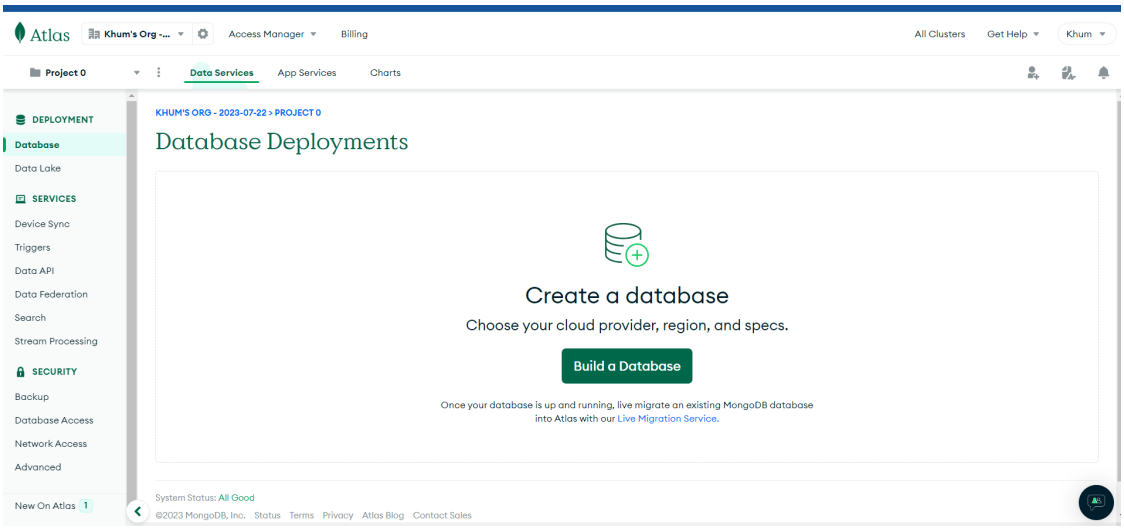
ReadMe Notes

Intelligent News Content Discover System

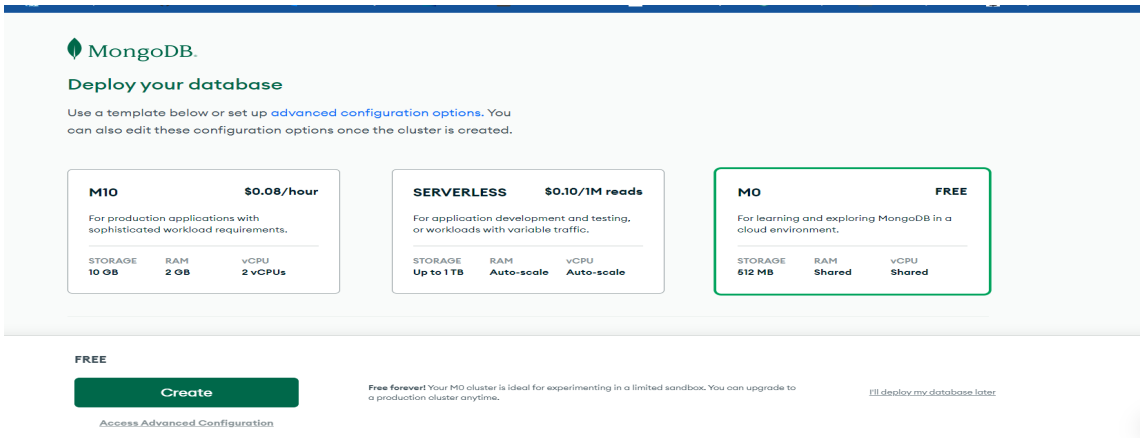
Initial steps for setup

MongoDB (Database Setup)

