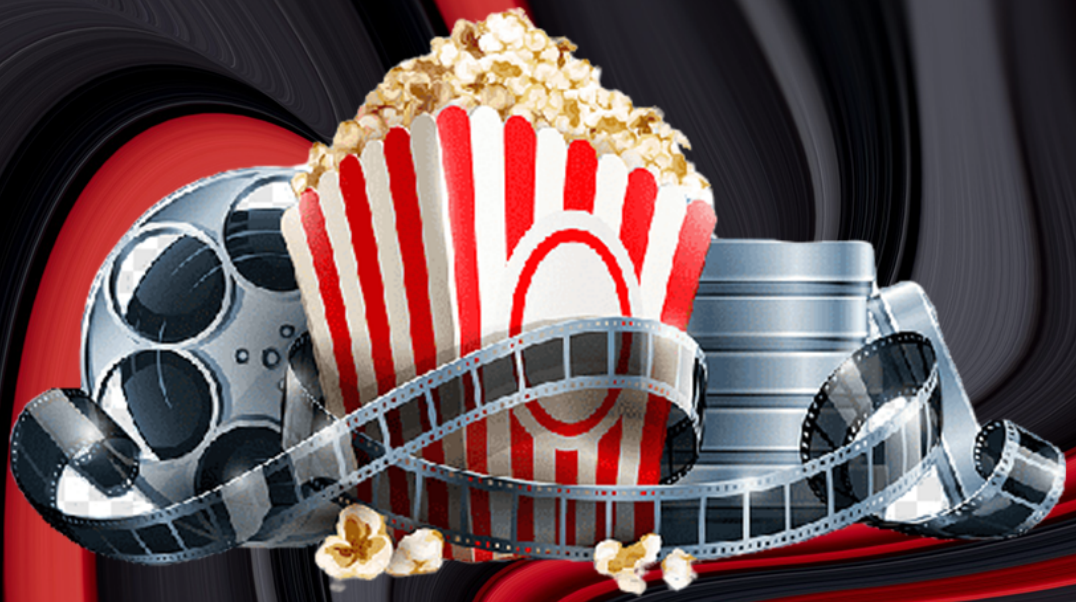
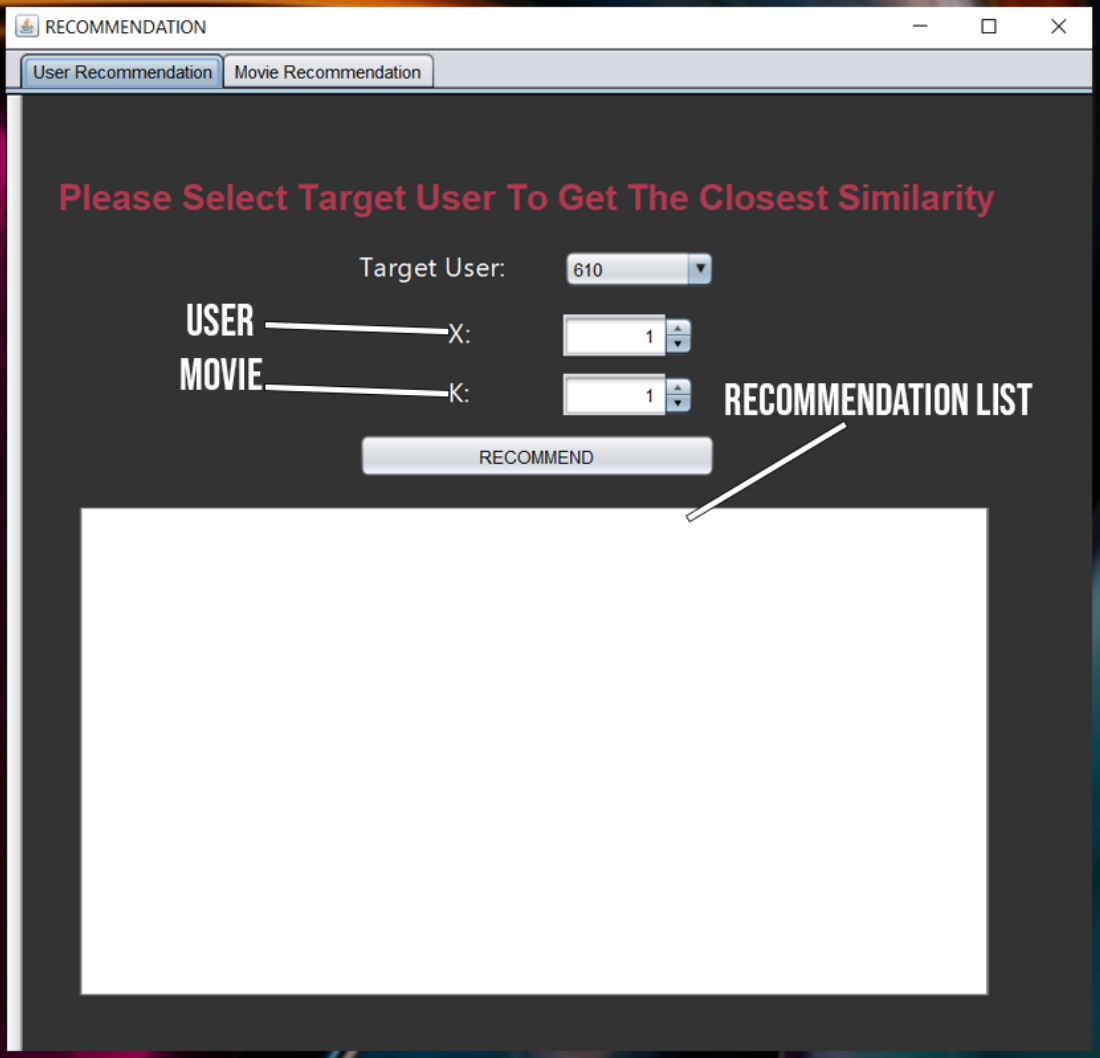
BLM19202E DATA STRUCTURES PROGRAMMING ASSIGNMENT #3
MOVIE RECOMMENDATION APPLICATION

