```
package DBconnector;

public class Movie {

int movie_id, user_Id;
float avg_rating;
String review, movieName, cast, genre, summary;

public int getMovie_id() {
return movie_id;
}
public void setMovie_id(int movie_id) {
this.movie_id = movie_id;
}
public float getAvg_rating() {
return avg_rating;
}
public void setAvg_rating(float avg_rating) {
this.avg_rating = avg_rating;
}
public int getUser_Id() {
return user_Id;
}
public void setUser_Id(int user_Id) {
this.user_Id = user_Id;
}
public String getMovieName() {
return movieName;
}
```

```
public void setMovieName(String movieName) {
this.movieName = movieName;
}
public String getGenre() {
return genre;
}
public void setGenre(String genre) {
this.genre = genre;
}
public String getSummary() {
return summary;
}
public void setSummary(String summary) {
this.summary = summary;
}

@Override
public String toString() {
return "Movie [movie_id=" + movie_id + ", avg_rating=" + avg_rating
+ ", user_Id=" + user_Id + ", movieName=" + movieName
+ ", cast=" + cast + ", genre=" + genre + ", summary="
+ summary + "]";
}

public Movie(){}

public Movie(String moviename, String summary, String cast, String genre, String review, int avgrating){
this.genre = genre;
this.summary = summary;
this.review = review;
this.movieName = moviename;
this.cast = cast;
this.avg_rating=avgrating;
}

public Movie(int movie_id, float avg_rating, int user_Id, String movieName,
        String genre, String summary) {
super();
this.movie_id = movie_id;
this.avg_rating = avg_rating;
this.user_Id = user_Id;
this.movieName = movieName;
this.genre = genre;
this.summary = summary;
}

}



package DBconnector;

import java.sql.*;
import java.util.*;

import DBconnector.Movie;

public class MovieDAO {
```

```
Connection conn;

public MovieDAO(){

conn = null;

try {
Class.forName("com.mysql.jdbc.Driver");
conn = DriverManager.getConnection(
"jdbc:mysql://localhost:3306/javadb", "root", "password");
}
catch(Exception e)
{}
}

public MovieDAO(String driver, String URL, String user, String pass){

conn=null;

try {
Class.forName(driver);
conn = DriverManager.getConnection(
URL, user, pass);
}
catch(Exception e)
{}
}

public int add(Movie m)
```

```
{
Statement stmt;
int s=0;
try {
stmt = conn.createStatement();
s = stmt.executeUpdate
("insert into movie values( "+m.getMovie_id()+", '"+m.getMovieName()+"', '"+m.getSummary()+
"', '"+m.getGenre()+"', "+m.getAvg_rating()+", "+m.getUser_Id()+" )");

} catch (SQLException e) {
}
return s;
}

public int update(Movie m)
{
Statement stmt;
int s=0;
try {
stmt = conn.createStatement();
s = stmt.executeUpdate
("Update movie set summary = '"+m.getSummary()+"', genre = '"+m.getGenre()+
"'  where movie_id="+m.getMovie_id() );
} catch (SQLException e1) {
}
return s;
}

public int delete(Movie m){
```

```
Statement stmt;
int s=0;
try {
stmt = conn.createStatement();
s = stmt.executeUpdate
("delete from movie where movie_id="+m.getMovie_id()+"");
} catch (SQLException e1) {
}
return s;
}

    public Movie getmoviebyid(int m_id){

Movie m1 = new Movie();

try {
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select * from movie where movie_id="+m_id+"");

    m1.movie_id=m_id;
m1.movieName=rs.getString(2);
m1.summary=rs.getString(3);
m1.genre=rs.getString(4);
m1.avg_rating=rs.getInt(5);
m1.user_Id=rs.getInt(6);
} catch (Exception e) {}
return m1;
}
```

```
 public Movie getmoviebyname(String m_name){

Movie m1 = new Movie();

try {
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select * from movie where movie_name="+m_name+"");

m1.movie_id=rs.getInt(1);
m1.movieName=m_name;
m1.summary=rs.getString(3);
m1.genre=rs.getString(4);
m1.avg_rating=rs.getInt(5);
m1.user_Id=rs.getInt(6);
} catch (Exception e) {}
return m1;
}

public List<Movie> movielist(){

List<Movie> movies = new ArrayList<>();

try {
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select * from movie");
while (rs.next())
movies.add( new Movie(rs.getInt(1), rs.getInt(5), rs.getInt(6),
rs.getString(2), rs.getString(3), rs.getString(4) ));
} catch (Exception e) {}
```

```
return movies;

}


  public Hashtable<String, String> movieByGenre(String gen){


        Hashtable<String, String> list = new Hashtable<String, String>();


try {

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select movie_name, Genre from movie where Genre='"+gen+"'");

while (rs.next())

list.put( rs.getString(1), rs.getString(2) );

} catch (Exception e) {}

return list;

}


  public void Top_ten(){


try {

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select movie_name, Average_rating from movie order by
Average_rating desc limit 10");

while (rs.next())

