A

PROJECT REPORT

FOR

SUBJECT: LAB II- PROJECT PHASE II

ON

“Relevance ranked QnA system for live YouTube’s embedded stream on an RWA”

Submitted in partial fulfillment of the requirement for the award of

Bachelor of Engineering

In

Computer Science and Engineering

Punyashlok Ahilyadevi Holkar Solapur University

By

Name	Roll. No.	Exam Seat No.
Anjali Chougule	BE CSE-40	1724618
Janvi Pampattiwar	BE CSE-41	1724477
Pragati Phand	BE CSE-42	1724480
DurreAfshan Shaikh	BE CSE-43	1724765

Under Guidance Of

Mr. P.S.R.Patnaik

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
WALCHAND INSTITUTE OF TECHNOLOGY
SOLAPUR - 413006
(2020-2021)

Certificate

This is to certify that the project entitled

“Relevance ranked QnA system for live YouTube’s embedded stream on an RWA”

Is submitted by

Name	Roll. No.	Exam Seat No.
Anjali Chougule	BE CSE-40	1724618
Janvi Pampattiwar	BE CSE-41	1724477
Pragati Phand	BE CSE-42	1724480
DurreAfshan Shaikh	BE CSE-43	1724765

Mr.P.S.R.Patnaik
Project Guide

Dr. Mrs. A.M.Pujar
Head CSE Dept

Dr. S. A. Halkude
Principal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
WALCHAND INSTITUTE OF TECHNOLOGY
SOLAPUR - 413006
(2020-2021)

Project Approval Sheet

The Project Entitled

"Relevance ranked QnA system for live YouTube's embedded stream on an RWA"

Submitted by

Name	Roll. No.	Exam Seat No.
Anjali Chougule	BE CSE-40	1724618
Janvi Pampattiwar	BE CSE-41	1724477
Pragati Phand	BE CSE-42	1724480
DurreAfshan Shaikh	BE CSE-43	1724765

"Is hereby approved in partial fulfillment for the degree of Bachelor of Computer Science and Engineering"

Mr.P.S.R.Patnaik
Project Guide

Dr. Mrs. A.M.Pujar
Head CSE Dept

Dr. S. A. Halkude
Principal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
WALCHAND INSTITUTE OF TECHNOLOGY
SOLAPUR - 413006
(2020 - 2021)

Acknowledgment

At the outset, we would like to take this opportunity to express our deep gratitude to our guide **Mr. P. S. R. Patnaik** of the CSE Department for his guidance and moral support throughout this successful completion of our project.

We heartily thank **Dr. Mrs. A. M. Pujar**, Head of CSE Dept for her moral support and for promoting us through the completion of our project.

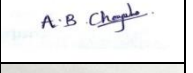
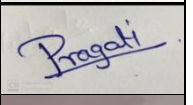
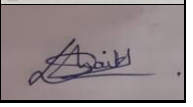
We would also like to thank our Principal **Dr. S. A. Halkude** and all staff members for their wholehearted co-operation in completing this project.

UNDERTAKING

We solemnly declare that project work presented in the report titled
"…….
Relevance ranked QnA system for live YouTube's embedded stream on an RWA
……."
is solely my project work with no significant contribution from any other person except for the project guide. Small contribution/help wherever taken has been duly acknowledged and that complete report has been written by the members of the project group.

We understand the zero-tolerance policy of the WIT, Solapur, and University towards plagiarism. Therefore, we as Authors of the above-titled report declare that no portion of the report has been plagiarized and any material used as a reference is properly referred/cited.

We undertake that if found guilty of any formal plagiarism in the above-titled report even after award of the degree, WIT, Solapur and Solapur University reserves the rights to withdraw/revoke the degree granted and that WIT, Solapur and the University have the right to publish our name on the website on which names of students are placed who submitted plagiarized report.

Name	Exam Number	University PRN Number	Signature
Anjali Chougule	1724618	2017032500246185	
Janvi Pampattiwar	1724477	2017032500244774	
Pragati Phand	1724480	2017032500244801	
DurreAfshan Shaikh	1724765	2017032500247656	

Date: 30/05/2021

Index

Sr. No.	Title		Page No.
1	**Abstract**		7
2	**Introduction**		8
	2.1	Front End	9
	2.2	Back End	10
3	**Methodology**		11
	3.1	System Architecture Overview (High-level Architecture)	11
	3.2	Low-Level Architecture	16
4	**Dependencies and Requirements**		18
	4.1	Libraries used	18
	4.2	Tools	18
	4.3	System Environment	19
5	**Instructions for Deployment**		26
6	**Summary**		28
7	**Future Scope**		29
8	**References**		30
9	**Plagiarism check**		31
10	**Student Details**		35

Abstract

Nowadays, all over the planet embedding live streaming YouTube video in our own responsive web site. Embedding videos is simply like making backlinks to your website. Like in SEO (Search Engine Optimization), embedding your videos in a very website behaves specifically sort of a back link and so facultative your videos to urge placed in program results and find a lot of views. A lot of number of views our videos receive, our video quality and complete image increase too. Not solely video quality, your product and website quality too increase so increasing your sales and profits.

Video embedding is the method of adding a video player to our web site victimisation of an internet video platform. There are several web sites that are building their own social media platforms, so it is as easy as copying and pasting a link. Video inserting works by adding associate degree embed code from your video hosting platform to the code of your web site. It permits you to integrate live streaming on our website.

Relevance ranking is to be provided for live chat of the live video stream. Relevance ranking is the process of sorting the chats so that those Questions which are most likely to be relevant to the topic are shown at the top of the chat window with the answer.

Introduction

In the real time all over the world live streaming video is embedded into their own responsive website. Since recent years have brought an increase in the popularity of video-sharing across hundreds of different platforms. This means that a number of people are sharing live and on demand videos regularly. Knowing how to broadcast live and embed live streaming video on your website is also becoming increasingly important for all kinds of broadcasters. People can not only watch live streaming video and live chat, but they can also leave blog post comments which mainly remain beyond the broadcast. People can also post supplemental information and links related to our livestream for viewers. If we stream using multiple platforms our blog is the place where fans can always find our latest broadcast.

As we know that many people make their own live streaming video. And the viewers who are watching their streaming they put or raise their question into the comment box. Since the questions which are put by the viewers are not in the relevance ranking so the host is not able to give each and every question in minimum time span.

So, we are developing our own responsive website so that we will provide a chat window which will display all the relevance ranked questions which will make the work of the host easier for answering the questions in minimum time span and also with the chat window we are also displaying the live streaming video.

Thus, this system aims to build a machine learning model that predicts the relevance of ranked questions.

Our system aims to give relevance ranked questions to the user on the topic related to streaming so that it makes the work of the host easier and also save his time from answering the same meaning questions.

To give a grouped and unique questions based on the relevance ranking, it is necessary to keep track of data such as:

- What type of data is taken from the comment box of the live streaming?
- What are the different types of questions asked by the viewer and there should be no emojis and different languages used?

Therefore, we have built a machine learning model which performs the task of giving relevance ranked questions based on the topic. So, we can build machine learning models for that. The built-in libraries used by the ML model for training are:

1] NLTK:

It is used for removing stop words.

2] Sklearn:

It is used for clustering.

3] Pytchat:

It is used for retrieving live chat from streaming.

2.1 Front End

2.1.1 HTML:

HTML is a standard markup language for creating web pages and web applications. It can be assisted by Cascading Style Sheets (CSS) and scripting languages like JavaScript. The front end of a project is built using HTML and CSS. The browser does not display the HTML tags but is utilized to build content of the page also used in plug-in development.

2.1.2 CSS:

CSS is used for styling the content of web pages including colors, layout, fonts. It will allow adapting the presentation of different CSS styles of documents written in a markup language like HTML.

2.1.3 jQuery:

jQuery is a fast, cross-platform, and feature-rich JavaScript library. Its main purpose is to provide an easy way to JavaScript on websites to make it more interactive, attractive, and also add animation. jQuery simplifies HTML document traversing, event handling, and Ajax interactions for rapid web development.

2.1.4 Bootstrap:

Bootstrap is a free front-end open-source HTML, CSS, JavaScript framework used for developing responsive web sites. It quickly designs and customizes mobile-first websites. It contains extensive prebuilt components, JavaScript plugins. It used to make the plugin front end more responsive.

2.2 Back End

2.2.1 Flask:

Flask is a popular lightweight python web application framework and baseband on the WSGI toolkit. Flask provides a simple template to build web applications. Flask can be used to save time building web applications after being imported into python. It has no database, abstraction layer, or form validation or any other components but flask supports extensions. Flask is used to build the REST API of the project with Oauth1. Oauth1 is Authentication level 1 used for authentication purposes.

Features of Flask:

- Integrated support for unit testing.
- Extensive documentation
- Unicode based

2.2.2 Python:

Pythonis a powerful, easy to learn programming language which contains high-level data structures. It can also be used to create web applications. It runs on different platforms friendly, has a simple syntax, and allows developers to write a program with fewer lines. It makes code short and versatile.

2.2.3 NumPy:

Numerical Python (NumPy) is a library consisting of multidimensional array objects and collections of routines used to manipulate those arrays. NumPy is the most fundamental and effective package deal for running with facts in python. It is handy for mathematical and logical operations and manipulation of arrays. And also used to perform operations on data.

2.2.4 PHP:

Personal Home Page (PHP) is a server-side scripting language used to develop static or dynamic web pages. PHP code may be embedded with HTML code. Used in combination with various web templates, web frameworks. PHP inbuilt has support for working hand in hand with MySQL. PHP is a cross-platform scripting language so it can work on different operating systems. Used for WordPress and WooCommerce Plugin development.