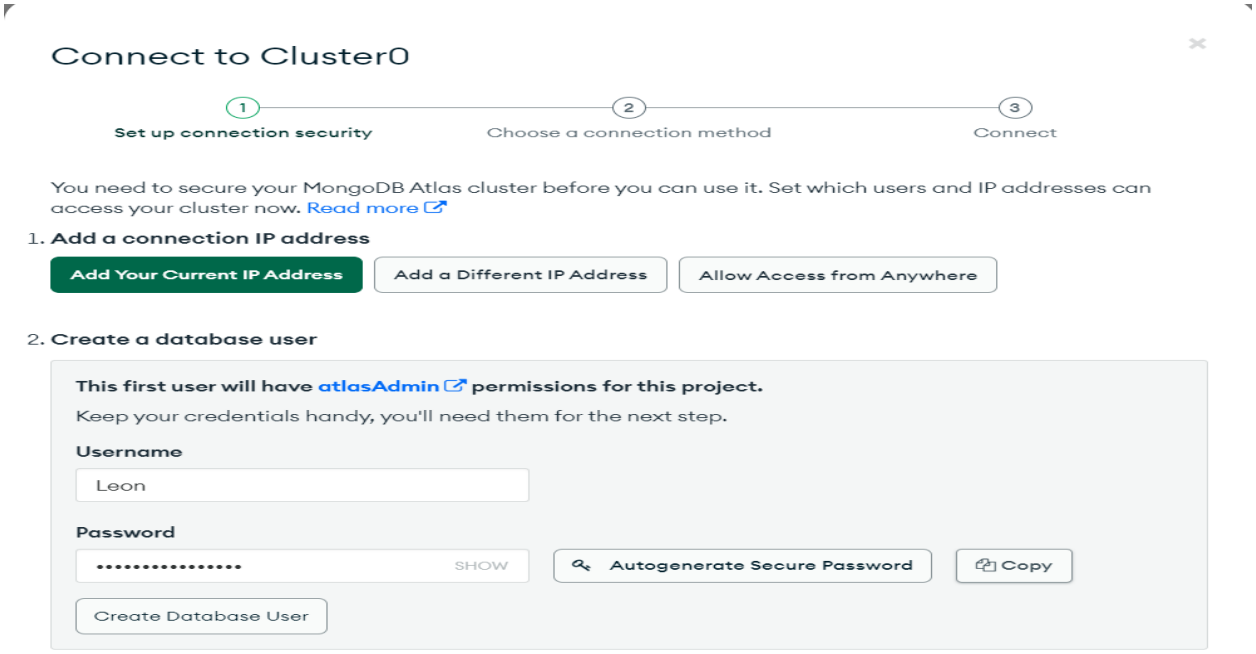
a. Create an account in MongoDB Atlas (https://account.mongodb.com/account/login)



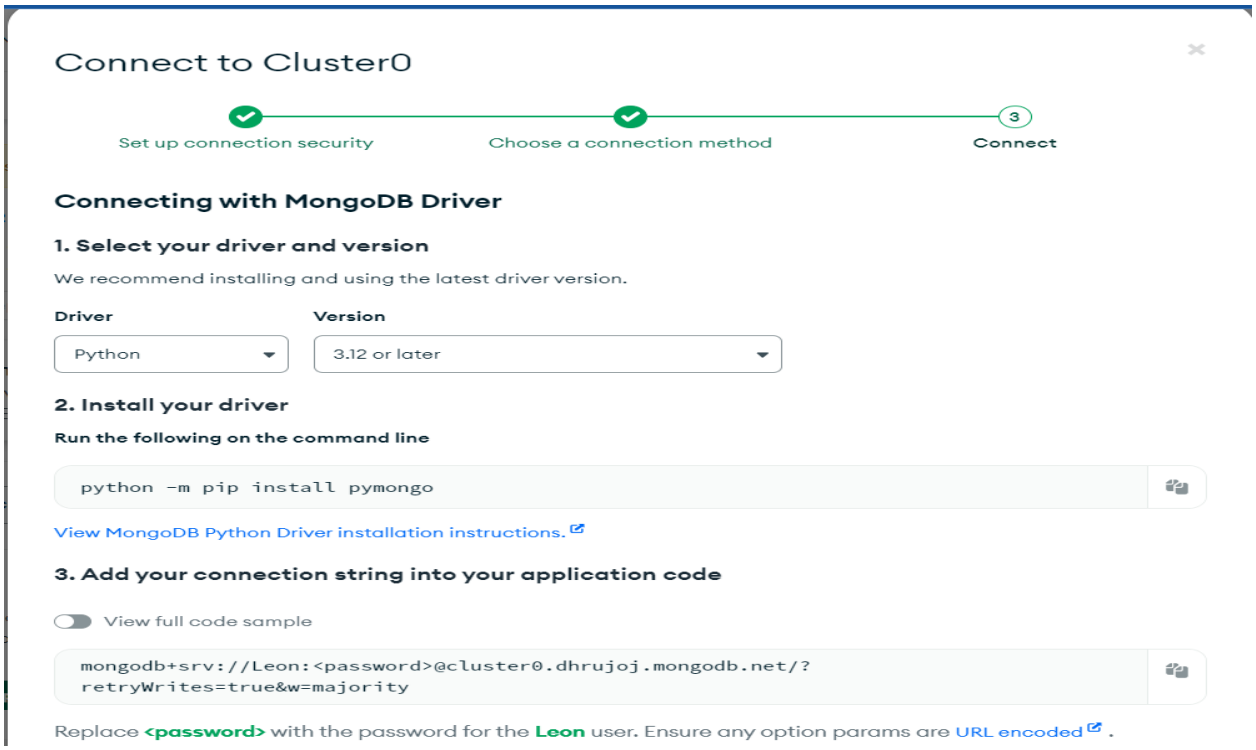
b. Create a project and deploy your MongoDB database by selecting your preferred configuration. For this example, a free database configuration is selected.



