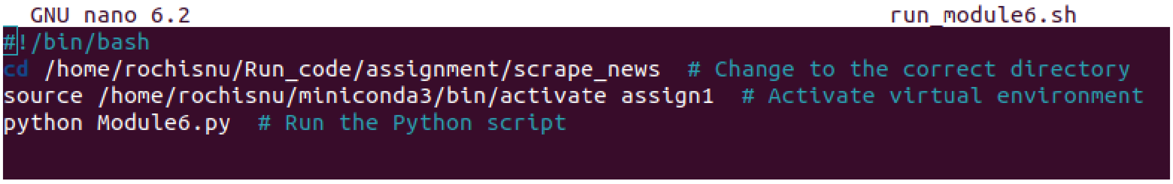
1. Loads configurations and initializes logging.

2. Scrape the Google News homepage and extract the 'Top Stories' link.

3. Scrapes headlines and images while ensuring a dedicated folder exists for storing images.
4. Connects to the PostgreSQL database and creates tables if not present.
5. Implements de-duplication before inserting new articles.
6. Logs execution details, errors, and skips duplicates to aid debugging.

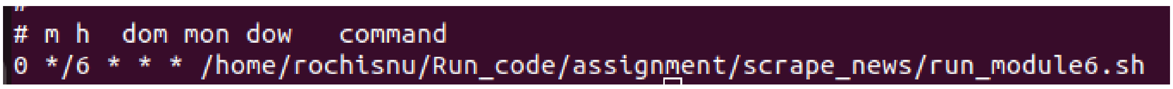
Automation via CronJob

7. The script is scheduled to run automatically every 6 hours.
8. Uses a Bash script run_module6.sh to activate the virtual environment (assign1) and execute the Python script.



```
  GNU nano 6.2                                                       run_module6.sh
#!/bin/bash
cd /home/rochisnu/Run_code/assignment/scrape_news  # Change to the correct directory
source /home/rochisnu/miniconda3/bin/activate assign1  # Activate virtual environment
python Module6.py  # Run the Python script
```

9. Crontab entry: 0 */6 * * * /path/to/run_module6.sh (every 6 hours)



```
# m h  dom mon dow   command
0 */6 * * * /home/rochisnu/Run_code/assignment/scrape_news/run_module6.sh
```