Computer Science Project Titled

“Multiplatform and User Customizable GUI Embedded Video Downloader”

Done By:

Aayush Rawal(Roll Number: 1)

Daksh Chhokra(Roll Number: 10)

Students of class 12th –B of

Amity International School, Sector-43, Gurgaon, Haryana

**Certificate**

This is to certify that Daksh Chhokra and Aayush Rawal of Class 12th-B of Amity International School, Sector-43, Gurgaon have prepared a project report based upon a project they developed called “Multiplatform and User Customizable GUI Embedded Video Downloader.” This report is a result of their efforts and endeavors. This report is found worthy of acceptance as a final project report for class 12 CBSE Board practical and has been prepared under my guidance

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed dlddcdcd

Ms. Anjali Mittal

PGT(Computer Science)

Department of Computer Science

Amity International School, Sector-43, Gurgaon

**Acknowledgments**

We would like to take this opportunity to express our profound gratitude and deep regards to our guide and mentor, our computer science teacher, Ms. Anjali Mittal (PGT Computer Science at Amity International School, Sector-43, Gurgaon) for her exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by her were integral to the success of this project.

We would also like to thank our school, Amity International School, Sector-43, Gurgaon, who has given us the chance to delve into this amazing topic of Computer Science, enabled us to create such a project.

**Inbuilt Python Modules Used**

* **Subprocess Module**
  + The subprocess module allows us to:
    - spawn new processes
    - connect to their input/output/error pipes
    - obtain their return codes
  + It offers a higher-level interface than some of the other available modules, and is intended to replace the following functions:
    - os.system()
    - os.spawn\*()
    - os.popen\*()
    - popen2.\*()
    - commands.\*()
  + We cannot use UNIX commands in our Python script as if they were Python code. For example, **echo name** is causing a syntax error because **echo** is not a built-in statement or function in Python. So, in Python script, we're using **print name** instead.
  + To run UNIX commands we need to create a **subprocess** that runs the command. The recommended approach to invoking **subprocesses** is to use the convenience functions for all use cases they can handle. Or we can use the underlying **Popen** interface can be used directly.
* **Platform Module**
  + This module tries to retrieve as much platform-identifying data as possible. It makes this information available via function APIs. If called from the command line, it prints the platform information concatenated as single string to stdout. The output format is useable as part of a filename.
* **OS Module**
  + The OS module in Python provides a way of using operating system dependent functionality. The functions that the OS module provides allows you to interface with the underlying operating system that Python is running on – be that Windows, Mac or Linux. You can find important information about your location or about the process.

**External Python Packages Used**

* **Tkinter Package**
  + Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps –

* + - Import the *Tkinter* module.
    - Create the GUI application main window.
    - Add one or more of the above-mentioned widgets to the GUI application.
    - Enter the main event loop to take action against each event triggered by the user.
* **yt-dl Package**
  + yt-dl is a external python package to detect and access video based file from popular video sharing websites.

**Files Generated**

The possible files generated by this program are

* **title.extension**
  + This is a video file
  + For example Thisisavideofile.mp4
* **title.jpeg**
  + This is an image file
  + It is generally the cover art for the requested video
* **title.srt**
  + This is an subtitle file
  + They are automatically embedded in the video

**General Principle and Working of the Code**

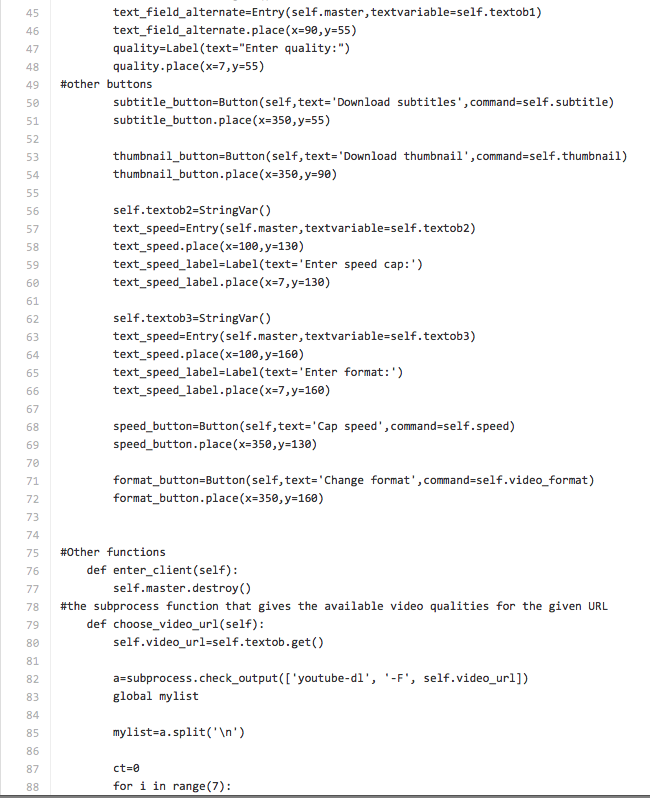
The code creates a robust graphic user interface, to help the user access and download videos of video sharing websites. This code is multiplatform, and works flawlessly on the Unix based Mac OSX, Windows 10, ad some select Linux kernels. Once the user enters the unique resource allocator of the video, the program returns all possible qualities available for retrieval. Then the user can choose whichever quality they like, and move to the other options. The user has the option to download, subtitles if they are available with the video. Also, the user can download the cover art of the video if they choose to do so, in a separate Joint **P**hotographic Experts Group(JPEG) format.