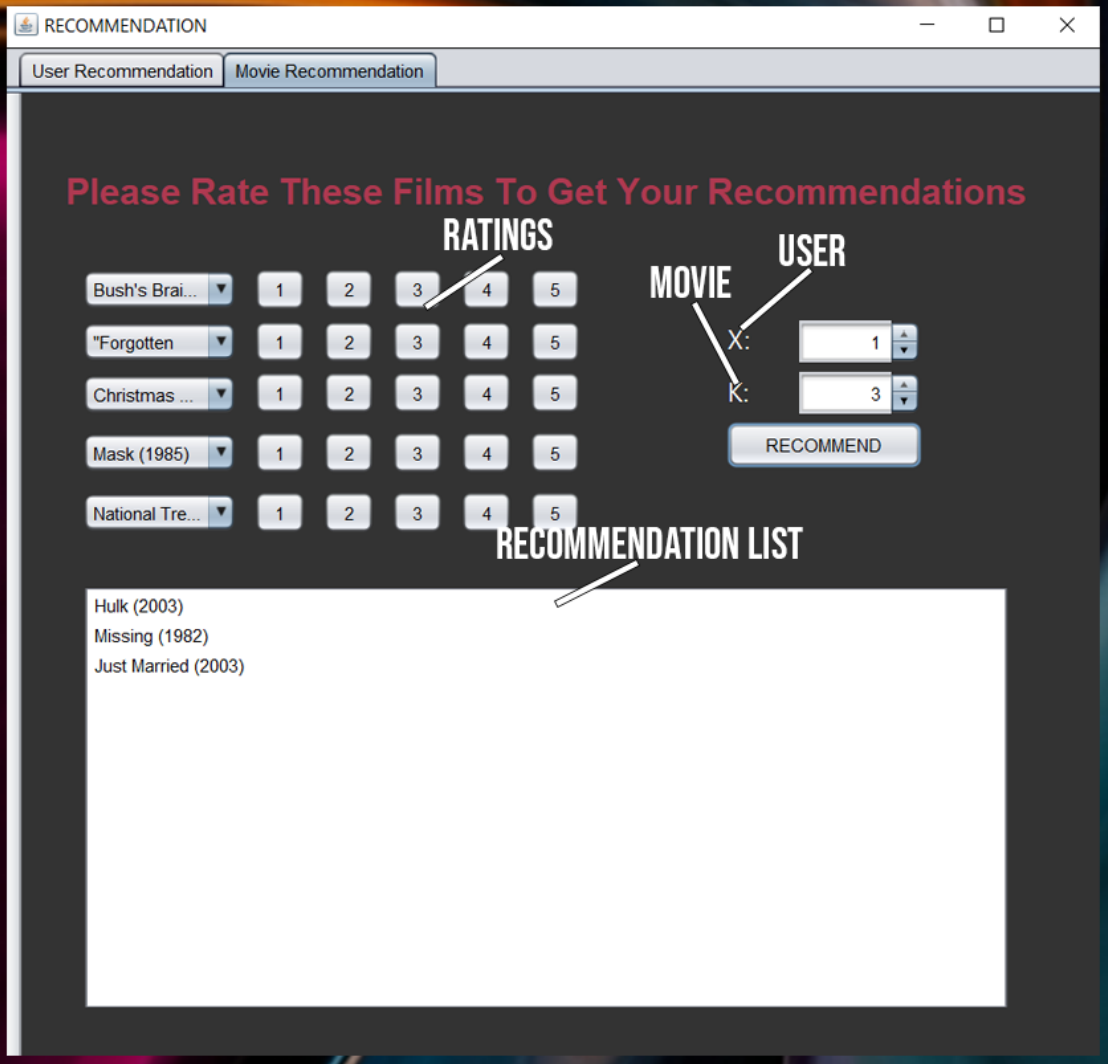


TARIK ALRAYAN - 2121221366

AHMED MUAZ ATIK - 2121221002


DESIGN


RECOMMENDATION
User Recommendation
Movie Recommendation
Please Select Target User To Get The Closest Similarity
Target User:
610
USER
X:
1
MOVIE
K:
1
RECOMMENDATION LIST
RECOMMEND


RECOMMENDATION
User Recommendation
Movie Recommendation
Please Rate These Films To Get Your Recommendations
RATINGS
USER
MOVIE
Bush's Brai...
"Forgotten
Christmas ...
Mask (1985)
National Tre...
1
2
3
4
5
X:
1
K:
3
RECOMMEND
RECOMMENDATION LIST
Hulk (2003)
Missing (1982)
Just Married (2003)

PSEUDOCODE

```
FUNCTION COMPUTESIMILARITY(USER1, USER2):
    RATINGS1 = USER1.GETRATINGS()
    RATINGS2 = USER2.GETRATINGS()

    COMMONMOVIES = INTERSECTION(RATINGS1.KEYSET(), RATINGS2.KEYSET())

    DOTPRODUCT = 0.0
    NORM1 = 0.0
    NORM2 = 0.0

    FOR MOVIEID IN COMMONMOVIES:
        RATING1 = RATINGS1.GET(MOVIEID)
        RATING2 = RATINGS2.GET(MOVIEID)

        DOTPRODUCT += RATING1 * RATING2
        NORM1 += RATING1^2
        NORM2 += RATING2^2

    IF NORM1 == 0.0 OR NORM2 == 0.0:
        RETURN 0.0 // NO SIMILARITY IF ONE OF THE USERS HAS NO RATINGS IN COMMON MOVIES.

    RETURN DOTPRODUCT / (SQRT(NORM1) * SQRT(NORM2))
```

IMPLEMENTATION DETAILS

```java
public void loadMovieNames(File f) {
    //load the movies names and ids from the movies file and store them into the movieNames table
    try (Scanner scanner = new Scanner( source: f)) {
        int line_Num = 0;
        while (scanner.hasNextLine()) {
            String line = scanner.nextLine();
            String[] parts = line.split( regex: ",");
            if (line_Num != 0) {
                if (parts.length >= 2) {
                    int movie_Id = Integer.parseInt(parts[0]);
                    String movie_Name = parts[1];
                    movieNames.put( key: movie_Id,  value: movie_Name);
