

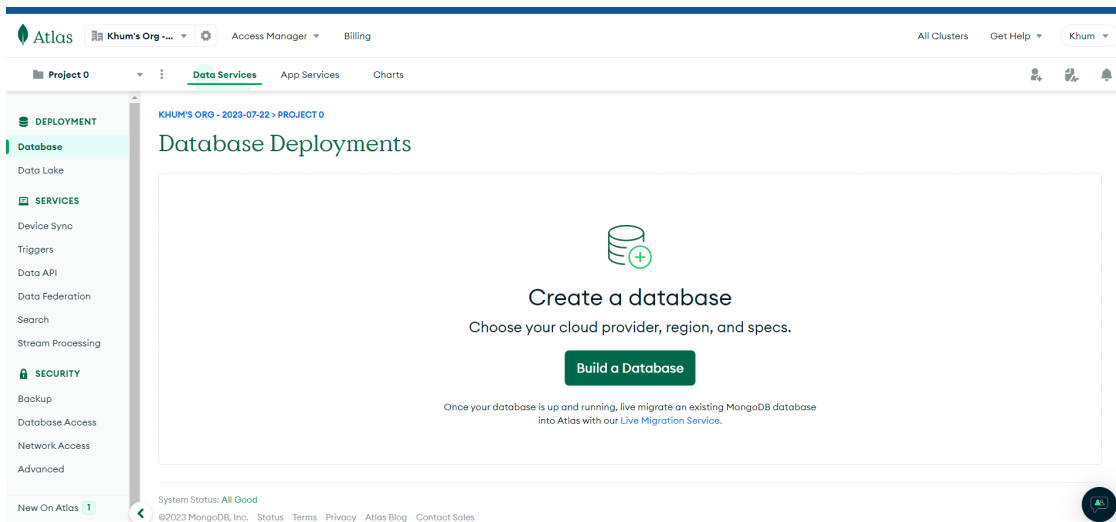
ReadMe Notes

Intelligent News Content Discover System

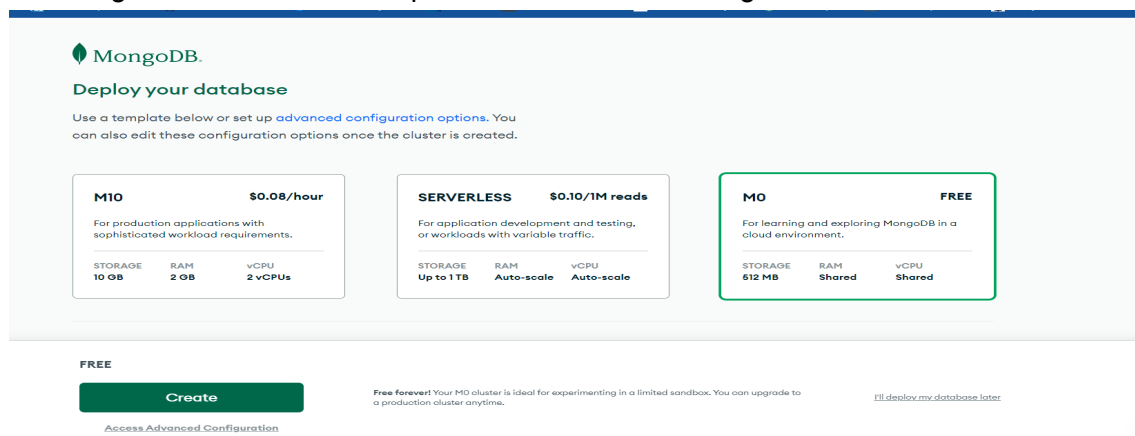
Initial steps for setup

MongoDB (Database Setup)

- a. Create an account in MongoDB Atlas (<https://account.mongodb.com/account/login>)



- b. Create a project and deploy your MongoDB database by selecting your preferred configuration. For this example, a free database configuration is selected.



- c. Create a connection IP address and database user

Connect to Cluster0



You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. [Read more](#)

1. Add a connection IP address

Add Your Current IP Address

Add a Different IP Address

Allow Access from Anywhere

2. Create a database user

This first user will have [atlasAdmin](#) permissions for this project.
Keep your credentials handy, you'll need them for the next step.

Username

Leon

Password

.....

SHOW

Autogenerate Secure Password

Copy

Create Database User

d. Copy the MongoDB connection string as the credential to connect to Python Flask.

Connect to Cluster0



Connecting with MongoDB Driver

1. Select your driver and version

We recommend installing and using the latest driver version.

