Ain Shams University
Faculty of Engineering
Computer and Systems Engineering Department

CSE 421: Selected Topics in Computer Engineering, Multimedia Systems
4th Year CSE, 2nd Semester 2020/2021

Content Based Multimedia System

Abstract

As the size of multimedia database grows, the difficulty in finding desired information increases and it becomes impractical to manually annotate all contents and attributes of the media. To copy with these challenges, content-based multimedia retrieval systems have been developed. The report provides a conceptual architecture for the design of content-based retrieval system. Essential components of retrieval system and their research issues, including feature extraction and representation, dimension reduction of input content, indexing, and query specifications, are discussed in this report. Many potential applications are introduced for several applications, such as medical diagnosis, intellectual property, and broadcasting archives. Content-based retrieval is a young research field and there exists many challenging research problems. Several research issues are also addressed for the future research.

The Multimedia System will comprise of:

- Data entry (allowing the user to add images & videos to the server)
- Feature extraction & comparing (Feature extraction of input content and comparing the database

Table Contents

1. Introduction

During the last years, rapid diffusion of inexpensive tools (mostly embedded in smartphones, tablets, and other common mobile devices) for image, video, and audio capturing, together with improvements in storage technologies, have favored the creation of massive collections of digital multimedia content. However, the difficulty of finding relevant items increases with the growth of the amount of available data. To face this problem, two main approaches are often employed, both involving the use of metadata: the first includes manually annotating multimedia content by means of textual information (e.g., titles, descriptive keywords from a limited vocabulary, and predetermined classification schemes), while the second amounts to using automated feature extraction and object recognition to classify content.
The latter is naturally related to content-based multimedia retrieval (CBMR) systems and often represents the only viable solution since text annotations are mostly nonexistent or incomplete; moreover, CBMR has proven to dramatically reduce time and effort to obtain multimedia information, whereas frequent annotation additions and updates to massive data stores often require entering manually all the attributes that might be needed for the queries.

1.1. Purpose

The purpose of this report is to convey all the necessary information needed to understand, run, and use the system.

Additionally, this document provides:

- Software design and architecture using UML notation.
- A definition of the system's capabilities.
- A description of the environment in which the system is expected to operate.

The document is intended to serve several groups of audiences:

First, the client for the project, the normal user in our case, is expected to review this document, as it will serve to establish a basis for agreement between the client and development team about the functionality to be provided by the system and also understand of what and how the system operate.

Finally, the developers will review the document to clarity their understanding of what the application does and how.

1.2. List of Definitions

Acronym	Definition
CBIR	Content Based Image Retrieval
CBVR	Content Based Video Retrieval
VDV	Video Database
IDB	Image Database
SIFT	Scale-Invariant Feature Transform
CBR	Content-Based Retrieval

1.3. Scope

The scope of the CBR system project is as follows:

- ➢ To assist the users in capturing the effort spent on their respective working areas.
- ➢ To utilize resources in an efficient manner by increasing their productivity through automation.
- ➢ The system generates types of information that can be used for various purposes.

The CBR System will be web-based application that allows users to search for similar contents and retrieve it in very fast way.

2. Detailed Description of The Project

2.1. Product Perspective

CBR is a stand-alone system used by the any user. The system is self-contained. However, it is possible to exchange data with other systems through external interface if required.

2.2. General Constraints

This system is a web-based application, there will be a need to provide an internet connection to the system.

CBR can potentially have hundreds of users but the more the users rely on the server the slower it will be. Therefore, the system should be designed to be easy to use, providing help instructions, and appropriate error messages for invalid user inputs.

CBR can hold any size of data (image or data). Therefore, the larger the size of especially the video the slower the preprocessing, processing, searching and retrieving the ideal content.

2.3. Environment Description

The project is a web-based environment written by using NodeJS and ExpressJs for backend development, Python for image and video preprocessing and SQLite for database development.

2.4. Assumptions and Dependencies

The following is a list of assumptions and dependencies that would affect the software requirements if they turned out to be false:

- Users have a basic understanding of PC and Windows.
- The user should have internet connection to enter the system.

3. Beneficiaries of The Project

The purpose of this system is to automate and help the user for searching a content using content CBR system.

The CBR system provides:

- ➢ Content based image retrieval system based on Color Space Algorithm.
- ➢ Content based image retrieval system based on histogram difference Algorithm.
- ➢ Content based image retrieval system based on SIFT.
- ➢ Content based Video retrieval system based on Histogram difference Algorithm.

The CBR systems can be used in many fields in daily life and facilitates a lot the work of trademarks, copyright strikes and logo's cloning.

Examples of the fields are:

- Trademarks, copyrights, and logos
- Retailing
- Fashion and fabric design
- Art galleries and museums
- Interior design or decorating

4. Detailed Analysis

The system consists of the following:

- Backend server where the user uploads his content whatever it is image or video
- Python backend code to manage the sqlite3 database
- Python preprocessing, feature extraction, searching and retrieving codes which is the CBIR and CBVR systems

The user uploads the image or the video with the type of the algorithm he needs to preprocess his content uploaded and then it has been passed by its reference to the system based on the content type, the system then applies the algorithm the user chooses and output the feature extracted.

If the content is video, the video first is analyzed using histogram distance algorithm and the key frames of the videos are extracted to be compared with the DB, so that save a lot of time and space while searching and retrieving and increasing the accuracy.

The feature extracted value then will be searched through the database to look for similarity, if there is one, the content which is similar will be retrieved to the user.

5. Detailed Description of Adopted Techniques

They are multiple ways to perform feature extraction in images such:

- Primitive features: (Mean Color (RGB), Histogram)
- Semantic features: (Color Layout, Texture)
- Domain specific features: (Face recognition)

