**Title - Youtube Harvesting**

**Description –**

Welcome to the YouTube Harvesting tool, a simple tool designed to help you efficiently collect and analyse data from YouTube. This tool provides the functionalities you need to gain valuable insights from YouTube's vast repository of videos, comments, and user interactions.

The YouTube Harvesting tool is a comprehensive application built to extract various types of data from YouTube. By leveraging the YouTube Data API, allows users to gather information such as video, channel, comments, likes, dislikes, and more.

**Table of Contents**

1. [**Aim**](https://github.com/PhonePe/pulse?tab=readme-ov-file#goal)
2. [**Guide**](https://github.com/PhonePe/pulse?tab=readme-ov-file#guide)
3. [**Code Details**](https://github.com/PhonePe/pulse?tab=readme-ov-file#documentation)
4. **Usage**
5. **Features**
6. **FAQs**
7. **License**
8. **AIM**

To create a Streamlit application that allows users to access and analyse data from multiple YouTube channels

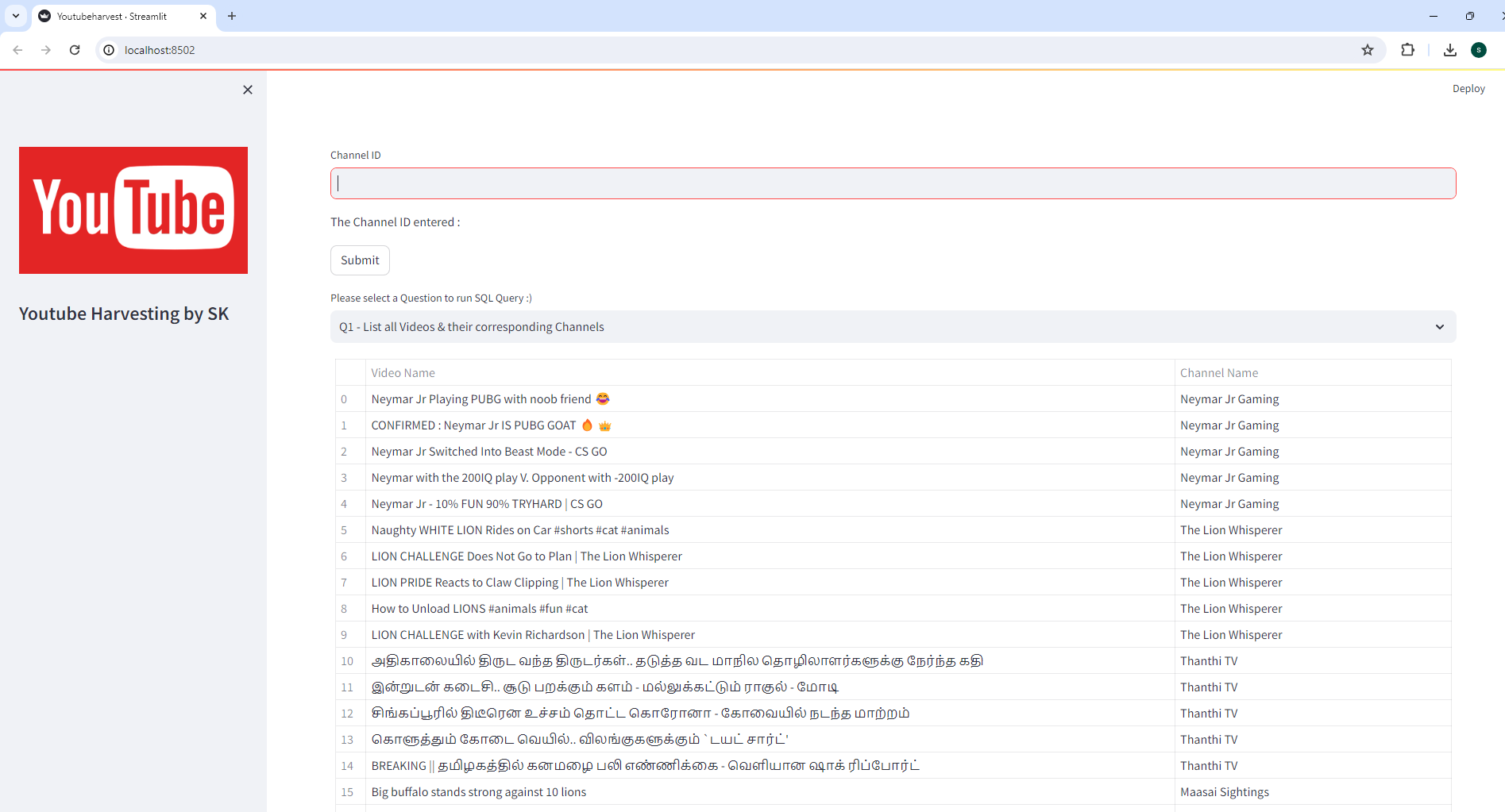
1. **Guide**

Prerequisite - you need the below file and software be downloaded / installed in your system.



Please follow the below steps

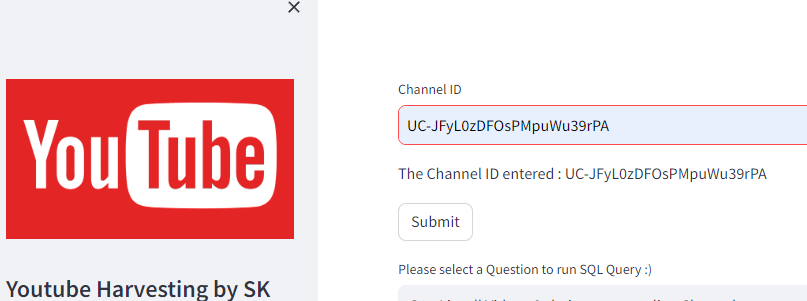
1. Open Visual studio code to execute the above python file.
2. Install the blow 2 packages in the terminal before executing the python file
   1. pip install google-api-python-client
   2. pip install isodate
3. Execute the program (python file) in the visual studio code
4. You will be prompted to run stream lit in the terminal - streamlit run Youtubeharvest.py
5. Once you execute the command the Youtube harvest app will be launched in the browser
6. Like below

****

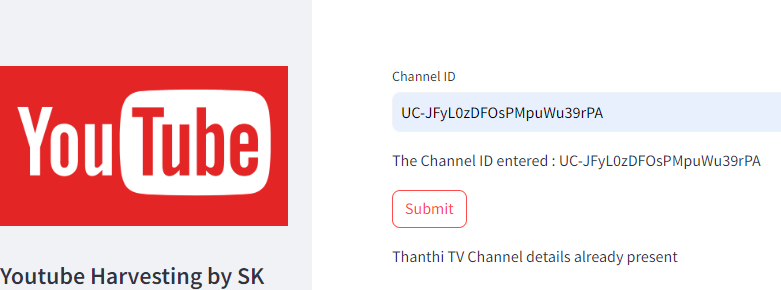
1. **Usage**

Examples of how to use the project, including code snippets and command-line usage.

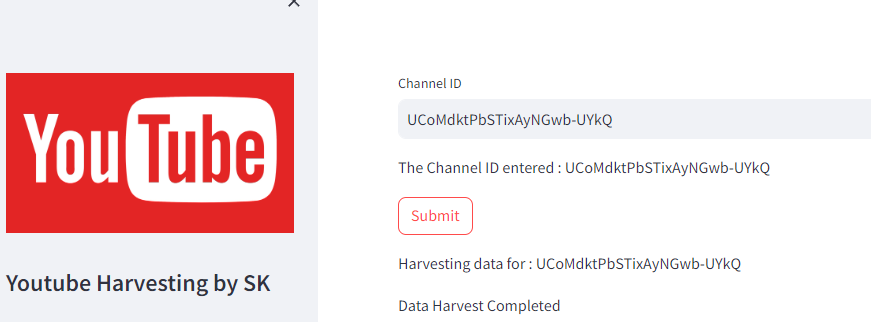
Once the YouTube Harvest app is up, you can enter a YouTube channel id to harvest the channel details.