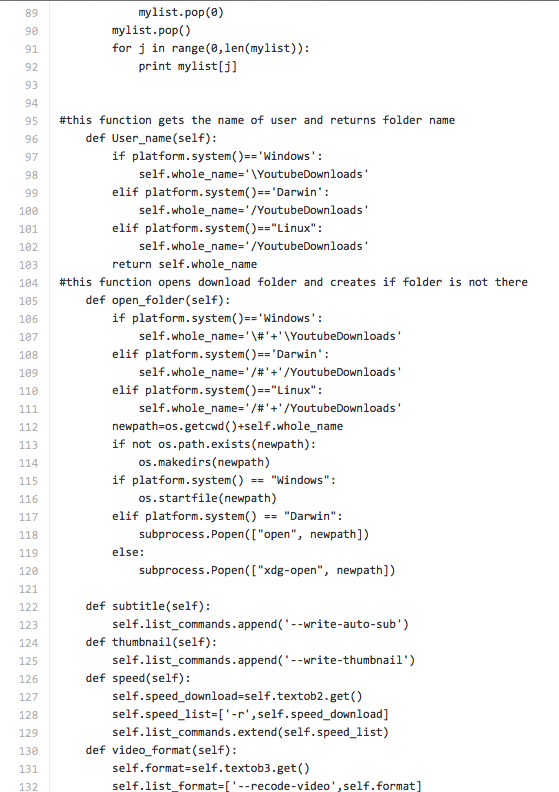
After that the user has the option to limit the video retrieval speed, by entering the numerical value of their highest allowed bitrate in bits per second. The final option is then, to choose the format the video, has to be converted into post the download.

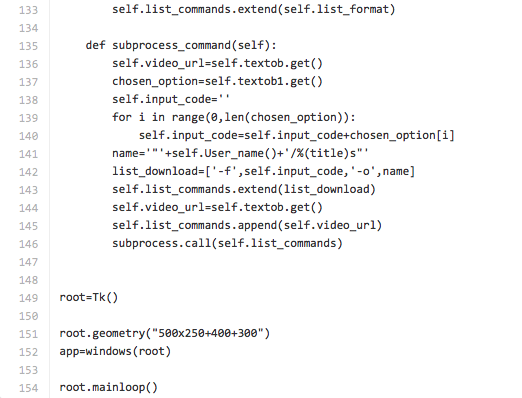
Finally, the user from within the application itself, can access the downloads folder to view, and if needed modify the downloaded items.

**The Code**

****



****

**The Graphic User Interface and Interpreter Output**

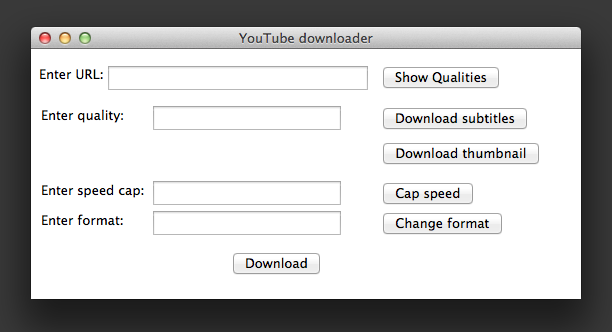


Figure 1: This is the first Graphical User Interface screen seen by the user

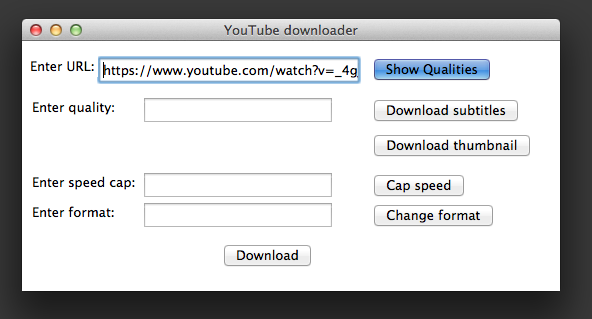


Figure 2: In this, the user enters the url of the video, and presses the show quality button

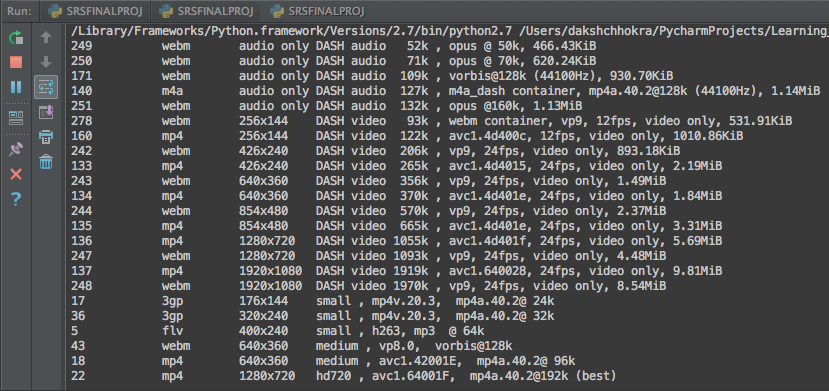


Figure 3: The output in the interpreter, which gives us all the qualities of the video available.