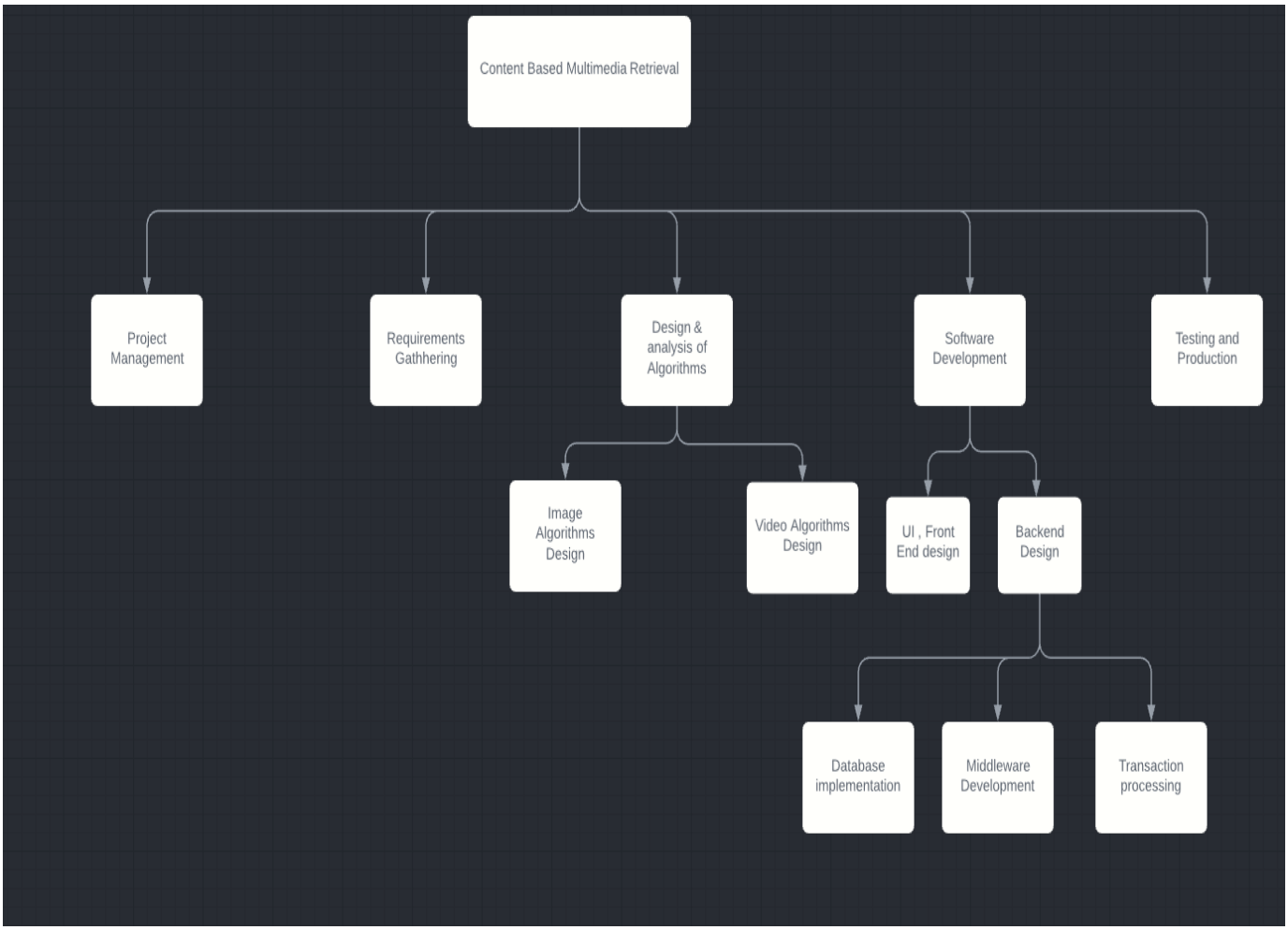
Our approaches were:

- Color Distance:
  First we convert the image from RGB mode to HVS mode which is the YCbCr representation of the image because it is more sensitive to color changes, then we divide the image into blocks to form a 5x5 grid resulting a 25 blocks, each block has three channels YCbCr, for each channel in each block we calculate the mean, standard deviation and skewness, so we have a feature vector with dimension (255,) for each image.
  Then when comparing with two images we simply calculate the distance between the feature vectors using L2 norm distance.
- Histogram Comparison:
  For each image we calculate the histogram for 256 bins and a fixed dimension of 180x256.
  Then when comparing the two images we calculate the similarity between the two histograms and if this similarity is less than a threshold then it is a good match to our image.
- SIFT algorithm:
  For each image we extract the corner points and their descriptors using the SIFT algorithm and save these descriptors.
  Then when comparing the two images we apply a brute force comparison between the descriptors to get the matched points, then we filter these matched ones to find which ones are a strong match, finally we return the ratio between the number of good matches and the total number of matches which represents the similarity between the two images.

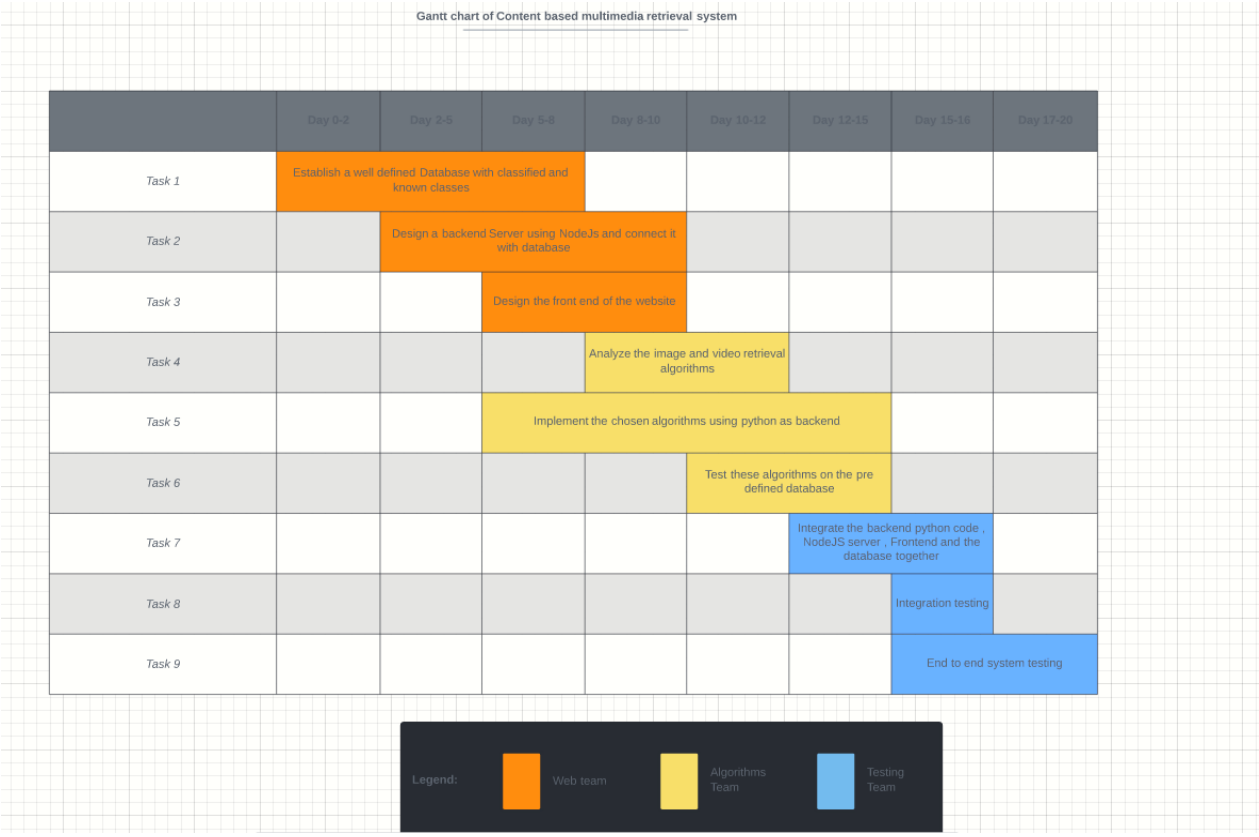
In CBVR we used only one approach to compare videos which is NVS:
For each frame in the query video, we find if there is at least a single match in any frame in the DB video, then calculate the ratio between the number of matches and the length of the frames. To calculate the match between two frames we use the histogram comparison technique.