Driver

Python

Version

3.12 or later

2. Install your driver

Run the following on the command line

```
python -m pip install pymongo
```

[View MongoDB Python Driver installation instructions.](#)

3. Add your connection string into your application code

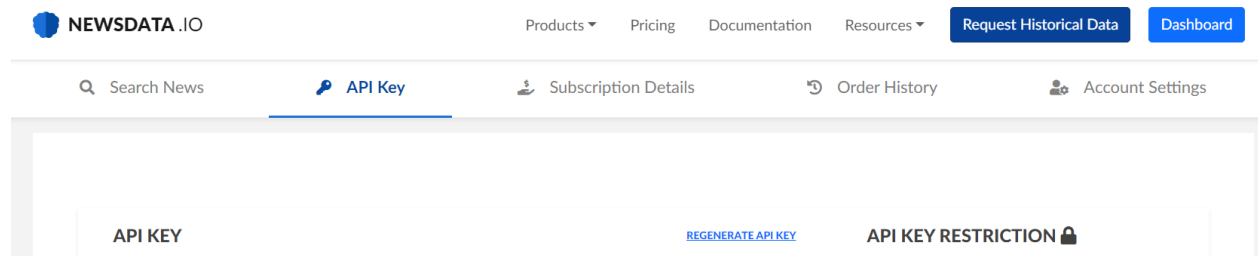
☐ View full code sample

```
mongodb+srv://Leon:<password>@cluster0.dhrujoj.mongodb.net/?  
retryWrites=true&w=majority
```

Replace **<password>** with the password for the **Leon** user. Ensure any option params are [URL encoded](#).

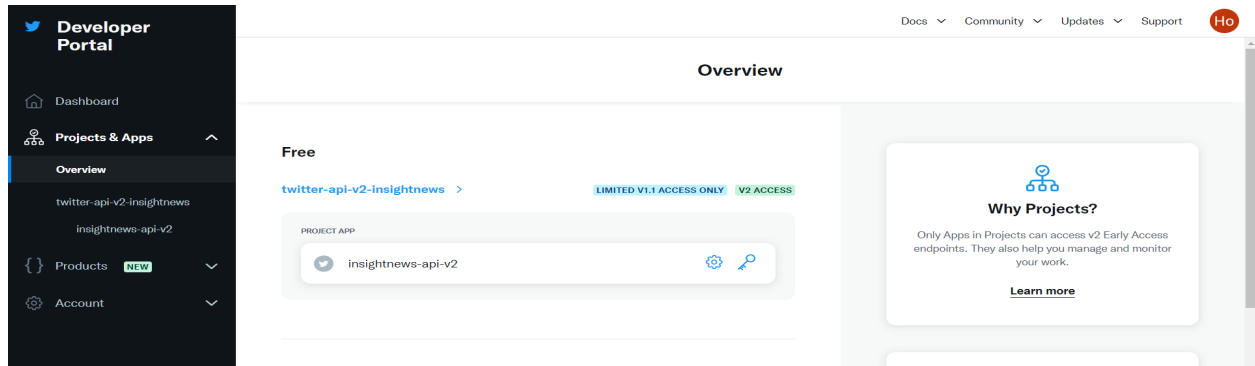
NewsData.io API (International News Articles Extraction)

- Create an account in newsdata.io (<https://newsdata.io/register>)
- Generate API key and copy the API key to Python scripts

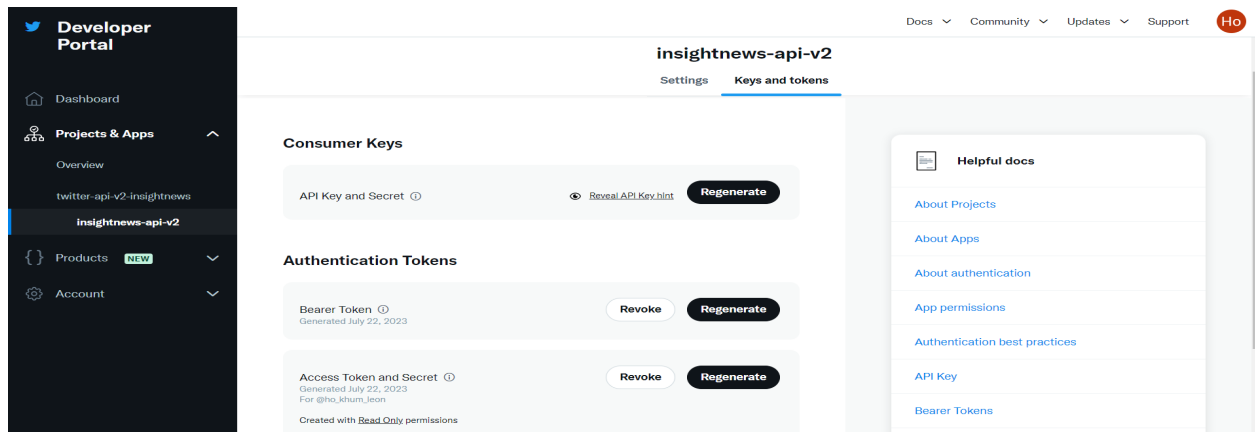


Twitter API (Tweets Extraction)