System.out.println(rs.getString(1) + " " + rs.getFloat(2));

} catch (Exception e) {}

}


  public List<String> Topten(String Gen){
        List<String> list = new ArrayList<String>();

try {
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select movie_name from movie where Genre='"+Gen+"' order by Average_rating desc limit 10");
while (rs.next()){
list.add(rs.getString(1));
}
} catch (Exception e) {}
return list;
}

  public List<String> listbyname(String moviename){

        List<String> list = new ArrayList<String>();

try {
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select movie_name from movie where movie_name='"+moviename+"'");
while (rs.next()){
list.add( rs.getString(1) );
}
} catch (Exception e) {}

return list;
}
```

```
    public List<Movie> moviedetail(int mid){

List<Movie> movies = new ArrayList<>();

try {
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select movie.movie_name, movie.Genre, movie.summary, movie.Average_rating, review.Review"
+ "from movie INNER JOIN review ON movie.movie_id = review.movie_id where movie.movie_id="+mid+"");
ResultSet rs1 = stmt.executeQuery("select Cast from cast where movie_id="+mid+"");

System.out.println(rs1.getString(1));
while (rs1.next()){
movies.add( new Movie( rs.getString(1), rs.getString(2),
rs1.getString(1), rs.getString(3), rs.getString(5), rs.getInt(4) ) );
}
} catch (Exception e) {}
return movies;
}

    public boolean movieExist(int mid){

              List<Integer> list = new ArrayList<>();

try {
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select movie_id from movie");
while (rs.next())
list.add( rs.getInt(1) );
} catch (Exception e) {}

for(int ele : list){
if(ele==mid) return true;
}

          return false;
   }

   public boolean movieexist(String mname){

          List<String> list = new ArrayList<>();

try {
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select movie_name from movie");
while (rs.next())
list.add( rs.getString(1) );
} catch (Exception e) {}

for(String ele : list){
if(ele==mname) return true;
}

          return false;
   }

public void finalise(){
```

```
try {
conn.close();
} catch (SQLException e) {}
}

}
```

```
package DBconnector;

public class User {

int user_id;
String userName, emailId;

public int getUser_id() {
return user_id;
}
public void setUser_id(int user_id) {
this.user_id = user_id;
}
public String getUserName() {
return userName;
}
public void setUserName(String userName) {
this.userName = userName;
}
public String getEmailId() {
return emailId;
}
public void setEmailId(String emailId) {
this.emailId = emailId;
}

@Override
public String toString() {
return "\nUser [user_id=" + user_id + ", userName=" + userName
+ ", emailId=" + emailId + "]";
}

public User(){}
public User(int user_id, String userName, String emailId) {
super();
this.user_id = user_id;
this.userName = userName;
this.emailId = emailId;
}

}
```

```
package DBconnector;

import java.sql.*;
import java.util.*;
```

```
public class UserDAO {

Connection conn;

public UserDAO(){

conn = null;

try {
Class.forName("com.mysql.jdbc.Driver");
conn = DriverManager.getConnection(
"jdbc:mysql://localhost:3306/javadb", "root", "password");
}
catch(Exception e)
{}
}

public UserDAO(String driver, String URL, String user, String pass){

conn=null;

try {
Class.forName(driver);
conn = DriverManager.getConnection(
URL, user, pass);
}
catch(Exception e)
{}
}

public int add(User u)
{
Statement stmt;
int s=0;
try {
stmt = conn.createStatement();
s = stmt.executeUpdate
  ("insert into users values( "+u.getUser_id()+", '"+u.getUserName()+"', '"+u.getEmailId()+"' )");
} catch (SQLException e) {
}
return s;
}

public List<User> userlist(){

List<User> list = new ArrayList<>();

try {
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select * from users");
while (rs.next())
list.add( new User(rs.getInt(1), rs.getString(2),
rs.getString(3) ));
} catch (Exception e) {}
return list;
}
```

```
public boolean userExist(int uid){

List<Integer> list = new ArrayList<>();

try {
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select user_id from users");
while (rs.next())
list.add( rs.getInt(1) );
} catch (Exception e) {}

for(int ele : list){
if(ele==uid) return true;
}

return false;
}

public void finalise(){
try {
conn.close();
} catch (SQLException e) {}
}

}
```

```
package DBconnector;
```

```
public class Cast {


int castid, userid, movieid;

String cast;

public int getCastid() {

return castid;

}

public void setCastid(int castid) {

this.castid = castid;

}

public int getUserid() {

return userid;

}

public void setUserid(int userid) {

this.userid = userid;

}

public int getMovieid() {

return movieid;

}