c. Create a connection IP address and database user

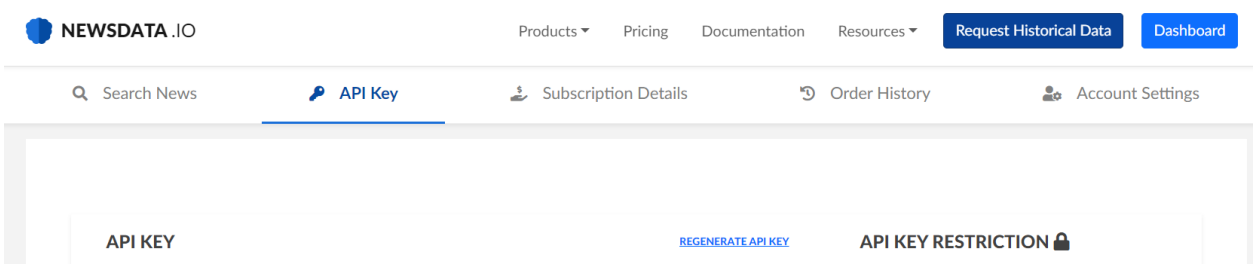


d. Copy the MongoDB connection string as the credential to connect to Python Flask.



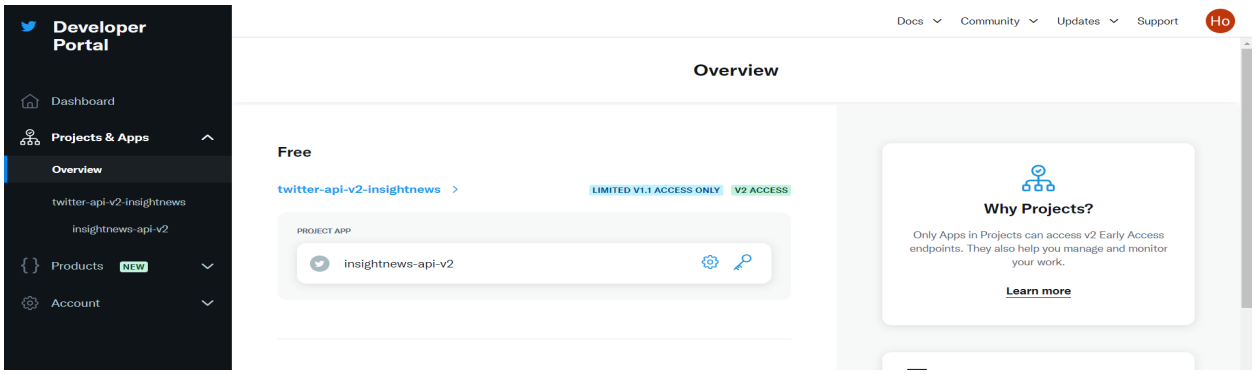
NewsData.io API (International News Articles Extraction)