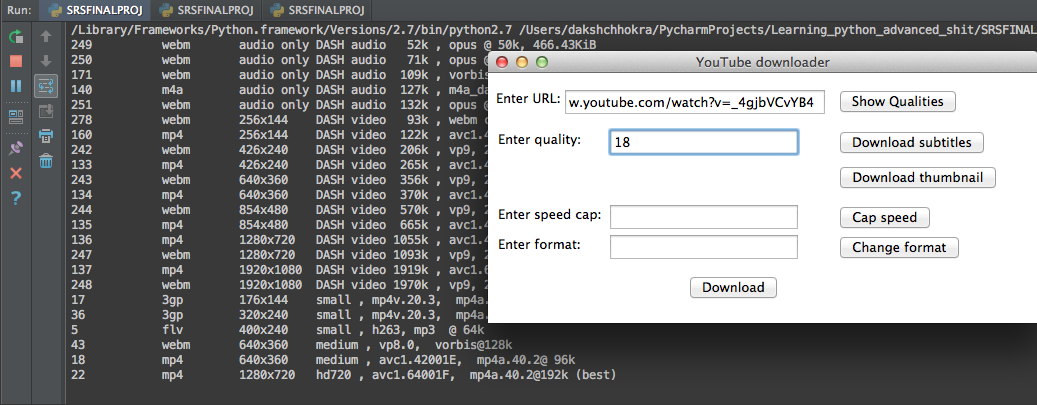


Figure 4: The user can then enter the numerical code corresponding to their preffered video format

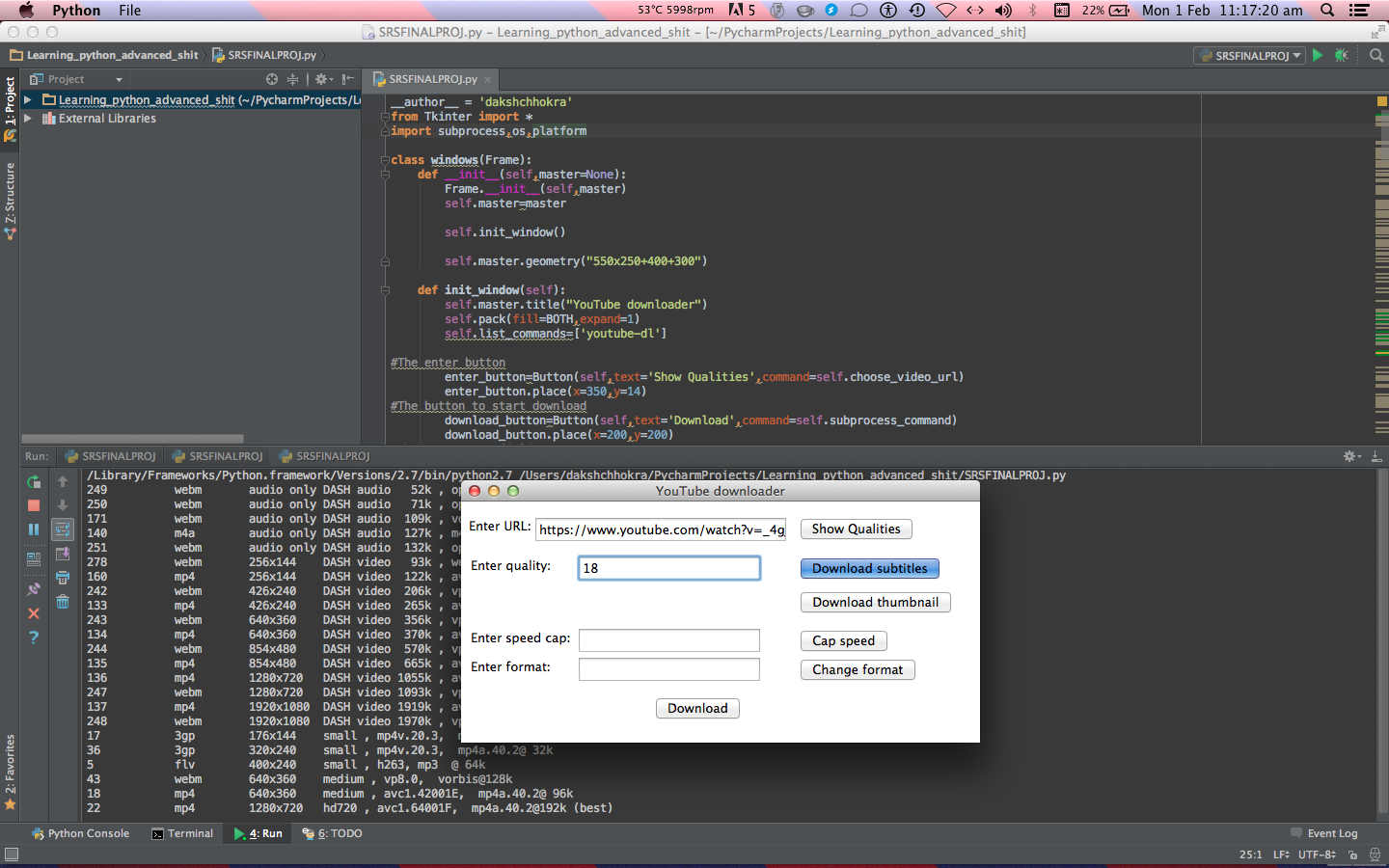


Figure 5: If the user chooses, and the video has them available, the user can elect to download accompanying subtitles