In the key frames extraction, we used the Sequential Comparison technique which is:
Assuming that first frame is a key frame and then for each later frame if there is a big difference between it the and the reference frame, then we assume that this frame is a new key frame and so on.

6. Task Breakdown Structure of The System

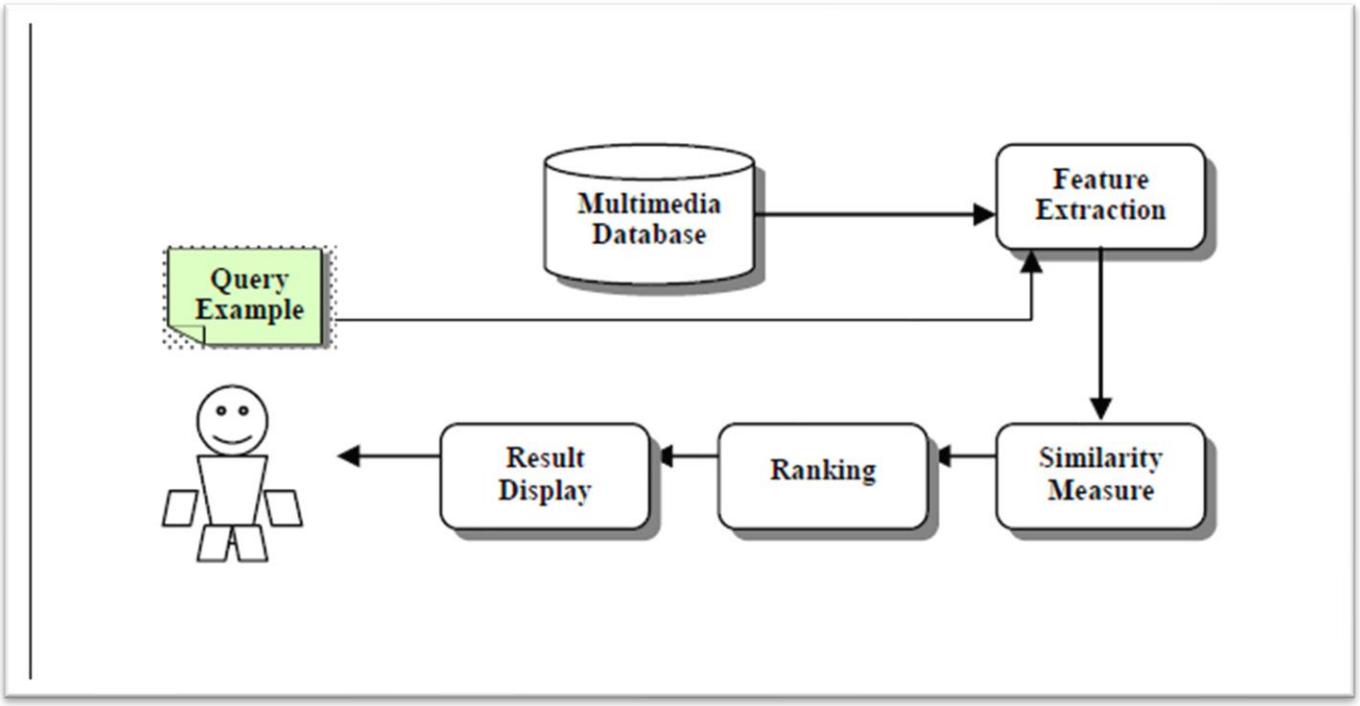


7. Time Plan and Team Members Roles:

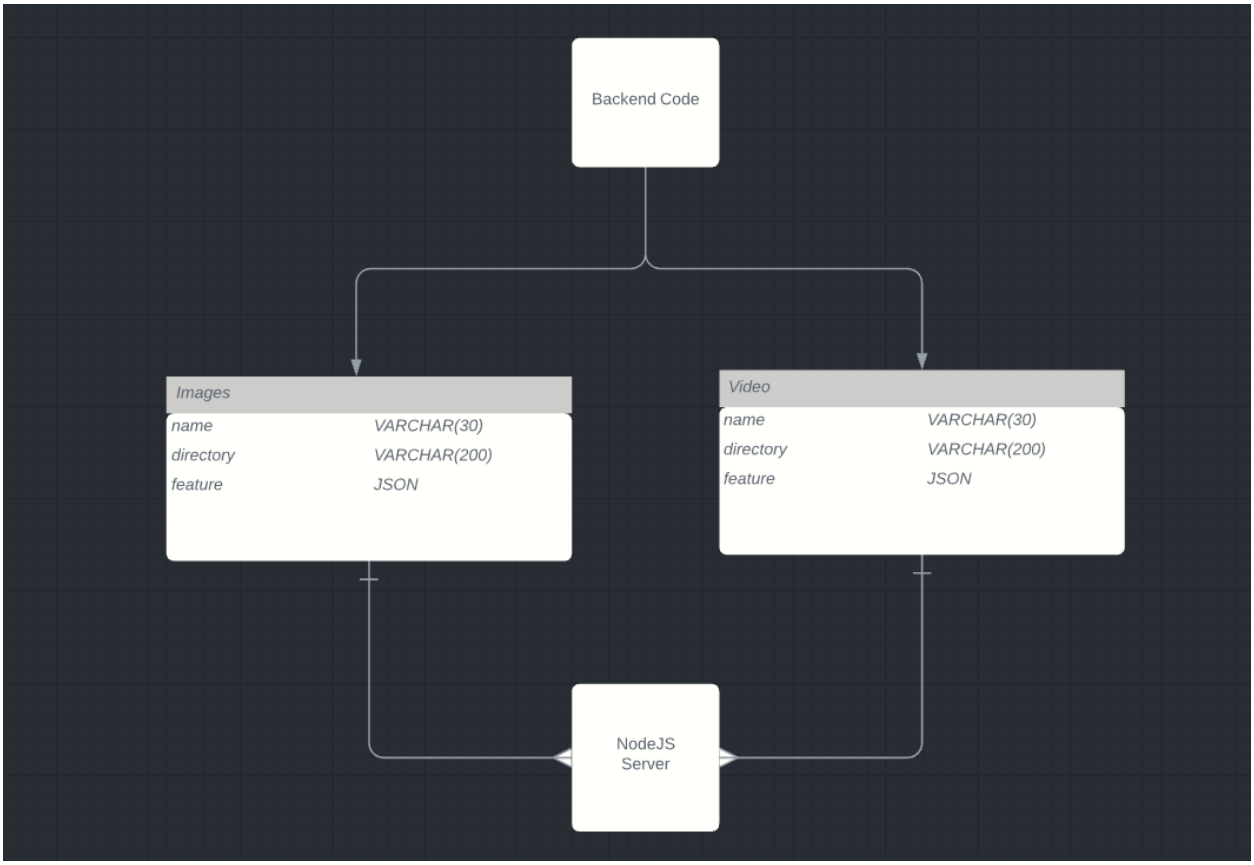


Name	ID	Role
Ahmed Mostafa Abd El-Fatah	1600195	SIFT Algorithm & CBVR
Ahmed Mohammed Salah	1600171	Color Distance, CBVR & integration
Fatma Elzahraa Mahmoud Esmail	1600966	Web interface & histogram comparison
Ibrahim Atef Abd-Elhalim	1600012	Histogram Comparison and Database
Kirolos Samuel Azmy	1601025	CBVR, Database& Report
Mariam Salah Abd-Elhamid Mohammed	1601371	Web interface, SIFT algorithm
Mark Sameh Azer	1601058	NodeJS server, Database and integration
Martina Fadi Fouad	1601053	Database & Testing

