Youtube App Documentation:

Objective:
The goal of the program is for the user to type in a search bar that will query videos from youtube services. The user will then be able to select their desired video from a drop box containing the results from their search. Once a video is selected, useful information will appear on screen such as publication date, thumbnail image, and the youtube video's url link. The user will then have the option to play their selected video in a web player.

Data Set:
The data include all info related to a Youtube video's snippet info. This includes the thumbnail image, url link, publication date, and title.

Ideal User:
Someone who would like to find all relevant data from a Youtube video without having to inspect the page themselves.

Ideal Context:
The selected video actually contains all information that is being pulled by the program.

Ideal Implementation:
Web based as this requires to connect to Youtube Services and play their videos using a pulled URL link.

API key instructions:

1.) Make a Google account if you don't already have one.
2.) Sign into Google Developers Console
3.) Create a new project, enabling google.com/youtube/v3
4.) Go to CREDENTIALS tab and press +create credential at top of screen
5.) Create API key.
6.) Copy paste api key into code that is commented "Replace with your own API key"

Note: The api has a query quota point system. The default is 10,000. Once this quota is reached you will have to wait one day before being able to use the API again.
For further information: https://developers.google.com/youtube/v3/determine_quota_cost

Postmortem:

Estimated time: 10+ Hours
Pre-Production: 1 hour
Production: 9+ hours

The largest struggle was just trying to figure out the Youtube api function. I watched several tutorial videos and read the official documentation.

I first created a console project that only purpose was to get the names of videos from a search. It had no user interface, being solely text with no other functionality. I then turn that code into a class within a window forms project. This is where most of my struggle came to play. I encountered a problem where my query call would infinitely loop. I believe this was the case because of threading issues from updating the gui so I researched on how to resolve it to no success. I eventually found out I needed to add a time limit to the query call on how long it will wait before searching and resolving the issue. It didnt take much time after that to add a Linq statement to search for matches of the user's selected video to retrieve and display its info.