SI 206 Final Project - BJPP

Members: Fay Piyathassrikul, Bridgit Jung

Harry Potter (API) and Rotten Tomatoes (BeautifulSoup)

Github repo: https://github.com/faypiya/SI206-FinalProj

1. **The goals for your project including what APIs/websites you planned to work with and what data you planned to gather**

For the API portion of the project, we worked with the Harry Potter API to gather data on the characters, one table generated from the characters from the whole franchise. The information gathered includes character id, name, house, patronus, student status and life status.

In regards to the BeautifulSoup portion, we worked with the "Best Movies at Home" (clicking on "Most popular streaming movies" in the menu leads to this) page of the Rotten Tomatoes website. We planned to gather data on each movie's title, Tomatometer score, and genres.

2. **The goals that were achieved including what APIs/websites you actually worked with and what data you did gather**

The initial plan for the API was to create one table. However, we created two tables to better meet the requirements of the project and to allow the database to join. We still managed to gather the character information from the character (whole franchise) collection. In addition to that, we gather some wand data from characters in the book series. Specifically, the wood, core, and length data were collected. For the BeautifulSoup portion, we were able to gather data on and work with each movie's title, Tomatometer score, and genres. However, we also gathered each movie's data_ems_id, which is the ID that Rotten Tomatoes gives each movie. We also worked with the "Certified Fresh" filter applied to the "Best Movies at Home" page.

3. The problems that you faced

The biggest problem we faced in the API portion was the wands table being partially complete. A lot of the data in the API was missing, in which some of the wand data were blank/null. Moreover, the character id weren't integer so executing the code caused a data mismatch. After a long process, we managed to fix the problem and convert the id column into a primary key integer using autoincrement.

One of the problems faced during the BeautifulSoup Portion was that the website's code was often inconsistent with its' element tags, so it made it difficult to scrape every movie's information (i.e. genres and the individual movie page links). I would often have to select for multiple variations of tags in order to get a complete set of the movies' information. Another problem I encountered the IDs provided by Rotten Tomatoes not being an integer—I still kept this information for the JOIN statement, but I was able unique integer IDs by using AUTOINCREMENT for reference.

4. The calculations from the data in the database (i.e. a screen shot) (10 points)

API:

calculations.txt

```
Proportion of Deceased Characters in Each Hogwarts House:
Gryffindor: 0.3797804532577904
Hufflepuff: 0.23784230406043438
Ravenclaw: 0.1308132672332389
Slytherin: 0.25156397544853637

Most Common Wand Cores in Gryffindor:
Gryffindor: unicorn tail-hair: 2
Gryffindor: phoenix feather: 1
Gryffindor: dragon heartstring: 1

```

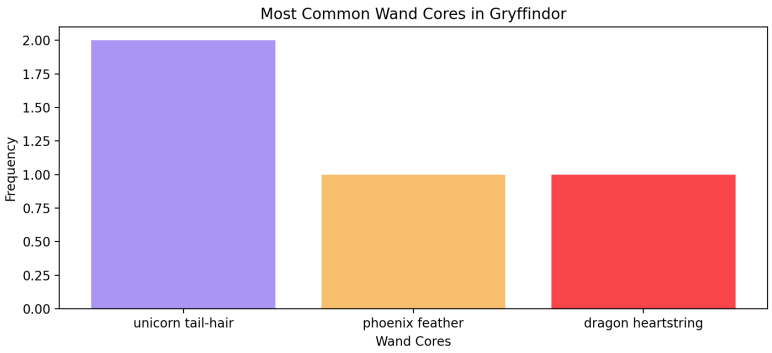
BeautifulSoup:

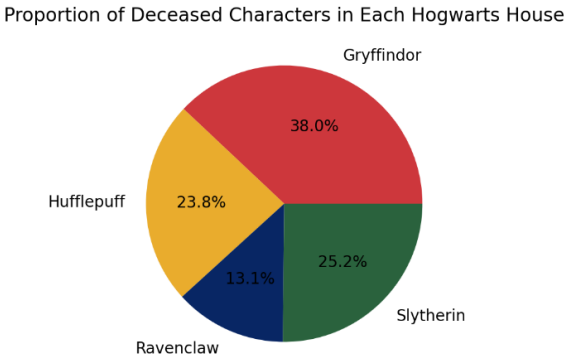
```
The average tomatometer of best movies that are certified fresh is 89.75.

How many times each genre shows up in Certfied Fresh Best Movies at Home:
Drama: 16
Sci-fi: 12
Horror: 8
Mystery: 12
Adventure: 8
Action: 8
Fantasy: 8
Romance: 4
Comedy: 12
Kids: 4
Documentary: 4
Music: 4
Biography: 4
```

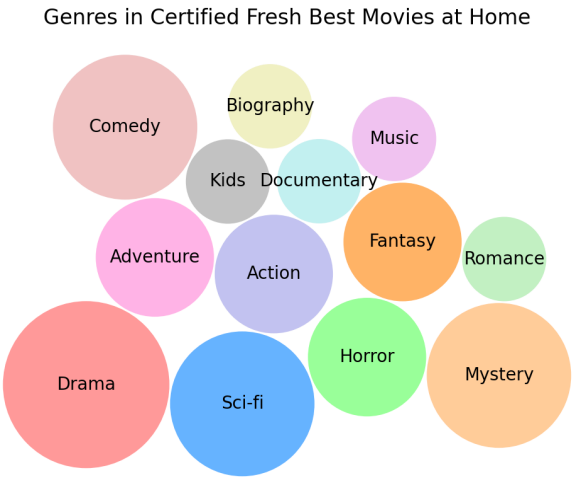
5. The visualization that you created (i.e. screen shot or image file) (10 points)

API:





BeautifulSoup:



6. Instructions for running your code (10 points)

API: Run the character_data file to create the characters table. Then, run the wand_data file to create the wands table. After both tables have been created, execute the data_processing

file to generate the text files with the calculations. The charts of the Harry Potter calculations will then be generated.

BeautifulSoup: Run the “data_database_rottentomato.py” file to collect the data from the Rotten Tomatoes website, put it into JSON files, then create the database. Afterwards, run the “calc_visual_rottentomato.py” file to create the text file with the calculations in it, in addition to generating the packed bubble chart.

7. Documentation for each function that you wrote. This includes describing the input and output for each function (20 points)

API

Function	**Input**	**Output**
call_api(url)	Url of the api url="https://hp-api.onrender.com/api/characters" url="https://hp-api.onrender.com/api/characters/students"	Character data in json format and character (book series) data in json format
query_function(statement: str)	a string representing a SQL query to execute on the "harry_potter.db" SQLite database.	A list of tuples, each row representing the query result. E.g. tuples of character ids
filter_to_25(characters: list, table: str):	Character list that contains dictionaries of the character/wand data Characters/wand table	The function returns a list of up to 25 filtered character dictionaries.
create_table_char():	No input	No output, but sets up the characters table
insert_character(characters: list): insert_wands(characters: list):	A list of dictionaries, each dictionary represents a character/wand, as input	No output, but inserts the data into the characters table
calculate_proportion()	No input	Returns a dictionary with house name as key and proportion of deceased characters in that house as value.

calculate_wand_material()	No input	Returns a dictionary with Gryffindor as key and count of core materials used by Gryffindor as value.
write_data_to_file(data, filename)	a dictionary of data (output of calculate_proportion) and a filename (calculations.txt)	Creates a text file with the calculations.
plot_deceased_by_house(deceased_by_house)	a dictionary of data (output of calculate_proportion)	Generates a pie chart of the proportions of deceased characters in each house
plot_wand_cores(wand_cores)	a dictionary of data (output of calculate_wand_material)	Generates a bar chart of the frequency of cores used by Gryffindor

