|  |  |
| --- | --- |
| A picture containing text, gear, metalware  Description automatically generated  **Project report** | **TITLE**:  **Movie Recommendation System**  **Instructor: Sir Shoaib Rauf**  **CL-2001** |

CL2001 PROJECT REPORT  
Movie recommendation system

FAST-NUCES BSCS FALL SEMESTER SECTION G

# Overview

## Group Members

|  |  |
| --- | --- |
|  | **21K-3170 Shaheer Badar (Leader)**  **21K-4676 Maryam Shahid**  **21K-4620 Laiba Tabraiz** |

## Project Description

|  |  |
| --- | --- |
|  | **An Offline Movie Recommendation System that helps user search for a movie according to his/her requirements and also recommends**  **according to their taste.** |

## CONCEPTS USED:

* **Binary Search Trees**

**Text

Description automatically generated**

**Text

Description automatically generated**

* **Singly Linked Lists**

**Text

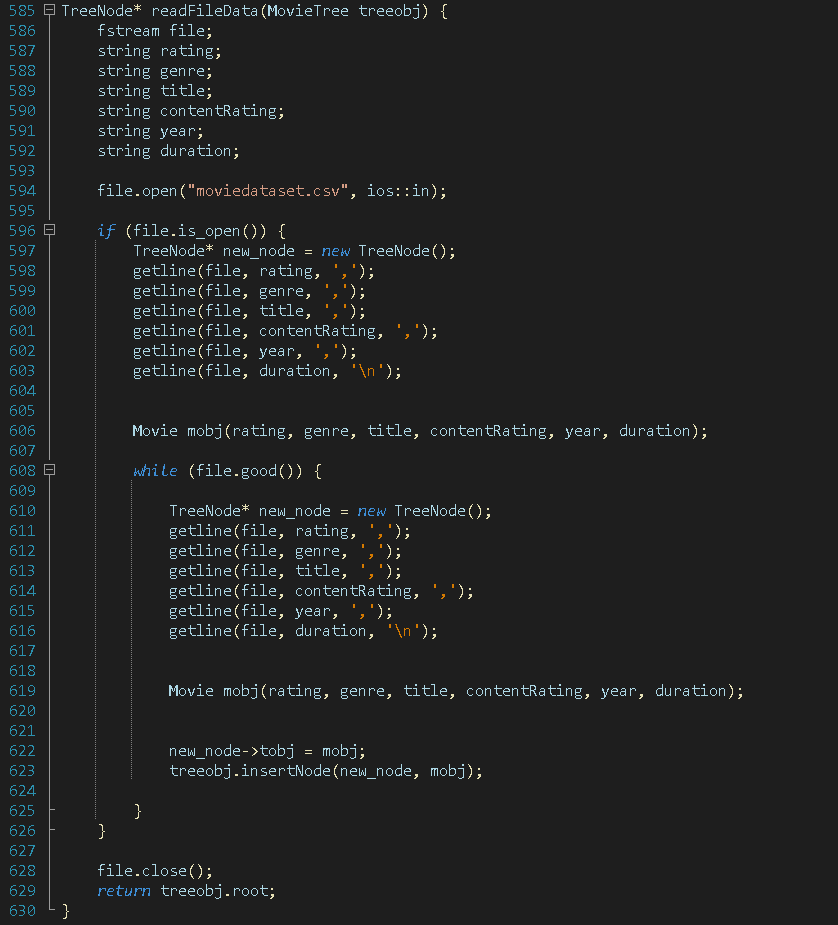
Description automatically generated**

* **Stacks**

**Text

Description automatically generated**

* **Filing**

****

* **String Streams**

**Text

Description automatically generated**

* **Classes and Objects**

**Text

Description automatically generated**

## LIBRARIES USED:

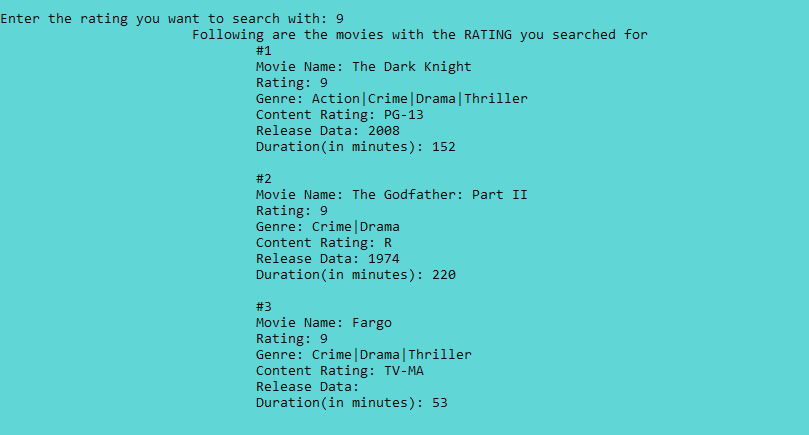