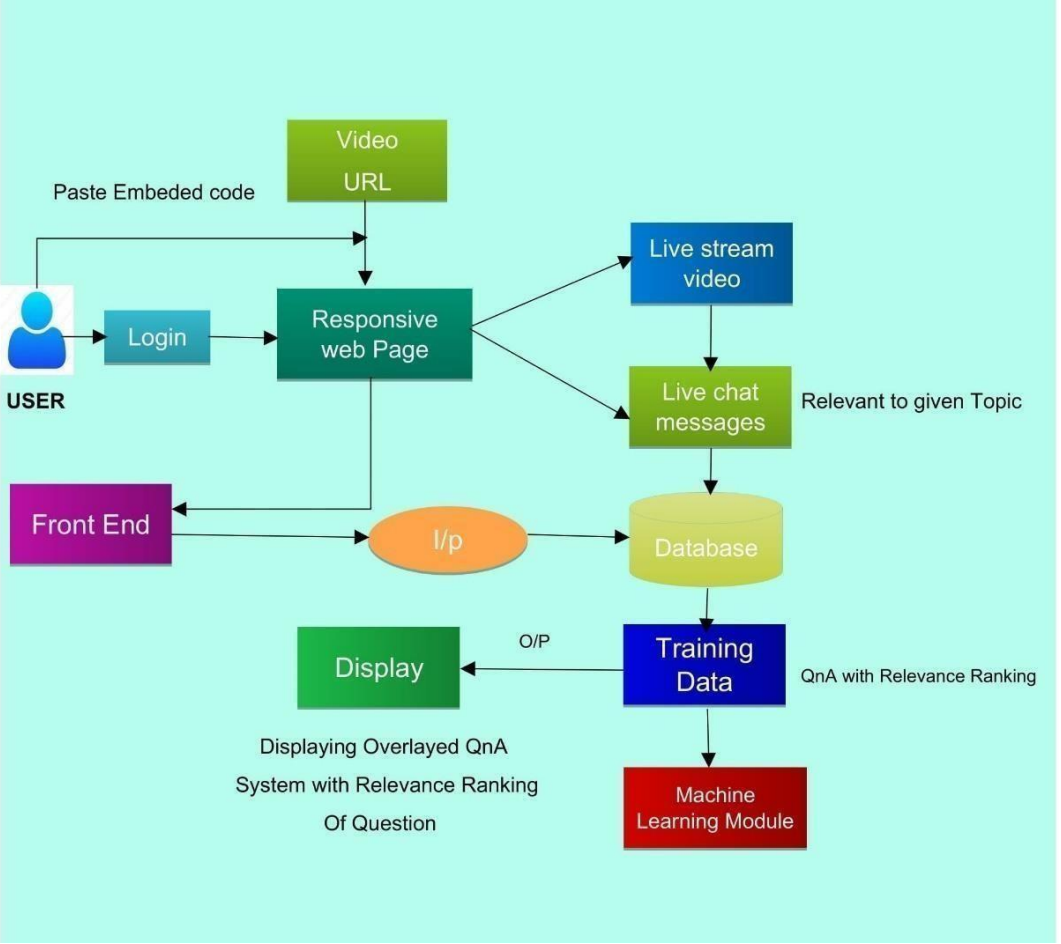
2.2.5 firebase(database):

Store and sync data with our NoSQL cloud database. Data is synced across all clients in real time, and remains available when your app goes offline.

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in real-time to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

Methodology

3.1 System Architecture Overview (High-level Architecture):



3.1.1 Data Collection:

Data should be collected from YouTube's live chat for pre-processing. The messages are collected from YouTube in systematic manner.

3.1.2 Data Pre-processing:

Real-world data is often incomplete, inconsistent, and lacking in certain behaviour trends and likely contains many errors. Data processing includes transforming raw data into an understandable format.

3.1.3 Prepared Data:

After data collection and data pre-processing the prepared data saved in the required format.

3.1.4 Splitting Data:

Partition of data into training data and test data. Splitting training data in
Training data and validation data.

3.1.5 Machine Learning Model:

ML model built using the ML algorithm. The different algorithms are suitable to solve the problems used and a model trained on the training dataset using that algorithm.

The algorithm giving a good performance is chosen.

3.1.6 NLTK (Natural Language ToolKit):

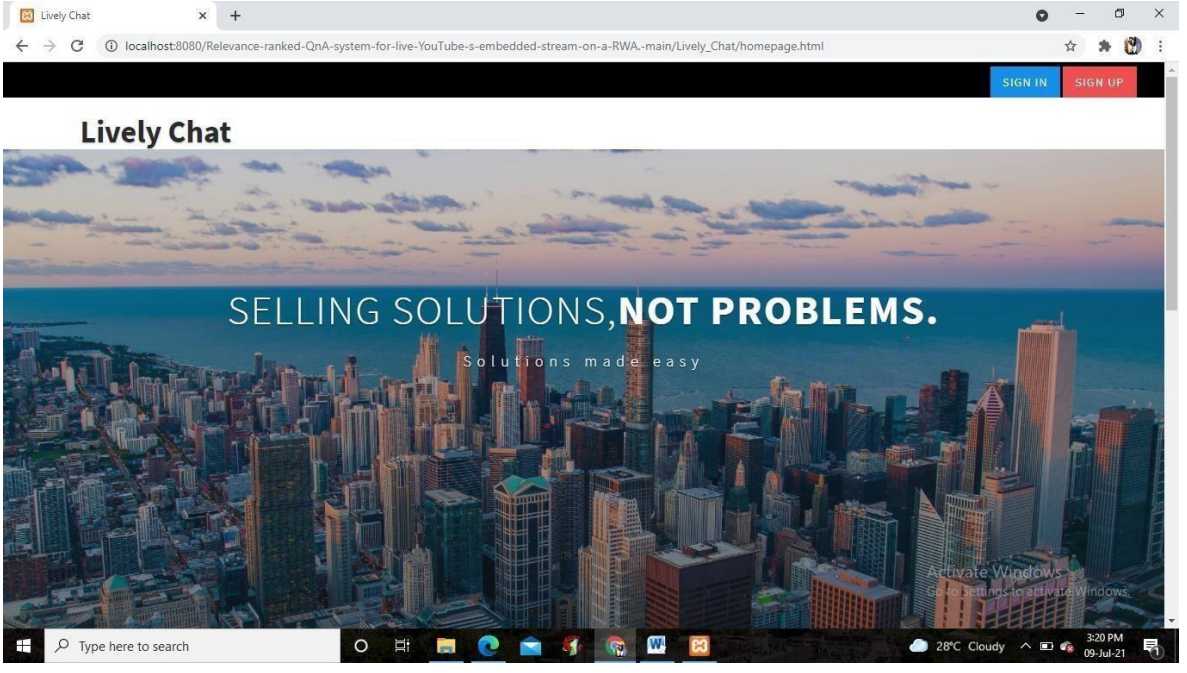
This library has been used in our project for removing stop words from the questions which have been retrieved from live chat. Stop words refers to the most common words that appear in our language. We see them often and are redundant, so it provides no real information. Hence, we remove them completely.

3.1.7 Ski kit learn (Sklearn):

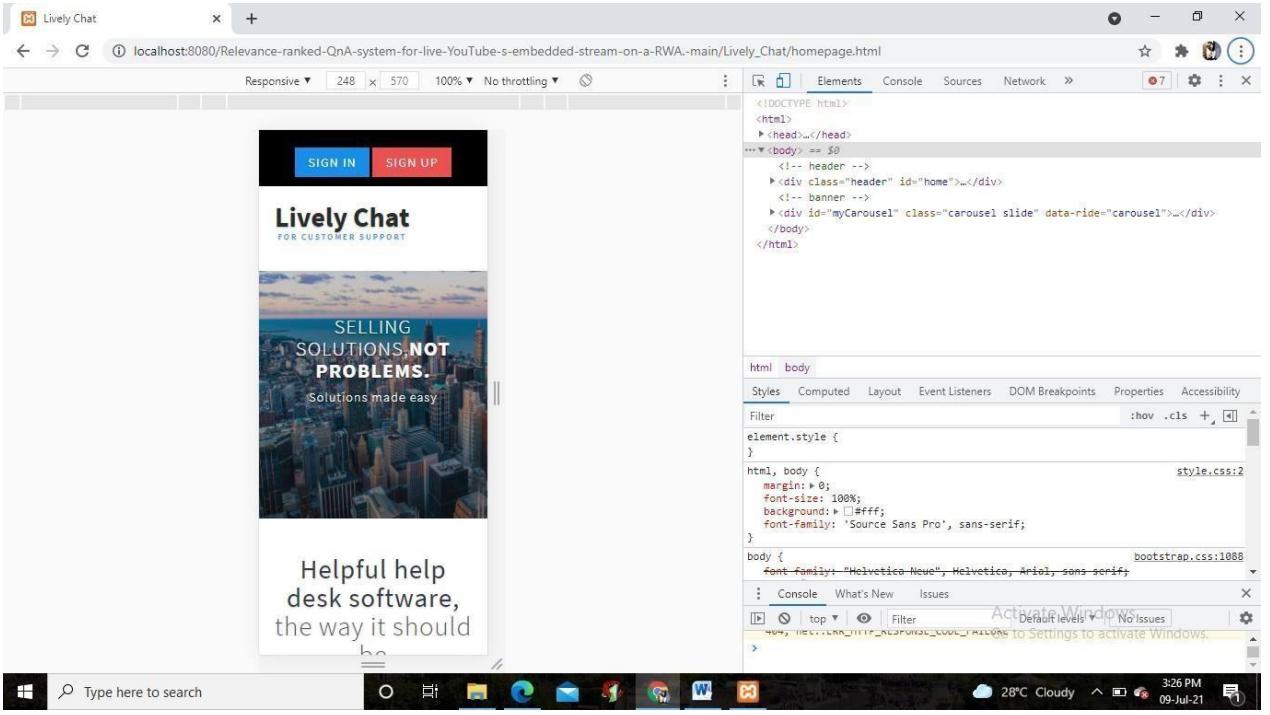
This library has been used in our project for clustering. For clustering we have K-mean clustering algorithm. The k-means clustering algorithm is an unsupervised clustering algorithm which determines the optimal number of clusters using the elbow method. It works iteratively by selecting a random coordinate of the cluster center and assign the data points to a cluster.

