Newspaper data crawler for Asgaard Lab Assignment

Crawler Development:	Python3
Database :	MySQL
Data Visualization:	PHP

asgaard_crawler.py

Python3 has been used to develop crawler specially article from newspaper. Two different newspapers are selected for crawling named as CNN (https://edition.cnn.com) and Daily Star BD (https://www.thedailystar.net/). However, this code has been designed as a generic one, where any other newspaper link can be used to crawl article. A library has been used to build with article of the targeted newspaper named as Newspaper3k which can be installed via pip in Linux.

Two python3 libraries are used to develop this code newspaper3k[1] and pymysql[2], which are.

```
import pymysql      # for MySQL database handle
import newspaper    # for newspaper content linkup
```

Program starts from the main function where, the prerequisite variable are declared, which must be modified based on your environment credentials. The prerequisites are given below.

```
topic = "Covid-19"

host = "localhost"
user = "root"
password = "1987"
database_name = "crawler"
table_name = "news"
```

Then the program connect with MySQL database with those prerequisites.

```
db = database_connection_start(host,user,password,database_name)

#function of mysql database connection
def database_connection_start(host, user, password,
database_name):
    db = pymysql.connect(host, user, password, database_name)
    cursor = db.cursor()
    cursor.execute("SELECT VERSION()")
    data = cursor.fetchone()
    print ("Database version : %s " % data)

    return db
```

Then database table needs to be truncated for clearing the existing articles.

```
database_clear(db, table_name)

# function of database content clear for fresh data
def database_clear(db, table_name):
    cursor = db.cursor()
    sql1 = "TRUNCATE TABLE %s"%table_name
    try:
        cursor.execute(sql1)
        db.commit()
    except:
        db.rollback()
```

1 https://github.com/codelucas/newspaper
2 https://github.com/PyMySQL/PyMySQL

Then the targeted newspaper named and links are stored inside a list.

```
newsList = [["cnn", "https://edition.cnn.com"], ["dailystar",
"https://www.thedailystar.net/"]]
```

Newspaper contents are loaded using a function from newspaper3k library named newspaper_article_build.

```
for newsInfo in newsList:
          newspaper_content_build =
newspaper_article_build(newsInfo[1])

#function of newspaper content loading based on the url
def newspaper_article_build(newspaper_url):
     newspaper_content_build = newspaper.build(newspaper_url,
memoize_articles=False)
     return newspaper_content_build
```

Then those articles are passed for processed.

```
individual_article_process_for_db_store(db, table_name,
newsInfo[0], newspaper_content_build, topic, 25)
```

Individual article is downloaded, parsed and checked whether it is related to the targeted topic. Then passed article's title, authors, and description with the article matched number through a database insertion function. All the processes are grouped inside a try-cacth exception handler to bypass unwanted issues. Articles are also prepossessed to fulfill the query execution.

```
try:
     article.download()
     article.parse()
     if article_topic_to_show in article.text:
          news_counter += 1
          id = news_counter

          title = article_preprocessing_for_database(article.title)
          description =
article_preprocessing_for_database(article.text)
          url = article.url
          authors = article_preprocessing_for_database(',
'.join([str(elem) for elem in article.authors]))

          article_insert_in_database(db, table_name,
id,title,description, url, authors)
except Exception as e:
     print (e)
     continue
```

Processed Articles are now ready to store in database. The insertion query is secured by try-catch exception handling for execution rollback.

```
#function of insert newspaper article to database table
def article_insert_in_database(db, table_name,
id,title,description, url, authors):

     cursor = db.cursor()
     sql = "INSERT INTO %s(id, title, description, authors) VALUES
('%s','%s','%s', '%s')"%(table_name,id, title,description,
authors)
```

```
    try:
        cursor.execute(sql)
        db.commit()
        print("OK; "+ table_name+ "; "  +title)
    except:
        print ("Problem; "+ table_name+ "; "  +title)
        db.rollback()
```

index.php

Raw PHP is used to write the content in web version. It creates the database connection at the very first of its code.

```
<?php
$servername = "localhost";
$username = "root";
$password = "1987";
$dbname = "crawler";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

Two newspapers are analyzed for article collection which are listed as a drop down in PHP.

```
select name="type" class="form-control mb-2">
     <option value="">Select Type</option>
     <option value="cnn">CNN</option>
     <option value="dailystar">Daily Star</option>
</select>
```

Then the visualization results are filtered based on this selection.

```
     $type = isset( $_POST['type'] ) ? $_POST['type'] : '';
     $sql = "SELECT * FROM news WHERE type= '". $type ."' ";
     $result = $conn→query($sql);
```

Finally results are displayed in a table with collapse div. When anyone click on the title, the entire description will be loaded.

```
<button class="btn" type="button" data-toggle="collapse" data-
target="#new_description<?php echo $row['id']; ?>" aria-
expanded="false" aria-controls="new_description">
<?php echo $row['title']; ?>
</button>
<?php     echo (is_null($row['authors'])) ? "" : $row['authors'];?>
</p>
<div class="collapse"id="new_description<?php echo $row['id'];?>">
<div class="card card-body">
<?php echo $row['description']; ?>
```