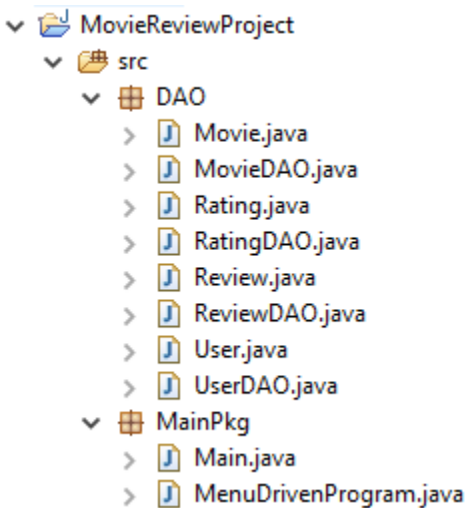


MovieReview console app in JAVA, using JDBC and MYSQL

Project Structure ->



Database schema ->

```
create table movie (
movie_id int auto_increment primary key,
title varchar(100), genre varchar(50), plot varchar(500));
```

```
create table movie_cast(
movie_id int, actor varchar(100),
FOREIGN KEY (movie_id) REFERENCES movie(movie_id));
```

```
create table user(
user_id int auto_increment primary key
,username varchar(100)
,name varchar(100));
```

```
create table rating(
movie_id int not null, user_id int not null,
ratinggiven int -- ( between 0-5)
,foreign key (movie_id) references movie(movie_id)
,foreign key (user_id) references user(user_id)
,primary key(movie_id,user_id));
```

```
create table review(
movie_id int not null, user_id int not null,
reviewgiven varchar(500)
,foreign key (movie_id) references movie(movie_id)
,foreign key (user_id) references user(user_id)
,primary key(movie_id,user_id));
```

We have used the Object Relational Mapping model, there were primarily 4 objects - movie, user, rating, review. We have tables for each one of them in our database, but since a movie's cast was a multivalued attribute, i.e. each movie consists of many actors we formed a separate table for it (movie_cast) which consists of a movie_id and the corresponding actors.

We formed a sample database, the some of the database entries looks like this -

movie table

movie_id	title	genre	plot
1	uri the surgical strike	action	Indian army execute a operation avenging the ...
2	the accidental prime minister	drama	explores manmohan singhs tenure as pm of indi...
3	gully boy	drama	A coming-of-age story based on the lives of str...
4	go goa gone	comedy	group of friends in a party on a remote island in...
5	happy new year	comedy	A gang of amateur dancers takes part in a danc...
6	jolly llb	comedy	A small time lawyer files a PIL in court against hi...
7	phillauri	fantasy	A man must marry a tree to ward off threats to ...
8	dangerous ishq	fantasy	searching for her love through ,many lives
9	horror story	horror	seven youngsters decide to spend a night in an ...
10	shaandar	romance	in the backdrop of a destination wedding in Lon...
11	raazi	thriller	kashmiri woman agrees to marry a Pakistani arm...
12	action replay	comedy	young man tries to revive his parents wilting ma...

movie_cast table

movie_id	actor
1	vicky kaushal
1	yami gautam
1	kirti kulhari
2	anupam kher
2	akshay khanna
2	abdul qadir
3	ranveer singh
3	alia bhatt
3	siddhant chaturvedi
4	saif ali khan
4	kunal khemu
4	vir das
5	shahrukh khan
5	deepkia padukone
5	boman irani
6	arshad warsi

user table

user_id	username	name
1	abhijain	abhishek jain
2	amikjain	amiksha jain
3	amita	amita
4	sharma_amay	amay sharma
5	kumarakshay	akshay kumar
6	nad_raf	rafael nadal
NULL	NULL	NULL

rating table

movie_id	user_id	ratinggiven
5	2	3
5	4	3
5	6	3
6	1	3
6	2	3
6	6	4
7	3	5
7	4	2
7	5	4
8	1	2

review table

movie_id	user_id	reviewgiven
3	1	not worth the money
3	4	inspirational movie, particulary for youth
3	5	very good depication of a personal journey, mu...
3	6	brilliant new age movie
12	2	hilarious science fiction movie
12	3	a must watch for everyone
12	6	absoulteiv non sense