- Register for a developer account in Twitter
(<https://developer.twitter.com/en/docs/authentication/guides/log-in-with-twitter>)
- Create a project app in Twitter Developer portal



- Create an app in the project to generate consumers key and authentication tokens for connection purposes. Copy those and attach to Python scripts



Steps to setup the environment in VSCode

Python version: 3.10

Back-end: Python Flask

Front-end: HTML, CSS, Javascript

List of Files:

Python:

1. app.py (**main logic of the application**)

HTML, CSS & Javascript

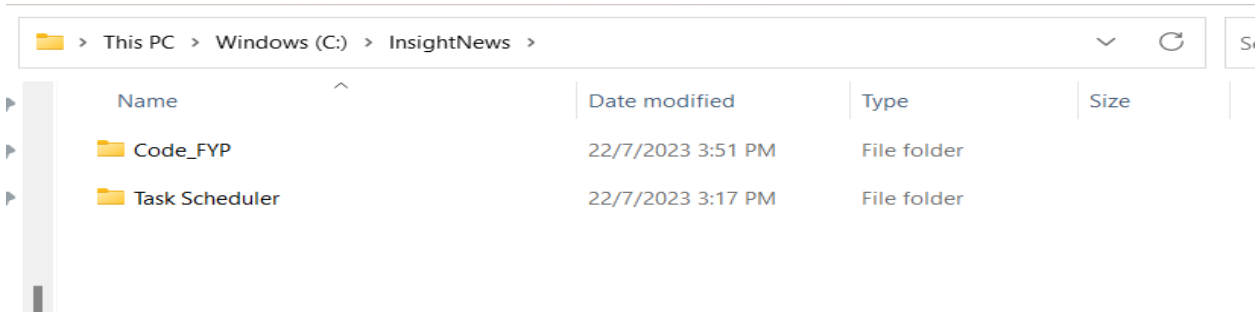
1. main_page.html (**home page/ landing page**)
2. forgot-password.html (**forgot password page**)
3. keyword-analysis.html (**keyword analysis page**)
4. Library.html (**library page**)
5. login-register.html (**authentication page**)
6. Multiview-comparison.html (**topic sentiment comparison page**)
7. Multiview-sentiment.html (**topic sentiment page**)
8. News-details.html (**news information page**)
9. pyLDAvis_3_2023.html (**topic modelling bubble chart visualization for March**)
10. pyLDAvis_4_2023.html (**topic modelling bubble chart visualization for April**)
11. pyLDAvis_5_2023.html (**topic modelling bubble chart visualization for May**)
12. pyLDAvis_6_2023.html (**topic modelling bubble chart visualization for June**)
13. pyLDAvis_7_2023.html (**topic modelling bubble chart visualization for July**)
14. search_result.html (**search results of news content page**)
15. topic_visualize.html (**bubble chart visualization for topic modelling results page**)
16. topic-analysis.html (**trending tweets for every topics page**)
17. user-profile.html (**user profile page**)
18. style_login_register.css (**style for authentication page**)
19. style_main.css (**style for every page except authentication page**)
20. app.js (**javascript for authentication page**)
21. script_main.js (**javascript for every page except authentication page**)
22. D3-legend.min.js (**javascript to display legend for bubble chart using d3.js**)
23. D3.min.js (**javascript to create bubble chart using d3.js**)

LDA model

1. lda_model_file_25_5_2023 (**optimal LDA model created**)

