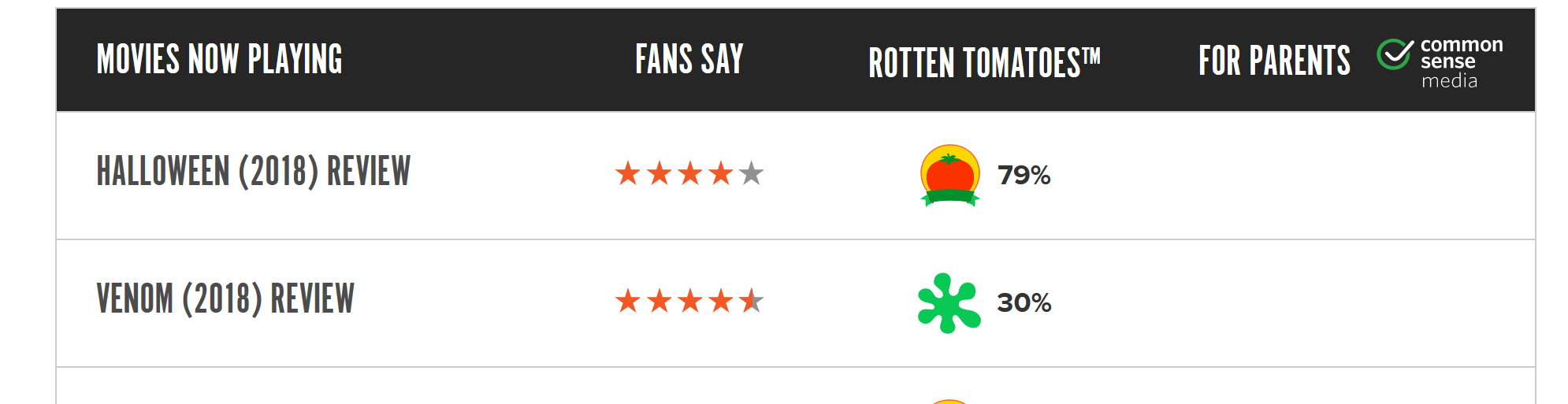
CIS 41B - Lab 3: Web access, Data storage, GUI

Write an application that lets the user search for information on movies that are currently playing in theaters.

The application has 2 separate parts: lab3front.py and lab3back.py.

lab3back.py will do the following steps:

* Go to [https://www.fandango.com/movie-reviews](https://www.fandango.com/movie-reviews%20) to get the list of current movies that are playing. Here is the part of the web page with data to extract:  
  

The data to be extracted are: the movie name (without the word review), the number of stars under the "FANS SAY" column (the ratings are 0 to 5.0, in increments of 0.5), and the link to the specific movie web page. There are 50 movies in the table shown on the page.

* Follow the link of each movie, we can find the release date and the genre of the movie. Here's an example for "Halloween" (in the spirit of this month):



Between the 2 web pages, the data that should be extracted for "Halloween" are: Halloween (2018), 4.0, October 19, 2018, and Horror.

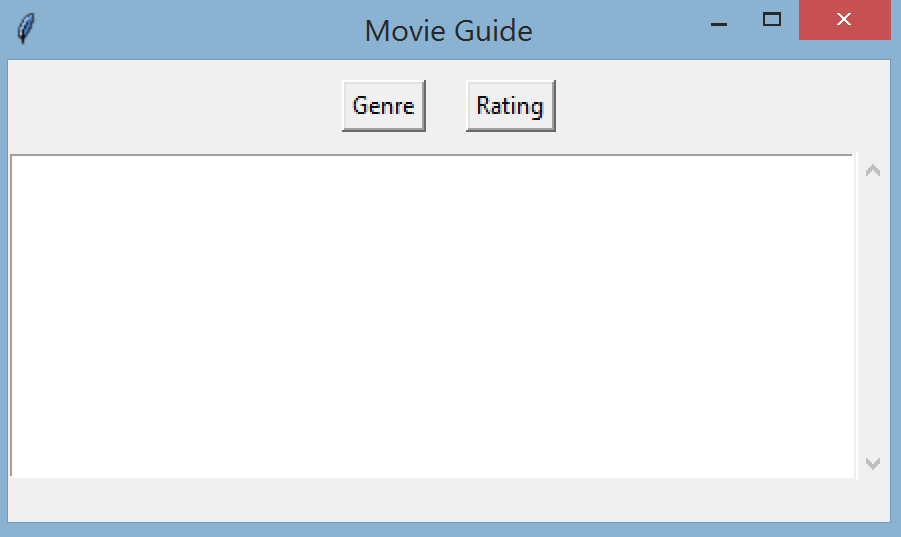
* It does take a little time to fetch the 4 data fields for all 50 movies, therefore we want to store the data away so that we don't have to wait every time we want to look up movie information.  
  Store the data in a pickle file, a JSON file, and an SQLite file.
* The only lab requirements for lab3back.py are:
  + All input data must come from the given fandango link and sub-links.
  + The 3 output files are created.
  + Use your knowledge of Python iterables and regular expression to make the data extraction efficient.

