SIT 102- INTRODUCTION TO PROGRAMMING:

PROJECT 2

SUMMARY:

In this task we prepared a c++ code and this code allows users to manage a movie database by performing various operations such as adding, removing, modifying movies, and displaying movies based on genre and rating. It provides a simple text-based interface for users to interact with the database.

The main functions used in this code are as follows:

Header Includes:

The code includes necessary headers like <vector>, <algorithm>, <string>, and "splashkit.h" to facilitate data structures, algorithms, and input/output operations.

1. <vector> : this header provides the 'vector' template class, which is a dynamic array that can resize itself automatically. In this code, it is used to store collections of movies and reviews.
2. <algorithm>: this header provides a collection of functions especially designed to be used on ranges of elements. In this code, it is used to perform operations like finding an element in a collection.
3. <string>: this header provides the 'string' class, which is a standard way of working with sequences of characters. In this code, it is used to handle movie names, descriptions, review texts and genre names.
4. "splashkit.h": this header includes declarations for splashkit functions, a library designed to simplify common programming tasks in c++. In this code, it is used for basic input/output operations like displaying messages and reading user inputs.

Enums and Structs:

Genre enum defines different movie genres.

Review struct holds review text and score.

Movie struct contains movie details like name, description, genre, and a vector of reviews.

Database Management:

The Database struct manages a vector of movies.

It provides functions to add, remove, modify, and display movies.

Additionally, it has functions to display movies by genre and filter movies by average rating.

User Interface:

The getUserInput function retrieves user input for strings.

displayMainMenu and displayGenreMenu functions display menus for various options.

getGenreFromUser function gets the genre choice from the user.

Main Program:

The main function initializes the movie database with sample movies.

It prompts the user for their name and displays a welcome message.

It runs a loop to display the main menu and execute user-selected options.

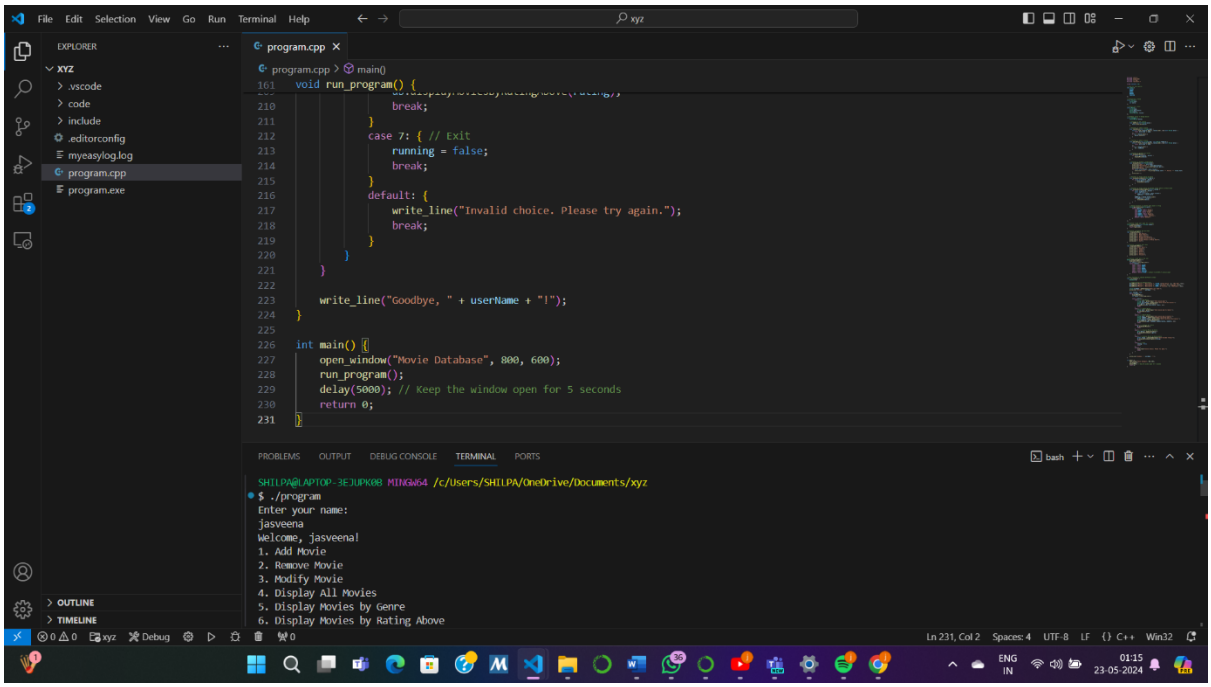
Options include adding, removing, modifying movies, displaying all movies, filtering movies by genre and rating, and exiting the program.

After exiting the loop, it displays a goodbye message.

SplashKit Functions:

open_window, write_line, and delay functions from SplashKit library are used for basic window handling and displaying messages.

LEARNING EVIDENCE:



REFLECTIONS :

1. How do you know that you have achieved the learning goals?

I know that I have achieved the learning goals because now I can effectively manage this database system and am able to successfully run the code knowing full well about the syntax, commands and the flow of the program.

2. What is the most important thing you learnt from this and why ?
   The most important thing I learnt from this task was how to organise and structure a c++ program to manage a database of complex data structures like movies and reviews. This involves defining structs, enums, and functions to manipulate and display the data. This is important because this is the basis of any c++ program and being familiar, efficient and having good knowledge about it can be quite beneficial later on if I need to develop more complex programs.
3. How does the content or skills learned here relate to things you already know ?
   Since it is not the first task I am doing in the c++ language . I am aware of the syntax and how to approach the programming for c++. I am also familiar with the concepts like structs,enums vectors and algorithms. It also build upon my understanding of user input/output handling and basic file organisation.
4. Where or when do you think it will be useful?
   This knowledge will be useful in various scenarios where I need to develop applications that involve managing collections of data, such as movie databases, inventory systems, or user management systems. Nevertheless, by understanding how to structure and manage data in a program is a fundamental skill that can be applied to a wide range of software development projects, making this learning highly practical and valuable.