CS261-Data Structure and Algorithms 

Mid Project Proposal (Fall 2021)

**Proposer Details**

|  |  |
| --- | --- |
| Group Number | 62 |
| Registration Number of Group Members | 2020-CS-58  2020-CS-89 |

**Proposal Details**

|  |  |
| --- | --- |
| ***Project*** |  |
| Proposed Project Title | Cine-Scrap |
| Executive Summary | Web scrapping is mostly used by Business companies and industries usually with or without any technical knowledge. This project is entirely based upon the scrapping of movies. It works in a way that it extracts the relevant data provided by the user, scrapes it, and will move it on the list shown on GUI. The purpose of web scrappers is to collect as much data as it can. Same is the case with this project, It will scrap data of almost 1 million movies / TV shows. It facilitates the user as much as it can i.e. It will ask user for the genre of movie:   * Comedy * Sci- Fi * Horror * Comedy-Romance * Fantasy etc.   Every movie has separate attributes through which 7 of them will be scrapped. The attributes for scrapping are:   * Title * Director * Ratings * Duration * Year * Type (Language) * Genre * Cast   Name of the movie/show  It is GUI based project. The Libraries used for this purpose will be:   * **Pandas** (in case to assemble the data in DataFrame) * **BeautifulSoap** ( that enables us to parse the HTML files ) * **WebDriver** ( that simply allows us to automate the chrome) * **Chrome.Options** (allows to manipulate the different properties of chrome) * **Sleep (**will helps to suspend the code execution for a desired time, as we need to pause it for some time )   The Library for GUI is **PyQt (**This allows flexibility for handling GUI events and makes the coding easier). There are different types of cinemas and theatres. This project will ask the name of the Cinema in which movie is being theatered. Cinema can be DS or Marvel or any required.  This gui based project comsists of viewing a list of thousands and millions of shows with sorting and searching techniques. Also it will pause scrapping if the user press the pause button. |
| ***Business Case*** |  |
| Outline the business need for the project | Film Marketing Companies uses the movie scrappers to analyze the main factors, measured by film ratings that affect the success of films.  Some social media scrappers uses it to differentiate between bestmovies. |
| End user of the product | Mostly Film Marketing Companies uses the movie scrappers. |
| Motivation for Project | We have always been a keen fan of movies, and like to explore great films by viewing webpages like imdb, allmovies, tmdb and so on. |
| State the level of impact expected should the project proceed and implications of not proceeding | At this stage, the implications are very classic hence at operational level. Expected is that the implementation would proceed. |
| ***Technical Details*** |  |
| Name of Entity | Movies, shows |
| Attributes of Entity  (Minimum seven attributes/rows can be increased) | |  |  |  | | --- | --- | --- | | **Name** | **Data Type** | **Description** | | Movie title | string | It will provide the name of the movie. | | Ratings | float | It will give the movie ratings in numbers. | | Director | string | It will provide the user, the director of the required movie. | | Cast | String | It will show the whole cast of movie as in website. | | Year | int | It will inform the released year of the movie. | | Genre | string | It will tell which type of the movie is, either it is comedy or action or any other | | Duration | string | The total time of the movie | |
| Sample of Scrapping Source |  |
| Github Repository Link | https://github.com/eshatanvr/CS261F21PID62 |
| Sorting Algorithms | Selection sort  Insertion sort  Merge Sort  Bubble  Quick sort  Heap Sort  Radix Sort  Counting Sort  Bucket Sort |
| |  |  | | --- | --- | | **Algorithms** | **Description** | | Selection Sort | The selection sort algorithm sorts an array by repeatedly identifying the least unsorted element. | | Insertion Sort | In insertion sort algorithm Values from the unsorted part are picked and placed at the correct position. | | Merge sort | Merge sort repeatedly divides a list into sublists then merges those sublists into a sorted list. | | Quick Sort | Like Merge sort, Quick sort also falls into the category of divide and conquer approach but it sets a pivot. | | Bubble Sort | Bubble sorting goes through the list repeatedly, compare and swaps adjacent elements . | | Heap Sort | Heapsort splits its input into a sorted and unsorted region, and recursively reduces the unsorted by obtaining the largest element from it. | | Bucket Sort | Bucket sort divides the elements of an array into a number of buckets. Each bucket is then individually sorted. | | Radix sort | Radix sort sorts by using the digits of individual numbers. Sorting is done from least to most significant digits. | | Counting Sort | A counting sort is a sorting algorithm that sorts an array's elements by counting the number of occurrences . | | |
| Searching Algorithms | Linear search  Binary search  Jump seach |
| Searching Filters for each data type | If string : we can give a string and ask to search names having this string in title  Or by director’s name  Or genre  If integer: we can ask to display list of movies of that year |
| Multi-Level Sorting | For example: first it will sorted by titles alphabetically and then by it will get the movies sorted by years |
| Any other features | Not yet decided. |
| ***Interfaces for your project*** |  |
| |  |  |  | | --- | --- | --- | | **UI Component Name** | **Type** | **Purpose of UI Component/Other details** | | Profomance | Progress bar | To check the progress of scrapping | | Start button | Button | To start scrapping | | Stop button | button | To stop scrappnig | | Pause button | Button | To pause scrapping | | Table | Table | To display scrapped items | | Scroll bar | Scroll bar | To scroll down the scrapped items | | |