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

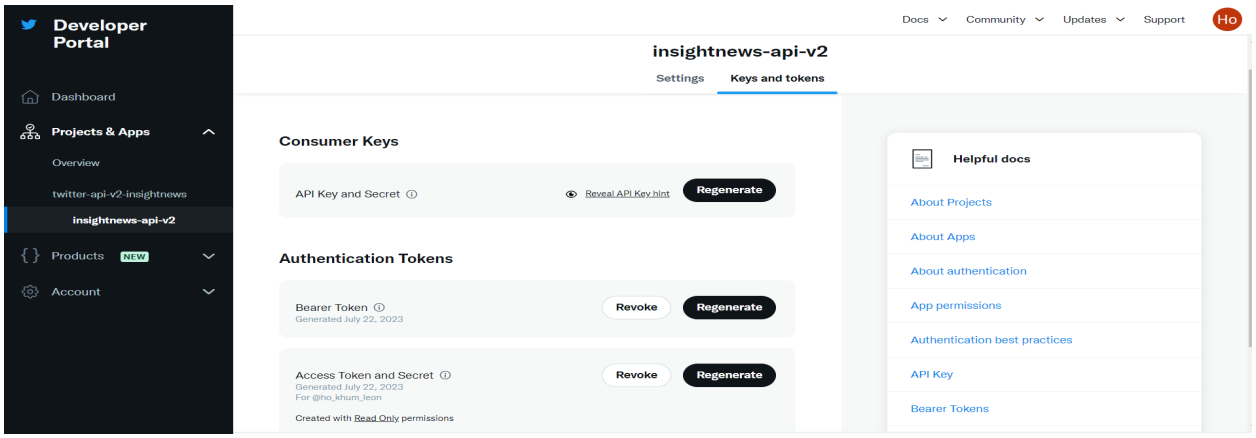


Twitter API (Tweets Extraction)

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



c. 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

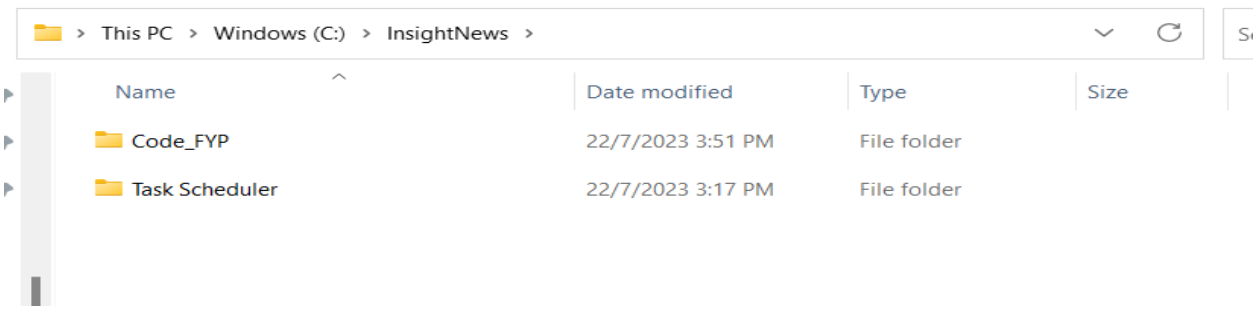
1. main_page.html **(home page/ landing page)**
2. forgot-password.html **(forgot password page)**
3. keyword-analysis.html **(keyword analysis page)**
4. Library.html **(library page)**
5. login-register.html **(authentication page)**
6. Multiview-comparison.html **(topic sentiment comparison page)**
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)**