3.2. Relevance-ranked-QnA-system Web Application:

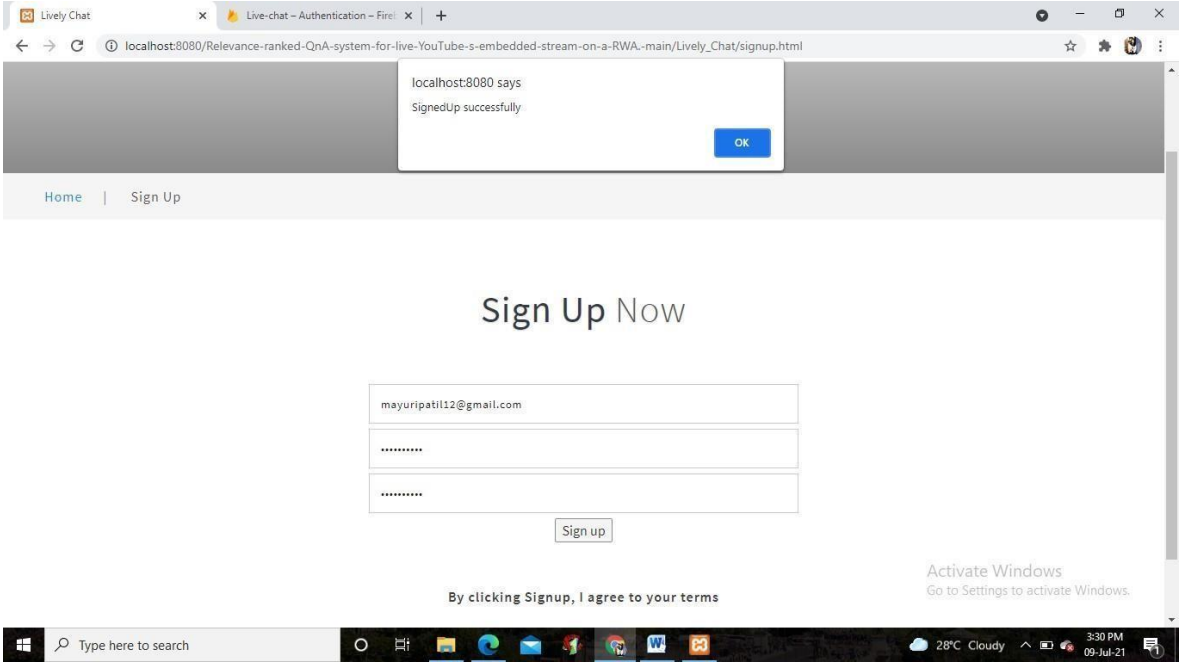
3.2.1 Homepage



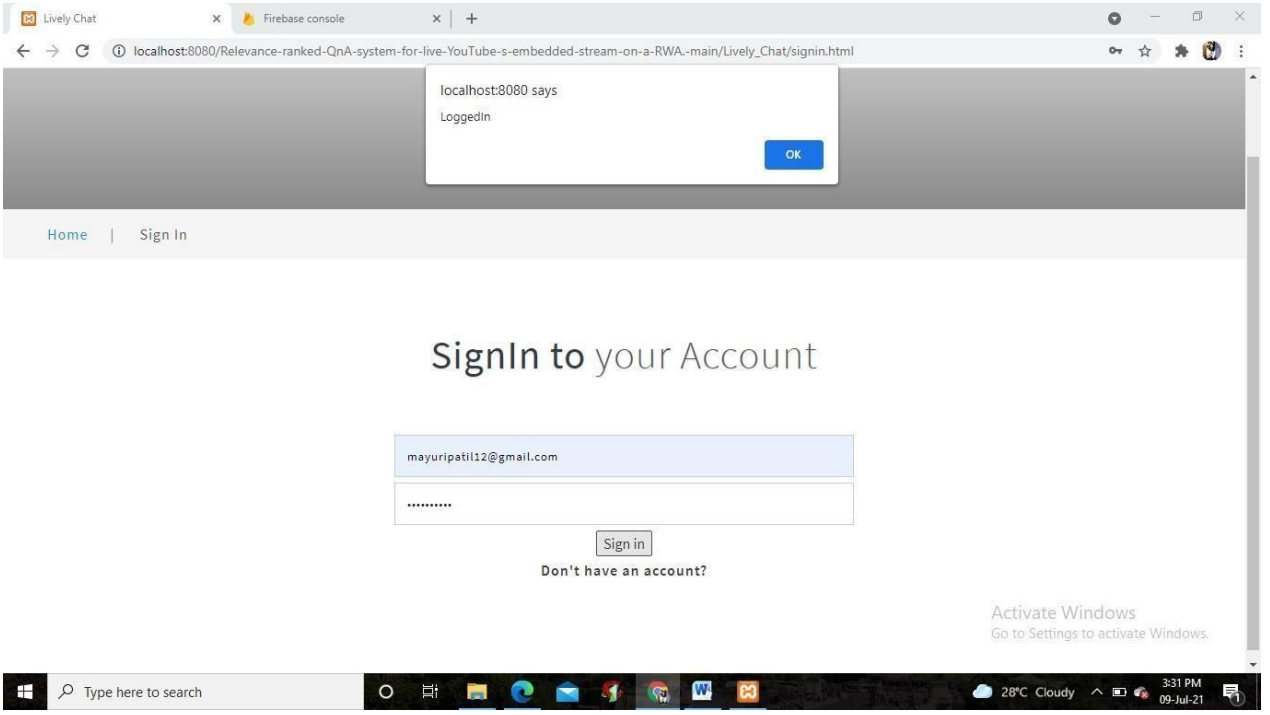
3.2.2 Responsive Web Application



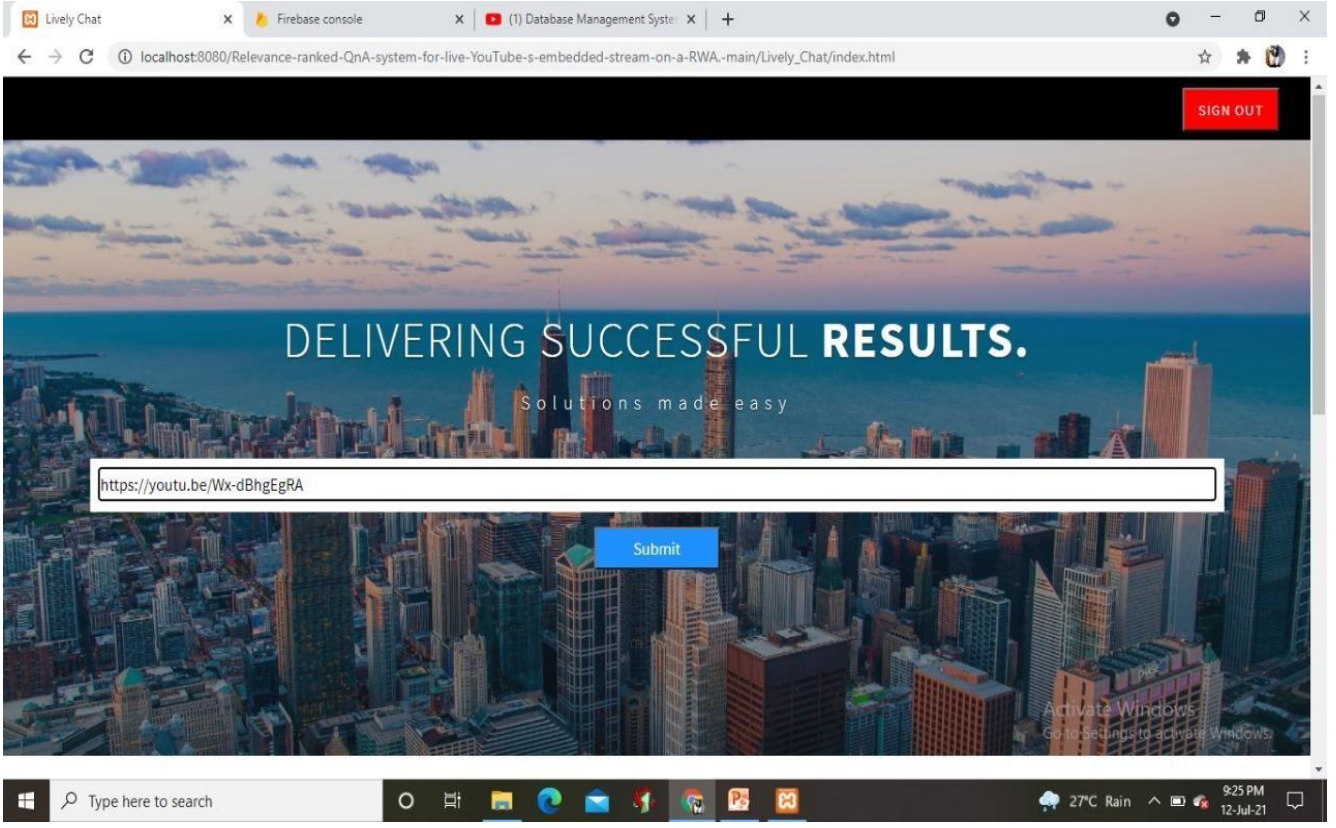
3.2.3 Sign up form

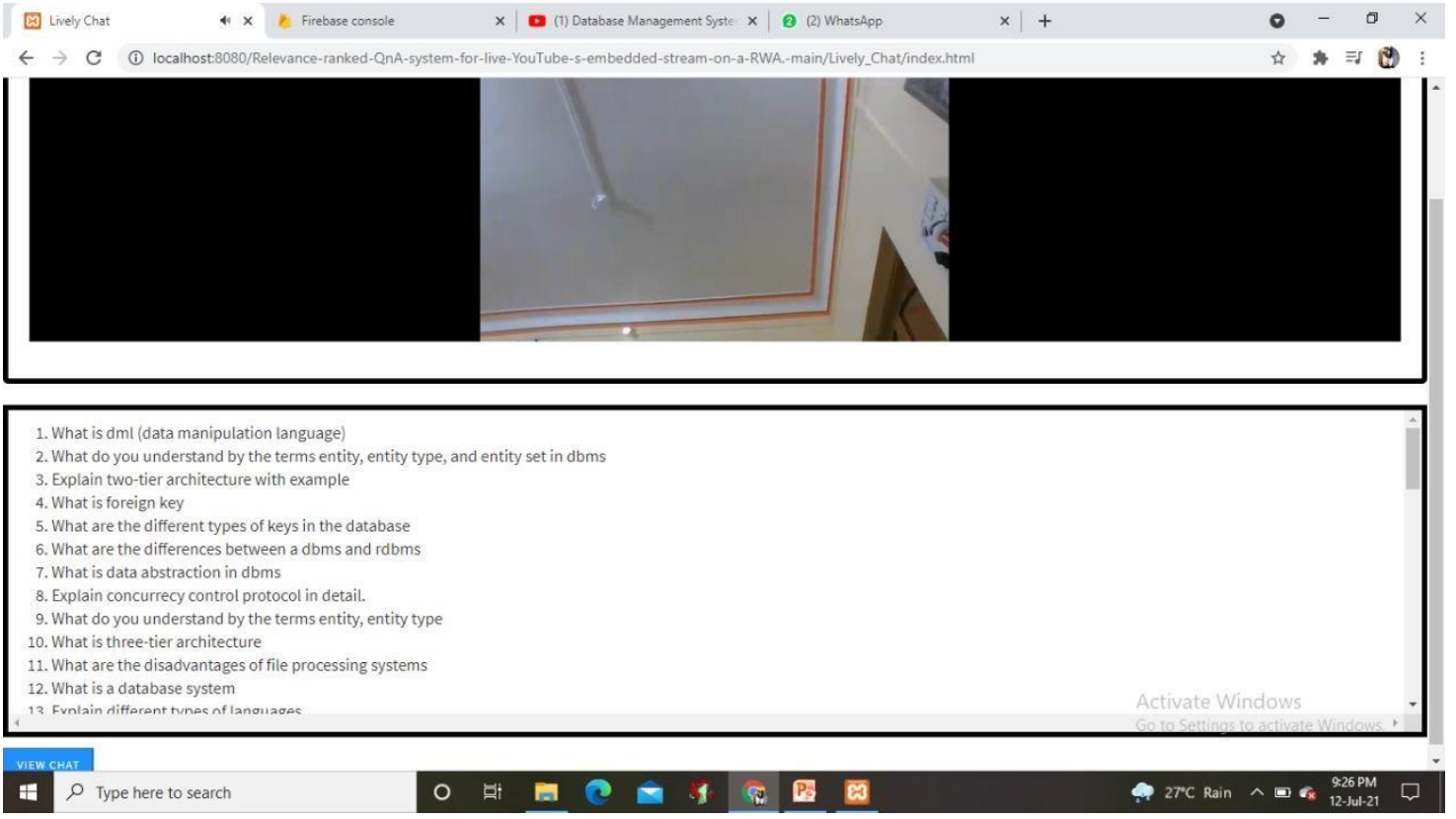


3.2.4 Sign in form



3.2.5 Main page:




Lively Chat
Firebase console
(1) Database Management Syste
(2) WhatsApp
localhost:8080/Relevance-ranked-QnA-system-for-live-YouTube-s-embedded-stream-on-a-RWA.-main/Lively_Chat/index.html
1. What is dml (data manipulation language)
2. What do you understand by the terms entity, entity type, and entity set in dbms
3. Explain two-tier architecture with example
4. What is foreign key
5. What are the different types of keys in the database
6. What are the differences between a dbms and rdbms
7. What is data abstraction in dbms
8. Explain concurrecy control protocol in detail.
9. What do you understand by the terms entity, entity type
10. What is three-tier architecture
11. What are the disadvantages of file processing systems
12. What is a database system
13. Explain different types of languages
Activate Windows
Go to Settings to activate Windows.
VIEW CHAT
Type here to search
27°C Rain
9:26 PM
12-Jul-21

3.2 Low-Level Architecture

3.2.1 Machine Learning Model

3.2.1.1 Data Exploration:

As the data required to train machine learning module is live chat fetched from YouTube stream. This fetched data consists of the emoji's, symbols, notations and so on. To improve the model efficiency, we need to remove all these unwanted data. Also, there is need to remove the stop words like and, or, if, of, etc which may change the meaning of the sentences.