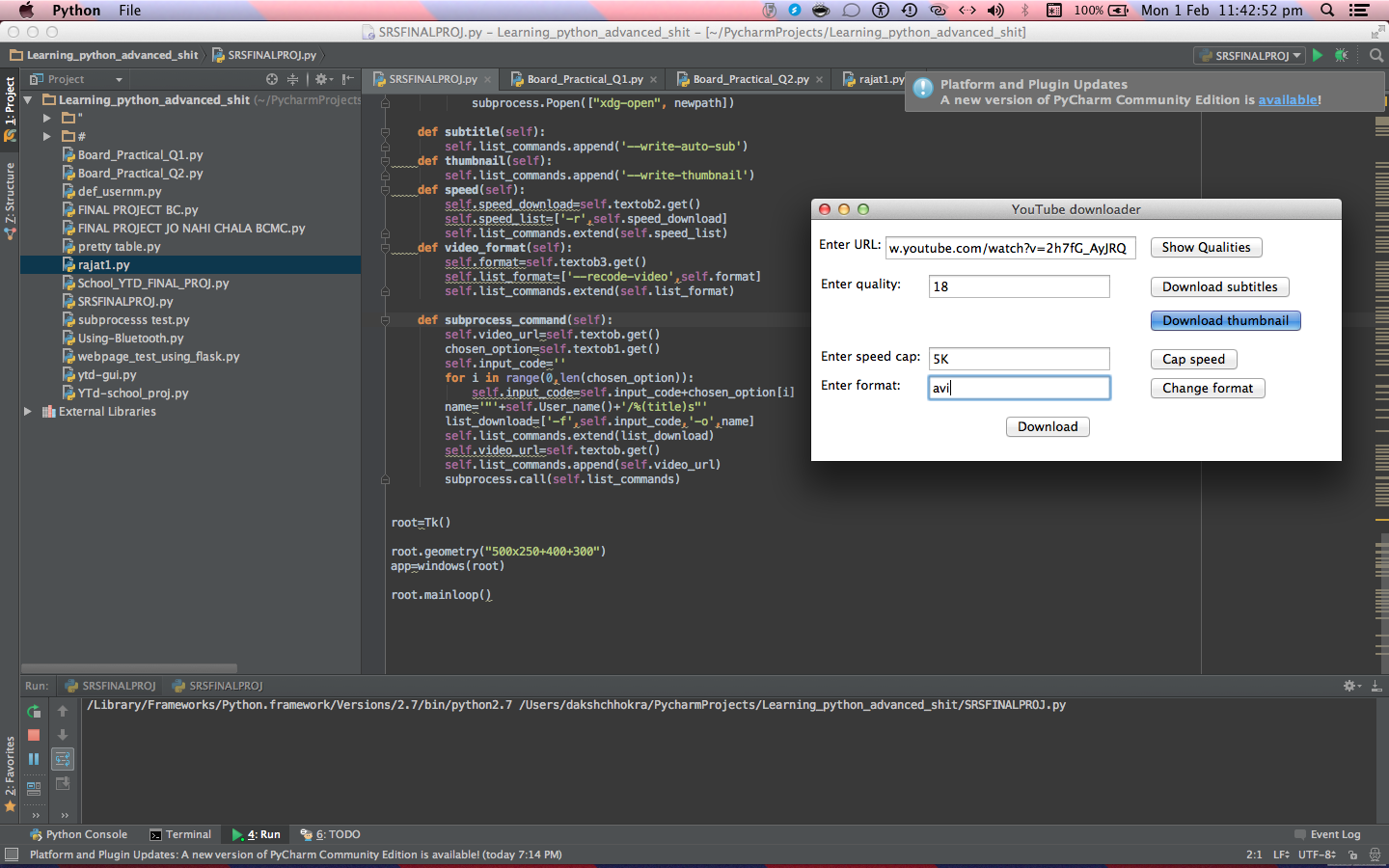


Figure 6: If the user chooses, they can also download the accompanying thumbnail of the video by pressing a button