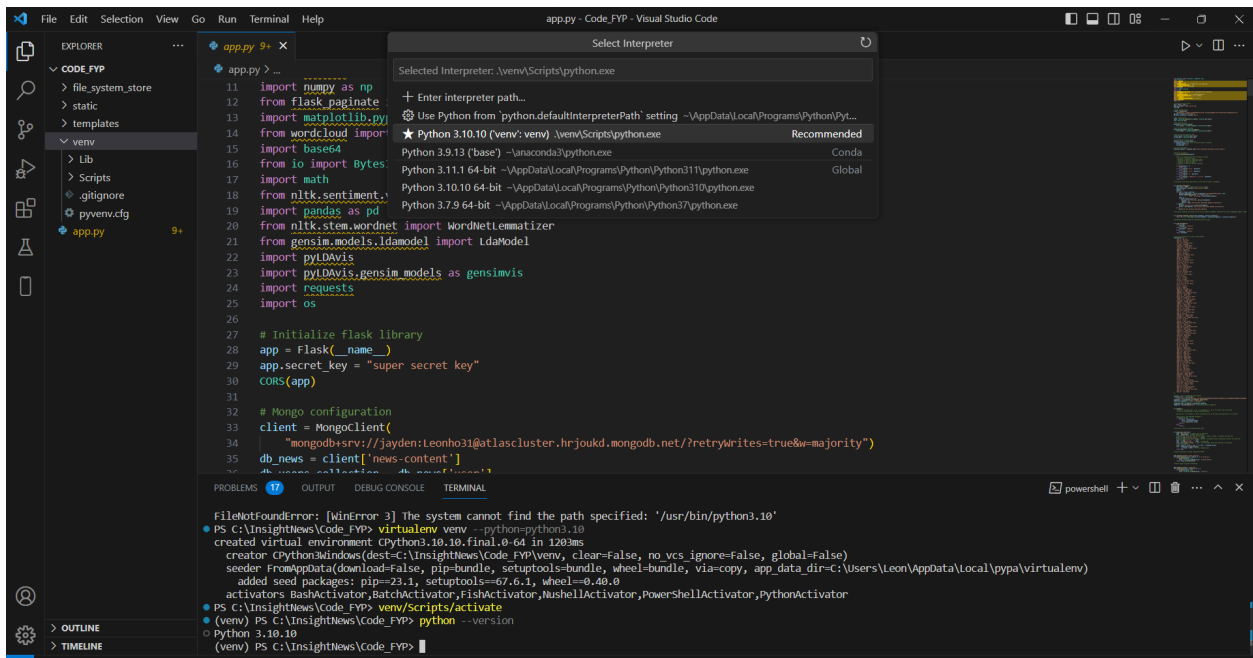
VSCode

- Create a folder in your local directory and insert all relevant code in the created folder



- Navigate to the desired folder and create a virtual environment with Python 3.10. Before creating the virtual environment, ensure that Python 3.10 has been installed in your local directory. If not, you may install it through <https://www.python.org/downloads/release/python-3100/>. You may create a virtual environment by entering the following command in the terminal. For this example, the name of the virtual environment is venv.

