**Data Structures and Algorithms – Project Report**

**Name:** Muhammad Abdullah Awan

**Class:** BSSE-3A

**Roll no:** SU92-BSSEM-F23-010

**Introduction:**

The Movie Management System is a C++ project designed to provide an efficient way to manage a personal movie database. The system leverages a doubly linked list data structure to perform various operations such as adding, searching, editing, deleting, and displaying movies. The primary objective is to create a user-friendly, console-based application with persistent data storage capabilities using file handling.

**Objectives:**

1. Implement a functional movie database using a doubly linked list.
2. Provide features for managing movie records such as add, search, edit, delete, and clear operations.
3. Ensure data persistence through file storage.
4. Design an interactive menu-driven console interface for users.

**Features:**

1. **Core Functionalities**

**Add Movies:** Users can add movies by specifying details such as name, genre, and rating.

**Display Movies:** All movies in the database can be viewed in a structured tabular format.

**Search Movies:**

1. By Name: Search for a specific movie by its title.
2. By Rating: Retrieve all movies matching a specific rating.

**Edit Movies:** Update details like name, genre, and rating of any existing movie.

**Delete Movies:** Remove a movie from the database using its name.

**Clear All Movies:** Delete the entire movie database in one command.

1. **Additional Features**

**File Persistence:** Save movie data to a text file (movies.txt) to maintain records between sessions.

Load existing data from the file when the application starts.

**Error Handling:** Display appropriate messages if operations like search, delete, or edit fail due to the absence of the specified movie.

**Movie Structure:**

Each movie is represented as a node in the doubly linked list with the following attributes:

**name:** The movie’s title.

**genre:** The category or genre of the movie (e.g., Action, Comedy).

**rating:** A numerical rating (e.g., 4.5).

**next and prev:** Pointers for traversing the list.

**MovieList Class:**

This class encapsulates all movie-related operations:

**addMovie():** Adds a new movie to the end of the list.

**displayMovies():** Displays all movies in the list.

**searchMovie():** Searches for a movie by name.

**searchMoviesByRating():** Finds movies with a specific rating.

**editMovie():** Updates the details of a specified movie.

**deleteMovie():** Removes a movie by name.

**clearMovies():** Deletes all movies in the list.

**saveToFile():** Writes movie data to a file for persistence.

**loadFromFile():** Reads movie data from the file at startup.

**GitHub:** <https://github.com/abdullahawan001/Movie-Recommendation-System.git>