To run and test our program, please follow these steps:

1. **Prepare a Directory**: Create a folder on your system to store the images and the collection.

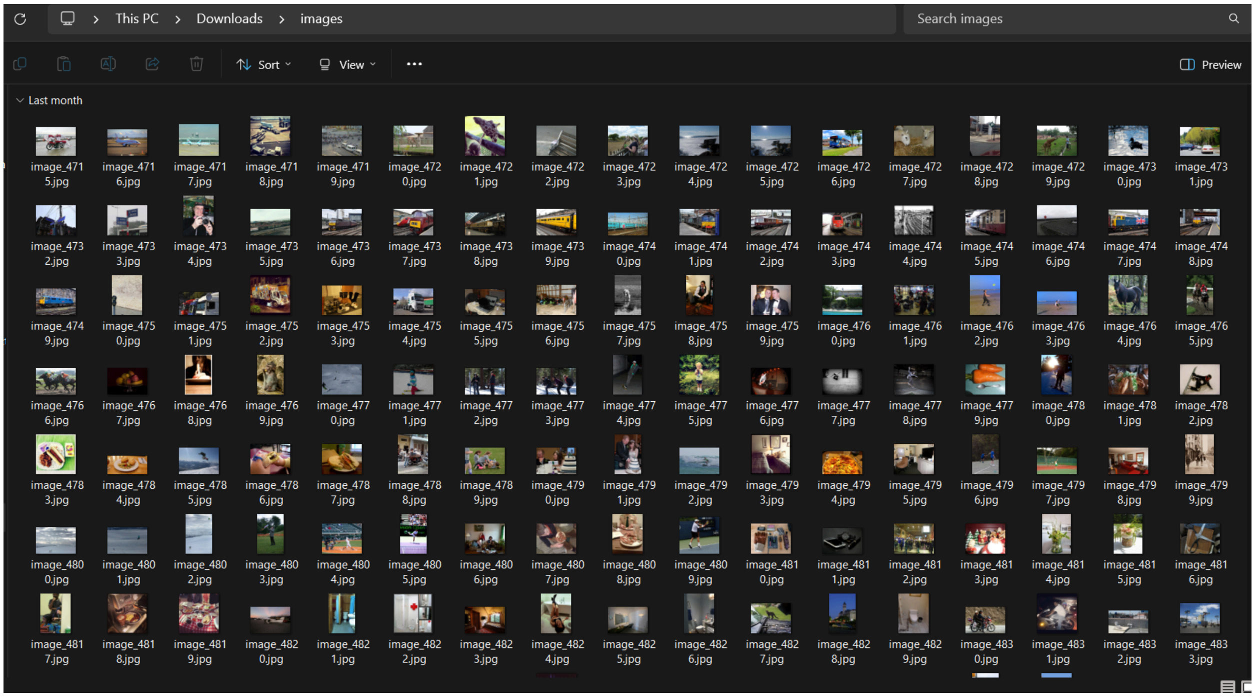
Down from the drive:
https://drive.google.com/drive/folders/17YUaZgvwhWeXwOL3qV3po88-9iGgn8pb?usp=sharing

2. **Update Dataset Path**: Open the `dataset.py` file and update the `PATH` variable to reflect the correct location of the collection folder on your machine.

3. **Execute the Application**: Navigate to the `Application/Retrieval_Demo` folder in your terminal and execute the following command to launch the web interface:

```
streamlit run front.py
```

Please ensure that the embedding vector matrices are saved within the specified directory.