                    line_Num++;
                }

            } else {
                line_Num++;
            }

        }
    } catch (FileNotFoundException e) {
        System.out.println( x: "File not found.");
    }

}
```

IN LOADMOVIENAMES FUNCTION, WE REACHED INSIDE THE FILE WITH SCANNER OBJECT.

WE SPLIT THE LINE WITH “,” CHAR.

WE INSERTED THE MOVIE_NAME, MOVIE_ID INTO THE MOVIENAMES HASHTABLE.

```java
public void LoadRatingsToTable(File f) {
    //load the main data file to the user tables
    try (Scanner scanner = new Scanner( source: f)) {
        int line_Num = 0;
        List<Integer> movie_Ids = new ArrayList<>();
        while (scanner.hasNextLine()) {
            String line = scanner.nextLine();

            if (line_Num == 0) {
                // get the movie ids from the first row and starts from index 1
                String[] movieIdValues = line.split( regex: ",");
                for (int i = 1; i < movieIdValues.length; i++) {
                    int movie_Id = Integer.parseInt(movieIdValues[i]);
                    movie_Ids.add( e: movie_Id);
                }
            } else {
                // if the index =0 that equal to the user id else movie rating
                String[] values = line.split( regex: ",");
                int user_Id = Integer.parseInt(values[0]);
                addUserToTable(user_Id,  table: users);

                for (int i = 1; i < values.length; i++) {
                    int movie_Id = movie_Ids.get(i - 1);
                    double rating = Double.parseDouble(values[i]);
                    addMovieRating(user_Id, movie_Id, rating,  table: users);
                }
            }

            line_Num++;
        }
    } catch (FileNotFoundException e) {
        System.out.println( x: "File not found.");
    }
}
```

IN LOADMOVIENAMES FUNCTION, WE REACHED INSIDE THE FILE WITH SCANNER OBJECT.

