DS PROJECT REPORT

Submitted to Ms. Mubashra

Group Members:

Noor Fatima (20K-0406)

Arooba Moin (20K-0213)

Bismah Akram(20K-0449)

Department of Computer Science
National University of Computer and Emerging Sciences-FAST
Karachi Campus

Table of Contents

INTRODUCTION	**2**
PROBLEM ANALYSIS	**3**
METHODOLOGY	**5**
IMPLEMENTATION	**6**
RESULTS	**7**
DATASET	**19**
REFERENCES	**20**
CONCLUSION	**20**

INTRODUCTION

This project is part of our Data Structures Course, the theme we were given was ‘Big Data’. We chose to use 2 datasets, we picked a Movies dataset, in which we chose to analyze information of each movie, and then gave the user the option to search a given movie from a massive database using the Movie ID, User ID, Movie Name, Genre, Rating, its Overview and then also delete movies simultaneously.

PROBLEM ANALYSIS

The user is given an overview of the menu, they are then asked what option they would like to select whether they would like to search for a movie using its user id, movie id, movie name or genre. They can also search for the highest rated movies of their chosen genre.

1. **Search by User's ID**

   By entering user's id number, a list of movies with the entered id number will be displayed. Enter the movie name from the list for that movie's details.

2. **Search by Movie's ID**

   By entering movie's id number, detail of that movie will be displayed.

3. **Search by Movie Name**

   By entering movie's name, details of that movie will be displayed.

4. **Search by Genre**

By entering any genre of your choice, details of all movies of the entered Genre will be displayed.

5. **Top five Rated Movies**

By choosing this option, top five movies in the list will be displayed along with their ratings. By entering the movie name from that list, details of that movie will be displayed.

6. **Search Highest Rated Movies by Genre**

By entering any genre of your choice, the highest rated movies in that particular genre will be displayed. By entering your preferred movie name, details of that movie will be displayed.

7. **Delete by Movie ID**

Users can also delete a movie by entering the movie id.

METHODOLOGY

A methodology is a set of standards and applications that guide you in organizing your projects to ensure their optimum performance. There are many different methodologies that can be adopted when working on a project depending on the budget, team size, flexibility, and timeline. Our group chose the Critical chain project management (CCPM) methodology.

This methodology consists of identifying and scheduling all of the critical tasks that compromise the project, as well as their dependencies. The critical path, longest sequence of critical tasks, can be visualized using PERT and Gantt charts.

This methodology was perfect for our project since our project had many depending modules, and we needed a visual map of the sequence. We also had a strict deadline and needed buffers to test out our ideas. These conditions were met by the CCPM methodology.

A Gantt chart is a horizontal bar chart developed as a production control tool, it is frequently used in project management, a Gantt chart provides a graphical illustration of a schedule that helps to plan, coordinate, and track specific tasks in a project. Below is the Gantt chart created for the database on “Project By 20k-0213 20k-0409 20k-0449”, where the cells shaded in green show the week number.

Identify Requirements											
Produce design											
Write Program code											
Test Modules											
Write Program code											
Add Data Files											
Test Modules											
Combine Modules and testing											
Write Report											
Week Number	1	2	3	4	5	6	7	8	9	10	11

IMPLEMENTATION

For this project the programming language used is C++. It can be used to develop operating systems, browsers, games and so on. C++ supports different ways of programming like procedural, object-oriented, functional, and so on. This makes C++ powerful as well as flexible.

The header files we used in this project are iostream, conio.h, fstream, string and stack. The iostream stands for standard input-output stream. This header file contains definitions to objects like cin, cout, etc. Conio.h is a C header file used mostly by MS-DOS compilers to provide console input/output. fstream, it represents both output Stream and input Stream. So it can read from files and write to files. The stack header file helps us implement the stack data structure in our program.