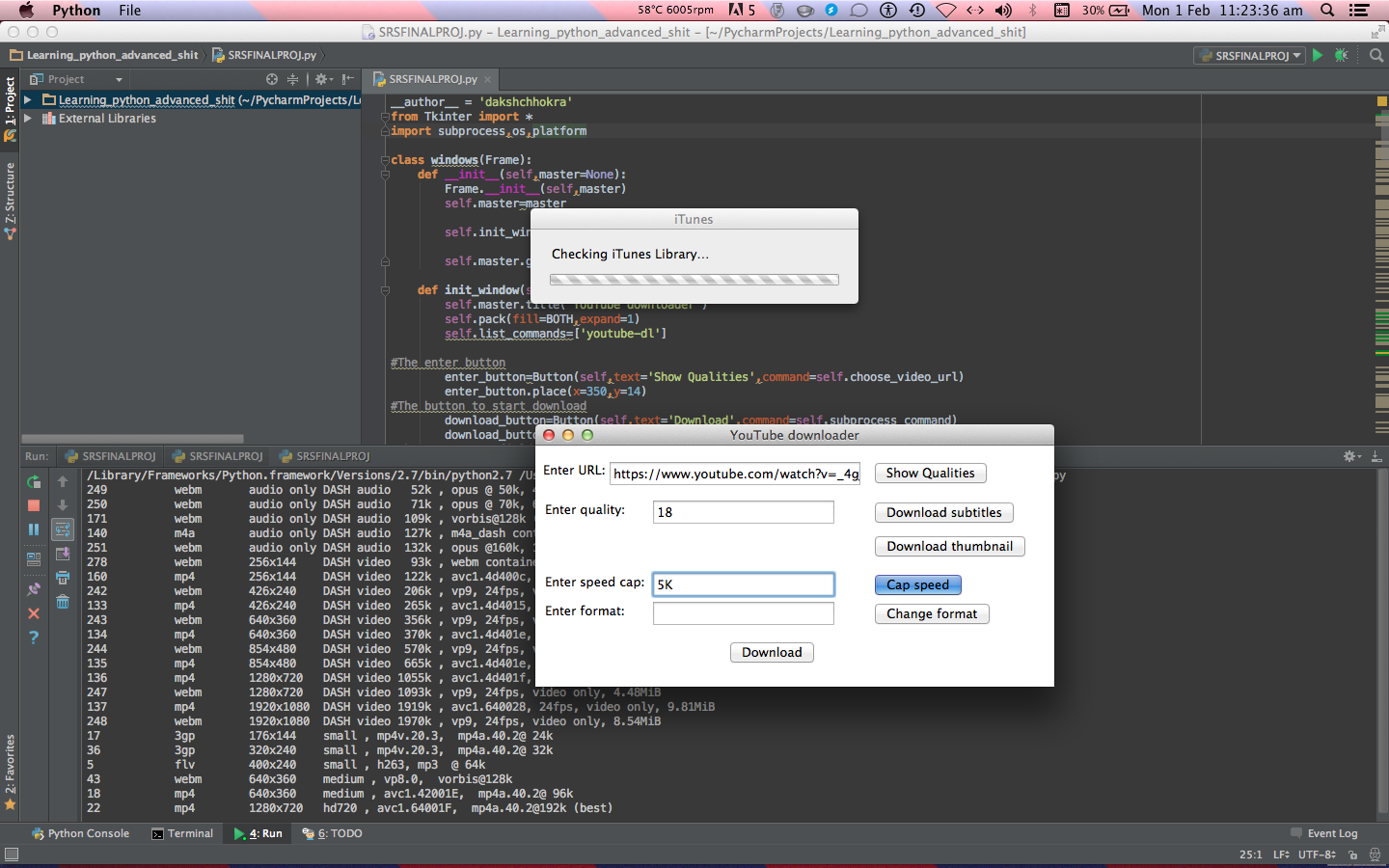


Figure 7: Then the user can cap the maximum download speed by entering the maximum permitted speed in bits per second