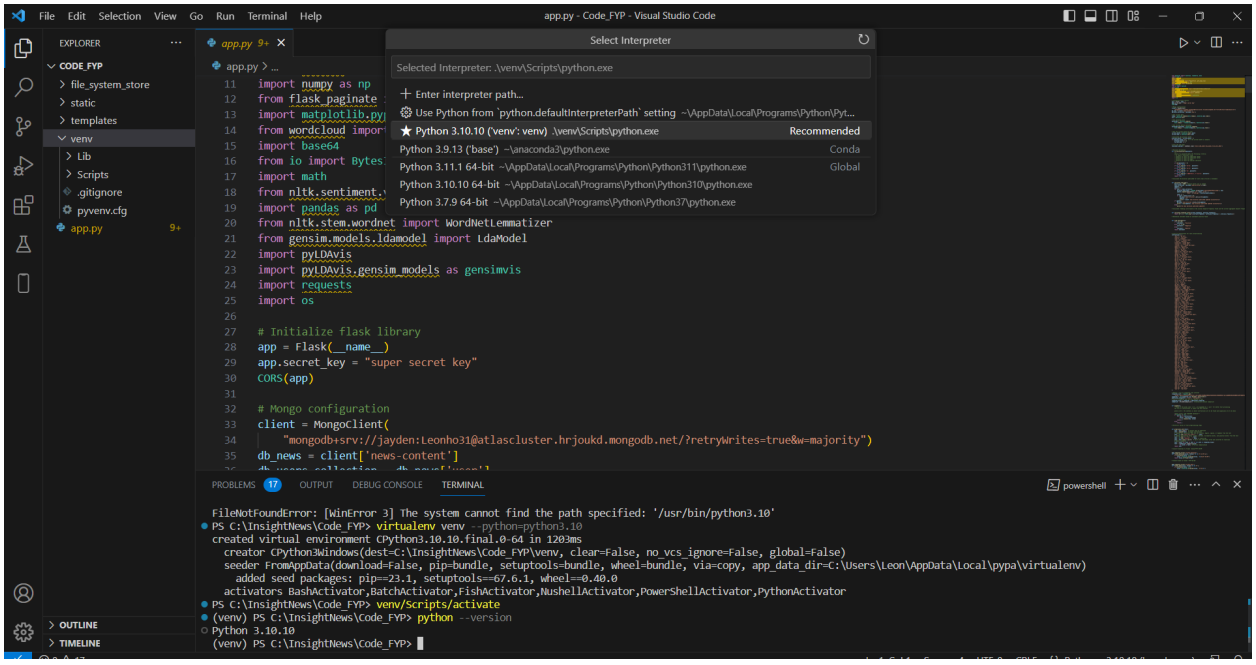
VSCode

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



b. 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

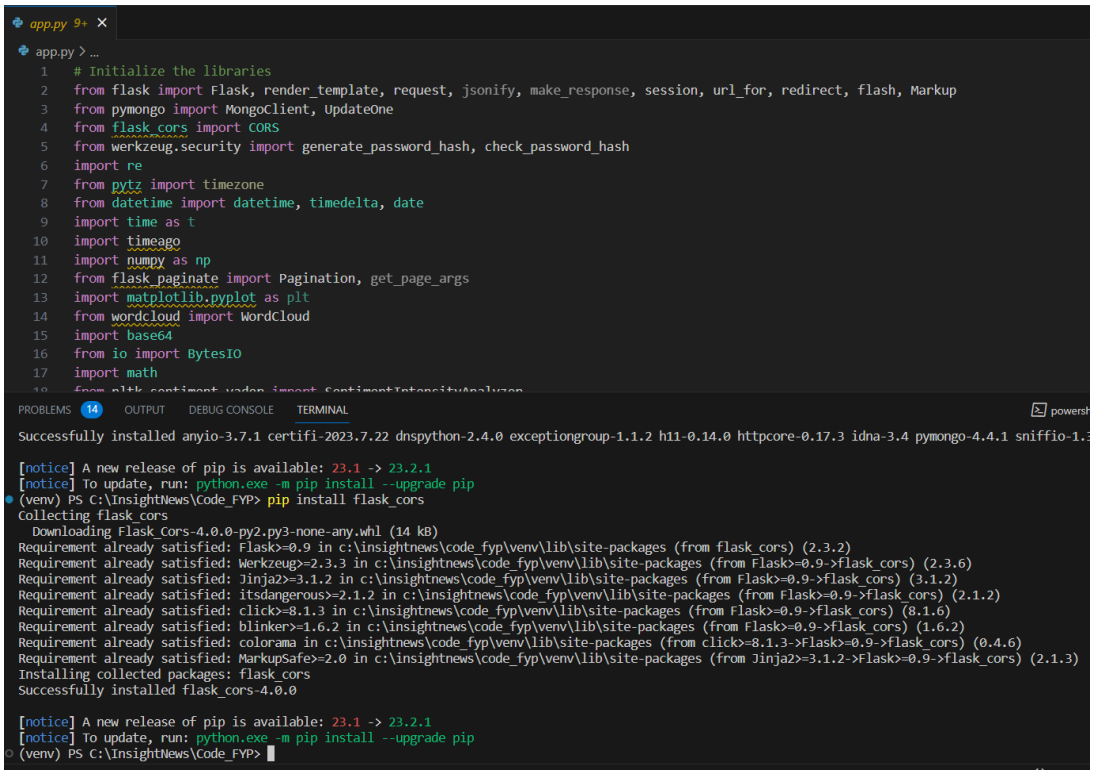


c. Activate the virtual environment and install all related libraries and dependencies