WE CREATED A MOVIE_IDS ARRAYLIST OBJECT. WE SPLIT THE LINE WITH “,” CHAR.

WITH ADDUSERTOTABLE FUNCTION, WE ADDED USER_ID INTO THE USERS.

WITH ADDMOVIERATING FUNCTION, WE ADDED USER_ID, MOVIE_ID AND RATING INTO THE USERS.

```java
public void LoadTargetsToTable(File f) {
    //load the main data file to the user tables
    try (Scanner scanner = new Scanner( source: f)) {
        int line_Num = 0;
        List<Integer> movie_Ids = new ArrayList<>();
        while (scanner.hasNextLine()) {
            String line = scanner.nextLine();

            if (line_Num == 0) {
                // get the movie ids from the first row and starts from index 1
                String[] movieIdValues = line.split( regex: ",");
                for (int i = 1; i < movieIdValues.length; i++) {
                    int movie_Id = Integer.parseInt(movieIdValues[i]);
                    movie_Ids.add( e: movie_Id);
                }
            } else {
                // if the index =0 that equal to the user id else movie rating
                String[] values = line.split( regex: ",");
                int user_Id = Integer.parseInt(values[0]);
                addUserToTable(user_Id,  table: targetUsers);

                for (int i = 1; i < values.length; i++) {
                    int movie_Id = movie_Ids.get(i - 1);
                    double rating = Double.parseDouble(values[i]);
                    addMovieRating(user_Id, movie_Id, rating,  table: targetUsers);
                }
            }

            line_Num++;
        }
    } catch (FileNotFoundException e) {
        System.out.println( x: "File not found.");
    }

}
```

IN LOADTARGETSTOTABLE FUNCTION, WE REACHED INSIDE THE FILE WITH SCANNER OBJECT.

WE CREATED A MOVIE_IDS ARRAYLIST OBJECT. WE SPLIT THE LINE WITH “,” CHAR.

WITH ADDUSERTOTABLE FUNCTION, WE ADDED USER_ID INTO THE TARGETUSERS.

WITH ADDMOVIERATING FUNCTION, WE ADDED USER_ID, MOVIE_ID AND RATING INTO THE TARGETUSERS.

```java
public ArrayList<String> getRecommendations(int targetUserId, int numSimilarUsers, int numRecommendedMovies) {
    User targetUser = null;
    for (User user : targetUsers.values()) {
        if (user.getUser_Id() == targetUserId) {
            targetUser = user;
        }
    }

    if (targetUser == null) {
        System.out.println( x: "Target user not found.");
        return new ArrayList<>(); // Return an empty list if target user not found
    }

    // Step 1: Compute similarity between each pair of users
    Heap<User> similarityHeap = new Heap<>();
    for (User user : users.values()) {
        double similarity = computeSimilarity( user1: user,  user2: targetUser);
        user.setSimilarity(similarity);
        similarityHeap.add( element: user);
    }

    // Step 2: Retrieve the most similar users from the heap
    ArrayList<User> mostSimilarUsers = new ArrayList<>();
    int count = Math.min( a: numSimilarUsers,  b: similarityHeap.size());
    for (int i = 0; i < count; i++) {
        User user = similarityHeap.removeTop();
        mostSimilarUsers.add( e: user);
    }

    // Step 3: Get the highest rated movies for the most similar users
    ArrayList<String> recommendedMovies = new ArrayList<>();
    for (User user : mostSimilarUsers) {
        Hashtable<Integer, Double> ratings = user.getRatings();
        ArrayList<Integer> topRatedMovies = getTopRatedMovies(ratings,  count: numRecommendedMovies);
        for (int movieId : topRatedMovies) {
            String movieName = movieNames.get( key: movieId);
            recommendedMovies.add( e: movieName);
        }
    }

    return recommendedMovies;
}
```

IN GETRECOMMENDATION FUNCTION, WE ARE TAKING 3 PARAMETERS.

AT THE BEGINNING WE ARE TRYING TO FIND THE USER INSIDE FROM THE TARGETUSER LIST.

IN STEP 1, WE ARE CALCULATING THE SIMILARITY BETWEEN USER AND TARGETUSER THEN ADDING THE RESULT IN THE HEAP.

IN STEP 2, WE ARE FINDING THE MOST SIMILIAR USERS TO TARGETUSER AND ADDING THE USERS INTO THE HEAP.