virtualenv venv --python=python3.10

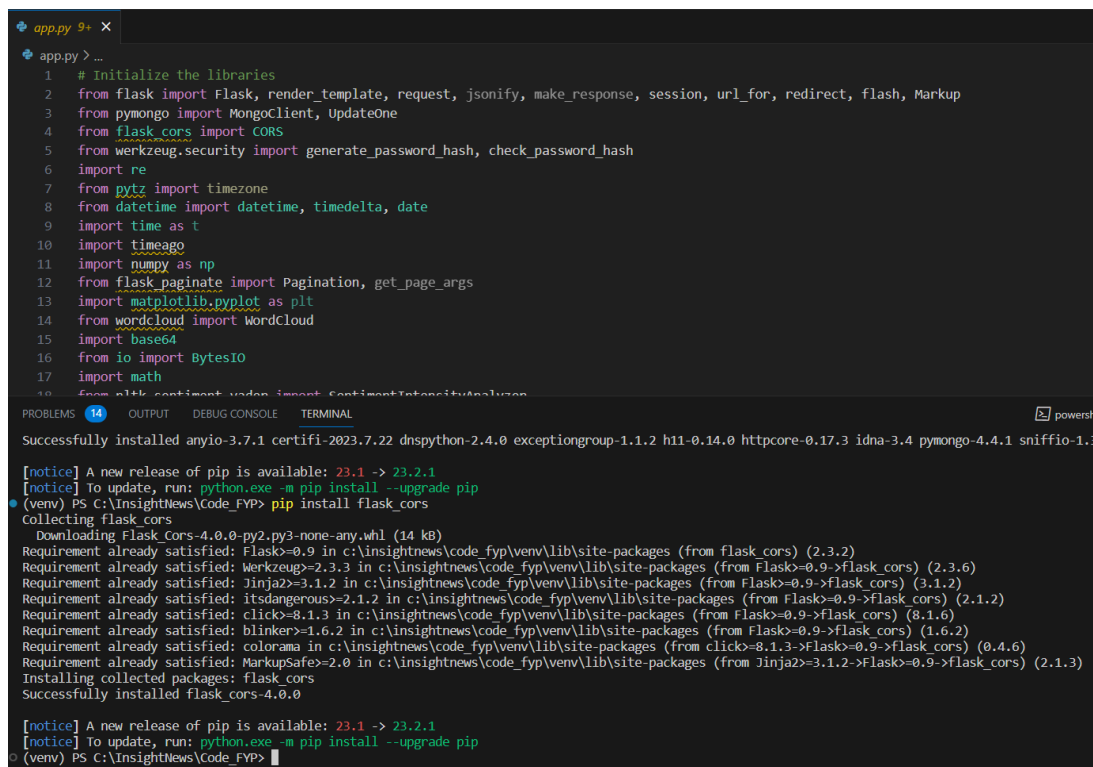


- Activate the virtual environment and install all related libraries and dependencies

Activate virtual environment command: ***venv/Scripts/activate***

Required Libraries:

1. Flask (**pip install flask**)
2. Pymongo (**pip install pymongo**)
3. Flask_cors (**pip install flask_cors**)
4. Timeago (**pip install timeago**)
5. Numpy (**pip install numpy**)
6. Flask_paginate (**pip install flask_paginate**)
7. Wordcloud (**pip install wordcloud**)
8. Nltk (**pip install nltk**)
9. Pandas (**pip install pandas**)
10. Gensim (**pip install gensim**)
11. pyLDAvis (**pip install pyLDAvis**)
12. Requests (**pip install requests**)



```
app.py 9+ x
app.py > ...
1 # Initialize the libraries
2 from flask import Flask, render_template, request, jsonify, make_response, session, url_for, redirect, flash, Markup
3 from pymongo import MongoClient, UpdateOne
4 from flask_cors import CORS
5 from werkzeug.security import generate_password_hash, check_password_hash
6 import re
7 from pytz import timezone
8 from datetime import datetime, timedelta, date
9 import time as t
10 import timeago
11 import numpy as np
12 from flask_paginate import Pagination, get_page_args
13 import matplotlib.pyplot as plt
14 from wordcloud import WordCloud
15 import base64
16 from io import BytesIO
17 import math
18 from nltk.sentiment.vader import SentimentIntensityAnalyzer

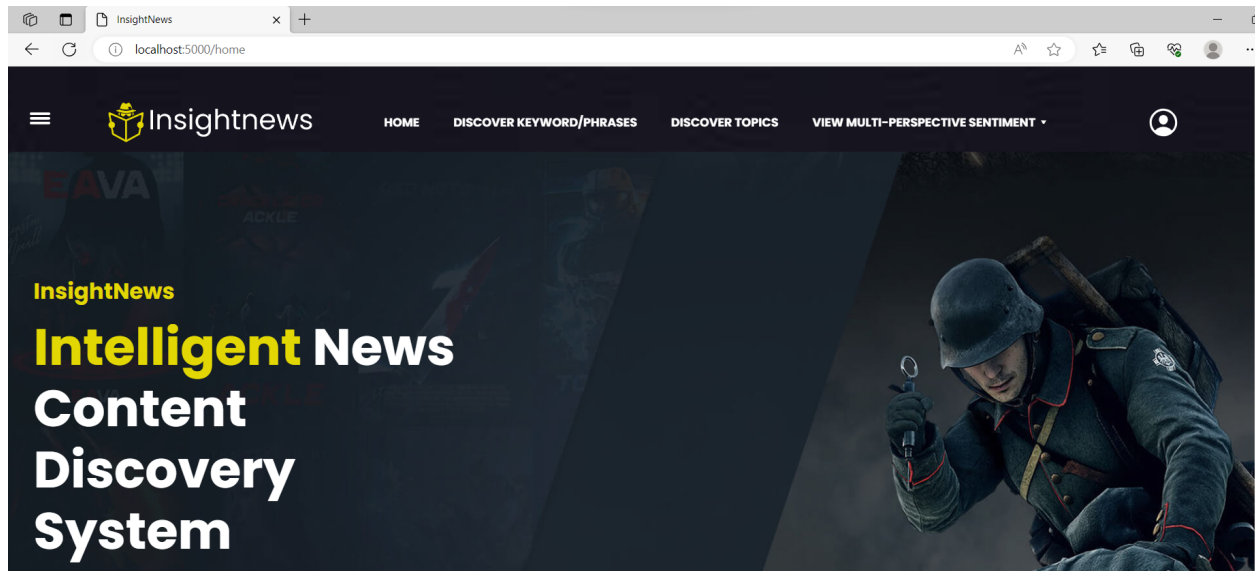
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL
Successfully installed anyio-3.7.1 certifi-2023.7.22 dnspython-2.4.0 exceptiongroup-1.1.2 h11-0.14.0 httpcore-0.17.3 idna-3.4 pymongo-4.4.1 sniffio-1.3.0

[notice] A new release of pip is available: 23.1 -> 23.2.1
[notice] To update, run: python.exe -m pip install --upgrade pip
(venv) PS C:\InsightNews\code_fyp> pip install flask_cors
Collecting flask_cors
  Downloading Flask_Cors-4.0.0-py2.py3-none-any.whl (14 kB)
Requirement already satisfied: Flask<0.9 in c:\insightnews\code_fyp\venv\lib\site-packages (from flask_cors) (2.3.2)
Requirement already satisfied: Werkzeug<2.3.3 in c:\insightnews\code_fyp\venv\lib\site-packages (from Flask<0.9->flask_cors) (2.3.6)
Requirement already satisfied: Jinja2<3.1.2 in c:\insightnews\code_fyp\venv\lib\site-packages (from Flask<0.9->flask_cors) (3.1.2)
Requirement already satisfied: itsdangerous<2.1.2 in c:\insightnews\code_fyp\venv\lib\site-packages (from Flask<0.9->flask_cors) (2.1.2)
Requirement already satisfied: click<8.1.3 in c:\insightnews\code_fyp\venv\lib\site-packages (from Flask<0.9->flask_cors) (8.1.6)
Requirement already satisfied: blinker<1.6.2 in c:\insightnews\code_fyp\venv\lib\site-packages (from Flask<0.9->flask_cors) (1.6.2)
Requirement already satisfied: colorama in c:\insightnews\code_fyp\venv\lib\site-packages (from click<8.1.3->Flask<0.9->flask_cors) (0.4.6)
Requirement already satisfied: MarkupSafe<2.0 in c:\insightnews\code_fyp\venv\lib\site-packages (from Jinja2<3.1.2->Flask<0.9->flask_cors) (2.1.3)
Installing collected packages: flask_cors
Successfully installed flask_cors-4.0.0

[notice] A new release of pip is available: 23.1 -> 23.2.1
[notice] To update, run: python.exe -m pip install --upgrade pip
(venv) PS C:\InsightNews\code_fyp>
```