3.2.1.2 Data Pre-processing:

The data provided to machine learning model is chats fetched from live YouTube video. This fetched data contains unwanted symbols, emoji's, notations and stop words which make model less efficient. To remove these unwanted data, we have used the machine learning algorithm.

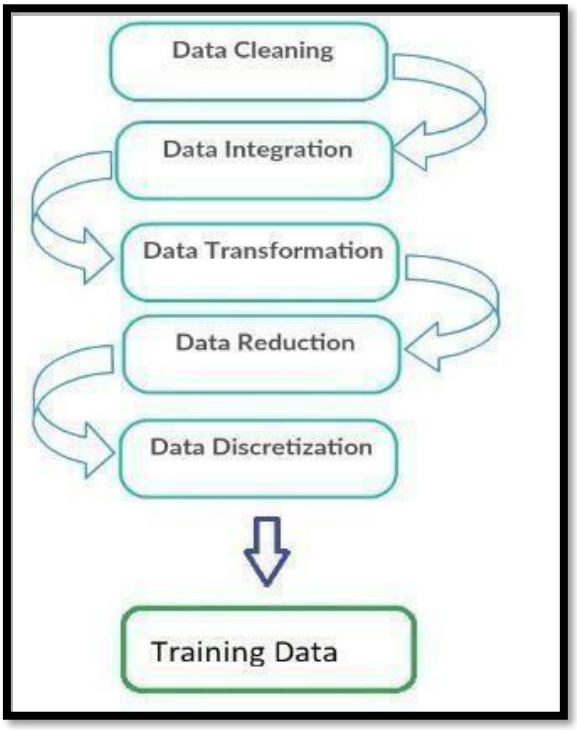


Fig. Data Pre-processing

3.2.1.3 Model Building:

Starting with model building, pytchat library is used for data fetching and different algorithms are applied over data and checked for accuracy of each. Different algorithms are considered and applied over it and checked for accuracy.

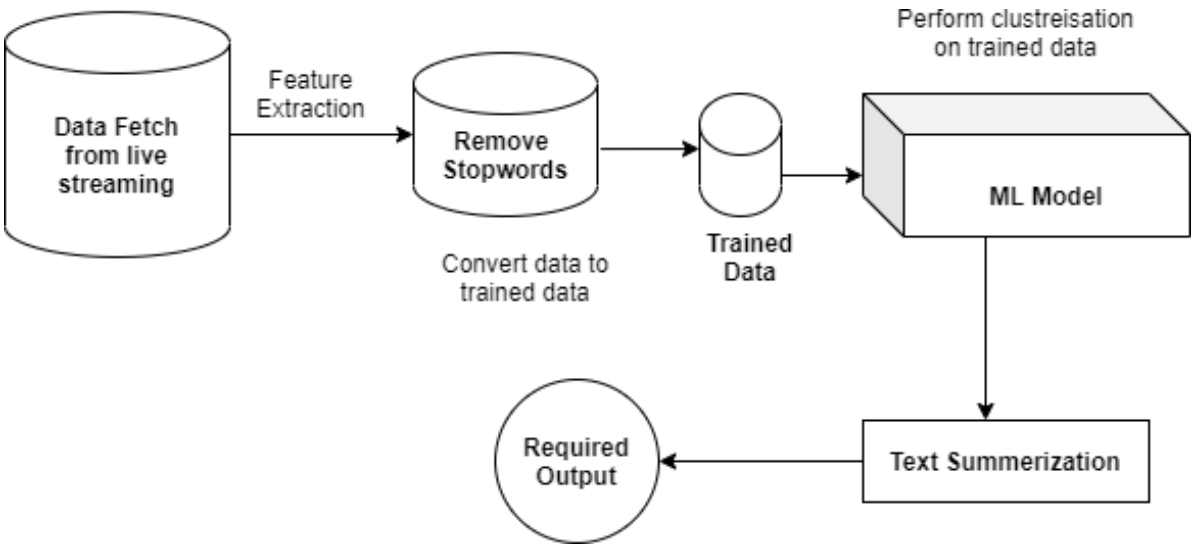


Fig. ML Model Overview

3.2.1.4 Accuracy

The data passed to machine learning model are chats after removing stop words. It gives clusters of similar data which are most closely related to each other. The accuracy achieved by the machine learning model is 82.72%.

3.2.1.5 Saving ML Model:

As once done with one-time training of the ML model and getting proper accuracy from it then it requires saving the ML model to use it.

3.2.2. Pytchat:

Pytchat library is used for retrieving data from live you tube stream.

3.2.3. Clusterization:

For clustering we have K-mean clustering algorithm. The k-means clustering algorithm is an unsupervised clustering algorithm which determines the optimal number of clusters using the elbow method. It works iteratively by selecting a random coordinate of the cluster center and assign the data points to a cluster.

3.2.4. Text Summarization:

After performing clustarization on live chat text summarization is used to summarize the organized data.