There is no requirement on how you structure your code, however, please use good programming and documentation practice. You can break up the code into functions or into classes. Please do not turn in a file that is just one long block of code.

Hint for working on this part of the lab: divide the work into 2 halves.  
1. Extract the data from the website and store into the pickle and JSON file  
2. Then use the pickle or JSON file as input to build the database. By not going to the web to fetch data, you save a lot of time each time each time that you re-run your program as you build the database.

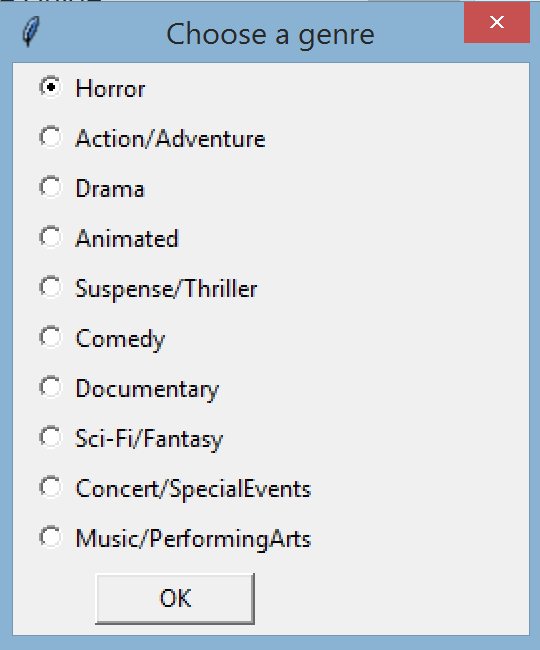
lab3front.py can only be done after lab3back.py is done. lab3front.py will do the following steps:

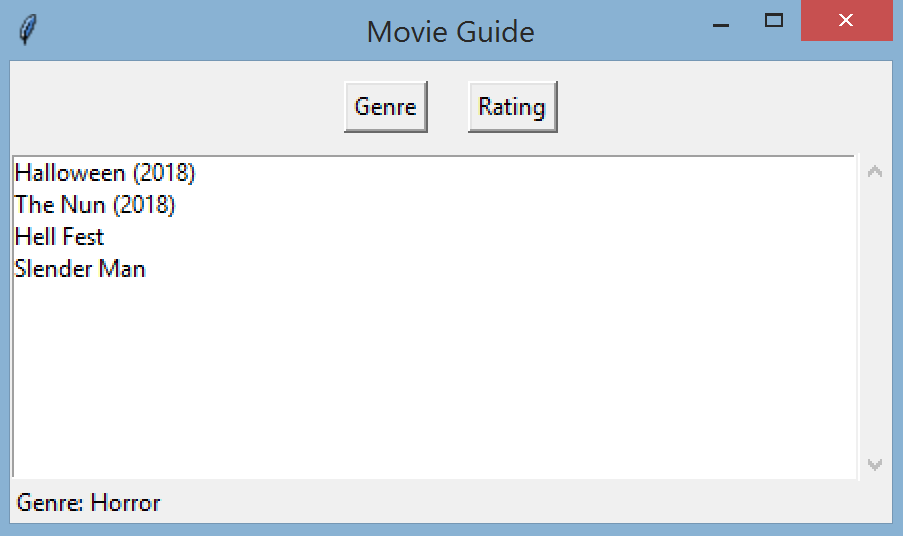
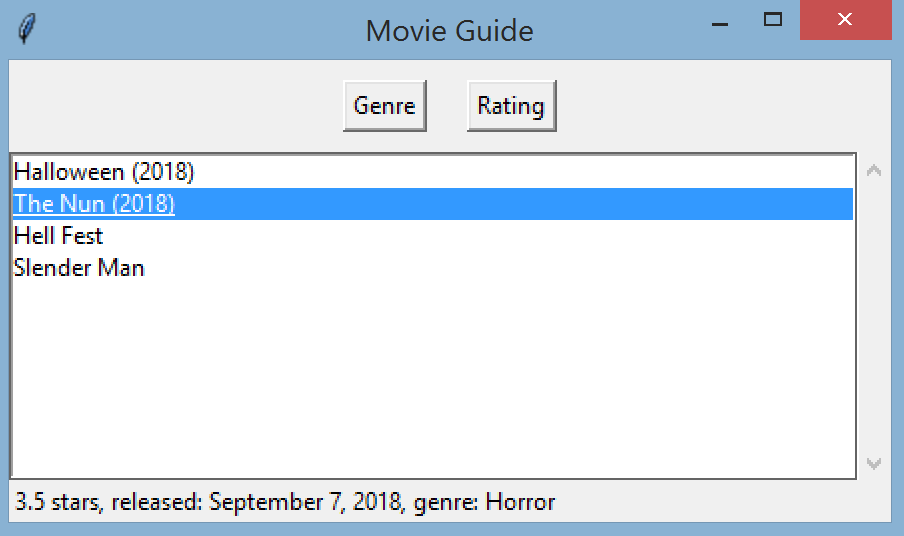
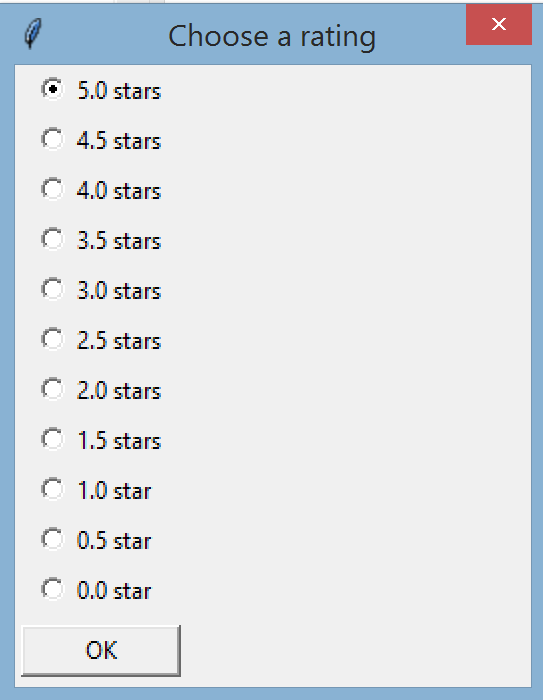
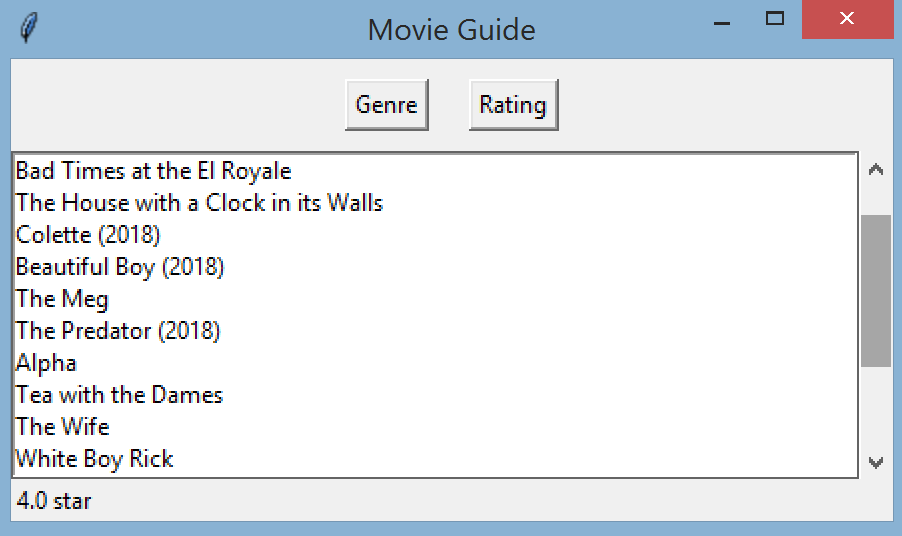
* Use the database that's built by lab3back.py for input. Do not go out to the web for data for this part.
* Create a GUI window for the user to choose to look up movies by genre or by rating.