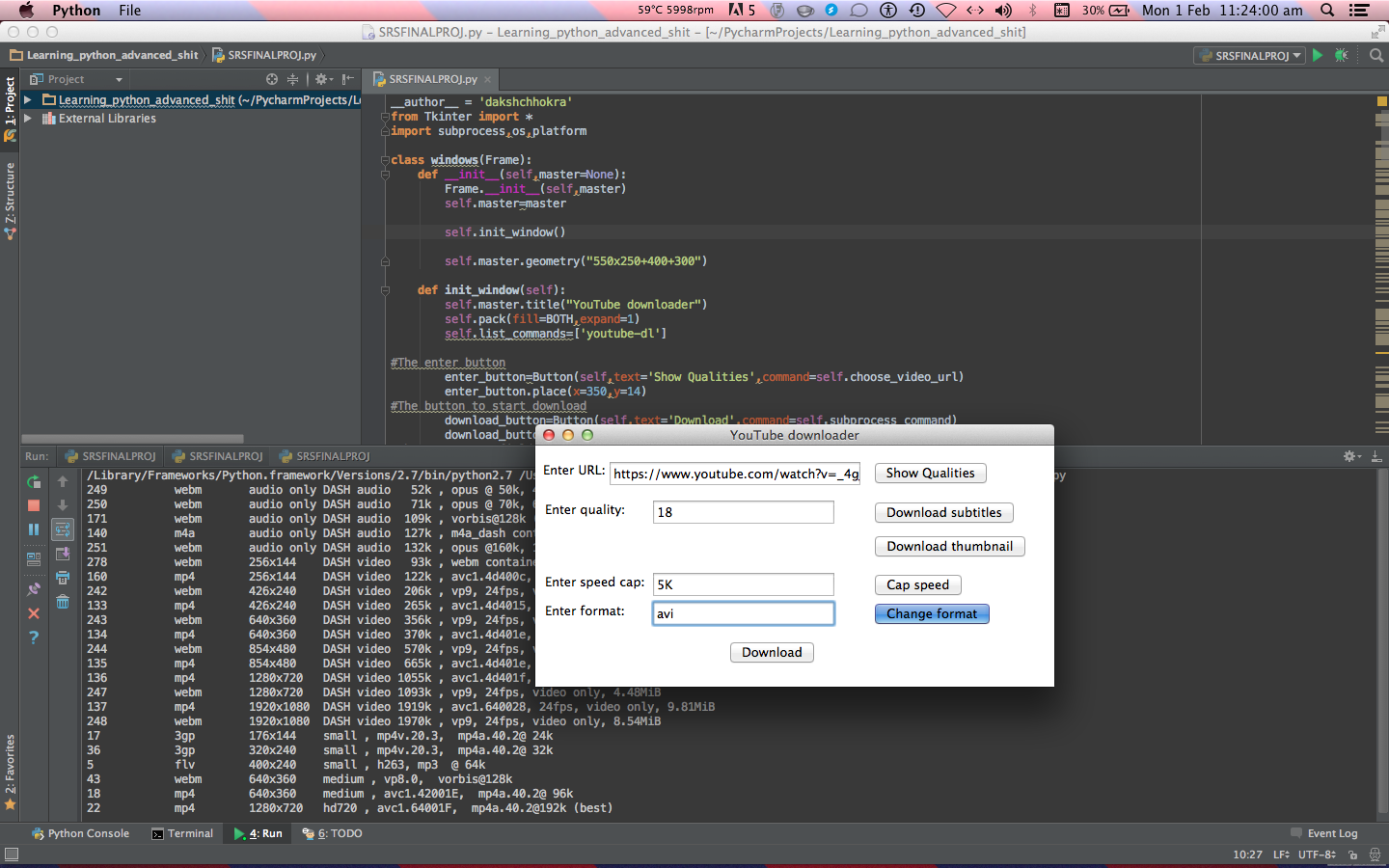


Figure 8: The user can then choose to change the format of the video post -processing if they choose to do so, by entering their preferred format in the text box.