- d. All the environment has been set up. Run the flask web application using the following command. The Flask development server is started, and the application is now running and accessible at the specified host and port (usually localhost:5000) by default.

Run Flask web application command: ***python app.py***



Existing Login Accounts

These are the existing login accounts that can readily login into the web application.

Email	Password
1. von@gmail.com	Leonho@13
2. abu@gmail.com	Leonho#13
3. hello@gmail.com	Hello@123

Steps to configure Task Scheduler

List of Files:

1. data_extraction_intelligent_components (**data extraction and intelligent components processes**)
2. news_duration_update (**update the duration published of news contents frequently**)
3. topic_modelling_visualization (**update the topic modelling every first day of the month**)

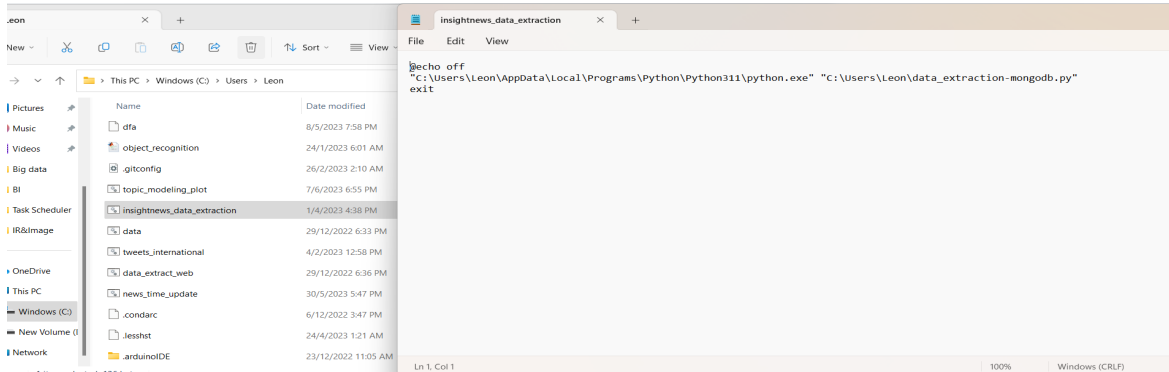
Task Scheduler

- a. Open the notepad and insert the following bat scripts and save the file as .bat extension. The script involves the location of your Python interpreter and Python scripts.