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

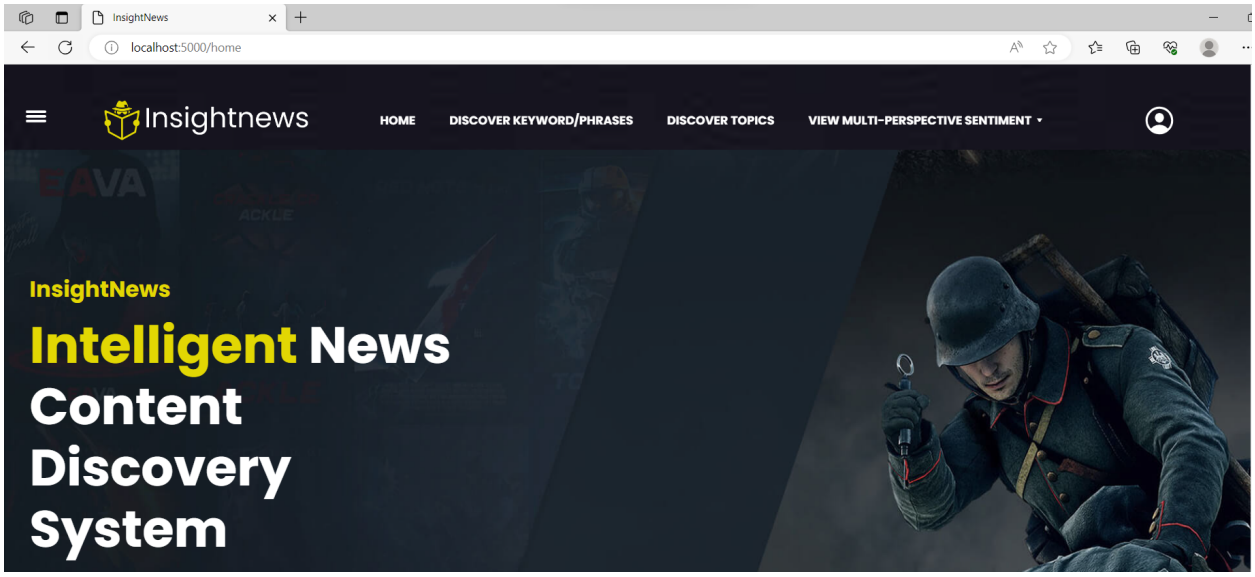
Required Libraries:

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)**



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:

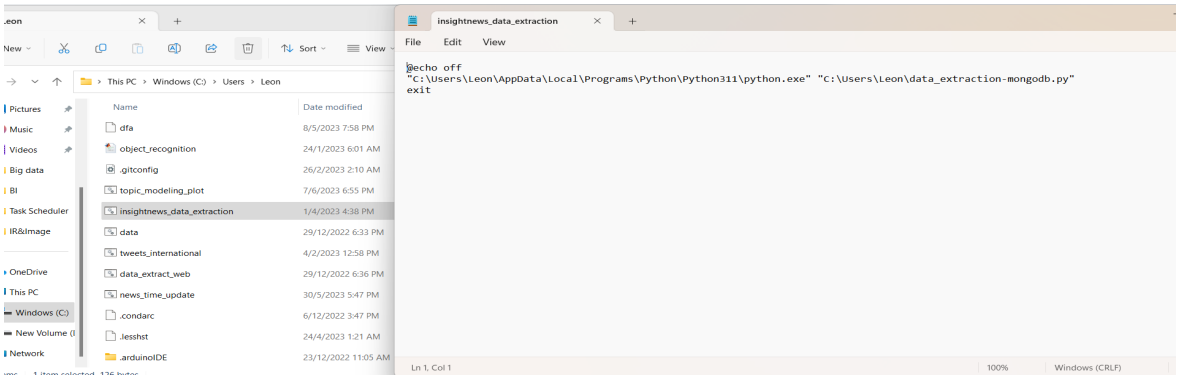
1. data_extraction_intelligent_components **(data extraction and intelligent components processes)**
2. news_duration_update **(update the duration published of news contents frequently)**
3. topic_modelling_visualization **(update the topic modelling every first day of the month)**