BeautifulSoup

Function	**Input**	**Output**
create_soup(url)	The “Best Movies at Home” page URL, Certified Fresh filter applied to “Best Movies at Home” URL	BeautifulSoup object of input webpage
get_movie_titles(url)	The “Best Movies at Home” page URL, Certified Fresh filter applied to “Best Movies at Home” URL	Movie_titles_list: List of movie titles found from the BeautifulSoup object created by create_soup(url)
get_tomatometers(url)	The “Best Movies at Home” page URL	Tomatometers_list: List of Tomatometer scores found from the BeautifulSoup object created by create_soup(url)
get_genres(url)	The “Best Movies at Home” page URL	Movie_genres_list: List of movie genres found from the BeautifulSoup object created by create_soup(url)
get_data_ems_ids(url)	The “Best Movies at Home” page URL, Certified Fresh filter applied to “Best Movies at Home” URL	Data_ems_id_list: List of movie data_ems_id’s found from the BeautifulSoup object created by create_soup(url)

best_movies_json(json_name)	Name of output JSON file (“movie_info.json”)	JSON file of “Best Movies at Home” with movie title, Tomatometer score, genres, data_ems_id
best_movies_fresh(json_name)	Name of output JSON file (“certified_fresh_movies.json”)	JSON file of Certified Fresh “Best Movies at Home” with movie title, data_ems_id
setUpDatabase(database_name)	Name of output database file (“rotten_tomatoes.db”)	Tuple containing (cur, conn): conn makes a connection to the database (“rotten_tomatoes.db) and cur executes commands.
create_best_movies_table(cur, conn)	Cur,conn created by setUpDatabase(database_name)	Table in the database of the “Best Movies at Home” information
certified_fresh_table(cur, conn)	Cur,conn created by setUpDatabase(database_name)	Table in the database of the Certified Fresh “Best Movies at Home” information
main()	None	All the functions in each Python file
calculations(database, outfile)	Name of database to take values from (“rotten_tomatoes.db”) and name of output text file (“calculations_rottentomato.txt”)	Text file with calculations (“calculations_rottentomato.txt”)
visualization(calculations_file)	Name of calculations text file (“calculations_rottentomato.txt”) to take values from	PNG file of packed bubble chart of the proportion of how many times each genre shows up in the in the Certified Fresh movies in 'Best Movies at Home'

8. You must also clearly document all resources you used. The documentation should be of the following form (20 points)

Date	**Issue description**	**Location of Resource**	**Result (did it solve the issue?)**

4/17/23	Wanted to remove newline characters within genre text	https://stackoverflow.com/questions/13298907/remove-all-newlines-from-inside-a-string	Yes
4/17/23	Unsure of how to get “href” attribute value from elements	https://stackoverflow.com/questions/53911695/scrape-urls-using-beautifulsoup-in-python-3	Yes
4/18/23	Forgot how to format JSON data when putting info in	https://stackoverflow.com/questions/9170288/pretty-print-json-data-to-a-file-using-python	Yes
4/18/23	Some movies didn’t have genres, was getting an error when scraping	https://runestone.academy/ns/books/published/Win23-SI206/conditional/tryExcept.html	Yes
4/18/23	Forgot how to put info into a JSON file	https://runestone.academy/ns/books/published/Win23-SI206/bsoup/plan10.html	Yes
04/18/23	Data mismatch when trying to generate a primary integer ID	https://stackoverflow.com/questions/56973115/integrityerror-datatype-mismatch	No
04/19/23	Unsure of how to create a pie chart	https://www.python-graph-gallery.com/pie-plot-matplotlib-basic	Yes
04/19/23	Unsure of how to create a unique ID column	https://dev.mysql.com/doc/refman/8.0/en/example-auto-increment.html	Yes
04/20/23	Database locked	https://stackoverflow.	Yes

	when executing code	com/questions/151026/how-do-i-unlock-a-sqlite-database	
04/20/23	Unsure of how to show values in pie chart	https://matplotlib.org/3.1.1/gallery/pie_and_polar_charts/pie_features.html	Yes
04/20/23	Unsure of how to create bar chart	https://www.geeksforgeeks.org/bar-plot-in-matplotlib/	Yes
4/21/23	I was getting a packed bubble chart with an X and Y axis	https://matplotlib.org/stable/gallery/misc/packed_bubbles.html	Yes
4/21/23	Plt.savefig was returning a blank image	https://stackoverflow.com/questions/9012487/matplotlib-pyplot-savefig-outputs-blank-image	Yes