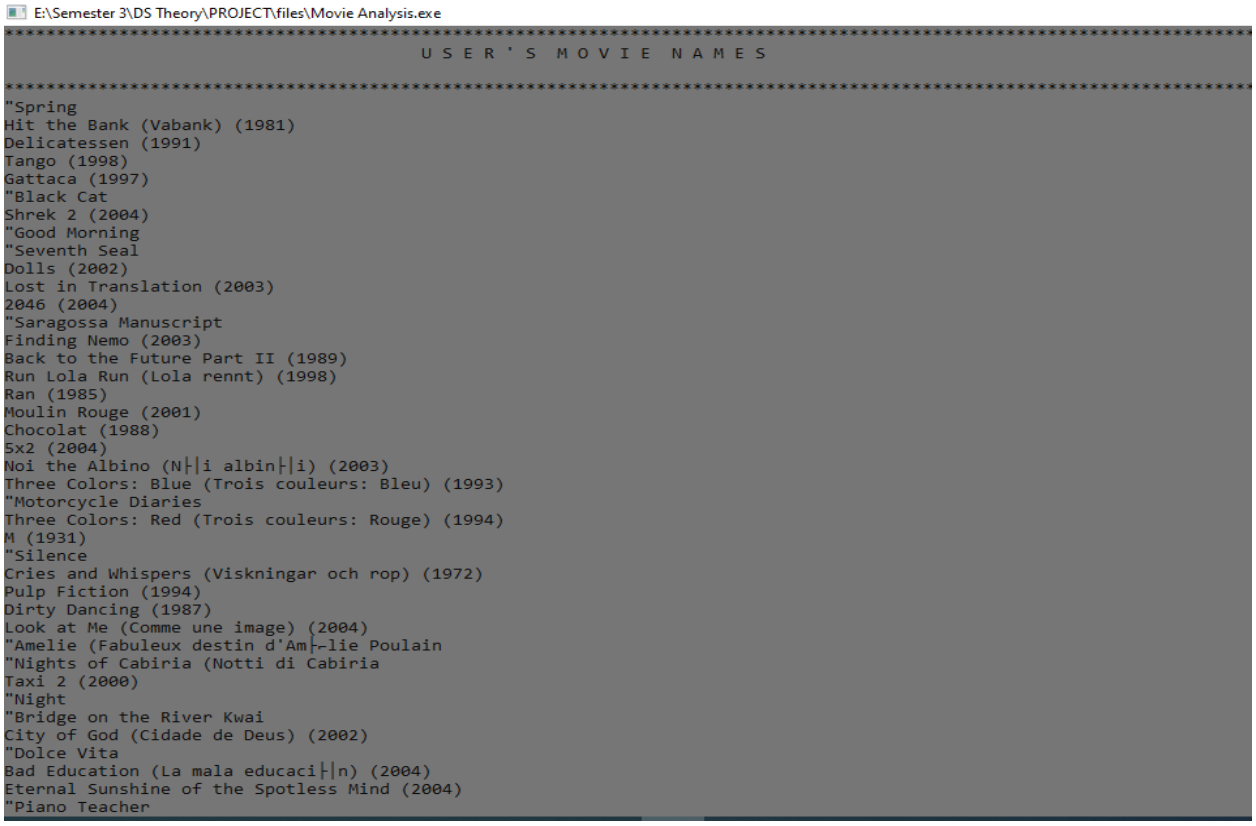
The data structure we are mainly using for searching and deletion in AVL Trees. This allows us to insert, search and delete data efficiently with a speed of O[logn]. Using AVL, the tree will self balance using rotations. Insertion was done using Movie ID as it is unique in our dataset and AVL allows unique data. For unique data, we have searched using recursive methods and for the data with duplications we have used stack by Depth First Search.

RESULTS

Menu

1. Searching by User ID:

List of Movies By User ID 1:



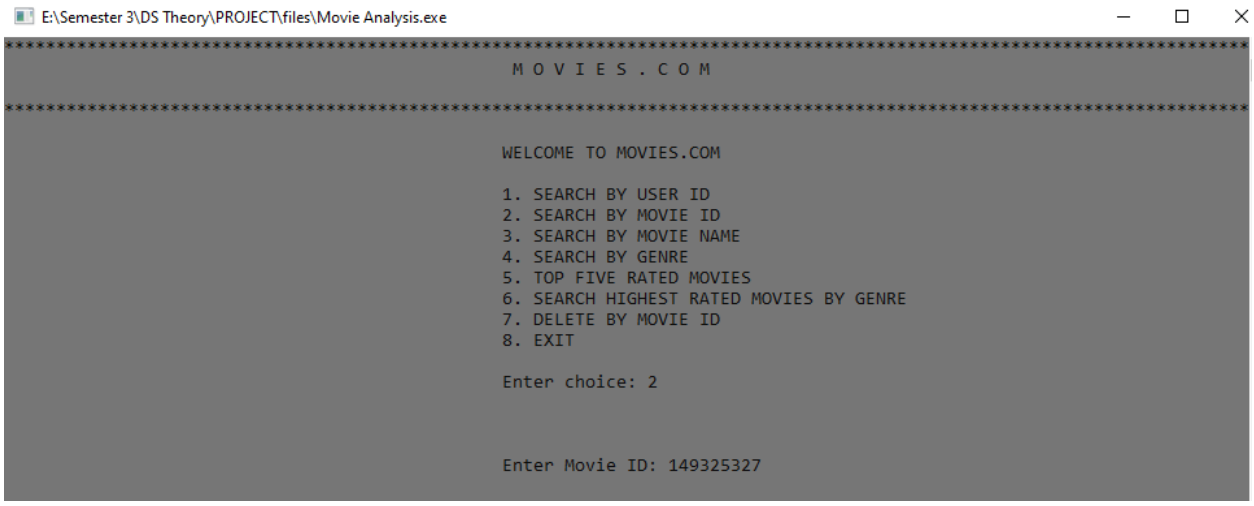
```
E:\Semester 3\DS Theory\PROJECT\files\Movie Analysis.exe
Eternal Sunshine of the Spotless Mind (2004)
"Piano Teacher
Swann in Love (Un amour de Swann) (1984)
Singin' in the Rain (1952)
Wild Strawberries (Smultronst├ñllet) (1957)
"Lord of the Rings: The Two Towers
Back to the Future Part III (1990)
Winter Light (Nattvardsg├ñsterna) (1963)
Fanny and Alexander (Fanny och Alexander) (1982)
Pirates of the Caribbean: The Curse of the Black Pearl (2003)
Requiem for a Dream (2000)
"M. HulotΓÇÖs Holiday (Mr. Hulot's Holiday) (Vacances de Monsieur Hulot
Naqoyqatsi (2002)
In the Mood For Love (Fa yeung nin wa) (2000)
Underground (1995)
"Miracle of Marcelino
Talk to Her (Hable con Ella) (2002)
"Strada
"NeverEnding Story
"Spanish Apartment
Port of Shadows (Quai des brumes) (1938)
"Virgin Spring
Through a Glass Darkly (S├Ñsom i en spegel) (1961)
Hour of the Wolf (Vargtimmen) (1968)
"Good bye
The Magician (1958)
"Passion of the Christ
Teddy Bear (Mis) (1981)
"Idiots
"Very Long Engagement
"Barbarian Invasions
Persona (1966)

                                        Enter the Movie Name for details: Persona (1966)
```