There's a listbox that will be used to display the movies that match the user's choice.

* When the user clicks on Genre, a dialog window shows up for the user to click on a choice and the OK button.

The radio buttons must be in the vertical direction, and note that the window title says "Choose a genre"  
 

* If the user selects Horror and clicks OK, then the dialog window closes and the main window shows the Horror movies in the listbox. The bottom of the window confirms / reminds the user of the genre he/she selected.  
   
* The user can click on one movie, and the bottom of the window shows all the informaton for that movie:  
   
* At this point the user can click on any of the movies in the listbox and the data at the bottom of the window will update for the currently selected movie.
* When the user clicks on Rating or on Genre again, the listbox clears and the bottom of the main window also clears, and the dialog window shows up.
* If the user clicks on Rating, then the dialog window opens with the number of stars listed in reverse order. Note that the title says "Choose a rating". Note also that there is no 's' after 'star' for 0.0, 0.5, and 1.0.   
  *[1pt Extra Credit: don't use an if else statement for this (and don't even think of hard coding 11 strings)]* 
* The user can click on a star rating and then click OK. The dialog window closes and the main window shows the result in the listbox. Just like with Genre, the bottom of the main window confirms / reminds the user of his/her choice. Note also that when the movie list is longer than the screen size, the scrollbar can be used.  
   
* If the user clicks on a movie at this point, then the bottom of the window shows the movie information, just as with the Genre.
* Additional description for the behavior of the dialog window:
  + When it appears, the user cannot select any widget in the main window.
  + If the user doesn't select any choice and instead clicks X to close the window, then nothing shows up in the listbox or at the bottom of the main window.
  + There should be only one dialog window class. From this one class you can instantiate a dialog object to work with Genre data or a dialog object to work with Rating data.
* The main window should be an object of your main window class, which is derived from tkinter's class.
* See version 2 of the GUI class notes for the 2 new widgets: listbox and scrollbar (slides 28-30).

There are 5 files to submit for this lab: lab3back.py, lab3front.py, pickle file, JSON file, SQLite file.  
The 2 .py files should be named as shown, but you can choose the names of the 3 data files.