* **#include<iostream>**
* **#include<string>**
* **#include<sstream>**
* **#include<stdlib.h>**
* **#include<fstream>**
* **#include<stack>**
* **#include<bits/stdc++.h>**
* **#include<dos.h>**
* **#include<conio.h>**
* **#include<windows.h>**

**Text

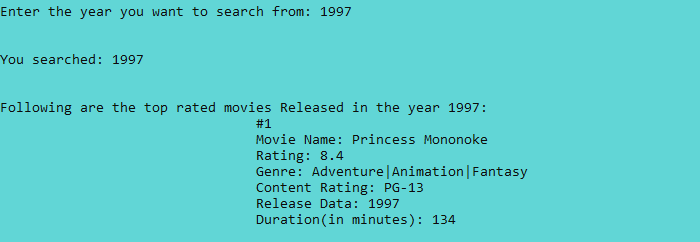
Description automatically generated**

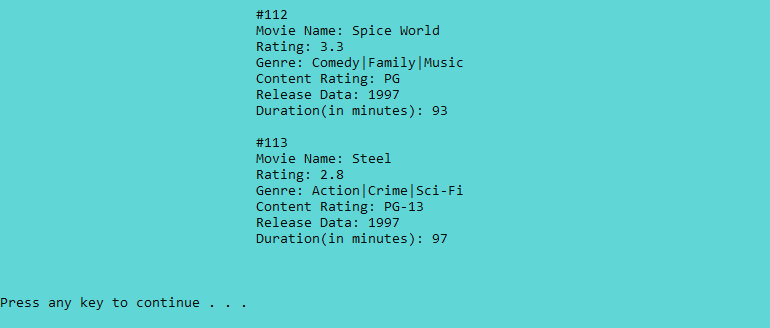
## FUNCTIONS:

* **Setter and Getter Functions of Class Movie.**
* **readFileData( ) –** filing function which reads data from file and stores in bst.
* **printTest( ) –** Testing function of class Movie used to check tree data.
* **Constructors (of all classes) –** Parameterized Constructors
* **appendNode( )** – adding duplicated data based on ratings in linked list.
* **printList( ) –** testing function of linked list.
* **isTreeEmpty( ) –** returns true or false if tree is empty or not respectively.
* **insertNode( ) –** add data to tree node.
* **Ratingsearch( ) –** search rating wise.

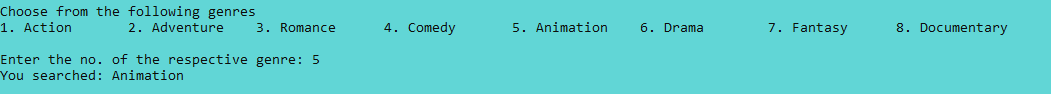


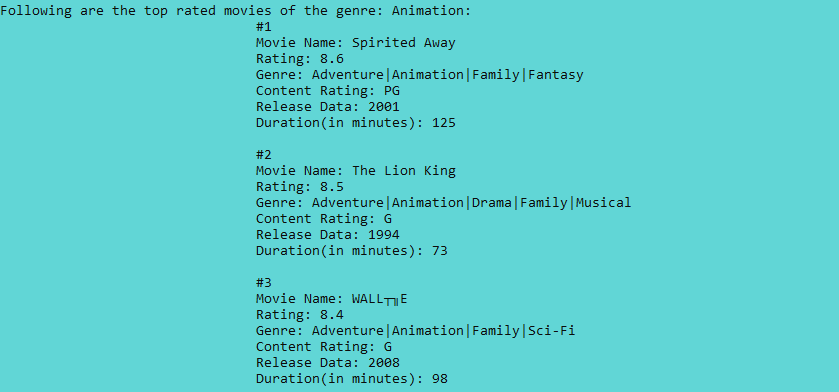
* **Yearsearch( ) –** search year wise.





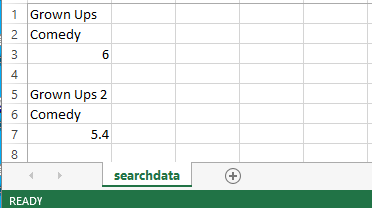
* **genresearch( ) –** search genre wise.