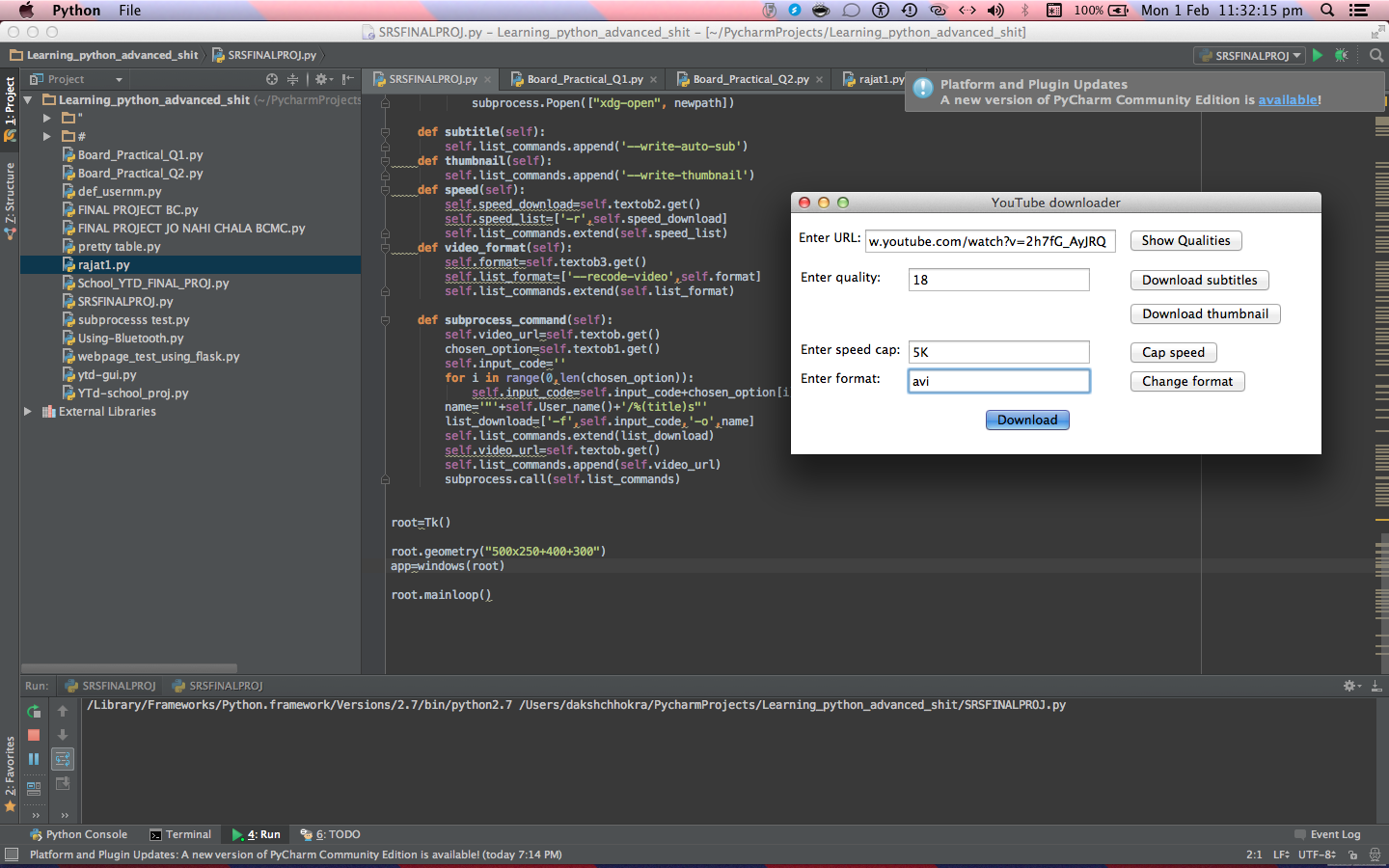


Figure 9: Then the user presses the download button



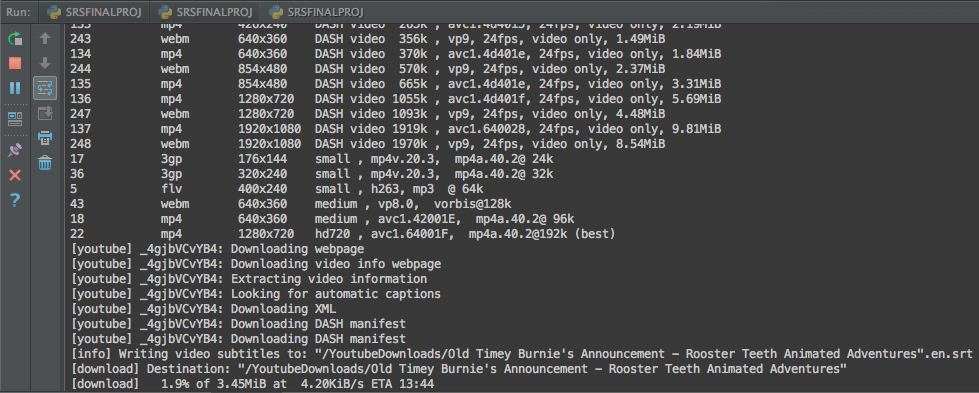


Figure 10: After we press download, the video retrieval process begins, and we get a progress percentage.

