Bonus Assignment #2

Assignment Deliverables:

You are required to submit a SINGLE Zip file that has the following deliverables are:

1. Your IPYNB script
2. All of your source code and output
3. Output report that has your assignment run saved in OUTPUT.pdf
4. Video recording of 15-20 minutes as a demo for the run of your assignment using Panopto

Post your assignment as a SINGLE ZIP file on Blackboard.

General Instructions:

1. **There is NO PARTIAL credit for the bonus assignment submission that has partial/incomplete code.**

2. **The grading for this assignment is BINARY: CREDIT or NO CREDIT**

3. All of your source code must be clearly documented and functional; **ZERO credit will be given to the submission that has nonfunctional code**.

4. Submit your comparative analysis report for the results you obtained for the experiments you executed.

5. **ZERO credit will be given to the submission that has NO comparative analysis report.**

6. Submit your **IPYNB** scripts

7. **Panopto Video recording** (15-20 minutes) of your run that has your code and your output.

Requirements:

1. Use Anaconda **Python** 3.11 to create IPYNB script: **GitHub_EdgeDB**

2. Download and install **Milvus** (https://milvus.io/docs/install_standalone-docker.md) on your development computer.

3. Compare and contrast the time needed to collect 1 Month data and 1 Year data from **GitHub** and store the data on **EdgeDB**.

4. For every **Date Property**, use **datetime** as the type for that property; do NOT use string for the date.

5. (***1 Week Data***) For **UNIT-TESTING** purposes while you are testing and debugging your code, create **GitHub_EdgeDB** IPYNB to collect all data for the **past week** for every repo listed below:
    a. React
    b. Selenium
    c. Python
    d. Keras
    e. OpenAI
    f. Docker
    g. Milvus

6. (***1 Year Data***) For **Final Delivery** of your assignment, modify the **GitHub_EdgeDB** IPYNB to collect all **issues** for the **past year**, from 11/5/2022 – 11/5/2023 for every repo listed:
    a. React
    b. Selenium
    c. Python
    d. Keras
    e. OpenAI
    f. Docker
    g. Milvus

7. Store the Data collected from **GitHub** on **EdgeDB**

8. Create an IPYNB script to read the data you stored for **GitHub** on **EdgeDB** and store it on **Milvus**, the vector database.

9. Use **OpenAI** "**text-embedding-ada-002**" (https://github.com/openai/openai-cookbook/blob/main/examples/vector_databases/milvus/Filtered_search_with_Milvus_and_OpenAI.ipynb) to create embeddings using for **GitHub Repos/Issues** data you read from **EDgeDB**

10. Store the **GitHub** data and its embeddings on **Milvus.**

11. Create and execute the **Filtered search** with **Milvus** and **OpenAI** embeddings to return the hits/matches in the **top 5 ranks** and their **scores** for the following queries:
    a. **"How are multiple choices streamed in openai "** issues creation date/last activity date between 1/2/2023 and 11/5/2023.
    b. **"What is the timeout in minutes for openai-python"** issues creation date/last activity date between 1/2/2023 and 11/5/2023.
    c. **"connect milvus timeout"** issues creation date/last activity date between 1/2/2023 and 11/5/2023.