Name:- Sakshi S Ghonge

Aim: To Create Program to perform a retrieving Image and Searching

Objective:

The fundamental need of any image retrieval model is to search and arrange the images that are in a visual semantic relationship with the query given by the user.

Most of the search engines on the Internet retrieve the images based on text-based approaches that require captions as input.

Theory :

Creating a program for image retrieval and searching involves several key components and steps. Here's a theoretical overview of the process:

Components of an Image Retrieval and Searching System:

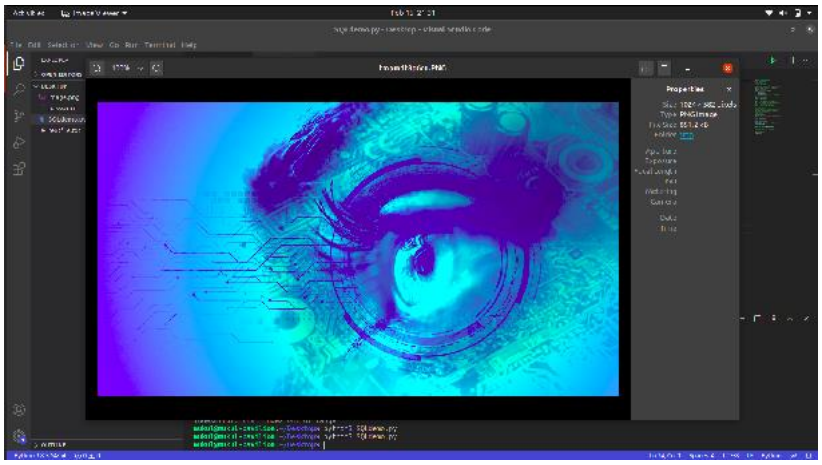
- Image Database: The system typically starts with an image database where you store and organize your images. These images can be stored in a directory structure on your file system, a cloud-based storage service, or a database.
- Image Features: To search for images, you need to extract meaningful features from them. These features can be visual (e.g., color histograms, texture descriptors, keypoints, deep learning embeddings) or metadata-based (e.g., tags, descriptions, timestamps).
- Indexing: To efficiently search through a large number of images, you'll need to create an index or database that maps image features to image files. This allows for faster searching and retrieval.
- Search Algorithm: Choose or design a search algorithm that takes a query (e.g., an image or keywords) and retrieves relevant images based on similarity measures. The choice of algorithm depends on the type of features used and the desired search criteria (e.g., similarity, relevance).

Code:

```
# Import the required modules

import mysql.connector

import base64

from PIL import Image

import io


# For security reasons, never expose your password

password = open('password','r').readline()
```

```
# Create a connection
mydb = mysql.connector.connect(
        host="localhost",
        user="root",
        password=password,
        database="studentdb" # Name of the database
)

# Create a cursor object
cursor = mydb.cursor()

# Prepare the query
query = 'SELECT PICTURE FROM PROFILE WHERE ID=100'

# Execute the query to get the file
cursor.execute(query)

data = cursor.fetchall()

# The returned data will be a list of list
image = data[0][0]

# Decode the string
binary_data = base64.b64decode(image)

# Convert the bytes into a PIL image
image = Image.open(io.BytesIO(binary_data))

# Display the image
image.show()
```

Output:



Conclusion :-

Building such a system requires careful planning, feature extraction, similarity calculation, and user interaction design. The specific implementation details will depend on your application and technology choices.