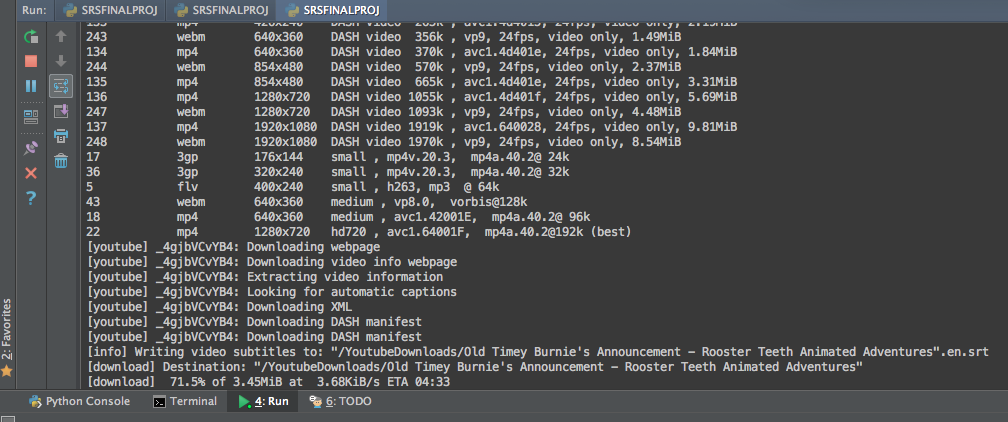


Figure 11: The download is in progress

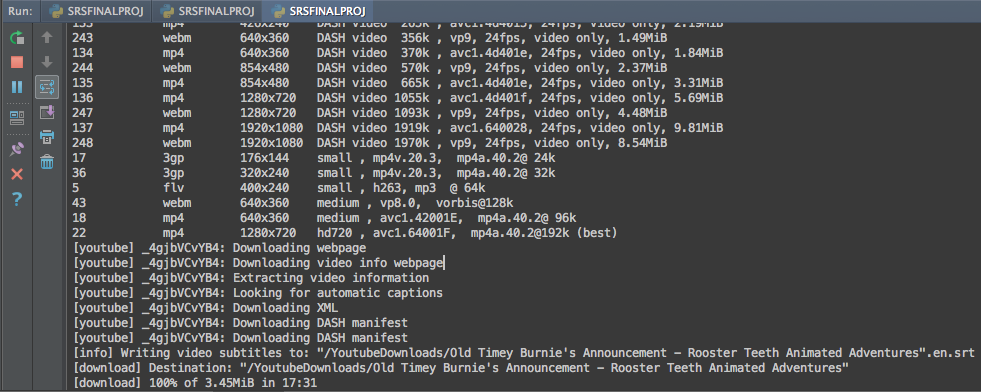


Figure 12: The download of the video is complete

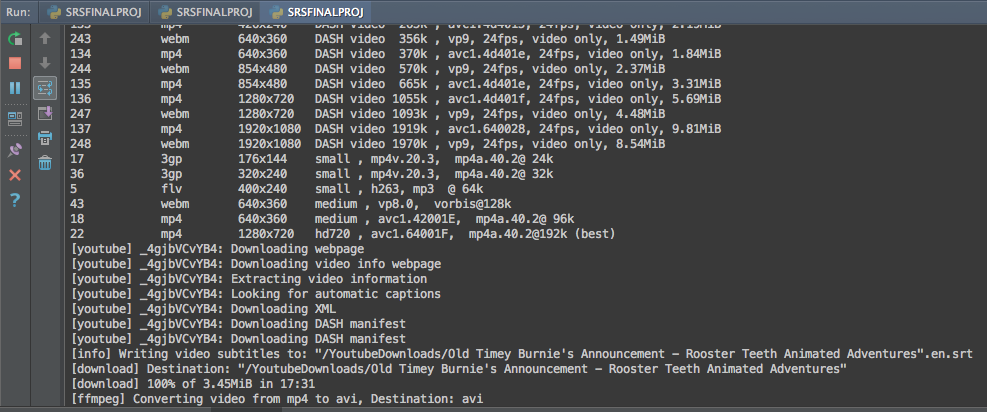


Figure 13: Once the download is complete, the conversion of the video is done.

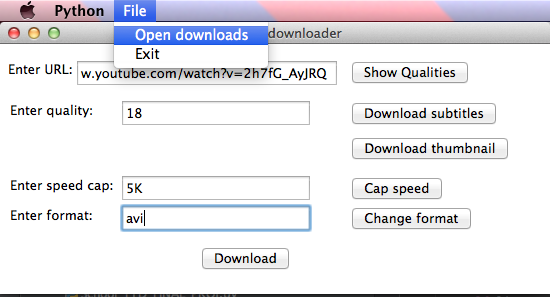


Figure 14: Then the downloads folder can be accessed to view the downloaded files

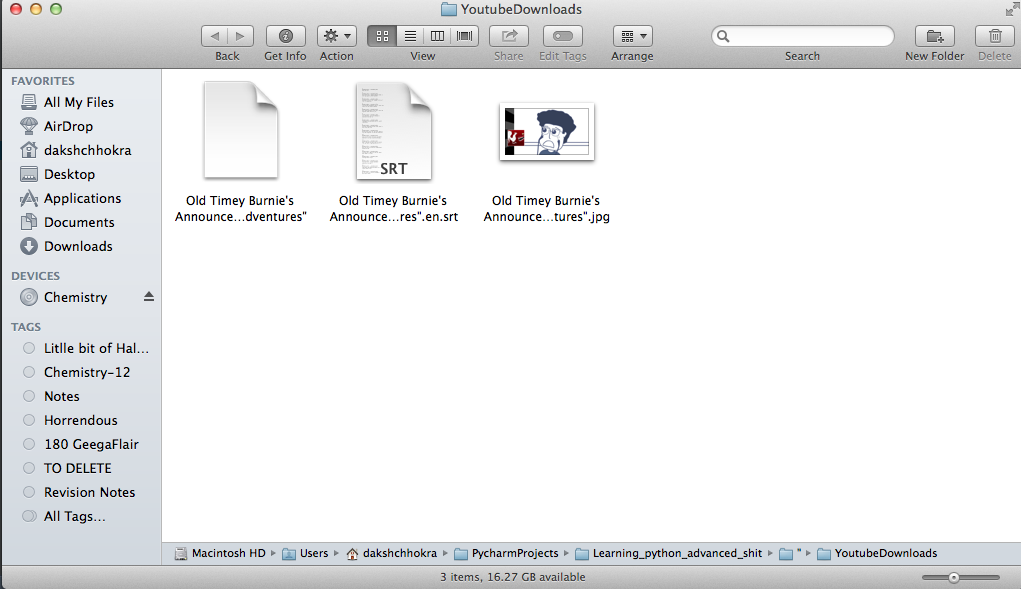


Figure 15: This is the downloads folder with the three files

**Bibliography**

* [**http://interactivepython.org/runestone/static/everyday/2015**](http://interactivepython.org/runestone/static/everyday/2015)
* [**https://docs.python.org/2/library/tkinter.html**](https://docs.python.org/2/library/tkinter.html)
* [**http://www.tkdocs.com/**](http://www.tkdocs.com/)
* [**http://effbot.org/tkinterbook/**](http://effbot.org/tkinterbook/)
* [**https://www.manning.com/books/python-and-tkinter-programming**](https://www.manning.com/books/python-and-tkinter-programming)
* [**https://www.google.co.in/search?q=subprocess&ie=utf-8&oe=utf-8&gws\_rd=cr&ei=tKGvVrj7McmSuAT78KCYCw**](https://www.google.co.in/search?q=subprocess&ie=utf-8&oe=utf-8&gws_rd=cr&ei=tKGvVrj7McmSuAT78KCYCw)
* [**http://rg3.name/201502261857.html**](http://rg3.name/201502261857.html)