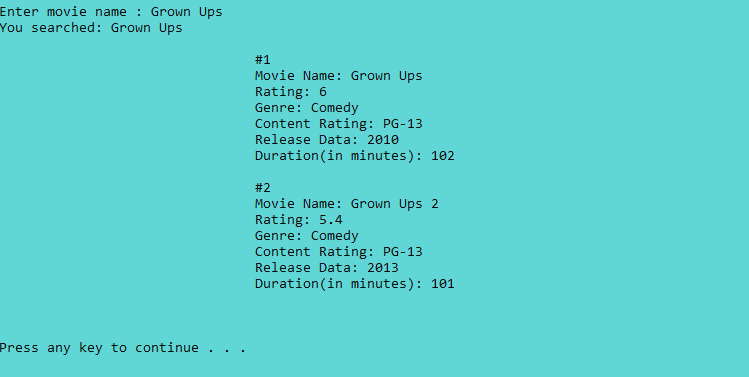




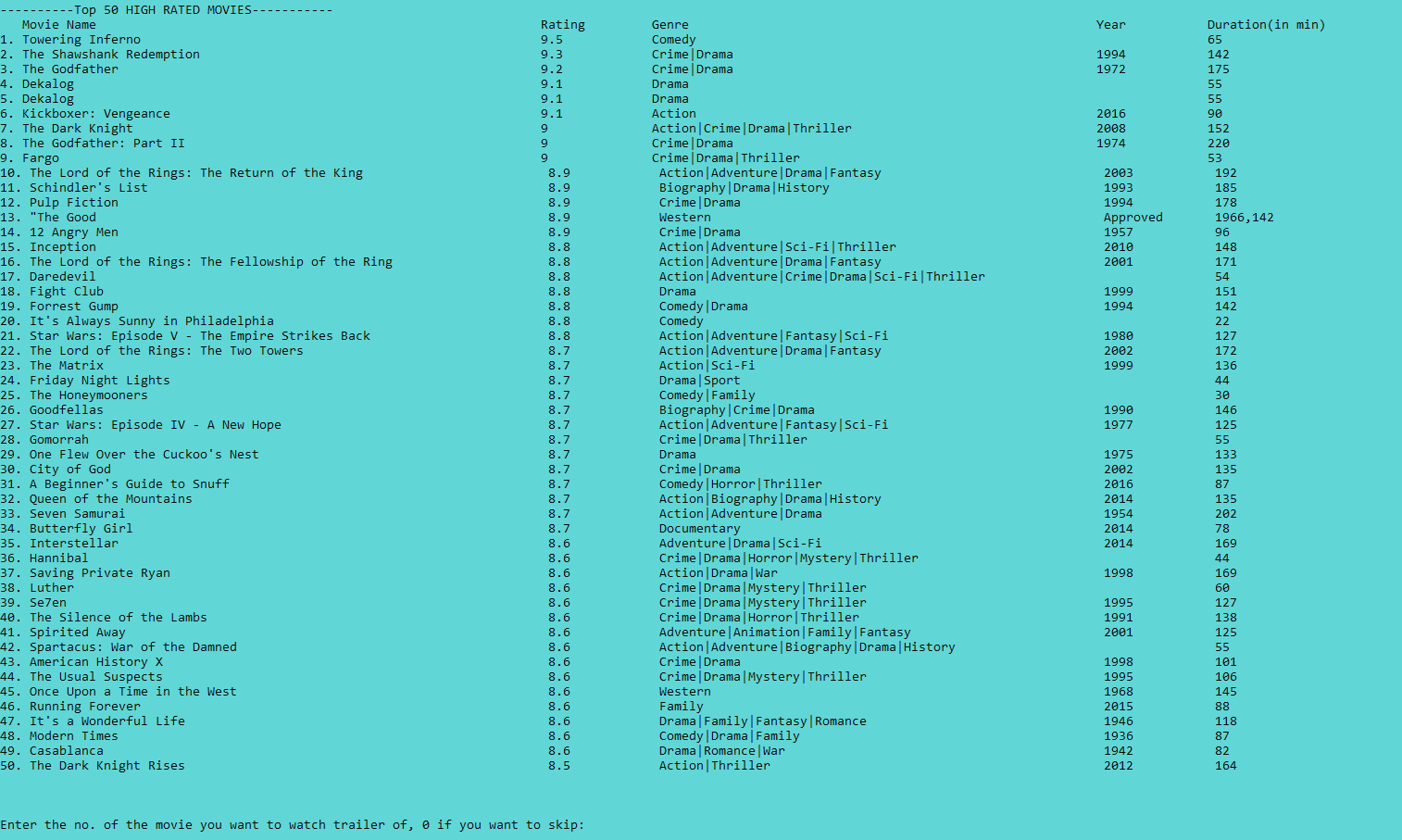
* **titlesearch( ) –** search using title name.