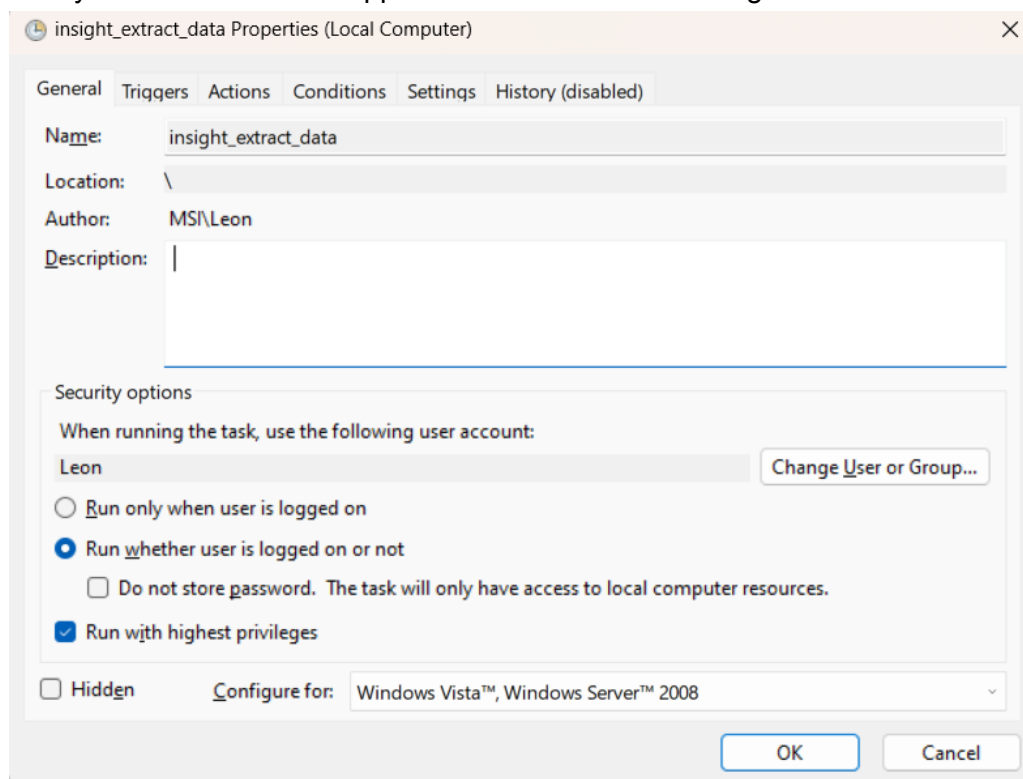
@echo off

"C:\Users\Leon\AppData\Local\Programs\Python\Python311\python.exe" <Python interpreter location>

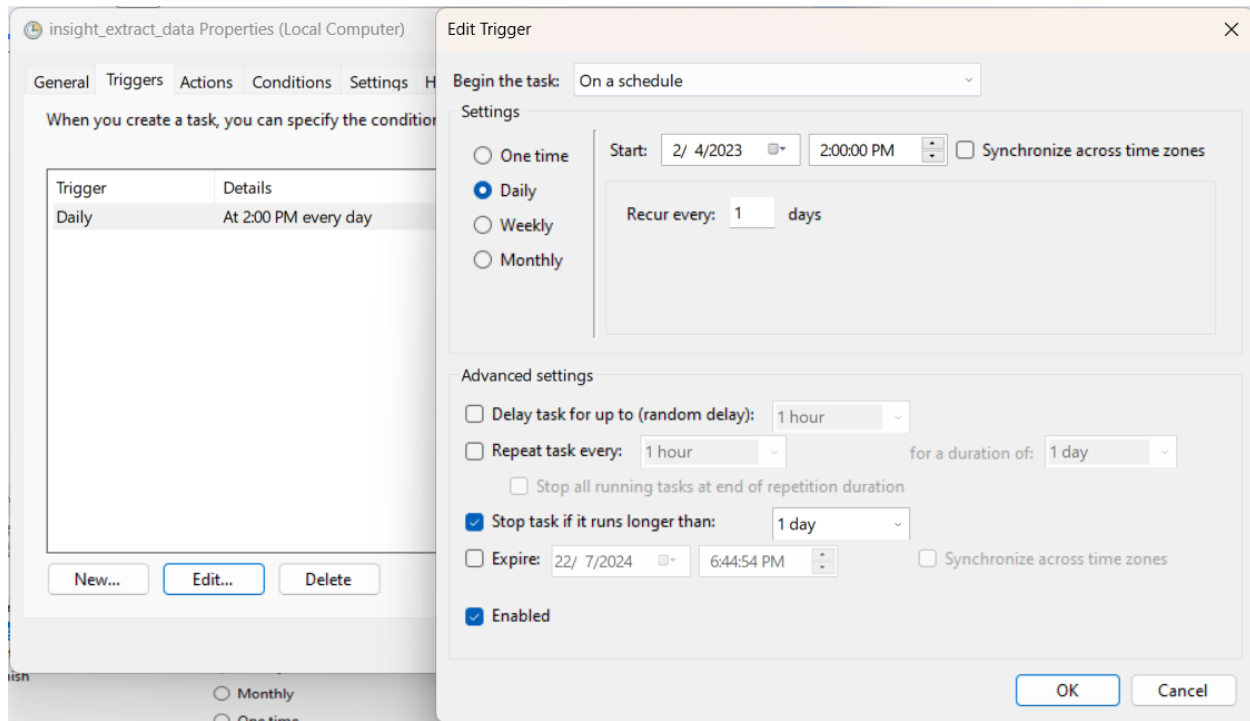
"C:\Users\Leon\data_extraction-mongodb.py" <Python scripts location>
exit