public void setMovieid(int movieid) {

this.movieid = movieid;

}

public String getCast() {

return cast;

}

public void setCast(String cast) {

this.cast = cast;

}
@Override

public String toString() {

return "Cast [castid=" + castid + ", userid=" + userid + ", movieid="

+ movieid + ", cast=" + cast + "]";

}


public Cast(){}


public Cast(int castid, int userid, int movieid, String cast) {

super();

this.castid = castid;

this.userid = userid;

this.movieid = movieid;

this.cast = cast;

}


}




package DBconnector;


import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;
```

```
}

import java.util.List;

public class CastDAO {

Connection conn;

public CastDAO(){

conn = null;

try {
Class.forName("com.mysql.jdbc.Driver");
conn = DriverManager.getConnection(
"jdbc:mysql://localhost:3306/javadb", "root", "password");
}
catch(Exception e)
{}
}

public CastDAO(String driver, String URL, String user, String pass){

conn=null;

try {
Class.forName(driver);
conn = DriverManager.getConnection(
URL, user, pass);
}
catch(Exception e)
```

```
{}
}

public int add(Cast c)
{
Statement stmt;
int s=0;
try {
stmt = conn.createStatement();
s = stmt.executeUpdate
("insert into cast values( "+c.getCastid()+", "+c.getMovieid()+
", '"+c.getCast()+"', "+c.getUserid()+" )");
} catch (SQLException e) {
}
return s;
}

public int update(Cast c)
{
Statement stmt;
int s=0;
try {
stmt = conn.createStatement();
s = stmt.executeUpdate
("Update cast set Cast = '"+c.getCast()+"', user_id = "+c.getUserid()+" where
movie_Id="+c.getMovieid());
} catch (SQLException e) {
}
return s;
```

```
}

public int delete(Cast c){
Statement stmt;
int s=0;
try {
stmt = conn.createStatement();
s = stmt.executeUpdate
("delete from cast where movie_id="+c.getMovieid()+"");
} catch (SQLException e1) {
}
return s;
}

public List<Cast> castlist(){

List<Cast> casts = new ArrayList<>();

try {
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select * from cast");
while (rs.next())
casts.add( new Cast(rs.getInt(1), rs.getInt(2), rs.getInt(4),
rs.getString(3) ));
} catch (Exception e) {}
return casts;
}

}
```

```
package DBconnector;

public class Review {

int review_Id, user_Id, movieId;
String reveiw;

public int getReview_Id() {
return review_Id;
}
public void setReview_Id(int review_Id) {
this.review_Id = review_Id;
}
public int getUser_Id() {
return user_Id;
}
public void setUser_Id(int user_Id) {
this.user_Id = user_Id;
}
public int getMovieId() {
return movieId;
}
public void setMovieId(int movieId) {
this.movieId = movieId;
}
public String getReveiw() {
return reveiw;
}
public void setReveiw(String reveiw) {
```

```
this.reveiw = reveiw;
}

@Override
public String toString() {
return "Review [review_Id=" + review_Id + ", user_Id=" + user_Id
+ ", movieId=" + movieId + ", reveiw=" + reveiw + "]";
}

public Review(){}

public Review(int review_Id, int user_Id, int movieId, String reveiw) {
super();
this.review_Id = review_Id;
this.user_Id = user_Id;
this.movieId = movieId;
this.reveiw = reveiw;
}
}



package DBconnector;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

public class ReviewDAO {

Connection conn;

public ReviewDAO(){

conn = null;

try {
Class.forName("com.mysql.jdbc.Driver");
conn = DriverManager.getConnection(
"jdbc:mysql://localhost:3306/javadb", "root", "password");
}
catch(Exception e)
{}
}

public ReviewDAO(String driver, String URL, String user, String pass){

conn=null;

try {
Class.forName(driver);
conn = DriverManager.getConnection(
URL, user, pass);
// here javadb is database name, root is username and password
```

```
}
catch(Exception e)
{}
}

public int add(Review r)
{
Statement stmt;
int s=0;
try {
stmt = conn.createStatement();
s = stmt.executeUpdate
("insert into review values( "+r.getReview_Id()+", "+r.getUser_Id()+", "+r.getMovieId()+", '"
+r.getReveiw()+"' )");

} catch (SQLException e) {
}
return s;
}

public int update(Review r)
{
Statement stmt;
int s=0;
try {
stmt = conn.createStatement();
s = stmt.executeUpdate
("Update review set review_Id = "+r.getReveiw());
} catch (SQLException e1) {
}

return s;

}


  public List<Review> reviewlist(){


List<Review> reviews = new ArrayList<>();


try {

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select * from review");

while (rs.next())

reviews.add( new Review(rs.getInt(1), rs.getInt(2), rs.getInt(4),

rs.getString(3) ));

} catch (Exception e) {}

return reviews;

}


public void finalise(){

try {

conn.close();

} catch (SQLException e) {}

}


}
```

```
package DBconnector;

public class Rating {

int ratingId, movieId, rating, user_Id;

public int getRatingId() {
return ratingId;
}

public void setRatingId(int ratingId) {