-stores the genre and title of searched movie in excel file for later recommendation

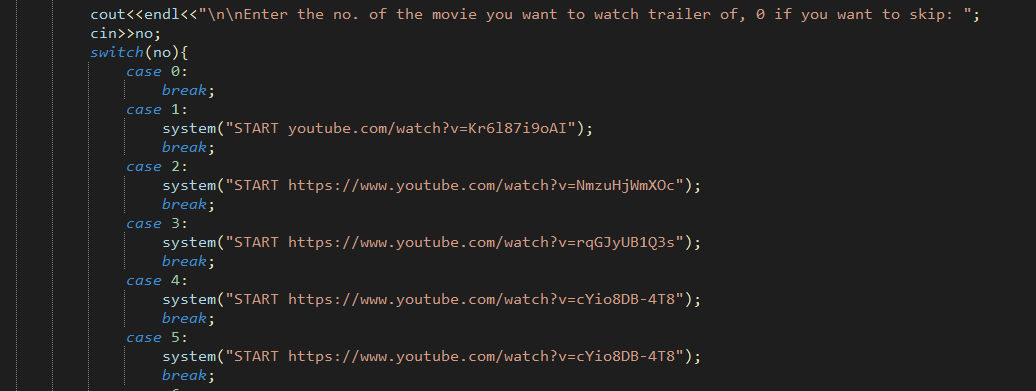




* **top50rated( ) –** to get top 50 rated movies.



-trailer can be watched if the user wants too

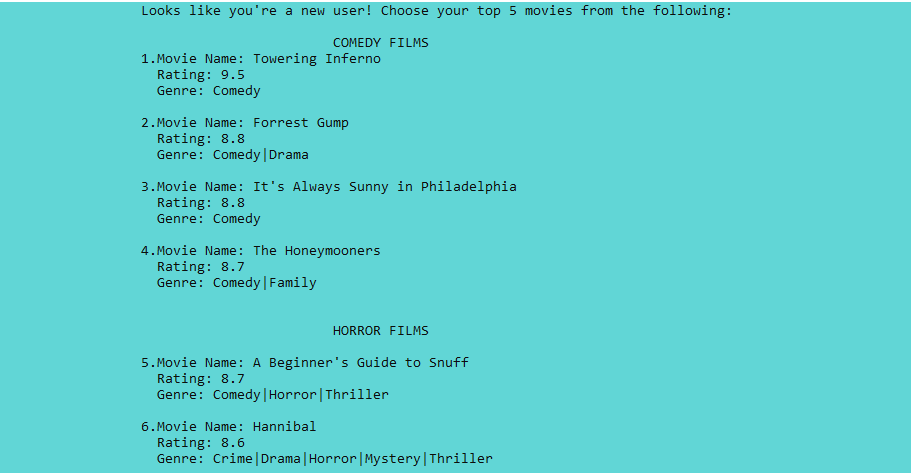


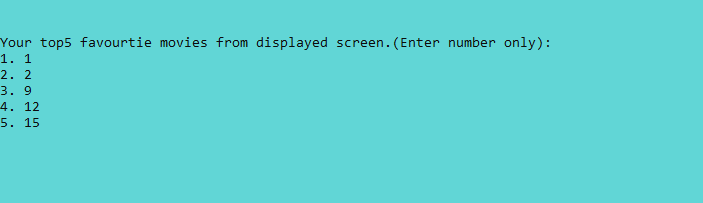
* **displayMovieToSelect( ) –** display prompt for user

-displays only when new account is created.

-displays top 4 movies for basic 5 genres.

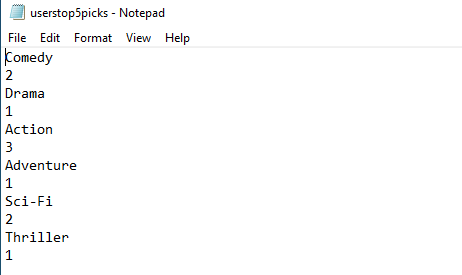
-asks the user to select their top5 favorite for recommendation later.