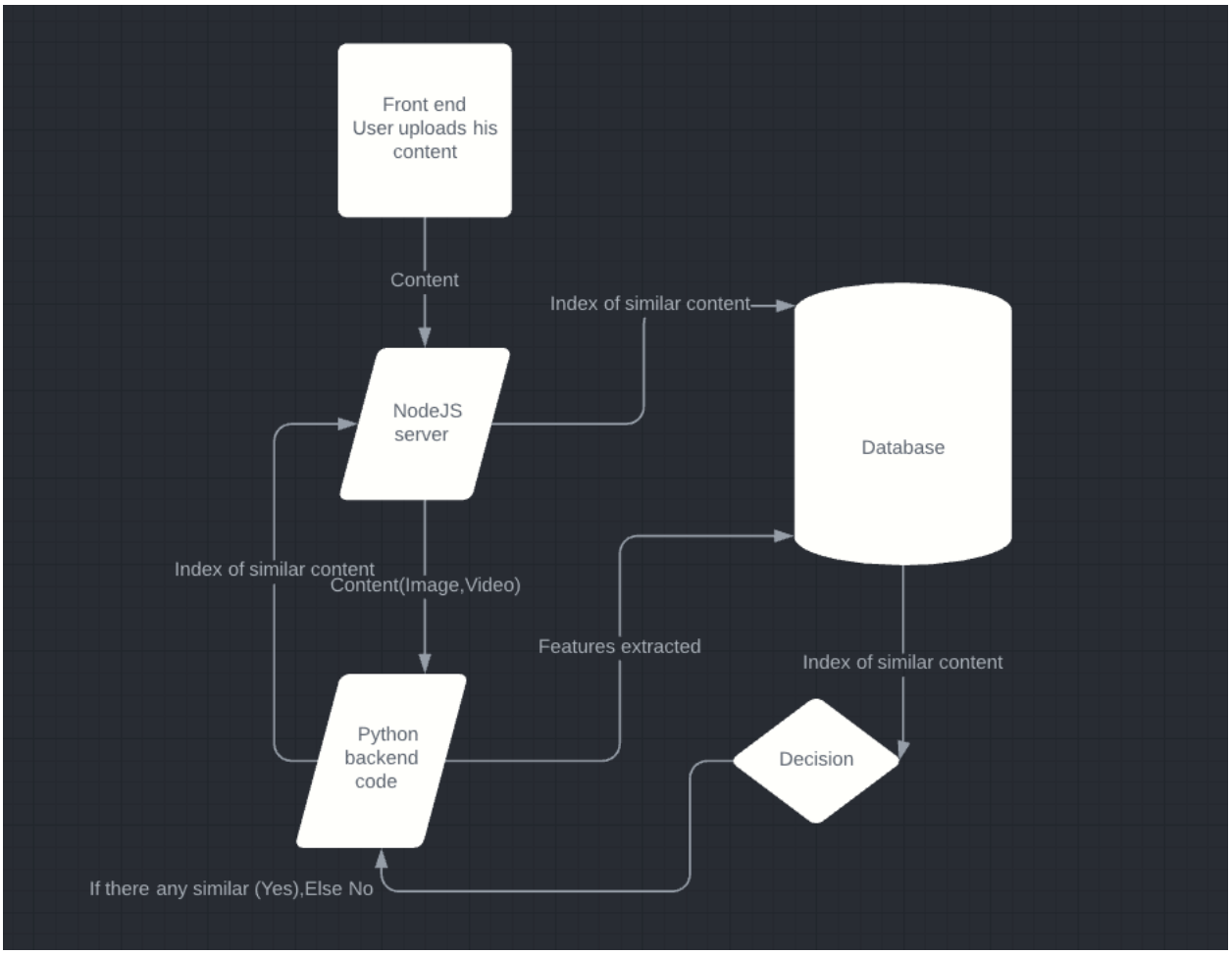
8. System Architecture



9. Multimedia Database Design

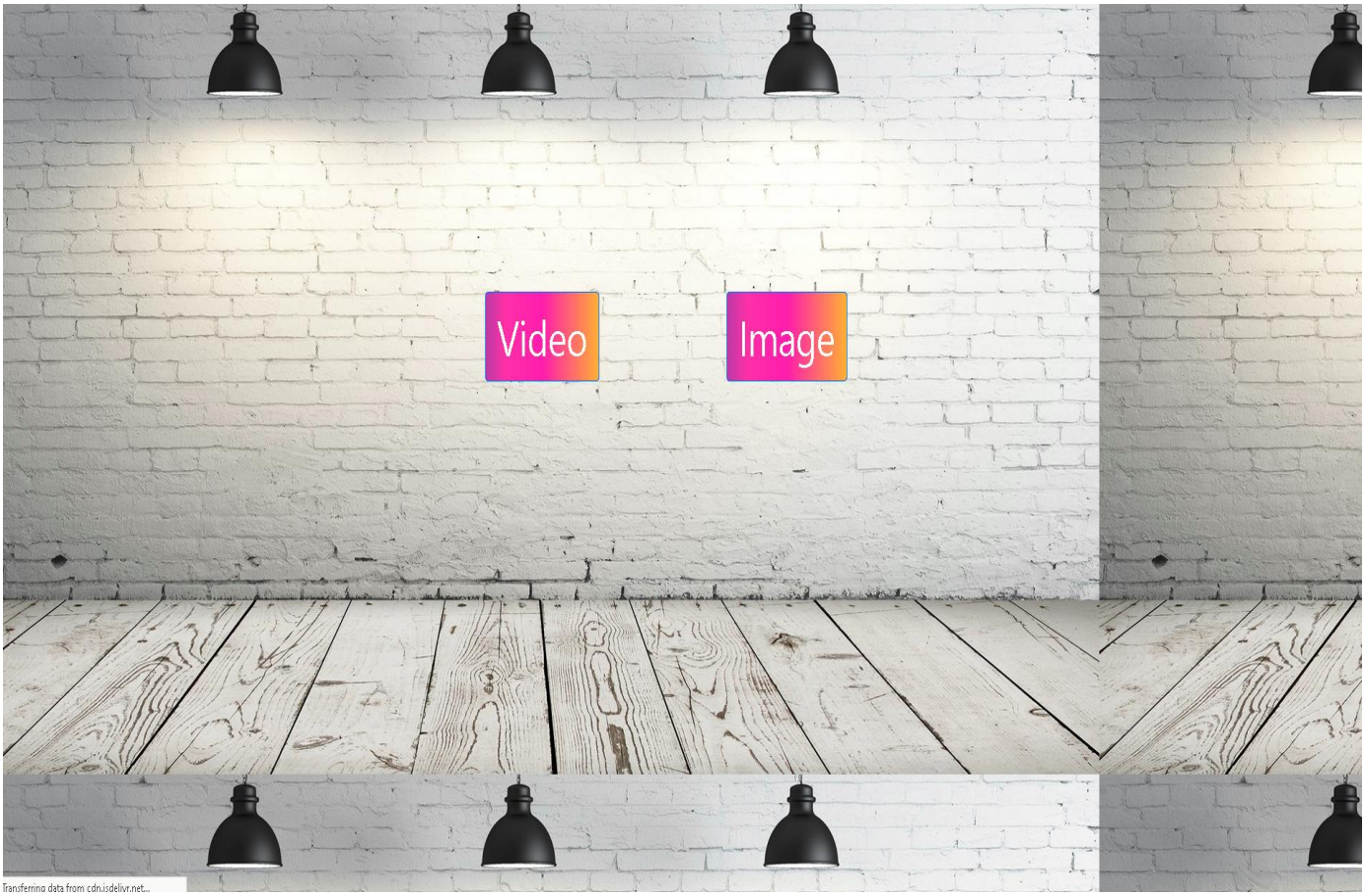


10. System Design

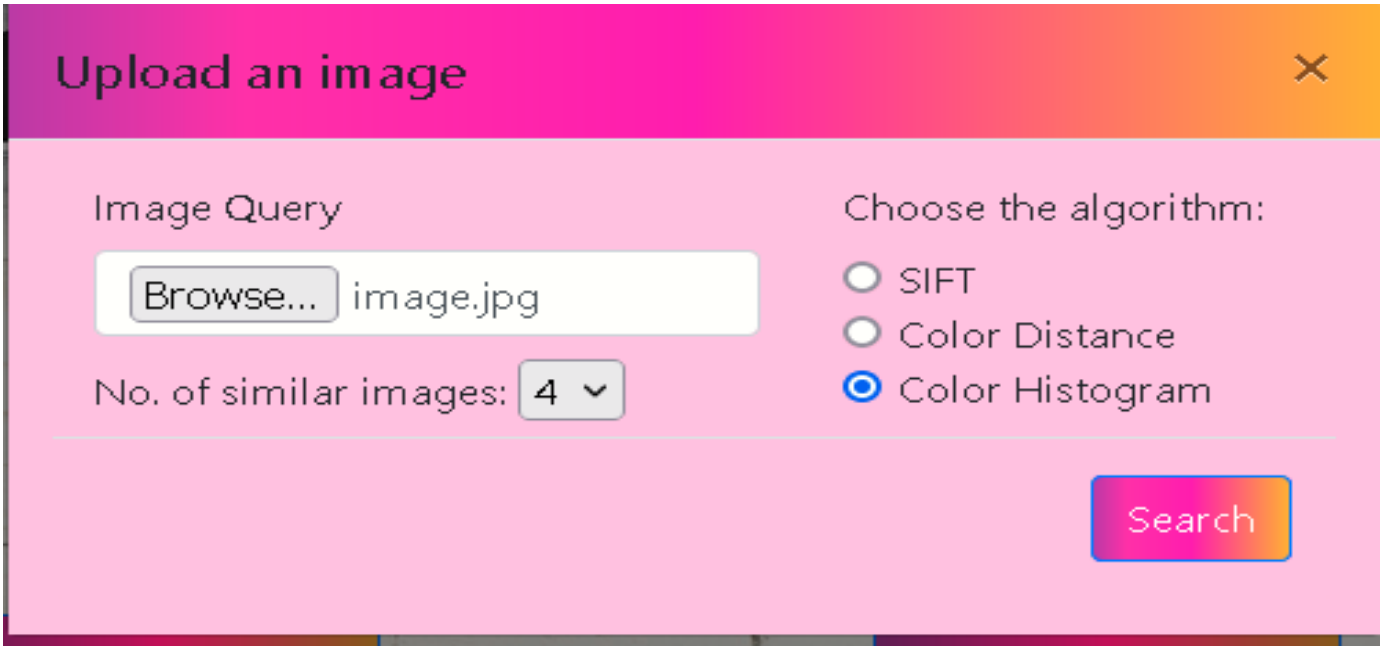


11. Testing Scenarios and Results

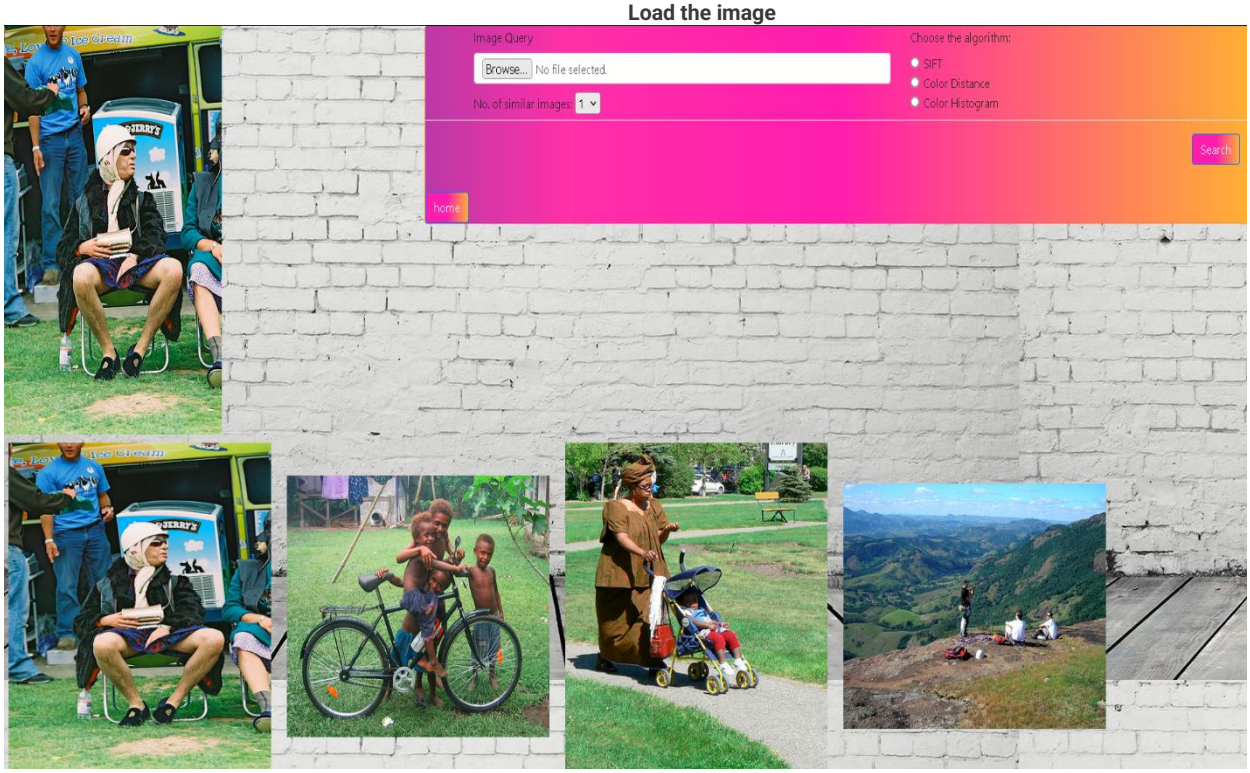