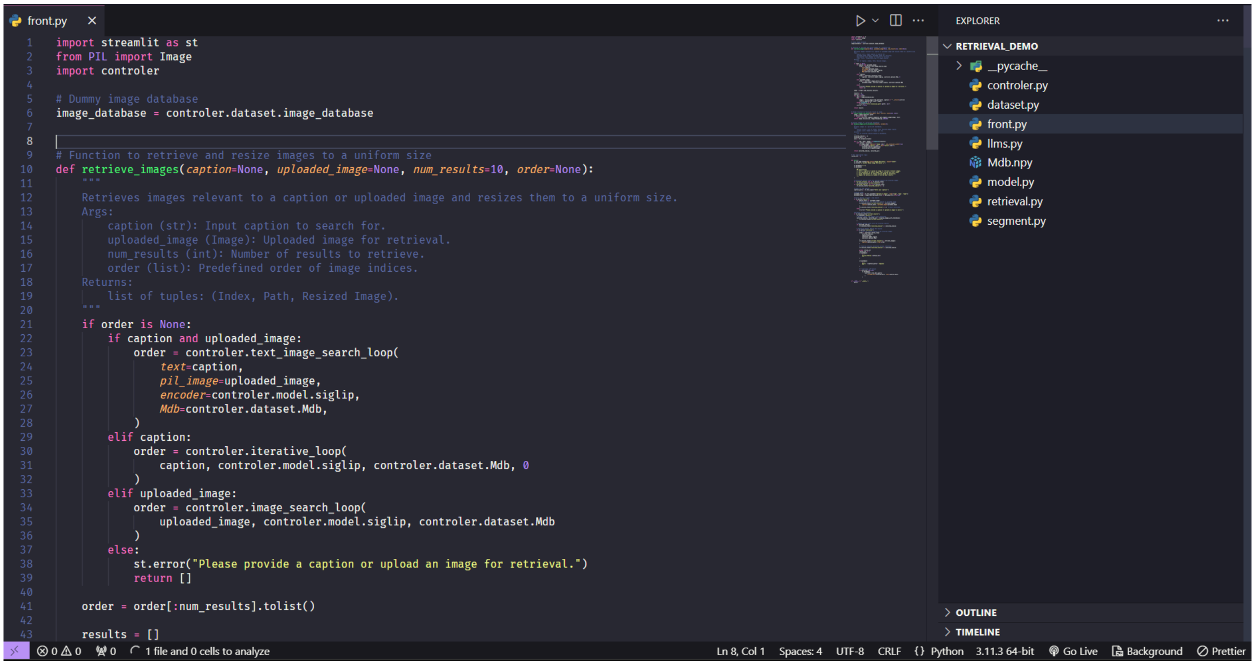


The software is designed following the **MVC (Model View Controller) architecture**:

- **front.py**: This component contains the Streamlit code responsible for designing the web interface.

- **Controller**: This layer synthesizes and manages interactions, coordinating the processing flow while abstracting unnecessary communications.

- **Additional Components**: The remaining components represent specific tasks, with each module divided into smaller units to handle distinct responsibilities.



```python
import streamlit as st
from PIL import Image
import controler

# Dummy image database
image_database = controler.dataset.image_database


# Function to retrieve and resize images to a uniform size
def retrieve_images(caption=None, uploaded_image=None, num_results=10, order=None):
    """
    Retrieves images relevant to a caption or uploaded image and resizes them to a uniform size.
    Args:
        caption (str): Input caption to search for.
        uploaded_image (Image): Uploaded image for retrieval.
        num_results (int): Number of results to retrieve.
        order (list): Predefined order of image indices.
    Returns:
        list of tuples: (Index, Path, Resized Image).
    """
    if order is None:
        if caption and uploaded_image:
            order = controler.text_image_search_loop(
                text=caption,
                pil_image=uploaded_image,
                encoder=controler.model.siglip,
                Mdb=controler.dataset.Mdb,
            )
        elif caption:
            order = controler.iterative_loop(
                caption, controler.model.siglip, controler.dataset.Mdb, 0
            )
        elif uploaded_image:
            order = controler.image_search_loop(
                uploaded_image, controler.model.siglip, controler.dataset.Mdb
            )
        else:
            st.error("Please provide a caption or upload an image for retrieval.")
            return []

    order = order[:num_results].tolist()

    results = []
```

Last but not least, due to the purpose of this demonstration, not all experimental evaluations are fully simulated within the application.

Also need to ensure that the following essential libraries are installed to run the code:

```python
import streamlit as st
from PIL import Image
import numpy as np
import google.generativeai as genai
import os
import time
from transformers import AutoProcessor, AutoModel, AutoTokenizer
import torch
from sklearn.metrics.pairwise import cosine_similarity
import cv2
from transformers import AutoProcessor, CLIPSegForImageSegmentation
import torch
```

These libraries have been mentioned in the report, if there is any problem, please contact us immediately via email: Quân Hoàng Ngọc or phone number: 0911869162. Thank for professor!