this.ratingId = ratingId;
}

public int getMovieId() {
return movieId;
}

public void setMovieId(int movieId) {
this.movieId = movieId;
}

public int getRating() {
return rating;
}

public void setRating(int rating) {
this.rating = rating;
}

public int getUser_Id() {
return user_Id;
}

public void setUser_Id(int user_Id) {
this.user_Id = user_Id;
}



@Override
public String toString() {
return "Rating [ratingId=" + ratingId + ", movieId=" + movieId
+ ", rating=" + rating + ", user_Id=" + user_Id + "]";
}

public Rating(){}

public Rating(int ratingId, int movieId, int rating, int user_Id) {
super();
this.ratingId = ratingId;
this.movieId = movieId;
this.rating = rating;
this.user_Id = user_Id;
}
}
```

```
package DBconnector;


import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;


public class RatingDAO {


Connection conn;


public RatingDAO(){


conn = null;


try {
Class.forName("com.mysql.jdbc.Driver");
conn = DriverManager.getConnection(
"jdbc:mysql://localhost:3306/javadb", "root", "password");
}
catch(Exception e)
{}
}


public RatingDAO(String driver, String URL, String user, String pass){
```

```
conn=null;

try {
Class.forName(driver);
conn = DriverManager.getConnection(
URL, user, pass);
}
catch(Exception e)
{}
}

public int add(Rating R)
{
Statement stmt;
int s=0;
try {
stmt = conn.createStatement();
s = stmt.executeUpdate
("insert into rating values( "+R.getRatingId()+", "+R.getMovieId()+
", "+R.getUser_Id()+", "+R.getRating()+" )");
} catch (SQLException e1) {
}
return s;
}

public int update(Rating R)
{
Statement stmt;
int s=0;
```

```
try {
stmt = conn.createStatement();
s = stmt.executeUpdate
("Update rating set rating = "+R.getRating()+
" where user_Id="+R.getUser_Id()+" and movie_Id="+R.getMovieId());
} catch (SQLException e1) {
}
return s;
}

public float getavgrating(int m_id){

	float avgrating=0;
 int count=0;
try {
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select Rating, movie_id from rating where movie_id="+m_id+"");
while (rs.next())
  avgrating += (float)rs.getInt(1);
  count++;
} catch (Exception e) {}
return (float)(avgrating/count);
}

public List<Rating> ratinglist(){

List<Rating> ratings = new ArrayList<>();

try {
```

```
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select * from rating");
while (rs.next())
  ratings.add( new Rating(rs.getInt(1), rs.getInt(2), rs.getInt(3), rs.getInt(4) ));
} catch (Exception e) {}
return ratings;
}

public void finalise(){
try {
conn.close();
} catch (SQLException e) {}
}

}
```

```
package DBconnector;

import java.util.List;
import java.util.Scanner;
```

```
    }



public class view {


MovieDAO mdao=new MovieDAO("com.mysql.jdbc.Driver","jdbc:mysql://localhost:3306/javadb",
"root", "password");

Scanner input = new Scanner(System.in);


public void fn(){

    System.out.println("Since you are not a registered user, you only have these options to choose: ");


   while(true){


          System.out.println("1. View a list of movies from a genre ");

   System.out.println("2. Veiw a list of top 10 movies overall or from a genre ");

   System.out.println("3. Search for a movie by name or/and by a genre ");

   System.out.println("4. View details of a movie, see its summary, cast, genre, reviews");

   System.out.println("5. Enter -1 to Exit");

      int k = input.nextInt();


    if(k==1){

 System.out.println("View a list of movies from a genre");

 System.out.println("Enter genre: ");

 String Genre = input.next();


 java.util.Hashtable<String, String> listByGenre = mdao.movieByGenre(Genre);


 if(listByGenre.isEmpty()) System.out.println("This Genre does not exist");

 else System.out.println(listByGenre);

    }
   else if(k==2){

 System.out.println("Veiw a list of top 10 movies overall or from a genre");
 int c=0;

 while(c!=2){

  System.out.println("Choose one: ");
  System.out.println("1. View top 10 \n2. View top 10 from Genre");
  int ch = input.nextInt();

if(ch==1){
System.out.println("Top ten movies by rating: ");
System.out.println("Movies and their Ratings");
mdao.Top_ten();
c=2;
}
else{
System.out.println("Enter Genre: ");
String gen=input.next();
List<String> list_ten = mdao.Topten(gen);

if(list_ten.isEmpty()) {
System.out.println("This Genre does not exist");
  System.out.println("1. Re-Enter another Genre\n2. Exit ");
  c=input.nextInt();
}
else {
```

```
System.out.println(list_ten);

c=2;

}

  }


}

   }


   else if(k==3){

System.out.println("Search for a movie by name or/and by a genre");

int c=0;


while(c!=2){

input.nextLine();

System.out.println("Enter movie name: ");

String mov=input.nextLine();


List<String> list = mdao.listbyname(mov);