Task Scheduler

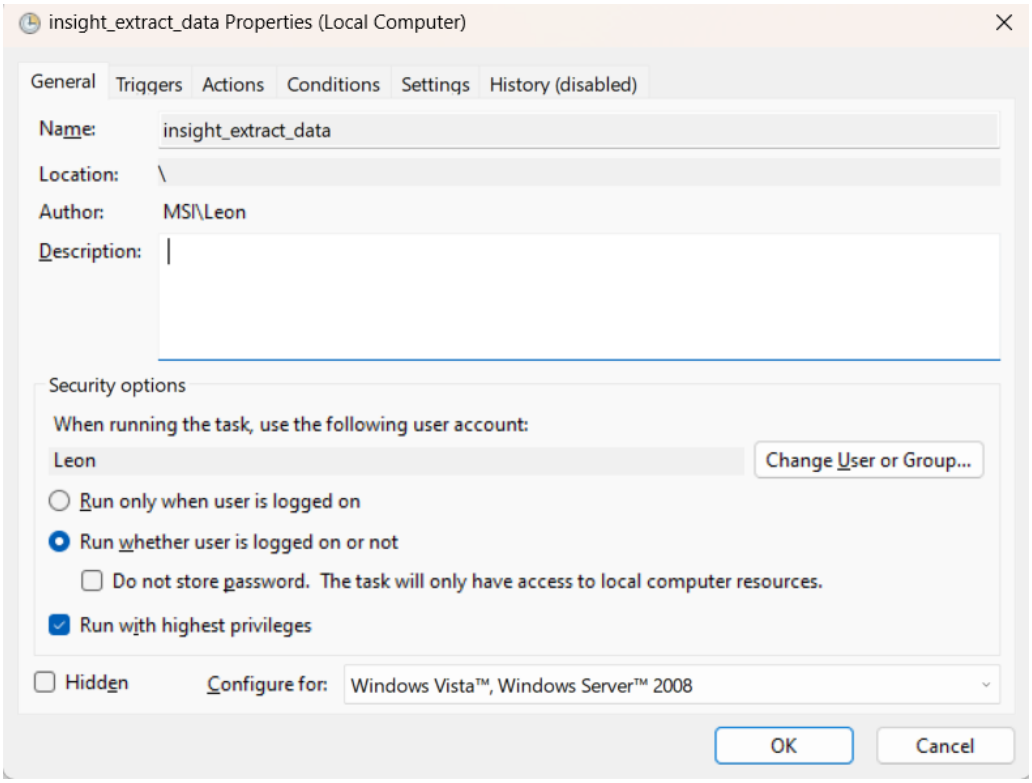
a. Open the notepad and insert the following bat scripts and save the file as .bat extension. The script involves the location of your Python interpreter and Python scripts.

@echo off

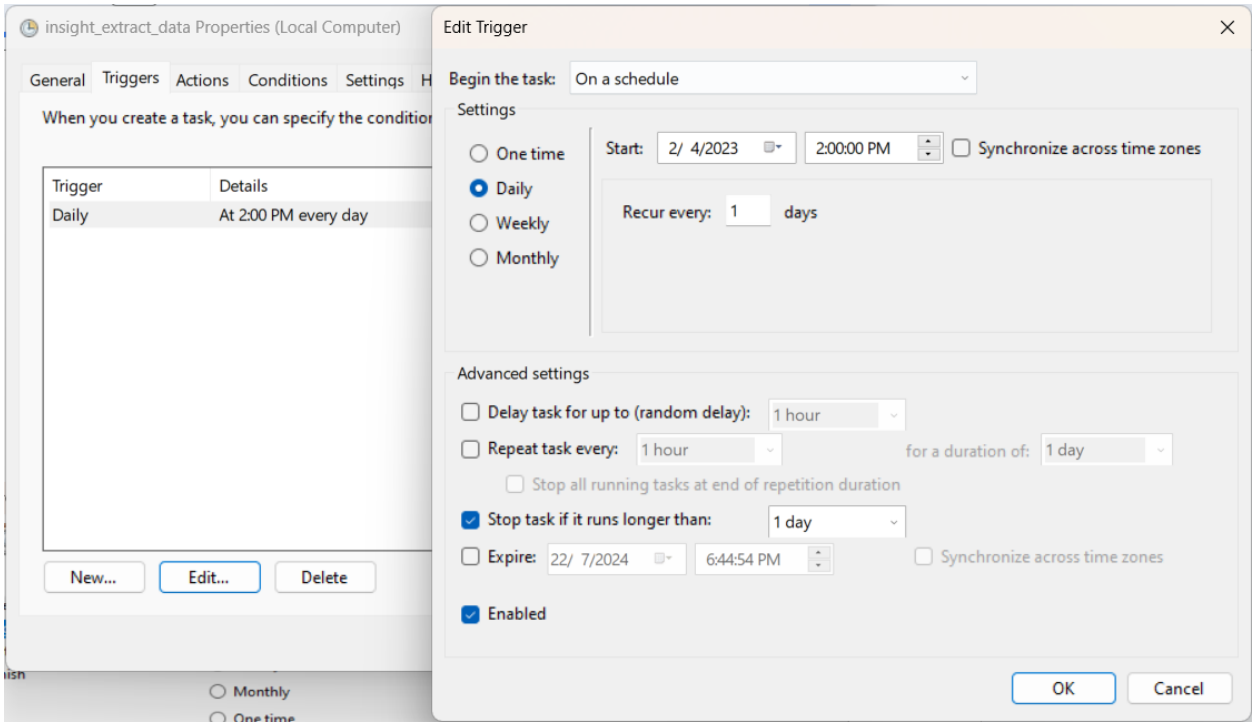
"C:\Users\Leon\AppData\Local\Programs\Python\Python311\python.exe" <Python interpreter location>
"C:\Users\Leon\data_extraction-mongodb.py" <Python scripts location>
exit