CODE -

Please skip to page 14 for output and screenshots

DAO Package

DAO.Movie

```
package DAO;
import java.util.List;
public class Movie {
    int movie_id;
    String title;
    String genre;
    String plot;
    List<String> cast;

    public Movie() {
    }
    public Movie(String title, String genre, String plot, List<String> cast) {
        this.title = title;
        this.genre = genre;
        this.plot = plot;
        this.cast = cast;
    }
    public Movie(int movie_id, String title, String genre, String plot,
        List<String> cast) {
        this.movie_id = movie_id;
        this.title = title;
        this.genre = genre;
        this.plot = plot;
        this.cast = cast;
    }
    public String getMovie_id() {
        return title;
    }
    public void setMovie_id(int id) {
        this.movie_id = id;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getGenre() {
        return genre;
    }
    public void setGenre(String genre) {
        this.genre = genre;
    }
    public String getPlot() {
        return plot;
    }
    public void setPlot(String plot) {
        this.plot = plot;
    }
    public List<String> getCast() {
        return cast;
    }
    public void setCast(List<String> cast) {
        this.cast = cast;
    }
    public String toString() {
```

```

        return "Movie [title=" + title + ", genre=" + genre + ",\n    plot=" + plot
            + ",\n    cast=" + cast + "]\n";
    }
}

```

DAO.MovieDAO

```

package DAO;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

public class MovieDAO {
    Connection con;
    public MovieDAO(Connection con) {
        this.con = con;
    }
    public int insertMovie(Movie m) {
        int s = 0;
        try {
            // to get id
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("select count(*) from movie");
            rs.next();
            m.movie_id = rs.getInt(1)+1;
            // add details of movie in movie table
            s = stmt.executeUpdate("insert into movie values (" + m.movie_id + ", "
                + m.getTitle() + ", " + m.getGenre() + ", " + m.getPlot()
                + ")");
            // add cast in movie_cast table
            for (String actor : m.getCast()) {
                stmt.executeUpdate("insert into movie_cast values (" + m.movie_id
                    + ", " + actor + ")");
            }
            rs.close();
            System.out.println("Movie successfully inserted");
        } catch (Exception e) {
            e.printStackTrace();
        }
        return s;
    }

    public int updateMovie(Movie m) {
        // movie name is unique, we assume cast can't be updated
        int s = 0;
        try {
            // update details of movie in movie table
            Statement stmt = con.createStatement();
            s = stmt.executeUpdate("update movie set title="
                + m.getTitle() + ", genre=" + m.getGenre() + ", plot="
                + m.getPlot() + " where movie_id=" + m.movie_id);
            System.out.println("Movie successfully updated");
        } catch (Exception e) {
            e.printStackTrace();
        }
        return s;
    }

    // search for a movie by name and/or genre

```

```

public Movie searchMovieByName(String title) {
    Movie m = null;
    try {
        Statement stmt = con.createStatement();
        List<String> cast = new ArrayList<>();
        ResultSet rs = stmt
            .executeQuery("select actor from movie_cast where movie_cast.movie_id
=(select movie.movie_id from movie where title='"
                                + title + "'");

        while (rs.next())
            cast.add(rs.getString(1));

        rs = stmt.executeQuery("select * from movie where title='" + title
                                + "' limit 1");
        while (rs.next())
            m = new Movie(rs.getInt(1), rs.getString(2), rs.getString(3),
                rs.getString(4), cast);

        rs.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return m;
}

public List<Movie> searchMovieByGenre(String genre) {
    List<Movie> movies = new ArrayList<>();
    try {
        Statement stmt = con.createStatement();
        String title;
        ResultSet rs = stmt
            .executeQuery("select title from movie where genre='"
                                + genre + "'");

        while (rs.next()) {
            title = rs.getString(1);
            movies.add(searchMovieByName(title));
        }
        rs.close();

    } catch (Exception e) {
        e.printStackTrace();
    }
    return movies;
}

// top 10 movies according to average rating
public List<Movie> searchTop10Movies() {
    List<Movie> movies = new ArrayList<>();
    try {
        Statement stmt = con.createStatement();
        String title;
        Statement stmttemp = con.createStatement();
        ResultSet rstemp;
        ResultSet rs = stmt
            .executeQuery("select movie_id from rating group by movie_id order by
avg(ratinggiven) desc limit 10");
        while (rs.next()) {
            rstemp = stmttemp
                .executeQuery("select title from movie where movie_id ="
                                + rs.getInt(1));

            rstemp.next();
            title = rstemp.getString(1);