The app immediately checks whether this channel details are already harvested, if available it will prompt a message. Else It will start the harvesting process

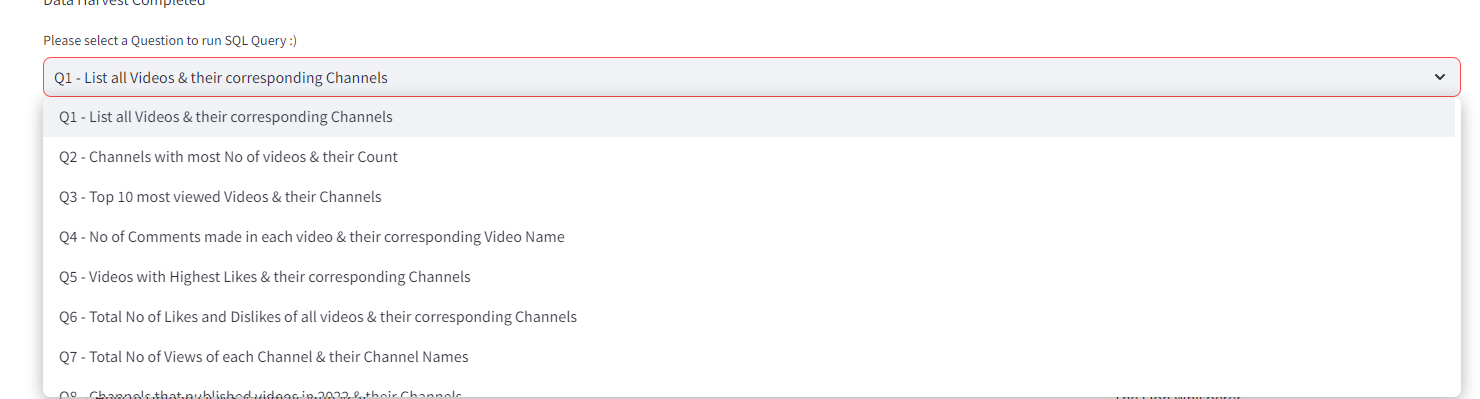


The program uses YouTube data API to retrieve all the details



Once the details are harvested the app prompts a message saying harvest completed.

The 10 built in statistical questions are available in a drop down, it can be selected to view the desired details.



1. **C**[**ode Details**](https://github.com/PhonePe/pulse?tab=readme-ov-file#documentation)

YouTube Request and response code sample

    #-----------------------------------------------------------------------------------------------------#

    # Channel request details

    #-----------------------------------------------------------------------------------------------------#

    chan\_request = youtube.channels().list(

        part="snippet,contentDetails,statistics",

        id=chan\_id

    )

    chan\_response = chan\_request.execute()

    #-----------------------------------------------------------------------------------------------------#

    # Playlist & Video request details

    #-----------------------------------------------------------------------------------------------------#

    playlst\_request = youtube.playlistItems().list(

        part="snippet,status,contentDetails",

        playlistId=chan\_playid

    )

    playlst\_response = playlst\_request.execute()

        video\_request = youtube.videos().list(

            part='snippet,contentDetails,statistics',

            id=video\_id

        )

        video\_response = video\_request.execute()

        #-----------------------------------------------------------------------------------------------------#

        # Comments request details

        #-----------------------------------------------------------------------------------------------------#

        try:

            comments\_request = youtube.commentThreads().list(

                part='snippet',

                videoId=video\_id

            )

            comments\_response = comments\_request.execute()

**5. Features**

1. Ability to input a YouTube channel ID and retrieve all the relevant data (Channel name, subscribers, total video count, playlist ID, video ID, likes, dislikes, comments of each video) using Google API.

2. Ability to collect data for many YouTube channels and store them in the data lake by clicking a button.

3. Store the data in a MYSQL

4. Ability to search and retrieve data from the SQL database using different search options, including joining tables to get channel details.

**6. FAQs**

**7. License**