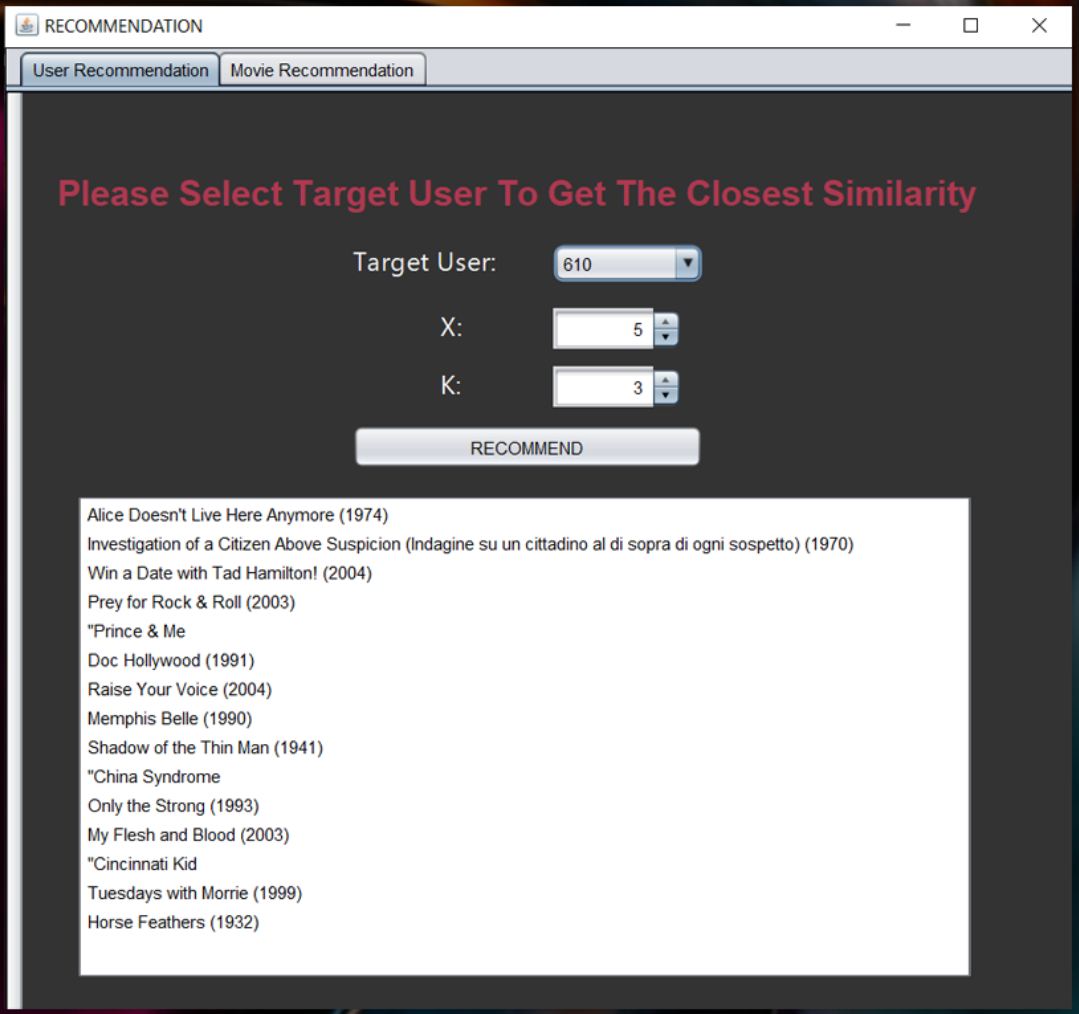
IN STEP 3, WE ARE GETTING THE HIGHEST RATED MOVIES FROM THE MOST SIMILIAR USER.