Dependencies and Requirements

4.1 Libraries used:

4.1.1 flask 1.1.2:

Flask is a lightweight WGSI (Web Server Gateway Interface) web application framework. It began as a simple wrapper around werkzeug and jinja and has become one of the most popular Python web application frameworks
License: BSD-3-Clause

4.1.2 numpy 1.18.1:

A fundamental package for array computing in python
License: OSI Approved (BSD)

4.1.3 scikit-learn 0.22.1:

Scikit-learn is a free software machine learning library for python programming. License: BSD 3-Clause

4.1.4 NLTK (Natural Language ToolKit):

NLTK is a leading platform for building Python programs to work with human language data.

4.1.5 requests 2.22.0:

Python HTTP for Humans.
License: Apache 2.0

4.1.6 TensorFlow 1.13.0:

TensorFlow is an open-source machine learning framework for everyone.
License: Apache 2.0

4.2 Tools:

- **IDE:**
  - Google Colaboratory.
  - Spider for the ML model
  - Visual Studio Code and Atom for plugin
- **Analysis:**
  - K-means algorithm for clustering.
- **Testing:**
  - Insomnia for testing Flask API.

4.3 System Environment:

- **Processor:**
  - 2.5 gigahertz (GHz) or faster processor.
- **RAM:**
  - 8 GB or more
- **Hard drive space:**
  - 48 GB for 64-bit OS or Higher
- **Operating Systems:**
  - Linux 18.04 or Higher
  - Windows 10
- **GPU:**
  - NVIDIA GTX 1050(4 GB) Compute Capability 3.5 or higher.
- **Language:**
  - Python 3.6.
- **Tool:**
  - Anaconda 3-5.2.0-Linux.
  - Anaconda3-5.2.0-Windows-x86_64.
  - Firebase connection.
- **Internet Connection:**
  - Internet connectivity is necessary to download some Libraries. Internet connection required during the training of the ML model.

Approach for relevance ranking :

Ranking questions are the question type aimed at getting respondents to order a list of answers into ranked order , providing quantitative research data. This question type allows respondents to identify which objects are **most** and **least** preferred.

Ranking questions compare individual elements to each other and rank them accordingly.

Firstly we retrieved live chat from YouTube live streaming and to find the relevant ranking of live chats we performed we performed **clusterization** and **text summarization** on the input live chat.

```
def accessingLiveChat(link):

  #link='https://youtu.be/AYsolQwS-lk'
  chat = LiveChat(video_id = link[17:])
  chat_list,count=[],0
  print('Stream started')
  while chat.is_alive():
    try:
      #print('----nside try block')
      data = chat.get()
      items = data.items
      for c in items:
        chat_list.append((str(f"{c.message}")+'?').lower())
        #print(chat_list[::-1][0])
        #fil_list.append(str(removeStopWords(chat_list[::-1][0])).lower())
        if count>20:
          makeClusters(chat_list)
        time.sleep(0)
        count+=1
      if not chat.is_alive():
        print('Live Stream Ended!!!')
        break
    except Exception:
      chat.terminate()
```

Steps for performing clusterization and text summarization on input data are as follows:

1. Clusterization:

"Clustering is an unsupervised learning algorithm which is used to cluster data into k groups without actually knowing which cluster the data belongs to."
For performing clustering on input data we have used the **K-means algorithm** over agglomerative hierarchical clustering. The main disadvantage of hierarchical clustering is that this algorithm can never undo what was done previously. This means that the input data will not form clusters on the basis of previous clusters which will not form one cluster for the same type of input data.
So instead of hierarchical clustering we have used the k-means algorithm.The advantage of k-means is that if variables are huge then it is faster than hierarchical clustering. This k-means helps to produce tighter clusters if the clusters are globular. Firstly, the k-means algorithm specifies no. of clusters k. Initialize centroids by first shuffling the dataset and then randomly selecting k data points for the centroids without replacement. Then keep iterating until there is no change to centroids.

After deciding K-means algorithm for clusterization. We will remove stopwords from the input data.**"Stopwords are the most common words used in any natural language.Since it provides no real information thus, we remove them completely."**
To perform the K-means algorithm for forming clusters we have removed stopward from input questions using **nltk** library. After successful removal of stopwords We have to fix the value of 'k' for formation of clusters.

Code for removal of stopwords is:

```
[ ]  def removeStopWords(recent_chat):
       stop_words = set(stopwords.words('english')+['?','what','how'])

       word_tokens = word_tokenize(recent_chat)

       filtered_sentence = [w for w in word_tokens if not w in stop_words]

       filtered_sentence = []

       for w in word_tokens:
           if w not in stop_words:
               filtered_sentence.append(w)
       #print(filtered_sentence)
       return filtered_sentence
```

The clusterization of questions is done on the basis of most common words in each questions for each cluster.This means that for grouping same type of questions we have to first fix the 'k' for performing K-means algorithm to form clusters having same keywords in input questions.So, we have tried many values of 'k' to make clusters of same meaning of questions to be in one cluster only.And thus, we get the fix value of 'k' which give us efficient result of forming same meaning of input questions in one cluster.

The clusters are mainly formed in higher and lower priority manner.The clusters are made using the keyword found in input sentences and also it depends on the number of same questions asked by the user. So, if the no of questions asked by the user having same keyword in the cluster than other cluster will have higher priority.And thus when the one cluster ranked lower than the other depends on the no of questions in that clusters are not having same keyword then it will have lower priority.

Hence, the ranking of questions depends on the questions based on keywords which shows the relevance of two or more sentences and no of questions asked by the user.

For Eg:
Suppose we are performing a live stream on “Database” then the user will put their questions in the comment box. After that we will retrieve live chat from YouTube and perform clusterization and text summarization on input data so that we can recognize higher and lower preferences among them.
Some of the questions asked by the user which form clusters are as follows:

```
li=['What is database',
    'what is database management system',
    'what is database management',
    'tell me about database management',
    'what are keys in database?',
    'what is primary key',
    'what is primary key in dbms',
    'what is foriengn key',
    'where is foreign key is used',
    'why foreign key is used'
  ]
makeClusters(li)
print()
```

The clusters formed after these input questions retrieved from the YouTube are:

```
What is database
----------------------------------------------------------------------------------------------------
Where is foreign key is used
Why foreign key is used
What is foriengn key
----------------------------------------------------------------------------------------------------
What are keys in database?
What is primary key
What is primary key in dbms
----------------------------------------------------------------------------------------------------
What is database management system
What is database management
Tell me about database management
----------------------------------------------------------------------------------------------------
```

After the formation of clusters text summarization is performed which shows relevance rank of questions.

```
 Tell me about database management
 what is foreign key

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py
```

2. Text Summarization:

"Text summarization refers to the technique of shortening long pieces of text. The intention is to create a coherent and fluent summary having only the main points outlined in the document."
NLTK library is used for text summarization.There are 5 steps followed for performing text summarization are:

1. **Create word frequency table:**
   We have created a dictionary for the word frequency table from the text.
   For this, we only use the words that are not part of the stopwords array.

```
[ ]  def _create_frequency_table(text_string) -> dict:

        stopWords = set(stopwords.words("english"))
        words = word_tokenize(text_string)
        ps = PorterStemmer()

        freqTable = dict()
        for word in words:
            word = ps.stem(word)
            if word in stopWords:
                continue
            if word in freqTable:
                freqTable[word] += 1
            else:
                freqTable[word] = 1

        return freqTable
```

2. **Tokenize the sentences:**
   After creating a frequency table we split the text_string in a set of sentences. For this, we have used the inbuilt method from the nltk.

```
# 2 Tokenize the sentences
sentences = sent_tokenize(text)
```

3. **Score the sentences: Term frequency**
   We're using the Term Frequency method to score each sentence.
   Basic Algorithm: score a sentence by its words, adding the frequency of every non-stop word in a sentence.

```
def _score_sentences(sentences, freqTable) -> dict:
    sentenceValue = dict()

    for sentence in sentences:
        word_count_in_sentence = (len(word_tokenize(sentence)))
        for wordValue in freqTable:
            if wordValue in sentence.lower():
                if sentence[:10] in sentenceValue:
                    sentenceValue[sentence[:10]] += freqTable[wordValue]
                else:
                    sentenceValue[sentence[:10]] = freqTable[wordValue]

        sentenceValue[sentence[:10]] = sentenceValue[sentence[:10]] // word_count_in_sentence

    return sentenceValue
```

4. **Find the threshold:**

Here, we are considering the average score of the sentences as a threshold. We can also use other methods to calculate the threshold.

```
def _find_average_score(sentenceValue) -> int:
    sumValues = 0
    for entry in sentenceValue:
        sumValues += sentenceValue[entry]

    # Average value of a sentence from original text
    average = int(sumValues / len(sentenceValue))

    return average
```

5. **Generate the summary:**

Basic Algorithm: Select a sentence for a summarization, If the sentence score is more than the average score.

```
def _generate_summary(sentences, sentenceValue, threshold):
    sentence_count = 0
    summary = ''

    for sentence in sentences:
        if sentence[:10] in sentenceValue and sentenceValue[sentence[:10]] > (threshold):
            summary += " " + sentence
            sentence_count += 1

    return summary
```

Limitations:

Our project has some limitations where this model gives less efficiency.The limitations are when the input data consists of emoji's after performing clusterisation on this emoji's this will reduce efficiency of our model.
When the data input by user is in another language rather than english this will cause problems with our model since we remove stopwords from input questions.Thus, this can also reduce efficiency of our model.
And if the input data consists of short form then also it will reduce efficiency since stopword method is not applicable on short form.

Thus, these are some limitations of our project which occurred while outputting relevance ranked questions.

Instructions for Deployment

Step 1: Download and install Visual Studio Code.

