Peter Mansbacher

Andres Dicochea

James Mortensen

Eric Slick

Getting Started

An overview of the Social Media Comparison Tool

Steps to Deploy Anew - Peter

-Order of set up

1. **Created scrapers in IDEs**
    - **1.1. Scrapers will need to access the database**
2. **Set up AWS accounts/group**
3. **Set up AWS Lightsail server (Should use windows instead of linux VM, might solve FB/Insta scraping errors)**
    - **3.1. Navigate to the Social-Media-Comparison-Tool/Access to Prototype/django&database_documentation.txt to see specifics on server and DB**
4. **As mentioned in django&database_documentation.txt, to create a new Django server, follow the guide in the link: https://docs.bitnami.com/aws/infrastructure/django/get-started/get-started/**
5. **Once the Django server has been set up, add views to hold the scrapers. Follow this link to learn how to create views: https://docs.djangoproject.com/en/4.0/topics/http/views/**
6. **The Django application will need forms, documentation on how to work with forms in Django can be found here: https://docs.djangoproject.com/en/4.0/topics/forms/**
    - **6.1. Forms will be used to get and post data**
7. **In order to access the database easily and test queries, we used MySQL Workbench**
    - **7.1. Here is a link on how to connect and access a database with the software: https://www.inmotionhosting.com/support/website/connect-database-remotely-mysql-workbench/**
8. **If any trouble is reached, check the project readme.txt for contacts and reach out to any of the numbers provided (not the client) for specifications**

Initialization –

-Data Andres

1. **To access data that is gathered from our program you will need to download MySQL Workbench at https://www.mysql.com/products/workbench/**
2. **Once downloaded, the credentials to access the database are:**
    - a. **EndPoint: ls-1ef1825172e62dcc237ee491d09a0c12aff562fe.cn5ycdfnko6g.us-east-1.rds.amazonaws.com**
    - b. **Port: 3306**
    - c. **Username: dbmasteruser**
    - d. **Password: q+o.H1sd$CRRZl&CSl>VK}-(~+t1ea&P**

3. **Once logged in, you will have access to the database with all the data that the scrapers were able to gather.**
4. **You are able to run any custom queries within this workbench to analyze/gather specific data that was scrapped from the searches that were ran on the website.**

-Constants/Settings James

1. Currently, the scrapers are hardcoded to only allow for 8,999 simultaneous requested searches. This can be changed, each scraper .py file includes several options for the setup of ChromeDriver like: **chromeOptions.add_argument('--remote-debugging-port=%s' % randrange(1000,9999))**. The range can be increased, just make sure your server has the capabilities to do so.
2. The information being displayed in the Results View can be easily modified in the views.py file. Add whatever information you would like displayed to the appropriate dictionary (see line 1051-1057 in views.py for the dictionary passed for brand 1 after a YouTube search).
3.

-Modifications Eric

1. The scrapers for Facebook, Instagram and TikTok should be routed throw a private proxy server for best/any results.
2. The passing of information into the database currently ignores duplicate entries. Duplicate entries are designated by either URLs or data specific IDs depending upon the platform. Recommend modifying the code for each scraper so that instead of ignoring duplicate entries, the information in the database is updated with the latest data.
3. Recommend adding an in between loading page after submitting a request to the scrapers. A loading page would be more intuitive for the user and