if(list.isEmpty()){

System.out.println("This movie does not exist");

System.out.println("1. Re-Enter anoter movie name\n2. Exit ");

c=input.nextInt();

}

else {

System.out.println(list);

c=2;

 }

    }
    }

    else if(k==4){
  System.out.println("View details of a movie ");
  System.out.println("Enter movie_id: ");
  int mid=input.nextInt();

  boolean f6=mdao.movieExist(mid);

  if(f6){
  List<Movie> movies = mdao.moviedetail(mid);
  System.out.println(movies);
  }
  else System.out.println("Movie does not exist");
   }

    else if(k==-1) break;

   else System.out.println("Invalid choice ");
   }
}
}
```

```
package DBconnector;
```

```
import java.sql.SQLException;

import java.sql.Statement;

import java.util.*;




public class Main {


public static void main(String[] args) {


MovieDAO mdao=new MovieDAO("com.mysql.jdbc.Driver","jdbc:mysql://localhost:3306/javadb",
"root", "password");

UserDAO udao=new UserDAO("com.mysql.jdbc.Driver","jdbc:mysql://localhost:3306/javadb", "root",
"password");

CastDAO cdao=new CastDAO("com.mysql.jdbc.Driver","jdbc:mysql://localhost:3306/javadb", "root",
"password");

ReviewDAO rdao=new ReviewDAO("com.mysql.jdbc.Driver","jdbc:mysql://localhost:3306/javadb",
"root", "password");

RatingDAO Rdao=new RatingDAO("com.mysql.jdbc.Driver","jdbc:mysql://localhost:3306/javadb",
"root", "password");



while(true){


Scanner input = new Scanner(System.in);

System.out.println("Choose any one: ");

System.out.println("1. Login ");

System.out.println("2. Signup ");

System.out.println("3. Other ");

System.out.println("4. Exit");
```

```
int checkuser = input.nextInt();

if(checkuser==1){

System.out.println("Login: ");
System.out.println("Enter user-id: ");

int id = input.nextInt();
boolean flag = udao.userExist(id);

if(flag==true){

System.out.println("Choose any one: ");
System.out.println("1. Edit\n2. View ");
int ch=input.nextInt();

if(ch==1){
System.out.println("Edit: ");
System.out.println("Choose any of the given options: ");
System.out.println("1. Add a new movie entry ");
System.out.println("2. Edit an existing movie entry ");
System.out.println("3. Delete an existing movie entry ");
System.out.println("4. Add cast to a movie ");
System.out.println("5. Add a review to a movie ");
System.out.println("6. Rate a movie ");

int k=input.nextInt();

if(k==1){
```

```
System.out.println("Add a new movie entry: ");
int c=0;

while(c!=2){

System.out.println("Enter movie-id: ");
int movid=input.nextInt();
boolean f = mdao.movieExist(movid);

if(f){
System.out.println("Movie already exists ");
System.out.println("Enter another movie-Id: ");
}
else {
System.out.println("Enter movie name: ");
String moviename=input.nextLine();
input.nextLine();
System.out.println("Type summary: ");
String summry=input.nextLine();
System.out.println("Enter movie genre: ");
String mgenre=input.nextLine();

Movie m1=new Movie(movid,0,id,moviename,summry,mgenre);
System.out.println(mdao.add(m1) + " record inserted");
List<Movie> mlist=mdao.movielist();
System.out.println(mlist);
c=2;
}
}
```

```
}
else if(k==2){
System.out.println("Edit an existing movie entry: ");

System.out.println("Enter movie-id: ");
int m_id=input.nextInt();

boolean f1=mdao.movieExist(m_id);

if(f1){
Movie m2=mdao.getmoviebyid(m_id);
System.out.println("Type Genre : ");
String c1=input.nextLine();
input.nextLine();
System.out.println("Type Summary: ");
String c2=input.nextLine();
Movie m22 = new Movie(m_id, m2.getAvg_rating(), id, m2.getMovieName(), c1, c2);
System.out.println(mdao.update(m22) + " record inserted");
System.out.println(mdao.movielist());
}
else System.out.println("Movie does not exist ");
}

else if(k==3){
System.out.println("Delete an existing movie entry: ");

System.out.println("Enter movie-id: ");
int m_id=input.nextInt();
```

```
boolean f2=mdao.movieExist(m_id);


if(f2){
Movie m3=mdao.getmoviebyid(m_id);
mdao.delete(m3);
System.out.println(mdao.movielist());
}
else System.out.println("Movie does not exist already ");


}
else if(k==4){
System.out.println("Add cast to a movie ");
System.out.println("Please type movie-name for which you want to add cast ");
   String mname=input.nextLine();


   boolean f3=mdao.movieexist(mname);


   if(f3){
           Movie m4=mdao.getmoviebyname(mname);
           Cast c1=new Cast();
           c1.castid=input.nextInt();
           c1.userid=id;
           c1.movieid=m4.movie_id;
           c1.cast=input.nextLine();
cdao.add(c1);
System.out.println(cdao.castlist());
   }
   else System.out.println("Movie does not exist ");
}
```

```
else if(k==5){
System.out.println("Add a review to a movie ");