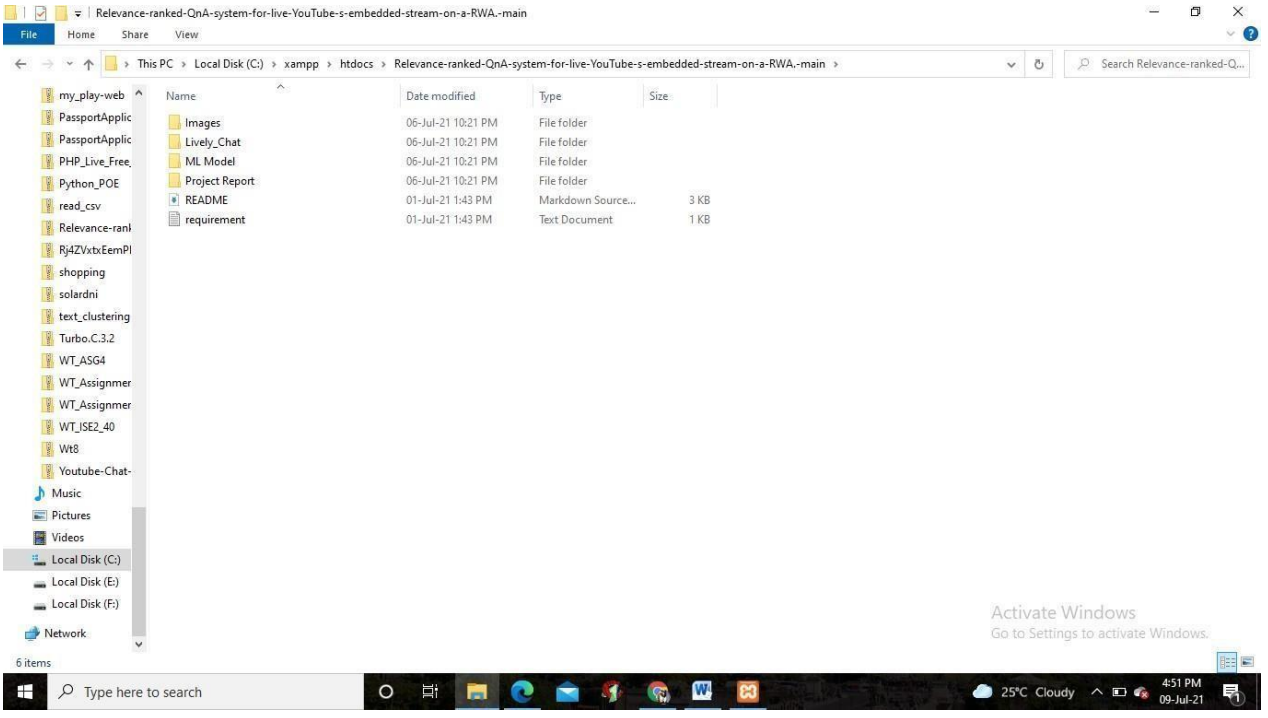
Step 2: Download and install higher version of xampp server.

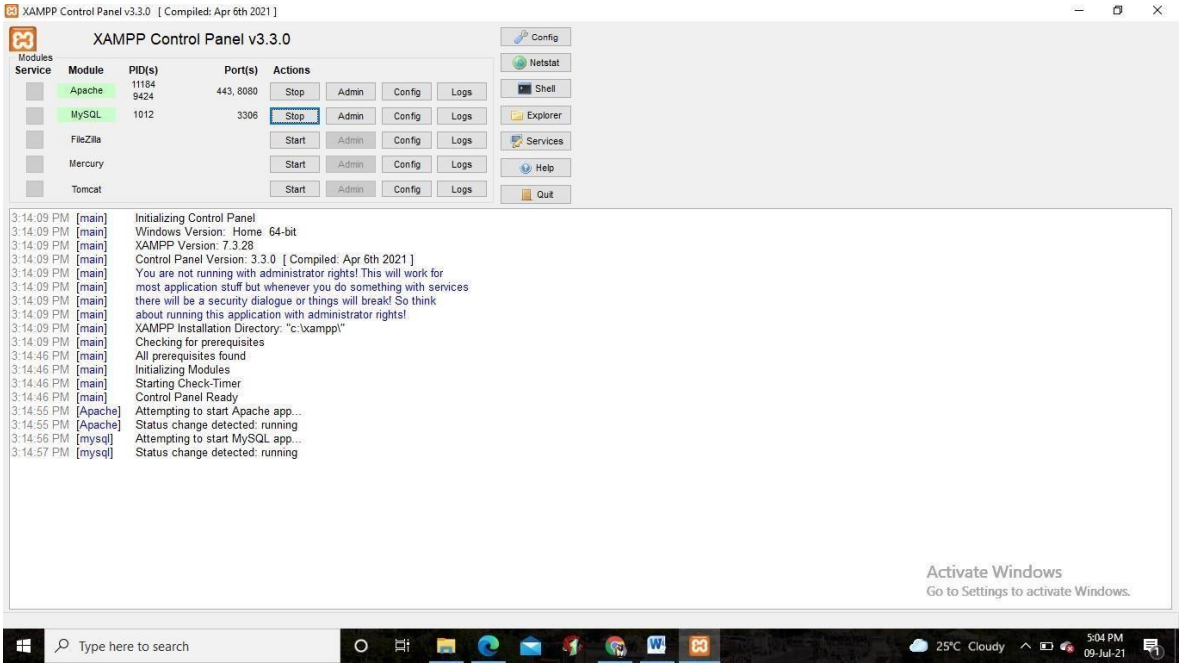
Step 3: Create a Virtual Environment and activate Virtual Environment

Step 4: Open directory where requirement.txt file located and run “pip install -r requirement.txt”. It will install all required packages for the project.

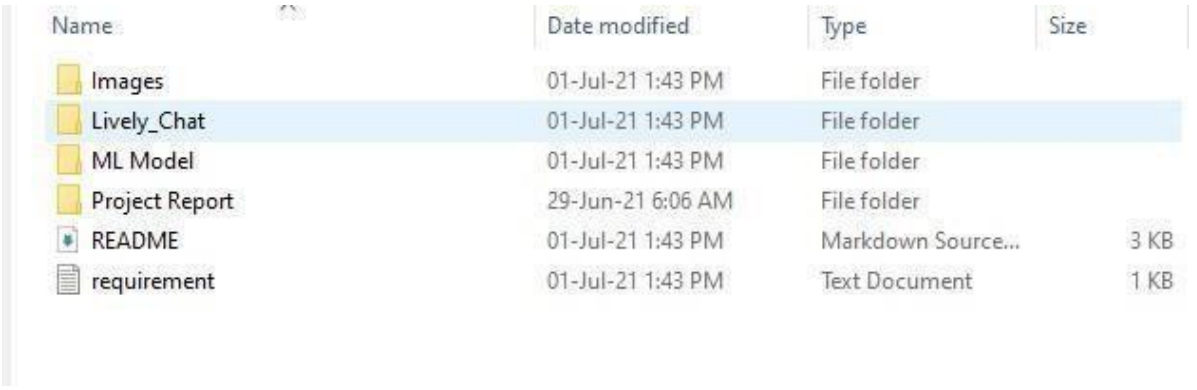
Step 5: Clone project repo to a file.

Step 6: Copy the folder to xampp/htdocs and start xampp server

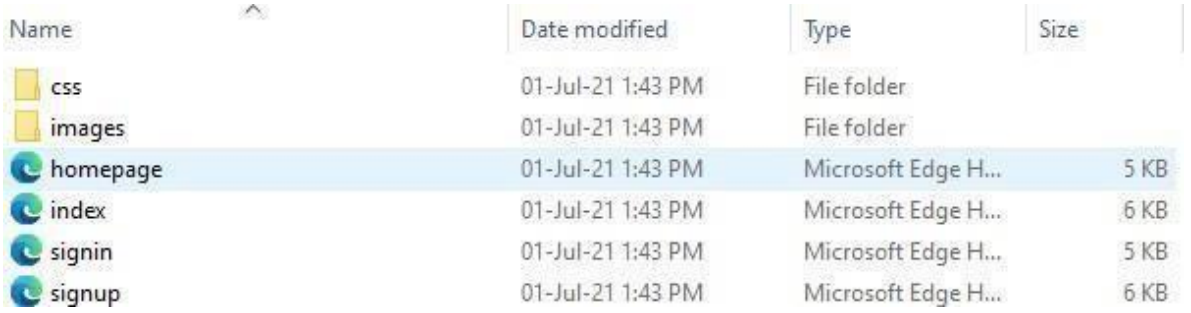




Step 7: Open the folder which contains the html and CSS files.



Name	Date modified	Type	Size
Images	01-Jul-21 1:43 PM	File folder	
Lively_Chat	01-Jul-21 1:43 PM	File folder	
ML Model	01-Jul-21 1:43 PM	File folder	
Project Report	29-Jun-21 6:06 AM	File folder	
README	01-Jul-21 1:43 PM	Markdown Source...	3 KB
requirement	01-Jul-21 1:43 PM	Text Document	1 KB



Name	Date modified	Type	Size
css	01-Jul-21 1:43 PM	File folder	
images	01-Jul-21 1:43 PM	File folder	
homepage	01-Jul-21 1:43 PM	Microsoft Edge H...	5 KB
index	01-Jul-21 1:43 PM	Microsoft Edge H...	6 KB
signin	01-Jul-21 1:43 PM	Microsoft Edge H...	5 KB
signup	01-Jul-21 1:43 PM	Microsoft Edge H...	6 KB

Step 8: Go to browser and type localhost://Relevance-ranked-QnA-system-for-live-YouTube-s-embedded-stream-on-a-RWA.-main

Summary

We have developed our own responsive website so that we will provide a chat window which will display all the relevance ranked questions which will make the work of the host easier for answering the questions in minimum time span and also with the chat window we are also displaying the live streaming video.

After finding many options we get pytchat library for retrieving the live chat of a live YouTube stream. The live chat is retrieved using python using pytchat library and this data does not give relevance rank so we need to put those questions with ranking.