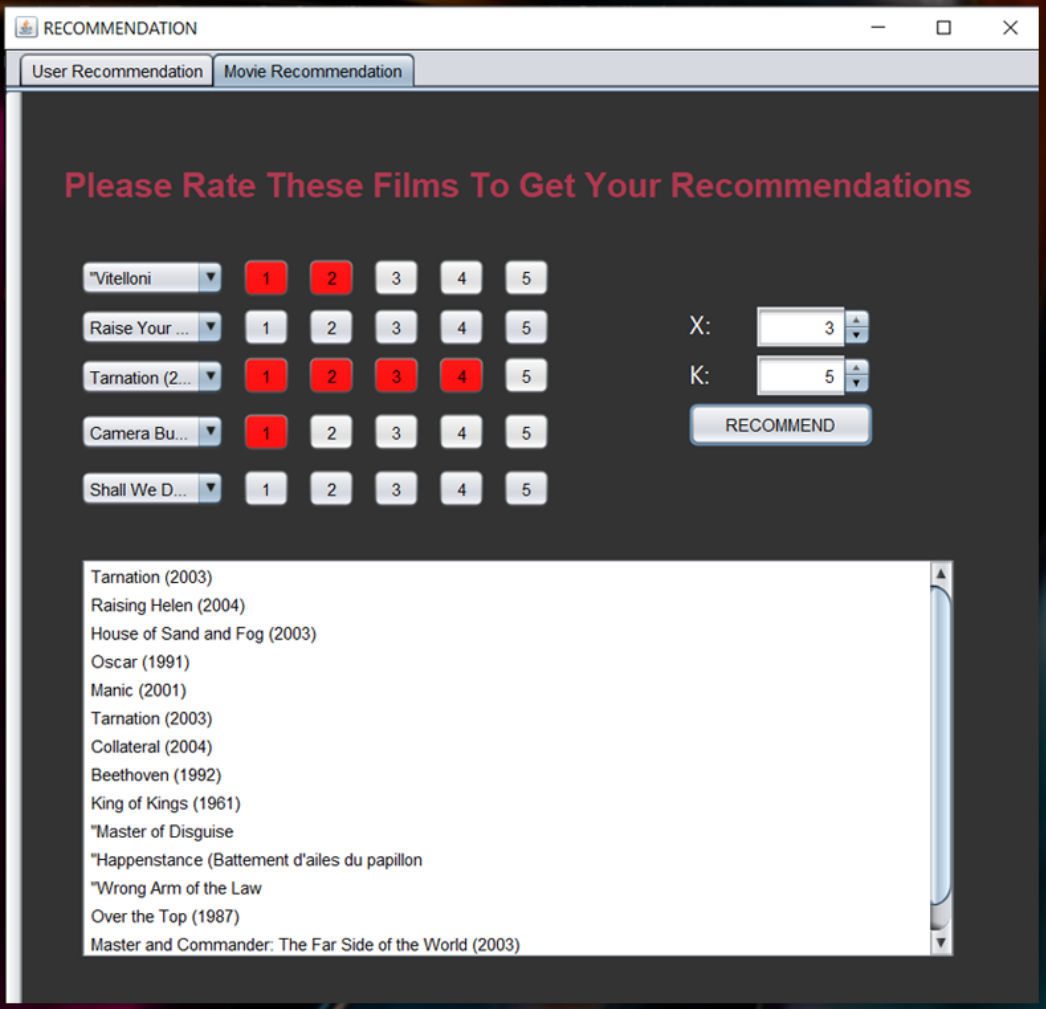
```java
public void loadMovieNames(File f) {
    //load the movies names and ids from the movies file and store them into the movieNames table
    try (Scanner scanner = new Scanner( source: f)) {
        int line_Num = 0;
        while (scanner.hasNextLine()) {
            String line = scanner.nextLine();
            String[] parts = line.split( regex: ",");
            if (line_Num != 0) {
                if (parts.length >= 2) {
                    int movie_Id = Integer.parseInt(parts[0]);
                    String movie_Name = parts[1];
                    movieNames.put( key: movie_Id,  value: movie_Name);
                    line_Num++;
                }

            } else {
                line_Num++;
            }

        }
    } catch (FileNotFoundException e) {
        System.out.println( x: "File not found.");
    }

}
```