  System.out.println("Please type movie-name for which you want to add a review ");
  String mname=input.nextLine();

  boolean f4=mdao.movieexist(mname);

  if(f4){
        Movie m5=mdao.getmoviebyname(mname);
        Review r1=new Review();
        r1.review_Id=input.nextInt();
        r1.user_Id=id;
        r1.movieId=m5.movie_id;
        r1.reveiw=input.nextLine();
rdao.add(r1);
System.out.println(rdao.reviewlist());
  }
  else System.out.println("Movie does not exist ");
}

else if(k==6){
System.out.println("Rate a movie ");
System.out.println("Please type movie-name for which you want to add a rating ");
  String mname=input.nextLine();

  boolean f5=mdao.movieexist(mname);

  if(f5){
```

```
            Movie m6=mdao.getmoviebyname(mname);

            Rating r2=new Rating();

            r2.ratingId=input.nextInt();

            r2.user_Id=id;

            r2.movieId=m6.movie_id;

            r2.rating=input.nextInt();

Rdao.add(r2);

m6.avg_rating=Rdao.getavgrating(r2.movieId);

mdao.update(m6);

System.out.println(Rdao.ratinglist());

System.out.println(mdao.movielist());

  }

  else System.out.println("Movie does not exist ");

}


else System.out.println("Invalid Input ");

}


else if(ch==2){

view ob = new view();

ob.fn();

}


else System.out.println("Invalid choice ");


}


else {

System.out.println("User does not exists ");
```

```
continue;

}


}


else if(checkuser==2){


System.out.println("Signup: ");

System.out.println("Enter user-id: ");

int uid = input.nextInt();

input.nextLine();

System.out.println("Enter user-name: ");

String uname = input.nextLine();

System.out.println("Enter email-id: ");

String emailid = input.nextLine();

User u = new User(uid,uname,emailid);

System.out.println(udao.add(u)+ "record inserted");

java.util.List<User> list=udao.userlist();

System.out.println(list);

}


else if(checkuser==3){

view obj = new view();

obj.fn();

}


else if (checkuser==4) break;


else System.out.println("Invalid Input ");
```

```
 }
   }
  }
```

OUTPUT

```
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is
`com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and manual loading of the
driver class is generally unnecessary.

Choose any one:

1. Login

2. Signup

3. Other

4. Exit

1

Login:

Enter user-id:

2

Choose any one:

1. Edit

2. View

1

Edit:

Choose any of the given options:

1. Add a new movie entry
```

```
2. Edit an existing movie entry
3. Delete an existing movie entry
4. Add cast to a movie
5. Add a review to a movie
6. Rate a movie
2
Edit an existing movie entry:
Enter movie-id:
6
Type Genre :
nkjnsekjnkj
Type Summary:
njknkjfs
1 record inserted
[Movie [movie_id=1, avg_rating=5.0, user_Id=1, movieName=Jagga jasoos, cast=null, genre=agga
Jasoos (transl.?Detective Jagga) is a 2017 Indian Hindi-language musical adventure comedy film[4]
written and directed by Anurag Basu, and produced by Siddharth Roy Kapur, Basu and Ranbir Kapoor.
It features Kapoor and Katrina Kaif in the lead roles, and tells the story of a teenage detective in
search of his missing father. It was released on 14 July 2017[5] to mixed reviews from critics, but
bagged ten nominations at the 63rd Filmfare Awards, winning four of them for the films music.,
summary=musical], Movie [movie_id=2, avg_rating=0.0, user_Id=2, movieName=Heroine, cast=null,
genre= Laura and Massimo are back and stronger than ever. But Massimos family ties and a
mysterious man bidding for Laura’s heart complicate the lovers lives, summary=Drama], Movie
[movie_id=3, avg_rating=0.0, user_Id=3, movieName=Through my window, cast=null, genre=Raquels
longtime crush on her next-door neighbor turns into something more when he starts developing
feelings for her, despite his familys objections., summary=Romantic], Movie [movie_id=4,
avg_rating=0.0, user_Id=4, movieName=Oxygen, cast=null, genre=After waking up in a cryogenic unit,
Liz fights to survive and remember who she is before her oxygen runs out., summary=Thrriller], Movie
[movie_id=5, avg_rating=0.0, user_Id=1, movieName=The man from toronto, cast=null, genre=A case
of mistaken identity forces a bumbling entrepreneur to team up with a notorious assassin known as
The Man from Toronto in hopes of staying alive., summary=Comedy], Movie [movie_id=6,
avg_rating=7.0, user_Id=2, movieName=RRR, cast=null, genre=njknkjfs, summary=], Movie
[movie_id=7, avg_rating=0.0, user_Id=1, movieName=, cast=null, genre=Action, summary=bjbke]]
Choose any one:
1. Login
```

```
2. Signup
3. Other
4. Exit
4
```

```
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is
`com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and manual
loading of the driver class is generally unnecessary.