- **Testing Images Retrieval:**



Opening the website

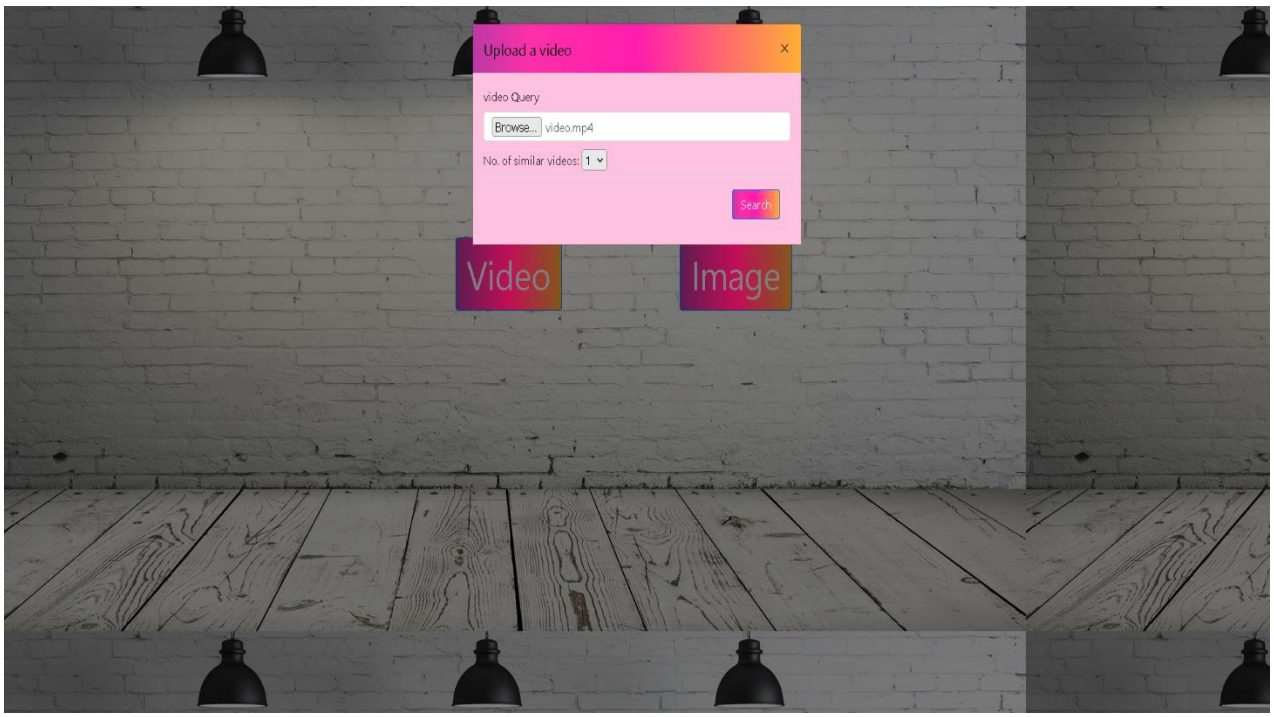


Load the image

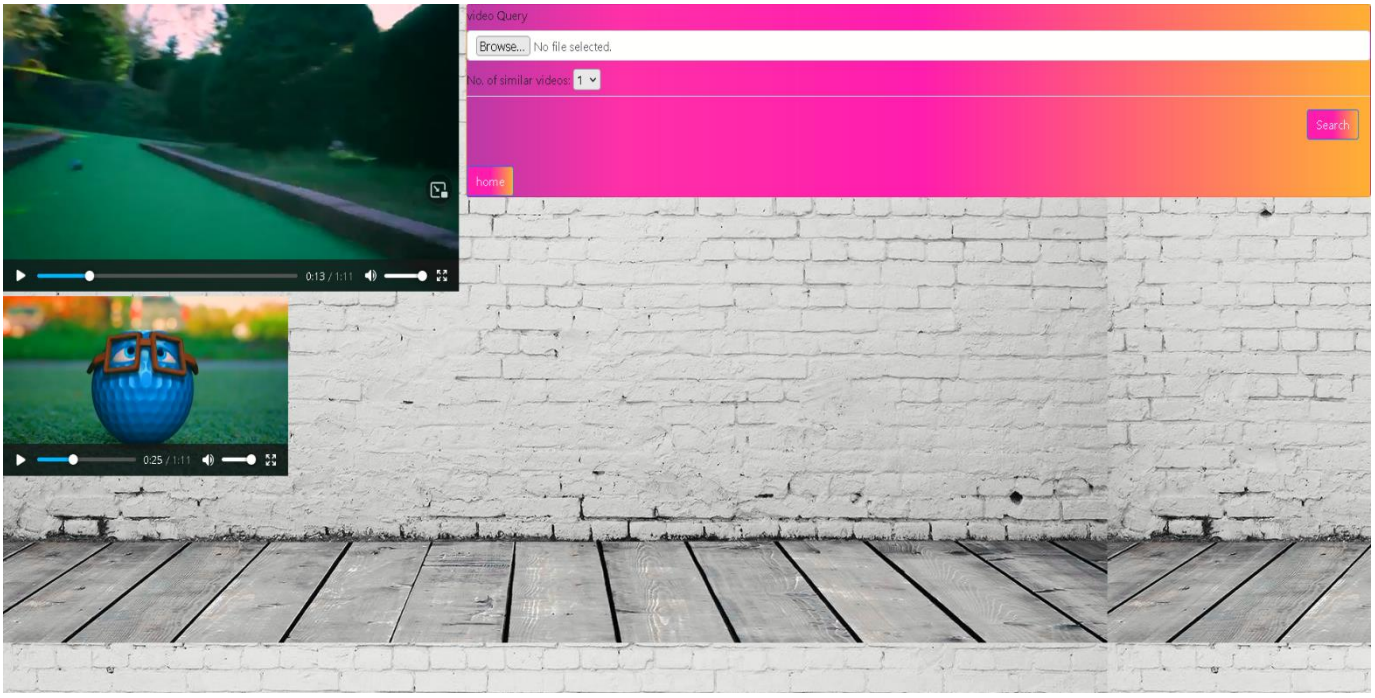


The images retrieved

- **Testing Videos Retrieval:**

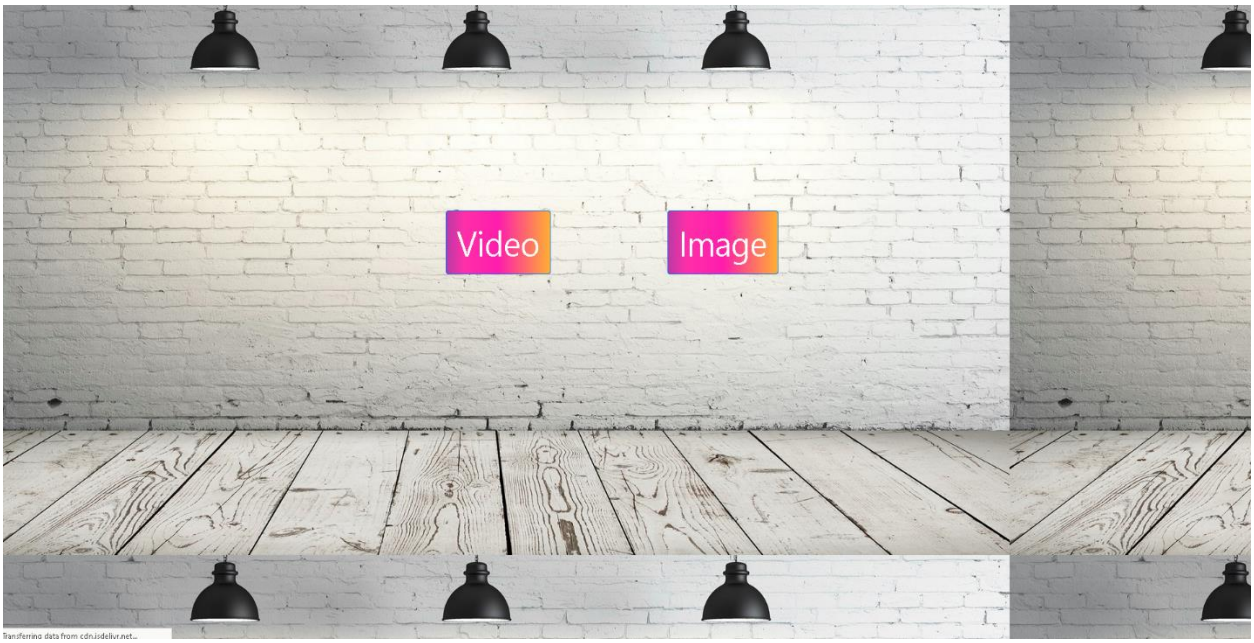


Upload The video