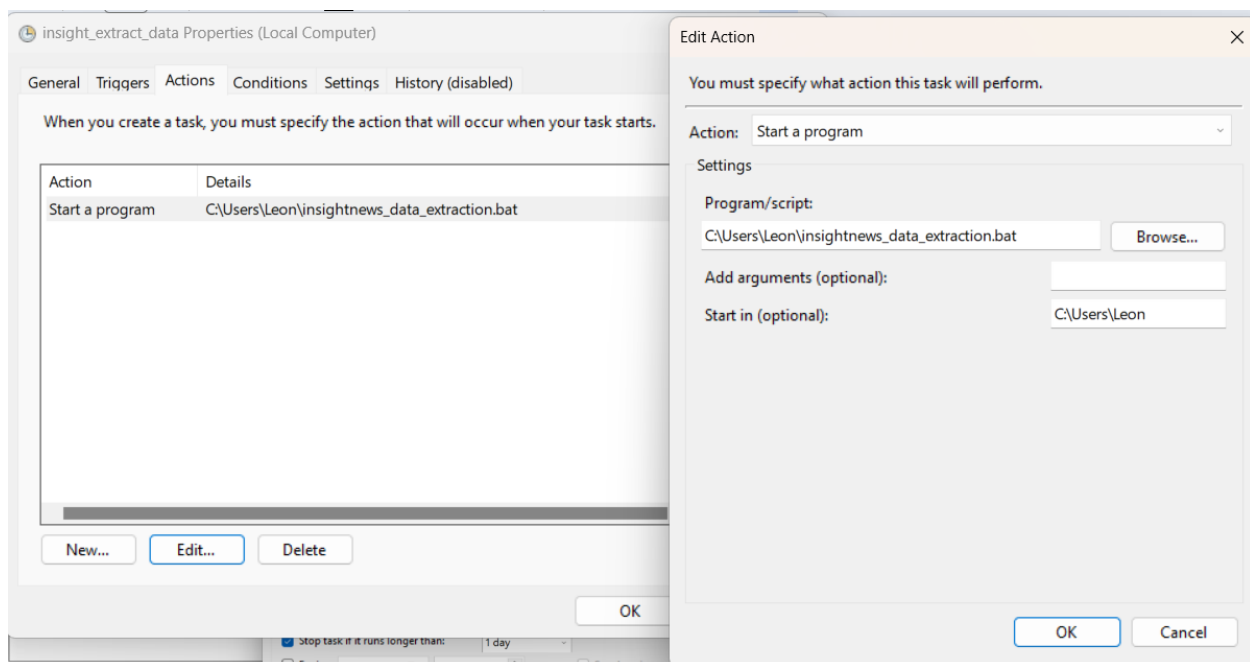
- b. Go to your Task Scheduler application and click on creating a task.



- c. Determine the trigger to automate the task



- d. Determine the action of your task in the task scheduler and insert the location of the .bat file for the task scheduler to run the file based on selected time interval.



- e. Now, the task scheduler will run the Python scripts automatically daily. The **data_extraction_intelligent_components python script** is to automate the process of data extraction and applying intelligent components. The **news_duration_update**

python script is to automate the process of updating the duration published for every news contents and topic model. The **topic_modelling_visualization python script** is to automate the process of creating a new bubble chart for topic modelling visualization on every first day of the new month automatically.