Choose any one:
1. Login
2. Signup
3. Other
4. Exit
2
Signup:
Enter user-id:
8
Enter user-name:
bhjbj
Enter email-id:
hjbjhef
1record inserted
[
User [user_id=1, userName=Neha Kumari, emailId=nk3456@gmail.com],
User [user_id=2, userName=Srishti Jha, emailId=srjh12@gmail.com],
User [user_id=3, userName=Vicky Gupta, emailId=vkg87@gmail.com],
User [user_id=4, userName=Avishi Raj, emailId=vish5674@gmail.com],
User [user_id=5, userName=varsha, emailId=gfhksdjsh],
User [user_id=6, userName=ananya agarwal, emailId=ag@23gmail.com],
User [user_id=7, userName=pushpa baraik, emailId=pb34@gmail.com],
User [user_id=8, userName=bhjbj, emailId=hjbjhef]]
Choose any one:
1. Login
2. Signup
3. Other
4. Exit

4
```

```
Choose any one:
1. Login
2. Signup
3. Other
4. Exit
3
Since you are not a registered user, you only have these options to choose:
1. View a list of movies from a genre
2. Veiw a list of top 10 movies overall or from a genre
3. Search for a movie by name or/and by a genre
4. View details of a movie, see its summary, cast, genre, reviews
5. Enter -1 to Exit
1
View a list of movies from a genre
Enter genre:
Comedy
{The man from toronto=Comedy}
1. View a list of movies from a genre
2. Veiw a list of top 10 movies overall or from a genre
3. Search for a movie by name or/and by a genre
4. View details of a movie, see its summary, cast, genre, reviews
5. Enter -1 to Exit

-1
Choose any one:
1. Login
```

```
2. Signup
3. Other
4. Exit
4
```

```
Choose any one:
1. Login
2. Signup
3. Other
4. Exit
3
Since you are not a registered user, you only have these options to choose:
1. View a list of movies from a genre
2. Veiw a list of top 10 movies overall or from a genre
3. Search for a movie by name or/and by a genre
4. View details of a movie, see its summary, cast, genre, reviews
5. Enter -1 to Exit
```

```
2
Veiw a list of top 10 movies overall or from a genre
Choose one:
1. View top 10
2. View top 10 from Genre
1
Top ten movies by rating:
Movies and their Ratings
RRR 7.85
Jagga jasoos 5.3
Heroine 0.0
Through my window 0.0
Oxygen 0.0
The man from toronto 0.0
0.0
1. View a list of movies from a genre
2. Veiw a list of top 10 movies overall or from a genre
3. Search for a movie by name or/and by a genre
4. View details of a movie, see its summary, cast, genre, reviews
5. Enter -1 to Exit
-1
Choose any one:
1. Login
2. Signup
3. Other
4. Exit
4
```

```
Choose any one:
1. Login
2. Signup
3. Other
4. Exit
3
Since you are not a registered user, you only have these options to choose:
1. View a list of movies from a genre
2. Veiw a list of top 10 movies overall or from a genre
3. Search for a movie by name or/and by a genre
4. View details of a movie, see its summary, cast, genre, reviews
5. Enter -1 to Exit
2
Veiw a list of top 10 movies overall or from a genre
Choose one:
1. View top 10
2. View top 10 from Genre
2
Enter Genre:
Comedy
[The man from toronto]
```

```
1. View a list of movies from a genre
2. Veiw a list of top 10 movies overall or from a genre
3. Search for a movie by name or/and by a genre
4. View details of a movie, see its summary, cast, genre, reviews
5. Enter -1 to Exit
-1
Choose any one:
1. Login
2. Signup
3. Other
4. Exit
4
```

```
Choose any one:
1. Login
2. Signup
3. Other
4. Exit
3
Since you are not a registered user, you only have these options to choose:
1. View a list of movies from a genre
```

```
2. Veiw a list of top 10 movies overall or from a genre
3. Search for a movie by name or/and by a genre
4. View details of a movie, see its summary, cast, genre, reviews
5. Enter -1 to Exit
3
Search for a movie by name or/and by a genre
Enter movie name:
RRR
[RRR]
1. View a list of movies from a genre
2. Veiw a list of top 10 movies overall or from a genre
3. Search for a movie by name or/and by a genre
4. View details of a movie, see its summary, cast, genre, reviews
5. Enter -1 to Exit

-1
Choose any one:
1. Login
2. Signup
3. Other
4. Exit
3
Since you are not a registered user, you only have these options to choose:
1. View a list of movies from a genre
2. Veiw a list of top 10 movies overall or from a genre
3. Search for a movie by name or/and by a genre
4. View details of a movie, see its summary, cast, genre, reviews
```

```
5. Enter -1 to Exit
4
View details of a movie
Enter movie_id:
1
[ Movie [movie_id=1, movie_name=Jagga jasoos, summary=Jagga jasoos is a drama based movie, Genre=Drama, Average_rating=5.3, user_id=2]]
1. View a list of movies from a genre
2. Veiw a list of top 10 movies overall or from a genre
3. Search for a movie by name or/and by a genre
4. View details of a movie, see its summary, cast, genre, reviews