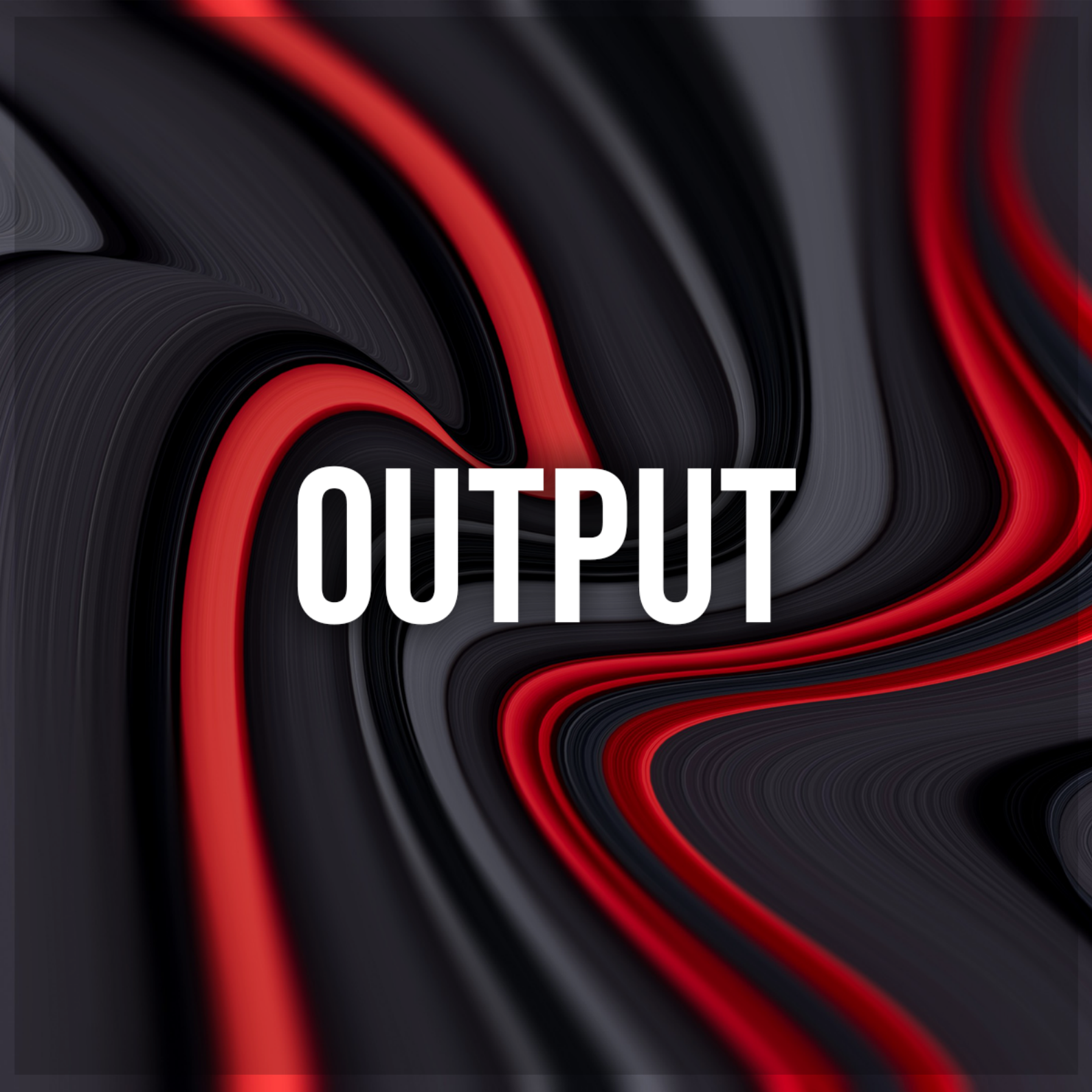
WE ARE GETTING THE MOVIE NAMES AND MOVIE IDS FROM THE FILE AND ADDING THEM INTO THE MOVIENAMES TABLE

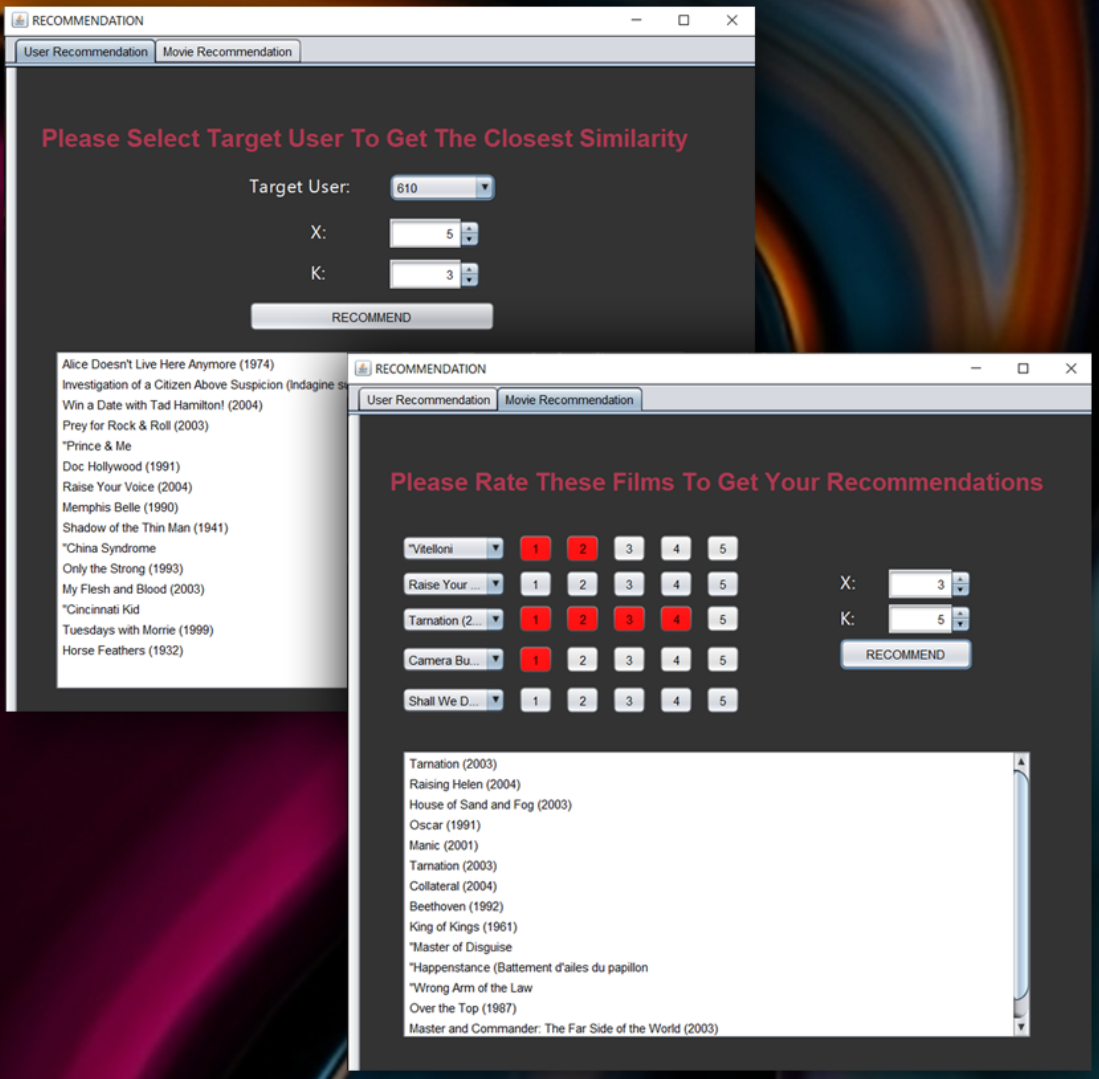
WE ARE REACHING THE FILE WITH SCANNER OBJECT AND SEPERATING THE LINES WITH “,” CHAR.