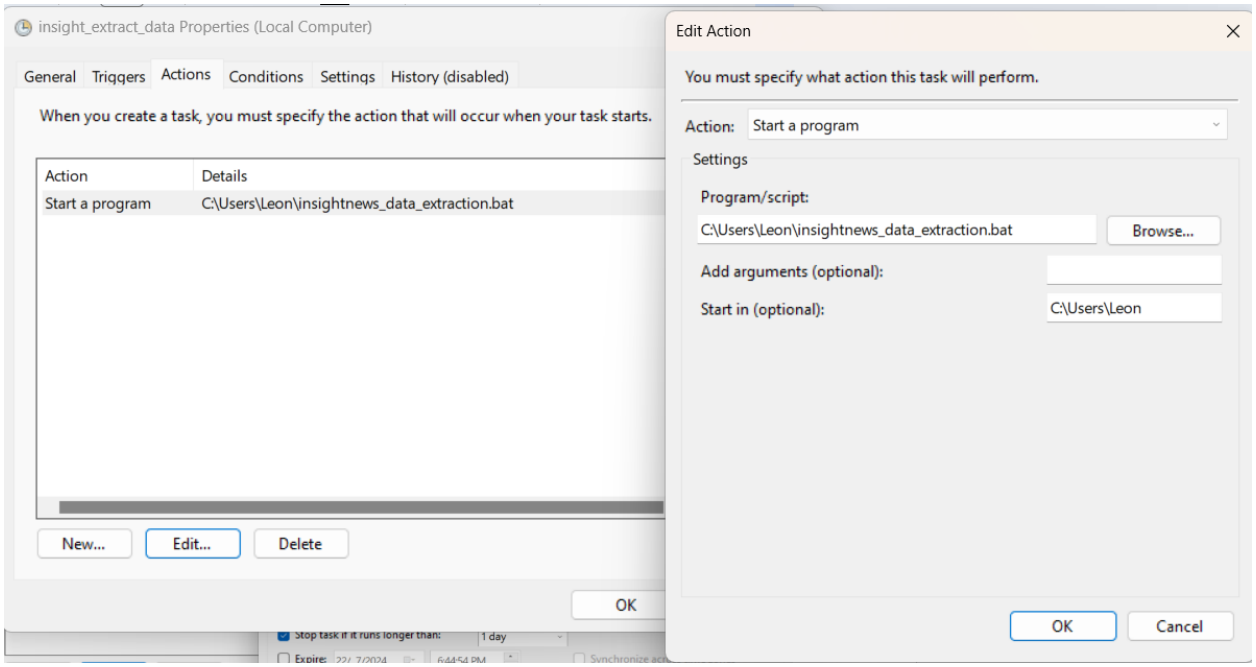
b. Go to your Task Scheduler application and click on creating a task.



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.