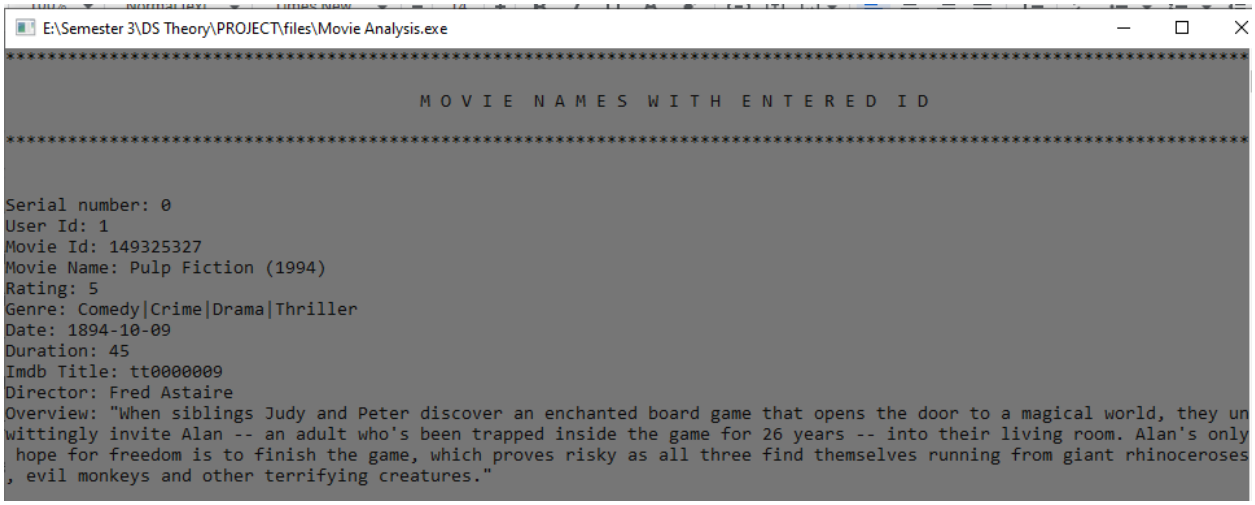
Movie details of entered Movie:

```
E:\Semester 3\DS Theory\PROJECT\files\Movie Analysis.exe
**************************************************************************************************************

                                    D E T A I L S  O F  T H I S  M O V I E

**************************************************************************************************************


Serial number: 47
User Id: 1
Movie Id: 93429989
Movie Name: Persona (1966)
Rating: 3.5
Genre: Drama
Date: 21-03-15
Duration: 195
Imdb Title: tt0004972
Director: Peter Lorre
Overview: "A prudish woman, working on tenure as a literacy professor at a large urban university, finds herself strangely attracted to a free-spirited, liberal woman w
hom works at a local carnival which comes to town."

                                    Press any key to continue.
```

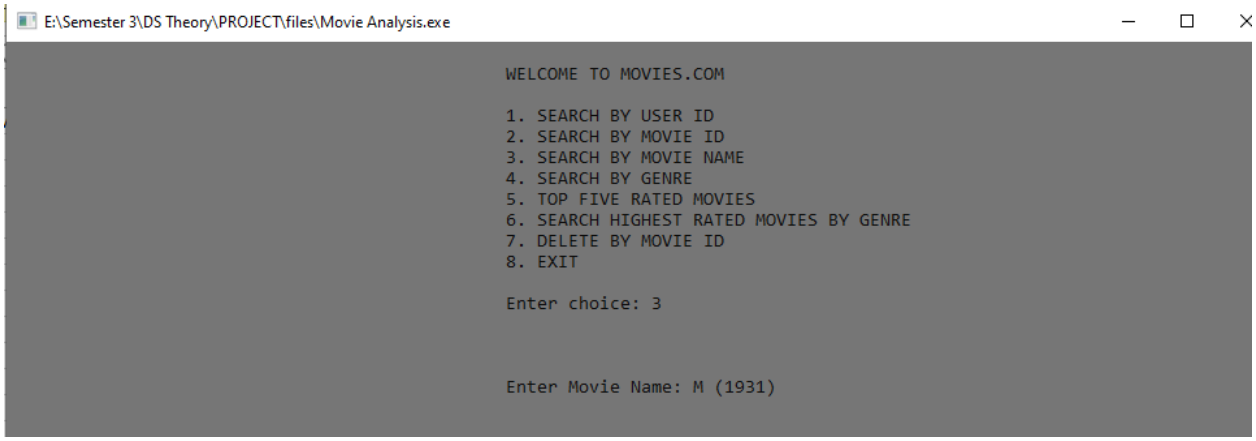
2. Searching by Movie ID:

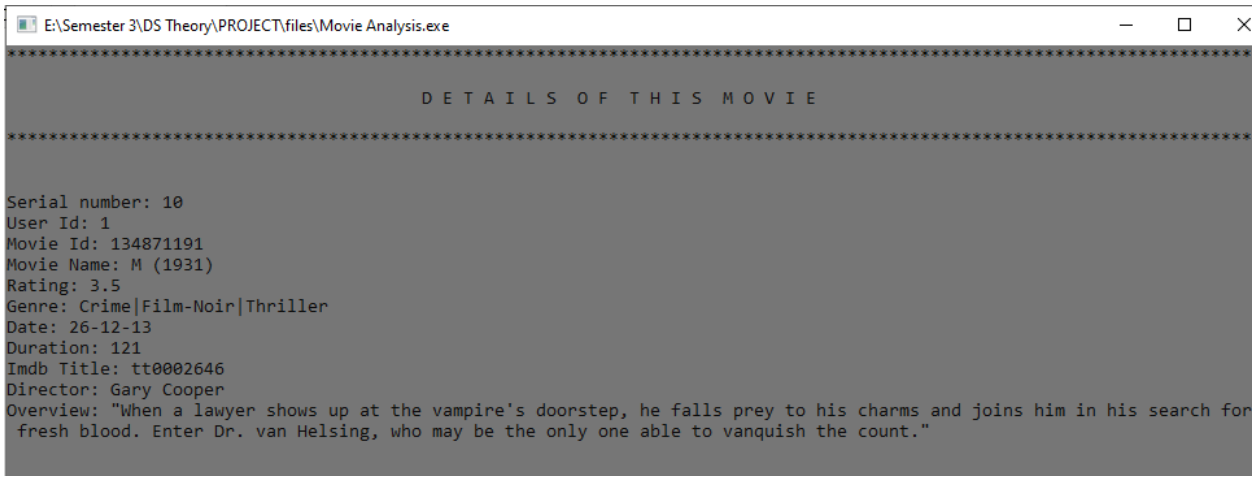
Movie details of entered movie id:





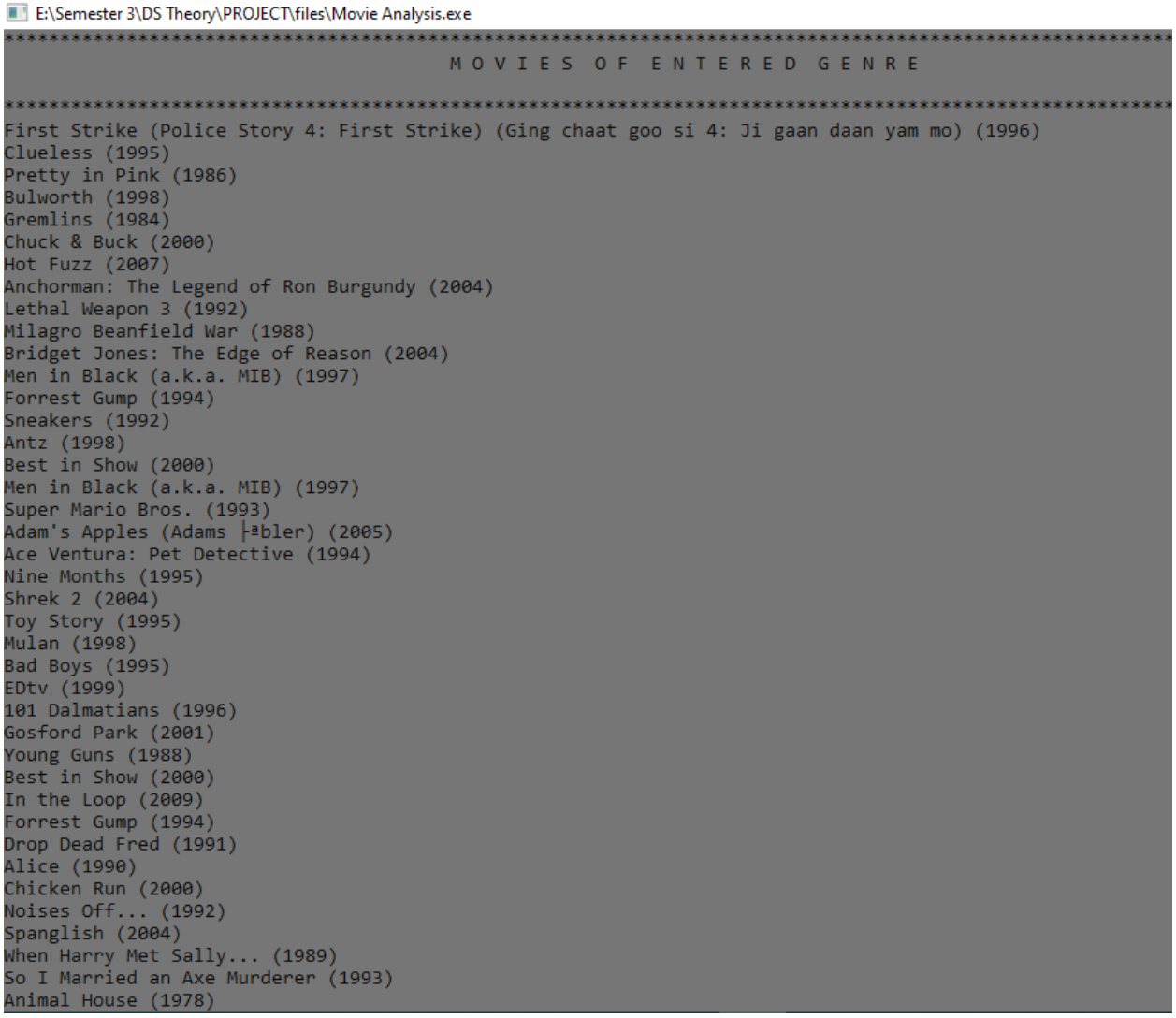
3. Searching by Movie Name:





4. Searching by Genre:

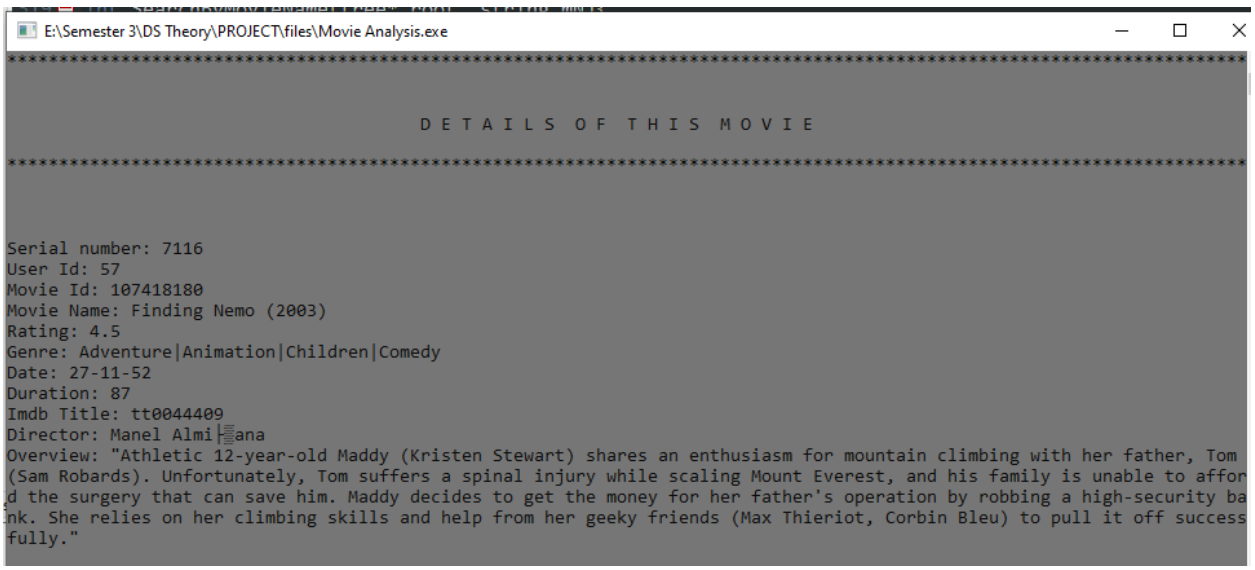
List of Movies of Entered Genre



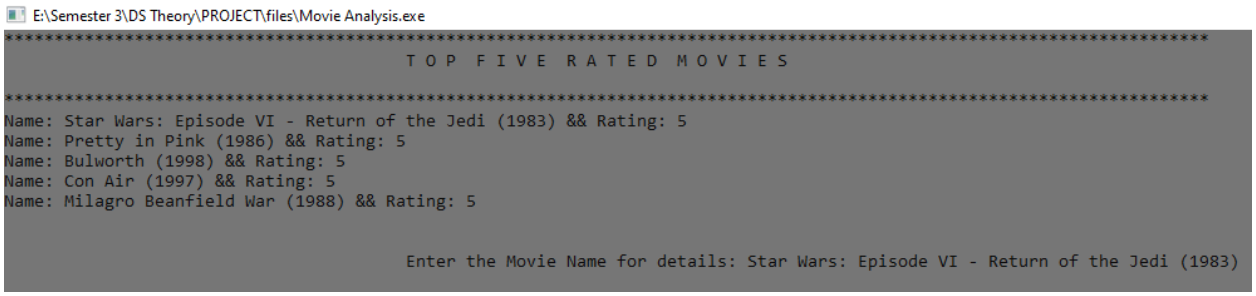
E:\Semester 3\DS Theory\PROJECT\files\Movie Analysis.exe

```
Shark Tale (2004)
Monty Python and the Holy Grail (1975)
Ed Wood (1994)
Miss Congeniality (2000)
Trainspotting (1996)
American Pie (1999)
Kick-Ass (2010)
Fargo (1996)
In Bruges (2008)
Four Weddings and a Funeral (1994)
Flirting With Disaster (1996)
Back to the Future (1985)
Lemony Snicket's A Series of Unfortunate Events (2004)
Michael (1996)
Shrek (2001)
21 Jump Street (2012)
Bad Santa (2003)
Blast from the Past (1999)
Camp Rock (2008)
What We Do in the Shadows (2014)
Junior (1994)
Finding Nemo (2003)
Back to the Future Part II (1989)
Men in Black (a.k.a. MIB) (1997)
Monty Python and the Holy Grail (1975)
Borat: Cultural Learnings of America for Make Benefit Glorious Nation of Kazakhstan (2006)
This Is Spinal Tap (1984)
Party Girl (1995)
Bring It On (2000)
French Kiss (1995)
Pulp Fiction (1994)
Cat Ballou (1965)
Drop Dead Gorgeous (1999)
Rat Race (2001)
Finding Nemo (2003)
You've Got Mail (1998)
Bringing Up Baby (1938)
Addams Family Values (1993)
Napoleon Dynamite (2004)



                                    Enter the Movie Name for details: Finding Nemo (2003)
```