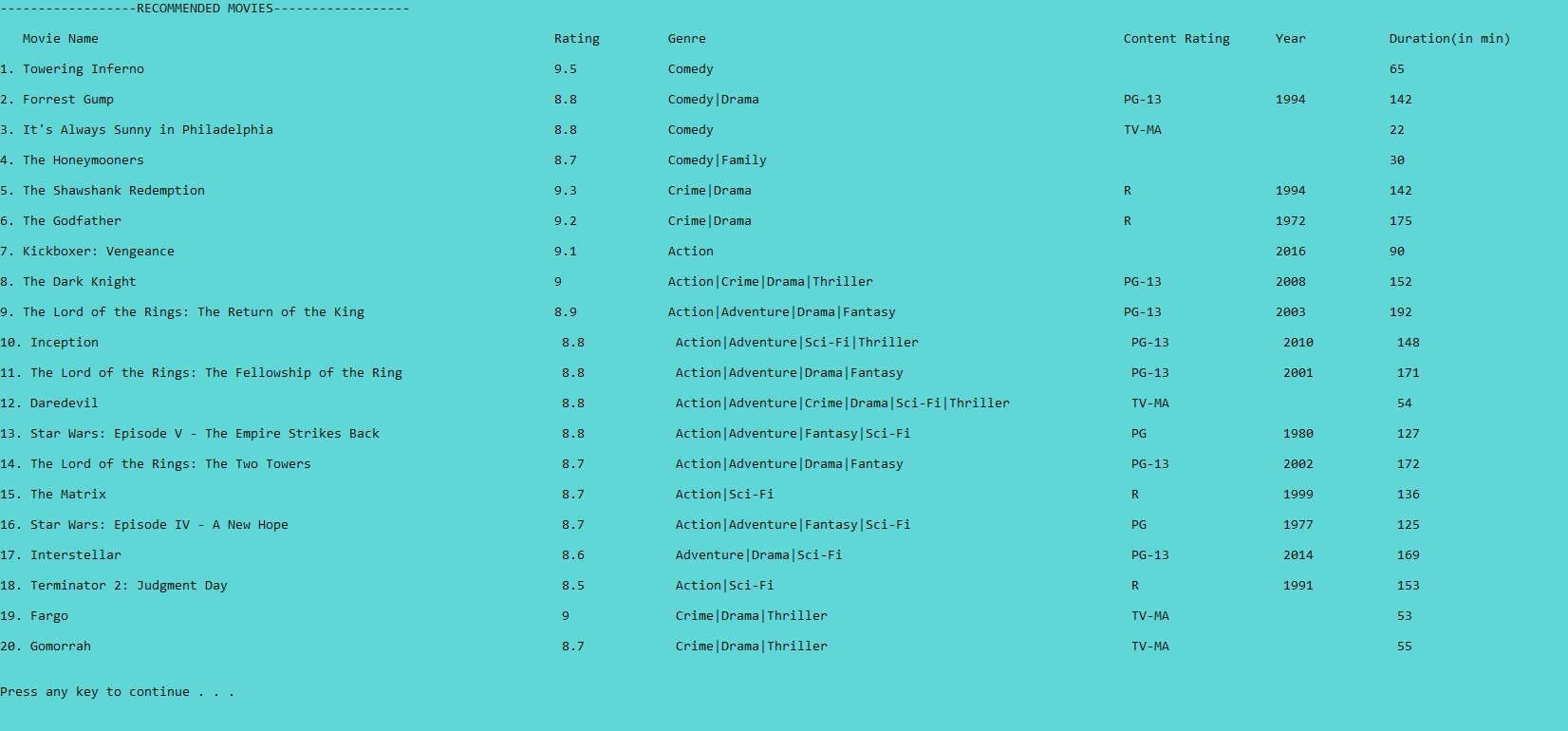
* **pickupMovie( ) –** function for user to select movie

-maintain the record for genre and each genre count of selected movies in text file.