With the help of NLTK (Natural Language Toolkit) we have removed stop words from it so that the most common words that appear in our language are removed. After that we have used Sklearn so that we can perform clustering and text summarization on that data after removal of stop words which we retrieved using pytchat and got the relevance ranked questions which are displayed in the web page which will be easier for the host to answer.

Thus, our website provides the relevance ranked questions based on the topic related to live streaming video.

Future Scope

As future work on identifying new features and related data to improve the performance of the ML model and increase the complexity of the ML model.

To extract chat with emoji and apply stemming and summarization on it.

Automatic QnA system on generated data.

To develop a mobile app version for this website.

References

[1] ”Embedding live stream on website”. [Online].Available: https://www.dacast.com/blog/how-to-embed-streaming-video-to-your-website/ [Accessed: 9-July-2021]

[2] ”Convert YouTube URL into embed code”. [Online].Available: https://stackoverflow.com/questions/21607808/convert-a-youtube-video-url-to-embed-code/21607897; [Accessed: 9-July-2021]

[3] ”Research of key approaches to responsive website development and their practical application”. [Online]. Available: https://www.researchgate.net/publication/311960073_Research_of_key_approaches_to_responsive_website_development_and_their_practical_application [Accessed: 9-July-2021]

[4] ”Responsive Web mode and application- IEEE Explore”. [Online]. Available: https://ieeexplore.ieee.org/document/6976522 [Accessed: 9-July-2021]

[5] ”Detailed study of clustering algorithm- IEEE Explore”. [Online]. Available: https://ieeexplore.ieee.org/document/8342454 [Accessed: 9-July-2021]

[6] ”Data Clustering K-means Algorithm- ReasearchGate”. [Online]. Available: https://www.researchgate.net/publication/261355115_Review_based_on_data_clustering_algorithms [Accessed: 9-July-2021]

[7] ”Making Sense of Text Clustering-Ignasius Harvey Jul 1 2020”. [Online]. Available: https://towardsdatascience.com/making-sense-of-text-clustering-ca649c190b20 [Accessed:9-July-2021]

[8] ”Clustering Similar sentences together using Machine Learning”. [Online]. Available: https://blog.eduonix.com/artificial-intelligence/clustering-similar-sentences-together-using-machine-learning/ [Accessed: 9-July-2021]

[9] ”Text Summarization”. [Online]. Available:https://paperswithcode.com/task/text-summarization [Accessed: 9-July-2021]

[10] ”Python Documentation”. [Online]. Available: https://www.python.org/; [Accessed: 9-July-2021]

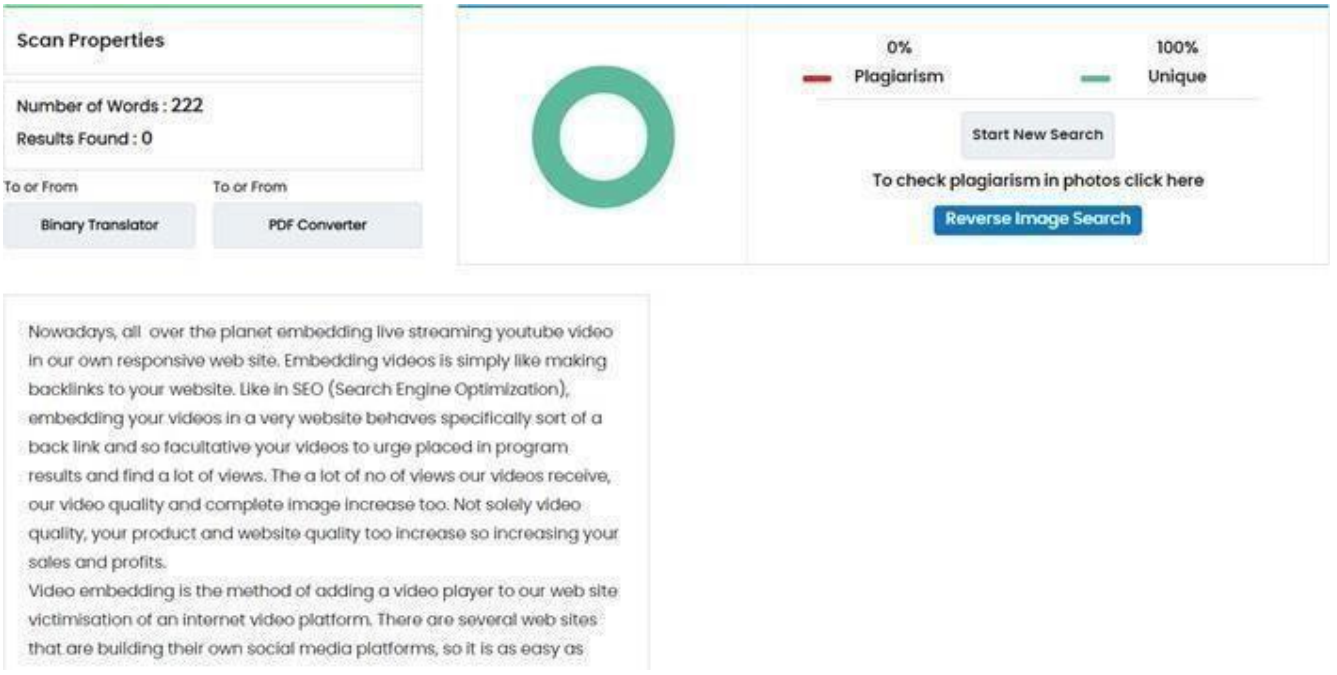
[11] ”Retrieve Live YouTube chats using pytchat”. [Online]. Available: https://github.com/taizan-hokuto/pytchat/wiki/LiveChat [Accessed: 9-July-2021]

[12] ” Authenticate with Firebase with a Email Using JavaScript “. [Online]. Available: http://www.gstatic.com/firebasejs/8.6.1/firebase.js [Accessed: 9-July-2021]

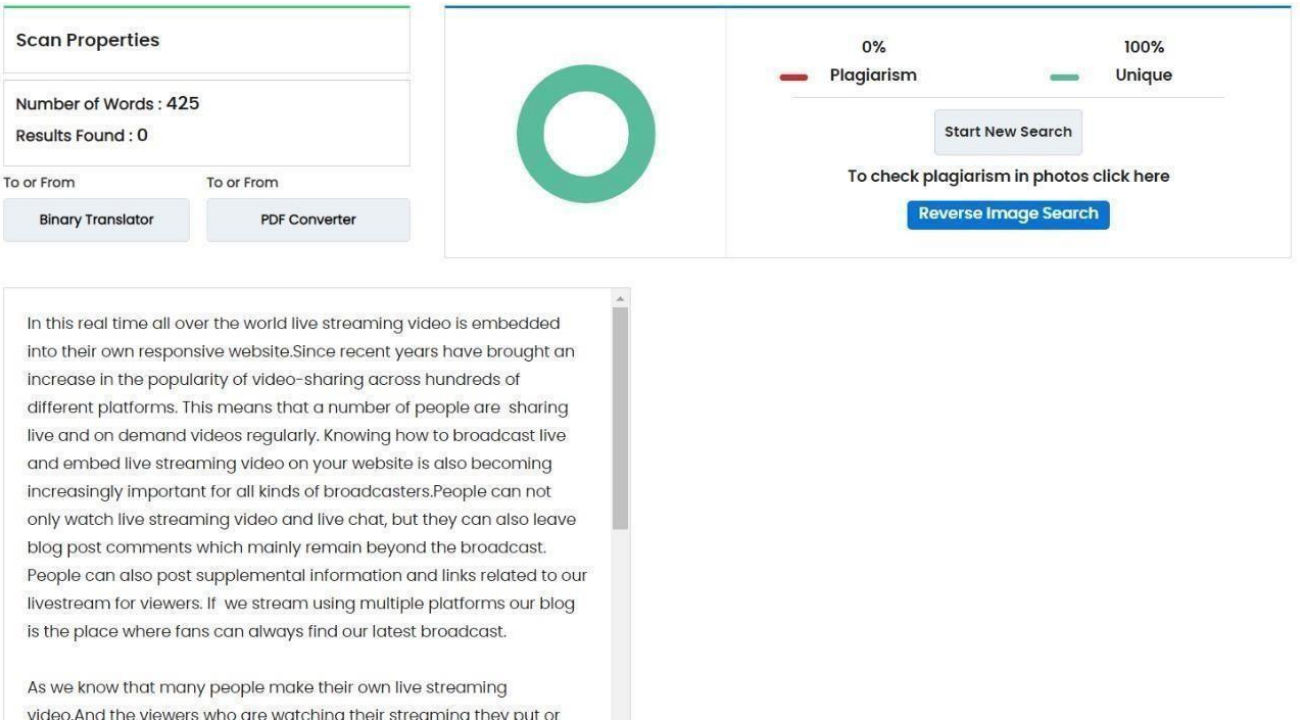
Plagiarism check

Source:https://www.duplichecker.com/

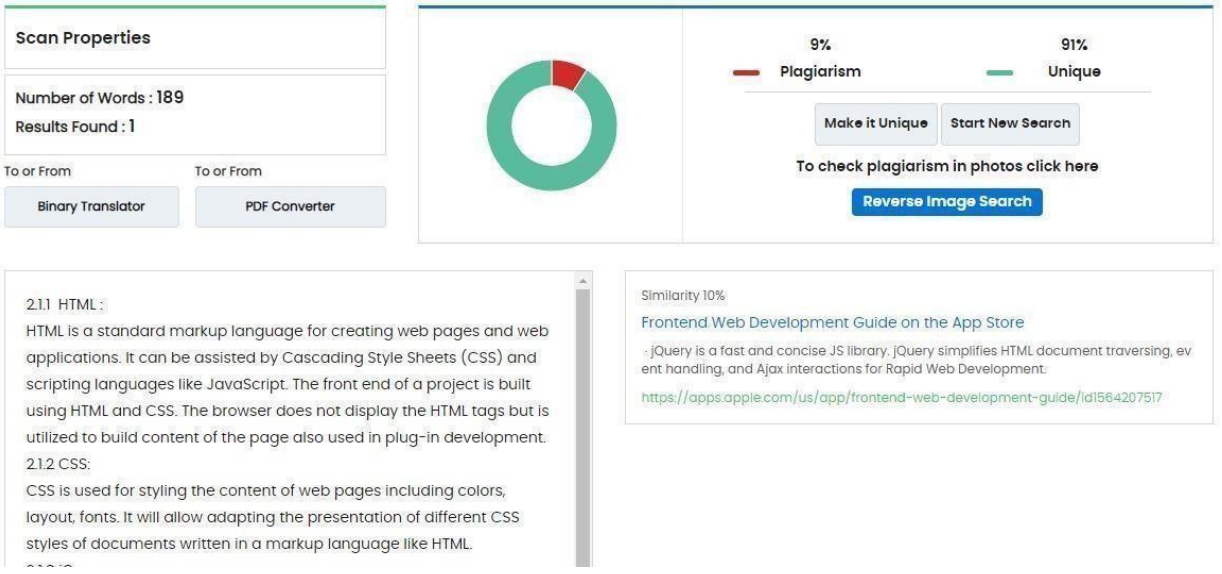
Abstract:



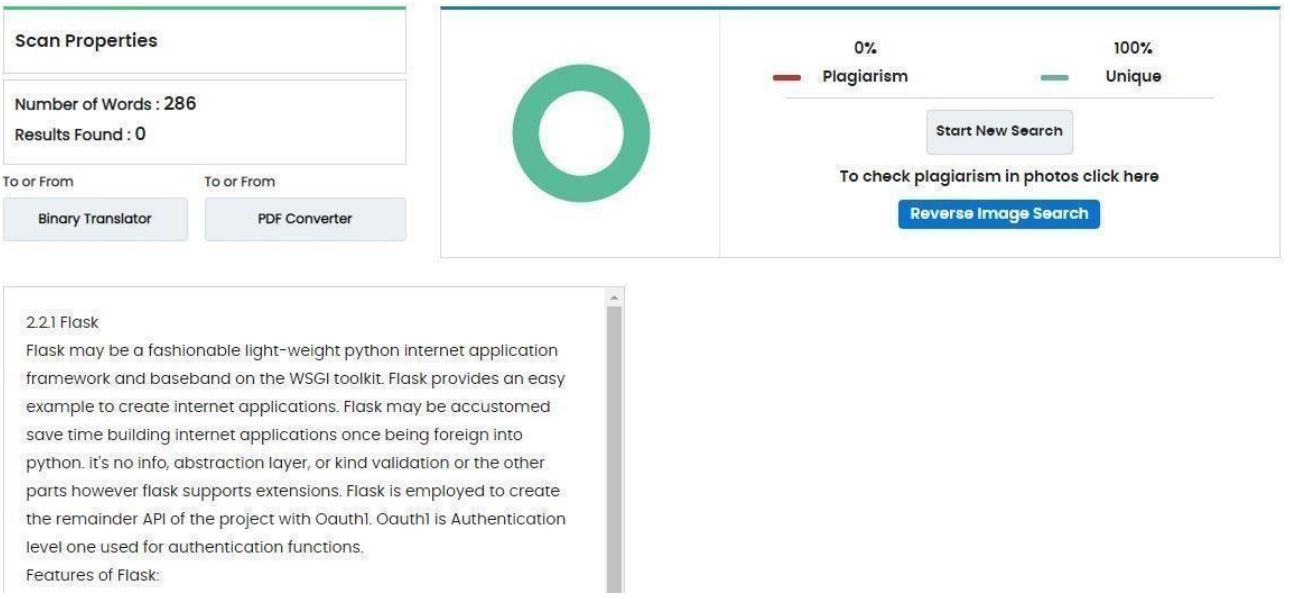
Introduction:



Front End:

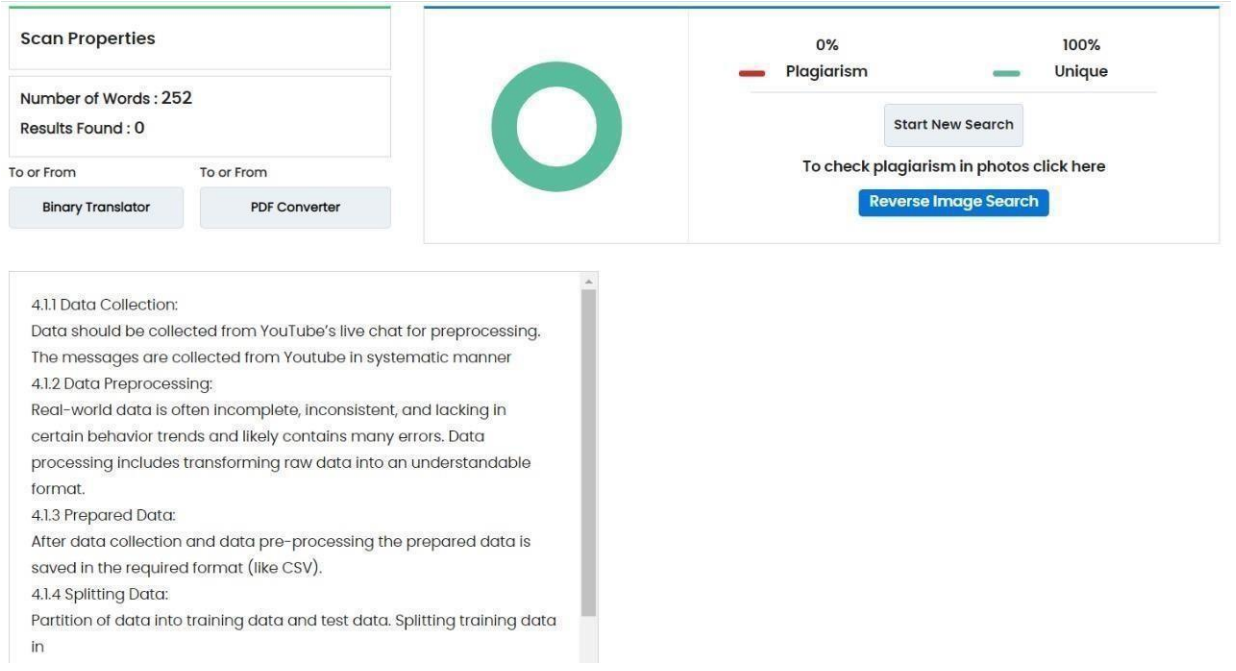


Back End:



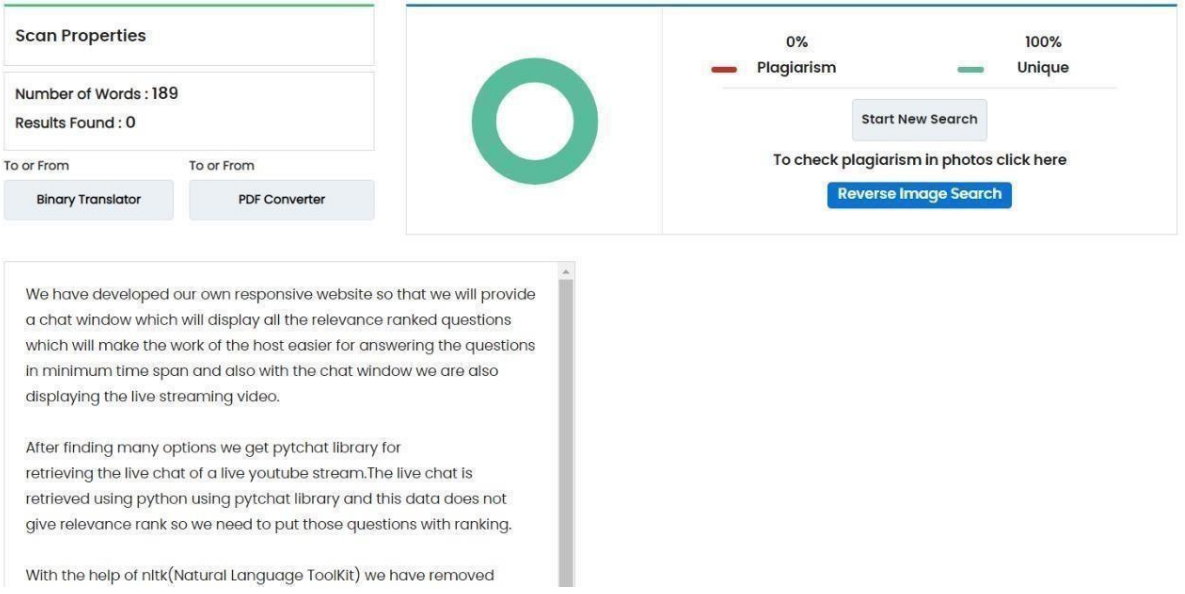
Methodology:

System Architecture Overview (High-level Architecture):



Low-Level Architecture:

Summary:



Student Details:

Name	Roll No.	Seat No.	Email ID	Contact No.
Anjali Chougule	40	1724618	anjalichougule23@gmail.com	7083233686
Janvi Pampattiwar	41	1724477	janvipampatiiwar123@gmail.com	8408954389
Pragati Phand	42	1724480	pragatiphand13@gmail.com	9067922831
DurreAfshan Shaikh	43	1724765	afshan8856@gmail.com	9518907974

Group Photo:

Anjali Chougule

Janvi Pampattiwar

Pragati Phand

DurreAfshan Shaikh

GitHub Link:

https://github.com/AnjaliChougule/Relevance-ranked-QnA-system-for-live-YouTube-s-embedded-stream-on-a-RWA