5. Enter -1 to Exit
-1
Choose any one:
1. Login
2. Signup
3. Other
4. Exit
4
```

```
Choose any one:
1. Login
2. Signup
3. Other
4. Exit
1
Login:
Enter user-id:
1
Choose any one:
1. Edit
2. View
1
Edit:
Choose any of the given options:
1. Add a new movie entry
2. Edit an existing movie entry
3. Delete an existing movie entry
4. Add cast to a movie
5. Add a review to a movie
6. Rate a movie
2
Edit an existing movie entry:
Enter movie-id:
```

```
4

Type Genre :

Comedy

Type Summary:

bjbkfe

1 record inserted

[Movie [movie_id=1, avg_rating=5.0, user_Id=1, movieName=Jagga jasoos, cast=null,
genre=agga Jasoos (transl.?Detective Jagga) is a 2017 Indian Hindi-language musical
adventure comedy film[4] written and directed by Anurag Basu, and produced by Siddharth
Roy Kapur, Basu and Ranbir Kapoor. It features Kapoor and Katrina Kaif in the lead roles, and
tells the story of a teenage detective in search of his missing father. It was released on 14 July
2017[5] to mixed reviews from critics, but bagged ten nominations at the 63rd Filmfare
Awards, winning four of them for the films music., summary=musical], Movie [movie_id=2,
avg_rating=0.0, user_Id=2, movieName=Heroine, cast=null, genre= Laura and Massimo are
back and stronger than ever. But Massimos family ties and a mysterious man bidding for
Laura’s heart complicate the lovers lives, summary=Drama], Movie [movie_id=3,
avg_rating=0.0, user_Id=3, movieName=Through my window, cast=null, genre=Raquels
longtime crush on her next-door neighbor turns into something more when he starts
developing feelings for her, despite his familys objections., summary=Romantic], Movie
[movie_id=4, avg_rating=0.0, user_Id=4, movieName=Oxygen, cast=null, genre=bjbkfe,
summary=], Movie [movie_id=5, avg_rating=0.0, user_Id=1, movieName=The man from
toronto, cast=null, genre=A case of mistaken identity forces a bumbling entrepreneur to team
up with a notorious assassin known as The Man from Toronto in hopes of staying alive.,
summary=Comedy], Movie [movie_id=6, avg_rating=7.0, user_Id=2, movieName=RRR,
cast=null, genre=njknkjfs, summary=], Movie [movie_id=7, avg_rating=0.0, user_Id=1,
movieName=, cast=null, genre=Action, summary=bjbke]]

Choose any one:

1. Login

2. Signup

3. Other

4. Exit

1

Login:
```

```
Enter user-id:
1
Choose any one:
1. Edit
2. View
1
Edit:
Choose any of the given options:
1. Add a new movie entry
2. Edit an existing movie entry
3. Delete an existing movie entry
4. Add cast to a movie
5. Add a review to a movie
6. Rate a movie
3
Delete an existing movie entry:
Enter movie-id:
1
1 record inserted
[Movie [movie_id=2, avg_rating=0.0, user_Id=2, movieName=Heroine, cast=null, genre=
Laura and Massimo are back and stronger than ever. But Massimos family ties and a
mysterious man bidding for Laura’s heart complicate the lovers lives, summary=Drama],
Movie [movie_id=3, avg_rating=0.0, user_Id=3, movieName=Through my window, cast=null,
genre=Raquels longtime crush on her next-door neighbor turns into something more when he
starts developing feelings for her, despite his familys objections., summary=Romantic], Movie
[movie_id=4, avg_rating=0.0, user_Id=4, movieName=Oxygen, cast=null, genre=bjbkfe,
summary=], Movie [movie_id=5, avg_rating=0.0, user_Id=1, movieName=The man from
toronto, cast=null, genre=A case of mistaken identity forces a bumbling entrepreneur to team
up with a notorious assassin known as The Man from Toronto in hopes of staying alive.,
summary=Comedy], Movie [movie_id=6, avg_rating=7.0, user_Id=2, movieName=RRR,
cast=null, genre=njknkjfs, summary=], Movie [movie_id=7, avg_rating=0.0, user_Id=1,
movieName=, cast=null, genre=Action, summary=bjbke]]

Choose any one:

1. Login

2. Signup

3. Other

4. Exit

1

Login:

Enter user-id:

2

Choose any one:

1. Edit

2. View

1

Edit:

Choose any of the given options:

1. Add a new movie entry

2. Edit an existing movie entry

3. Delete an existing movie entry

4. Add cast to a movie

5. Add a review to a movie

6. Rate a movie

5

Add a review to a movie

Please type movie-name for which you want to add a review

Movie does not exist
```

```
Choose any one:
1. Login
2. Signup
3. Other
4. Exit
1
Login:
Enter user-id:
2
Choose any one:
1. Edit
2. View
2
Since you are not a registered user, you only have these options to choose:
1. View a list of movies from a genre
2. Veiw a list of top 10 movies overall or from a genre
3. Search for a movie by name or/and by a genre
4. View details of a movie, see its summary, cast, genre, reviews
5. Enter -1 to Exit
-1
Choose any one:
1. Login
2. Signup
3. Other
4. Exit
4
```