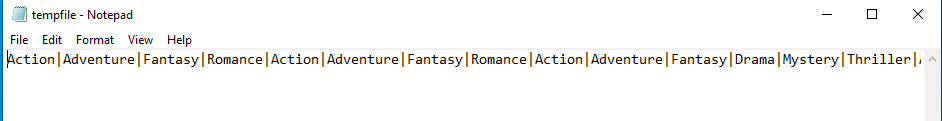


* **recommend( ) –** the function for recommending

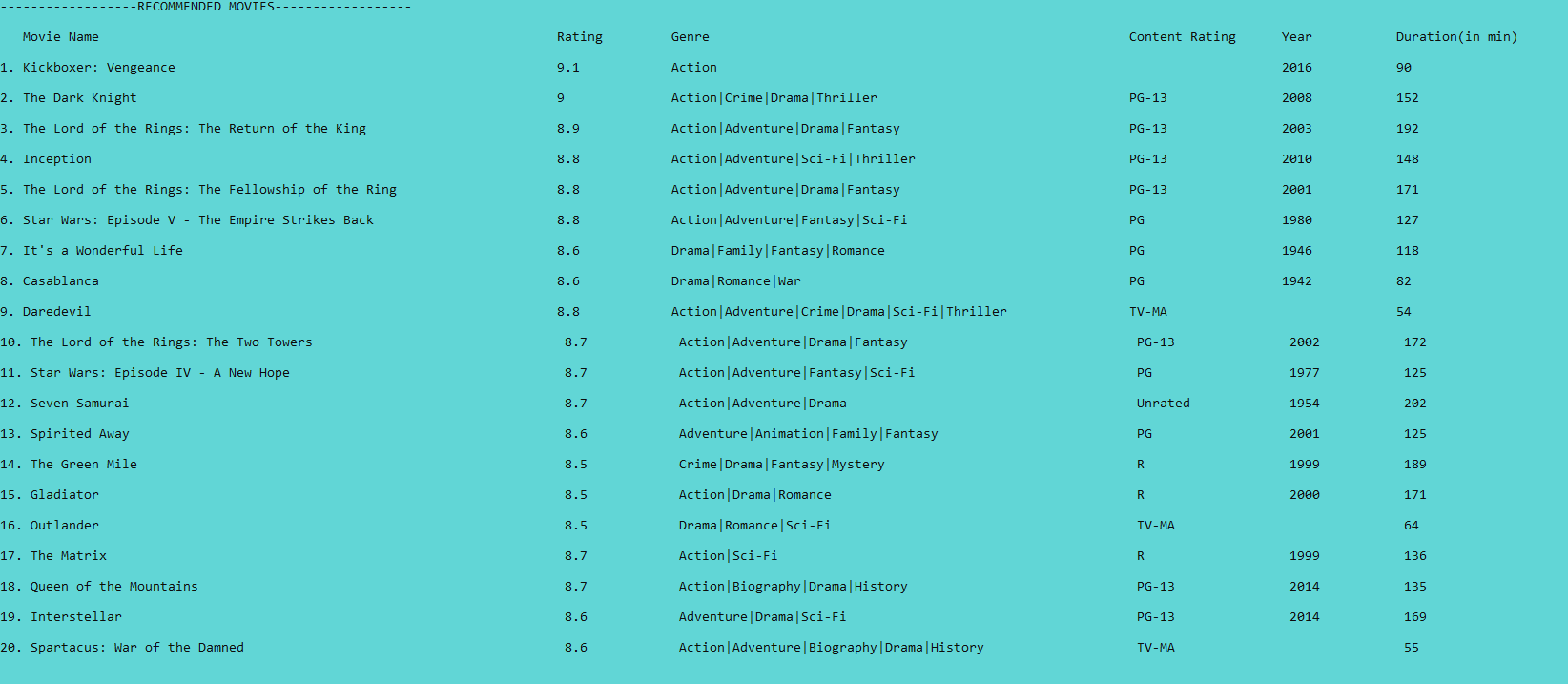
**-**on the basis of selected genres



**-**on the basis of the genre of most recent searched movies







* **frontEnd( ) –** the main menu of the system
* **main( )**

## VARIABLES:

bool flag;

string displayed\_mtitle[25];

string displayed\_mgenre[25];

string rec\_genre[25];

int gcount[25] = {0};

int gindex= 0;

static int displayed\_mcount;

static int displayed\_mcount2;

static int displayed\_mcount3;

static int cc;

static int i;

string rating;

string genre;

string title;

string contentRating;

string year;

string duration;

int option1, option2 = 1;

bool duplicate;

fstream userfile, searchfile, tempfile,top5picksfile,watchlist

string username, pass, tempu, tempp, temptitle, tempgenre[5], tempg;

string genrearray[8]={"Action","Adventure","Romance", "Comedy","Animation","Drama","Fantasy","Documentary"};

int length1, length2;