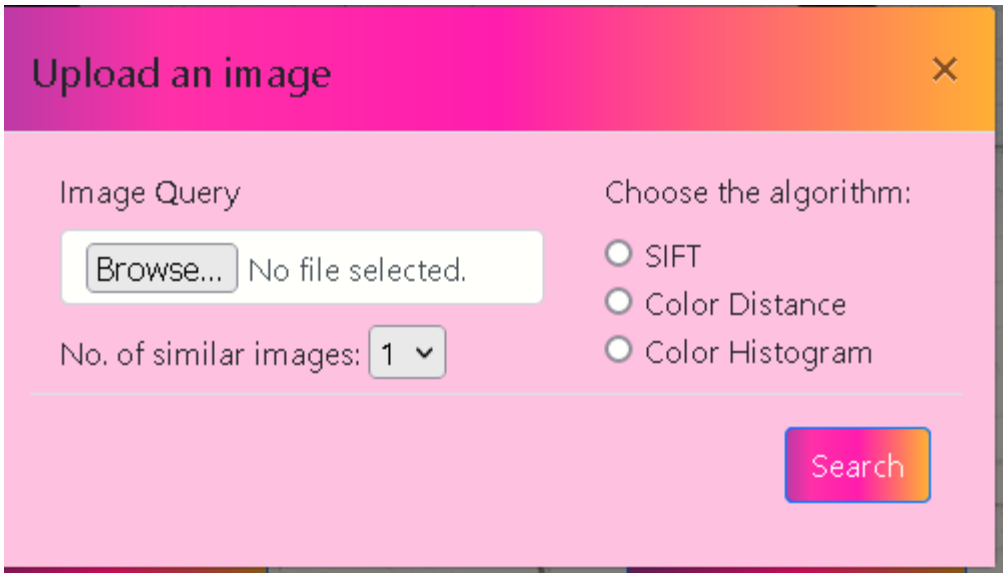


The output videos retrieved

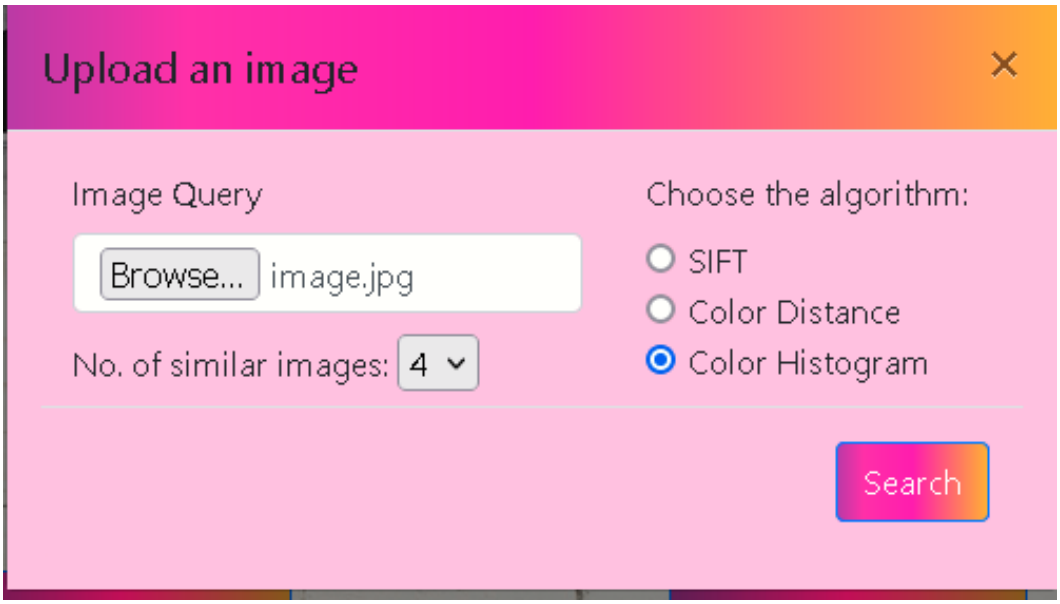
12. End User Guide



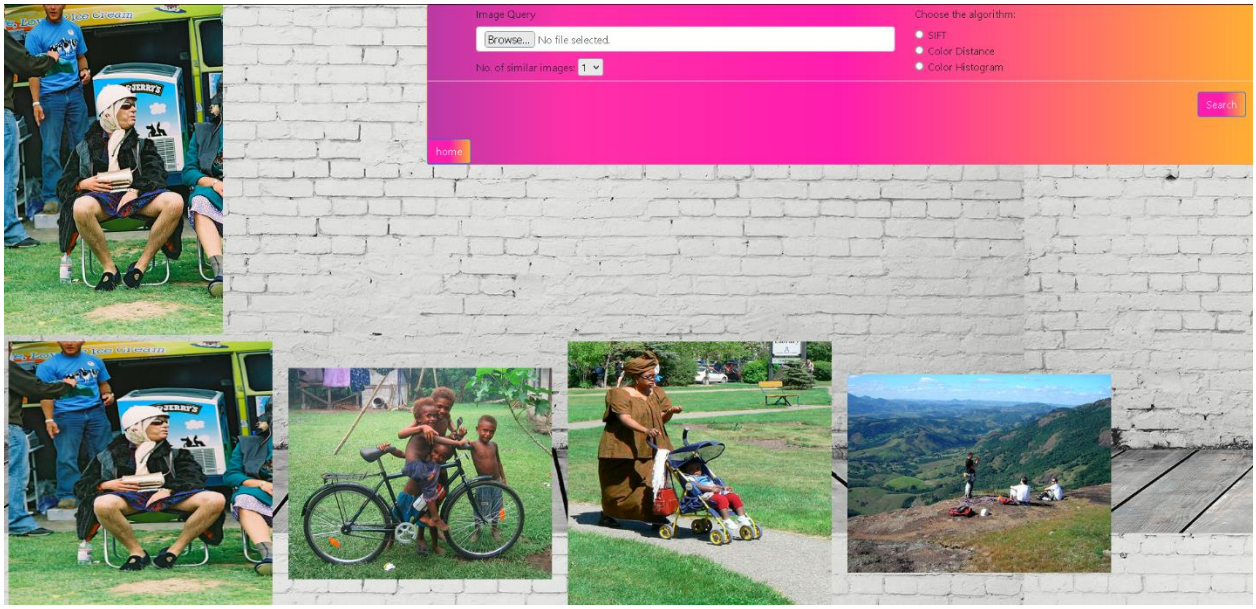
First Choose the content you want to retrieve



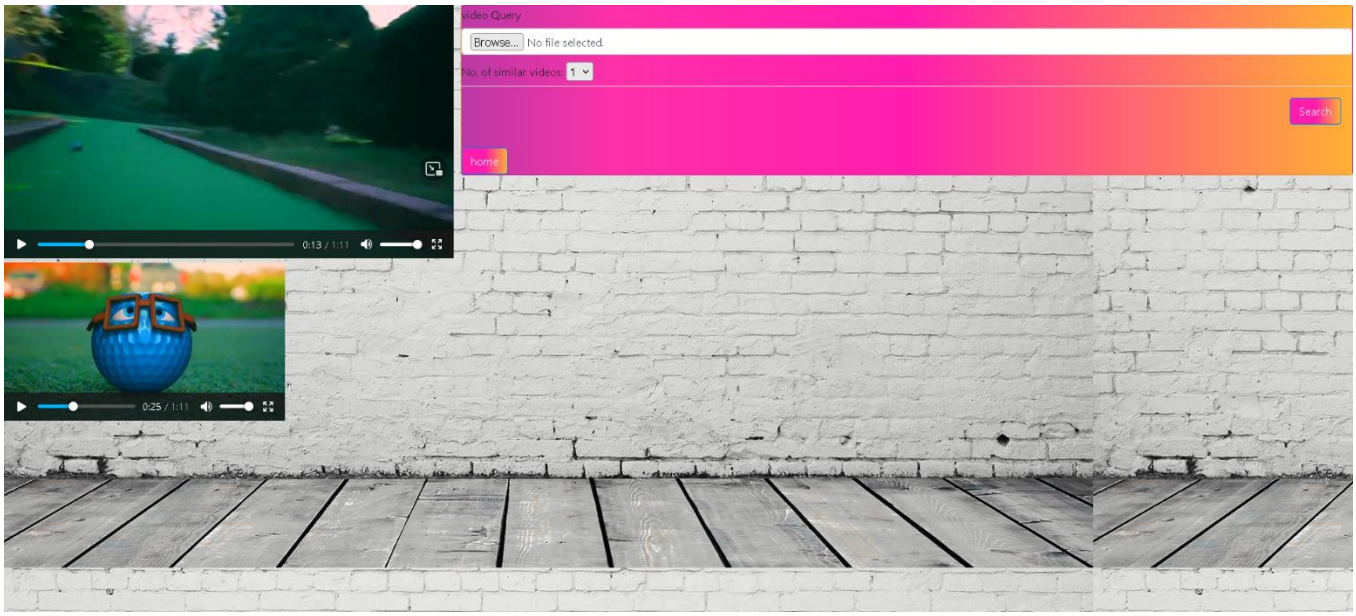
If you choose image, The system provides you 3 algorithms to choose from them and a threshold of maximum images retrieved



Here an example of image above where histogram is chosen as an algorithm and a number of 4 images to retrieve



An output of example above with an option that makes the user complete using the system as an image retrieval and also a home button is provided here to make the user go back if he wants to try video retrieval system



Here an example of Video retrieval system with only one algorithm and also maximum number of videos to retrieve