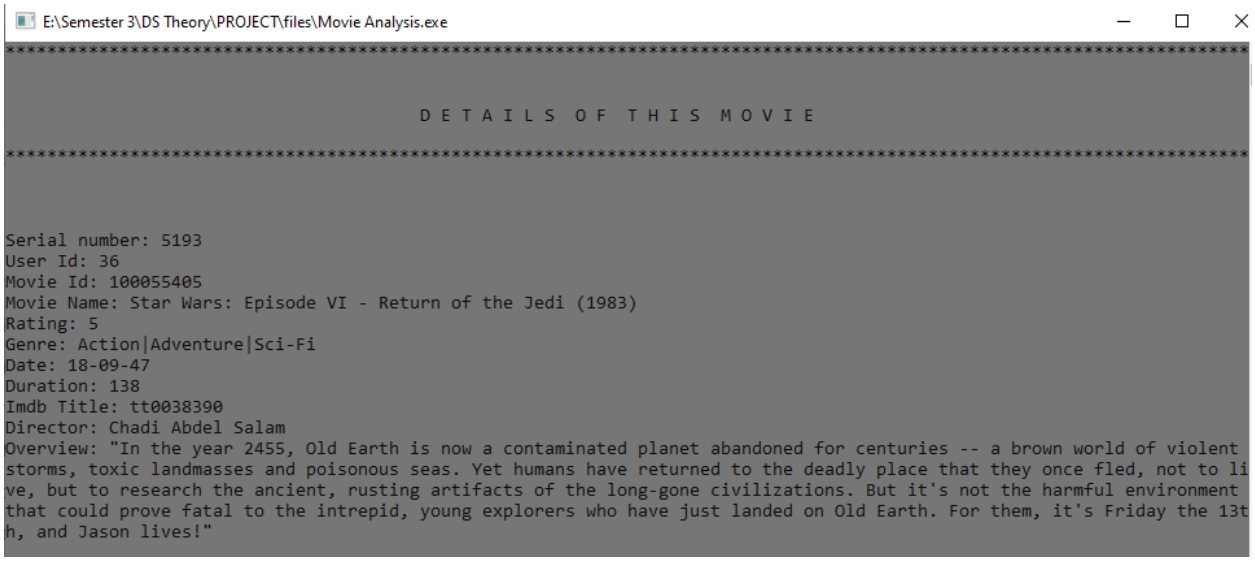
Details of entered Movie:



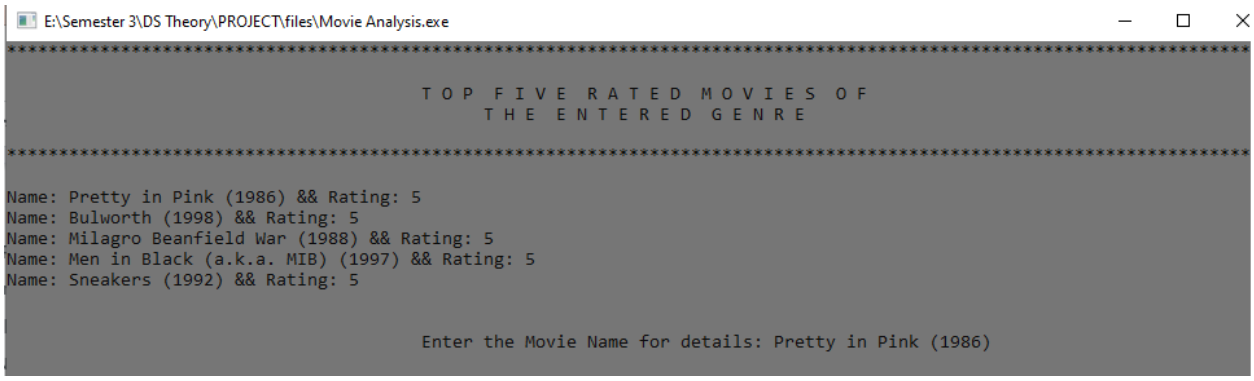
5. Top Five Rated Movies:



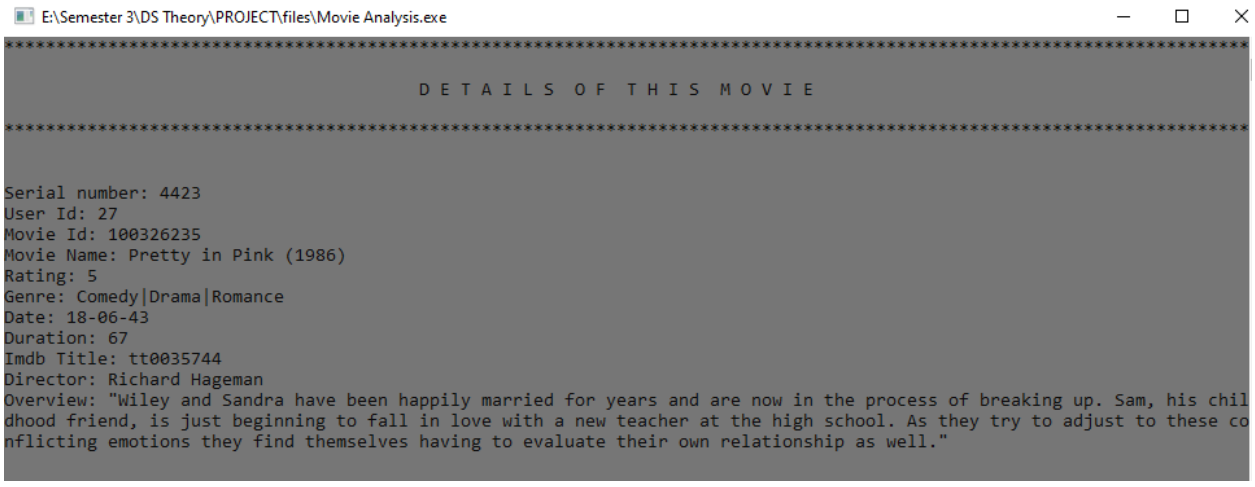
Movie details of entered movie name:



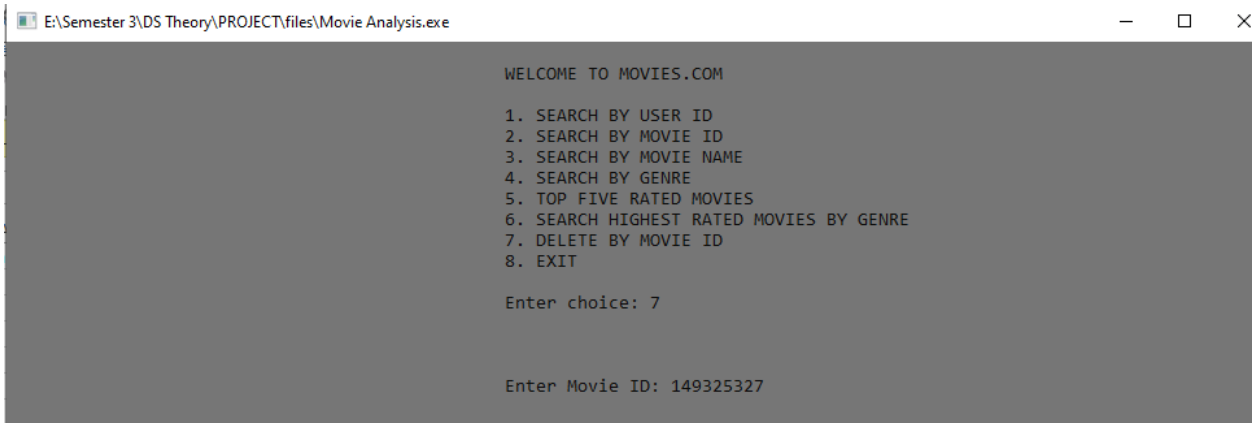
6. Searching Highest Rated Movies by Genre:

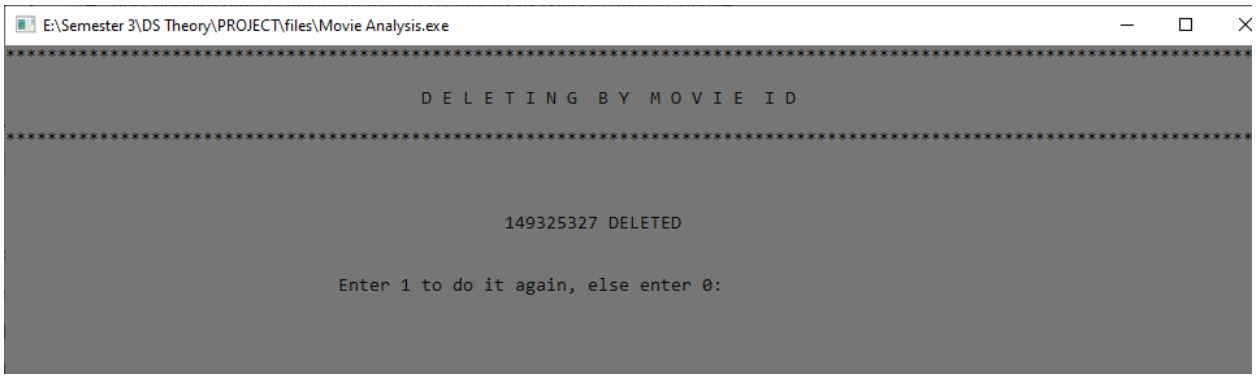


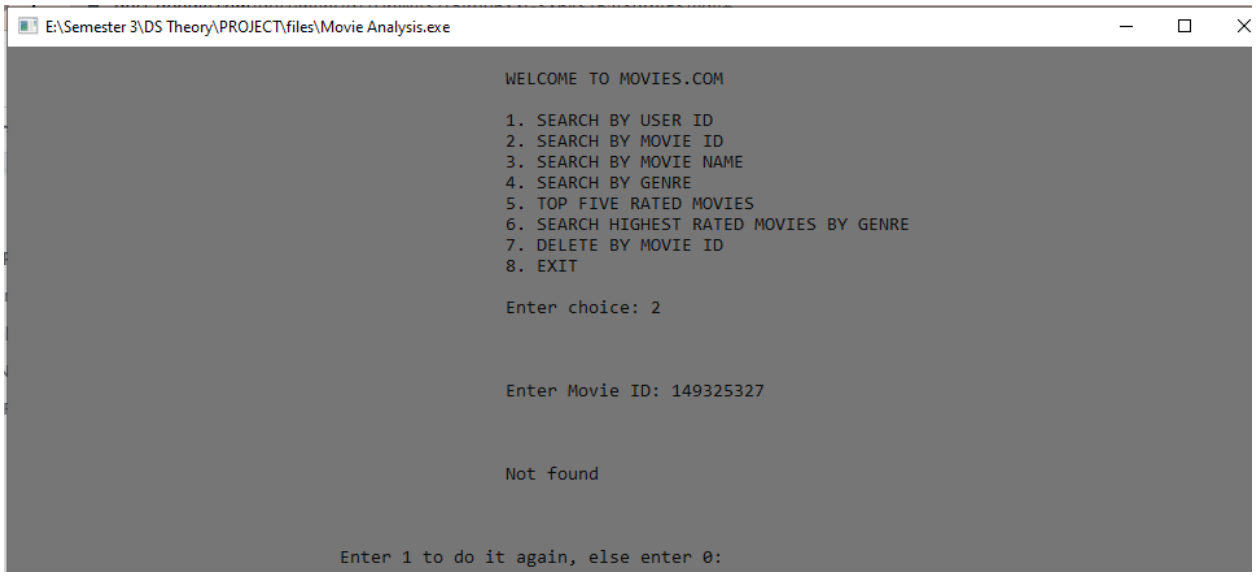
Movie details of entered movie:



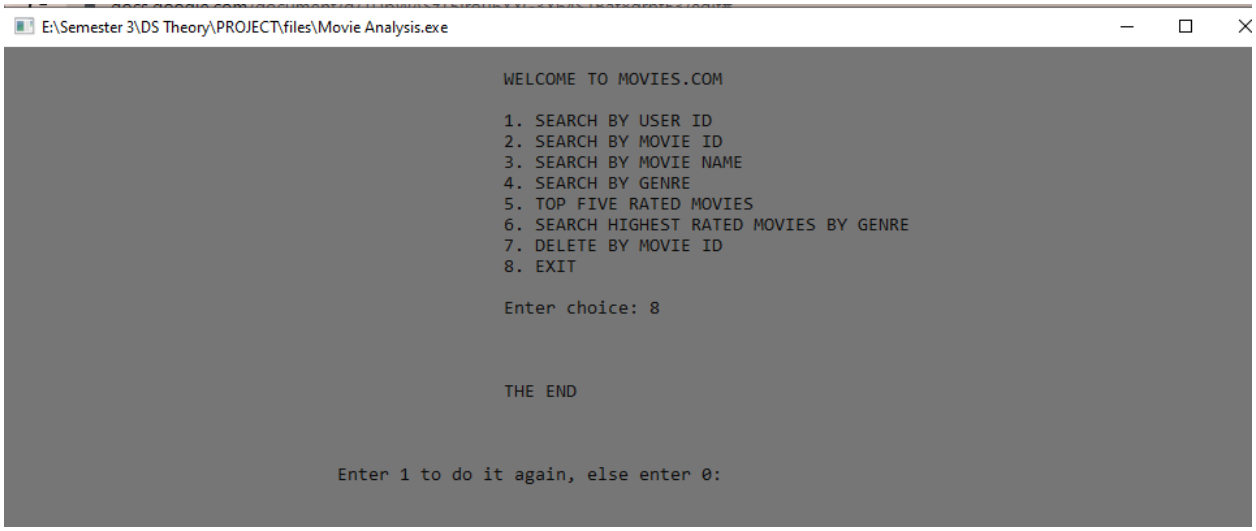
7. Deleting the movie by movie id:




E:\Semester 3\DS Theory\PROJECT\files\Movie Analysis.exe
D E L E T I N G B Y M O V I E I D
149325327 DELETED
Enter 1 to do it again, else enter 0:


E:\Semester 3\DS Theory\PROJECT\files\Movie Analysis.exe
WELCOME TO MOVIES.COM
1. SEARCH BY USER ID
2. SEARCH BY MOVIE ID
3. SEARCH BY MOVIE NAME
4. SEARCH BY GENRE
5. TOP FIVE RATED MOVIES
6. SEARCH HIGHEST RATED MOVIES BY GENRE
7. DELETE BY MOVIE ID
8. EXIT
Enter choice: 2
Enter Movie ID: 149325327
Not found
Enter 1 to do it again, else enter 0:

8. Exit


E:\Semester 3\DS Theory\PROJECT\files\Movie Analysis.exe
WELCOME TO MOVIES.COM
1. SEARCH BY USER ID
2. SEARCH BY MOVIE ID
3. SEARCH BY MOVIE NAME
4. SEARCH BY GENRE
5. TOP FIVE RATED MOVIES
6. SEARCH HIGHEST RATED MOVIES BY GENRE
7. DELETE BY MOVIE ID
8. EXIT
Enter choice: 8
THE END
Enter 1 to do it again, else enter 0:

DATASET

For this program, we are using two datasets that contain more than a million movie details. Combined together, these dataset have attributes like the unique ID, the name of the movie, release date, MOVIE ID, Rating, Duration, Director name and, an overview of a particular movie. This data is randomly sorted and the user cannot look up any useful information in this dataset without filtering so our software filters the data for the users and only displays the information which they want.

1) INSERTION:

All the movies are inserted in the BST. We have implemented AVL in BST which works on a self-balancing Binary Search Tree. It reduces the time complexity of all the operations which will be performed on the BST like insertion, deletion, and searching. All of these operations will work on O(logn) speed.

2) DELETION:

This functionality can only be done using Movie ID. The application will allow them to delete any movie based on its unique Movie ID.

3) SEARCHING:

Users are allowed to search in the data set via multiple options which are all stated in the problem analysis section. Due to the AVL tree, users can search data at a speed of O(logn).

REFERENCES

- www.kaggle.com

CONCLUSION

By using our concepts of object oriented programming and Data Structures, we completed our movie analysis by filtering the dataset and we showed the user what movies are available in the dataset by searching the movie using the ID and deleting it as well.