THEN WE ARE PUTTING THE MOVIEID AND MOVIENAME INTO THE MOVIENAMES TABLE WITH PUT FUNCTION.


RECOMMENDATION
User Recommendation
Movie Recommendation
Please Select Target User To Get The Closest Similarity
Target User:
610
X:
5
K:
3
RECOMMEND
Alice Doesn't Live Here Anymore (1974)
Investigation of a Citizen Above Suspicion (Indagine su un cittadino al di sopra di ogni sospetto) (1970)
Win a Date with Tad Hamilton! (2004)
Prey for Rock & Roll (2003)
"Prince & Me
Doc Hollywood (1991)
Raise Your Voice (2004)
Memphis Belle (1990)
Shadow of the Thin Man (1941)
"China Syndrome
Only the Strong (1993)
My Flesh and Blood (2003)
"Cincinnati Kid
Tuesdays with Morrie (1999)
Horse Feathers (1932)


RECOMMENDATION
User Recommendation
Movie Recommendation
Please Rate These Films To Get Your Recommendations
"Vitelloni
Raise Your ...
Tarnation (2...
Camera Bu...
Shall We D...
1
2
3
4
5
X:
3
K:
5
RECOMMEND
Tarnation (2003)
Raising Helen (2004)
House of Sand and Fog (2003)
Oscar (1991)
Manic (2001)
Tarnation (2003)
Collateral (2004)
Beethoven (1992)
King of Kings (1961)
"Master of Disguise
"Happenstance (Battement d'ailes du papillon
"Wrong Arm of the Law
Over the Top (1987)
Master and Commander: The Far Side of the World (2003)


OUTPUT


RECOMMENDATION
User Recommendation
Movie Recommendation
Please Select Target User To Get The Closest Similarity
Target User:
610
X:
5
K:
3
RECOMMEND
Alice Doesn't Live Here Anymore (1974)
Investigation of a Citizen Above Suspicion (Indagine
Win a Date with Tad Hamilton! (2004)
Prey for Rock & Roll (2003)
"Prince & Me
Doc Hollywood (1991)
Raise Your Voice (2004)
Memphis Belle (1990)
Shadow of the Thin Man (1941)
"China Syndrome
Only the Strong (1993)
My Flesh and Blood (2003)
"Cincinnati Kid
Tuesdays with Morrie (1999)
Horse Feathers (1932)
RECOMMENDATION
User Recommendation
Movie Recommendation
Please Rate These Films To Get Your Recommendations
"Vitelloni
Raise Your ...
Tarnation (2...
Camera Bu...
Shall We D...
X:
3
K:
5
RECOMMEND
Tarnation (2003)
Raising Helen (2004)
House of Sand and Fog (2003)
Oscar (1991)
Manic (2001)
Tarnation (2003)
Collateral (2004)
Beethoven (1992)
King of Kings (1961)
"Master of Disguise
"Happenstance (Battement d'ailes du papillon
"Wrong Arm of the Law
Over the Top (1987)
Master and Commander: The Far Side of the World (2003)