```

```

        movies.add(searchMovieByName(title));
    }

    } catch (Exception e) {
        e.printStackTrace();
    }
    return movies;
}
}

```

DAO.Rating

```

package DAO;
public class Rating {
    int movie_id;
    int user_id;
    int ratinggiven;

    Rating() {
    }
    public Rating(int user_id, int rating){
        this.user_id=user_id;
        this.ratinggiven = rating;
    }
    public Rating(int movie_id, int user_id, int rating) {
        this.movie_id = movie_id;
        this.user_id = user_id;
        this.ratinggiven = rating;
    }
    public int getMovie_id() {
        return movie_id;
    }
    public void setMovie_id(int movie_id) {
        this.movie_id = movie_id;
    }
    public int getUser_id() {
        return user_id;
    }
    public void setUser_id(int user_id) {
        this.user_id = user_id;
    }
    public int getRatinggiven() {
        return ratinggiven;
    }
    public void setRatinggiven(int ratinggiven) {
        this.ratinggiven = ratinggiven;
    }
    public String toString() {
        return "Rating [user_id=" + user_id
            + ", ratinggiven=" + ratinggiven + "]";
    }
}
}

```

DAO.RatingDAO

```

package DAO;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;

```

```

import java.util.List;

public class RatingDAO {
    Connection con;
    public RatingDAO(Connection con) {
        this.con = con;
    }
    public int insertRating(String title,String username,int rating) {
        int s = 0;
        try {
            Rating r = new Rating();
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("select movie_id from movie where title='"+title+"'");
            rs.next(); r.setMovie_id(rs.getInt(1));
            rs = stmt.executeQuery("select user_id from user where username='"+username+"'");
            rs.next(); r.setUser_id(rs.getInt(1));
            r.setRatinggiven(rating);
            s = stmt.executeUpdate("insert into rating values (" + r.movie_id
                                + "," + r.user_id + "," + r.ratinggiven + ")");
            rs.close();
            System.out.println("Rating successfully given");
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return s;
    }
    public List<Rating> searchRating(String title){
        Statement stmt;
        List<Rating> ratings = new ArrayList<>();
        try {
            int id;
            stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("select movie_id from movie where title='"+title+"'");
            rs.next(); id = rs.getInt(1);
            rs = stmt.executeQuery("select user_id,ratinggiven from rating where movie_id="+id);
            while(rs.next()){
                ratings.add(new Rating(rs.getInt(1),rs.getInt(2)));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return ratings;
    }
}

```

DAO.Review

```

package DAO;
public class Review {
    int movie_id;
    int user_id;
    String reviewgiven;
    Review() {
    }
    public Review(int user_id, String review){
        this.user_id=user_id;
        this.reviewgiven = review;
    }
    public Review(int movie_id, int user_id, String review) {
        this.movie_id = movie_id;
        this.user_id = user_id;
        this.reviewgiven = review;
    }
}

```

```

    }
    public int getMovie_id() {
        return movie_id;
    }
    public void setMovie_id(int movie_id) {
        this.movie_id = movie_id;
    }
    public int getUser_id() {
        return user_id;
    }
    public void setUser_id(int user_id) {
        this.user_id = user_id;
    }
    public String getReviewgiven() {
        return reviewgiven;
    }
    public void setReviewgiven(String reviewgiven) {
        this.reviewgiven = reviewgiven;
    }
    public String toString() {
        return "Review [user_id=" + user_id
            + ", reviewgiven=" + reviewgiven + "]";
    }
}

```

DAO.ReviewDAO

```

package DAO;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
public class ReviewDAO {
    Connection con;
    public ReviewDAO(Connection con) {
        this.con = con;
    }
    public int insertReview(String title,String username,String review) {
        int s = 0;
        try {
            Review r = new Review();
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("select movie_id from movie where title='"+title+"'");
            rs.next(); r.setMovie_id(rs.getInt(1));
            rs = stmt.executeQuery("select user_id from user where username='"+username+"'");
            rs.next(); r.setUser_id(rs.getInt(1));
            r.setReviewgiven(review);
            s = stmt.executeUpdate("insert into review values (" + r.movie_id
                + "," + r.user_id + "," + r.reviewgiven + ")");
            rs.close();
            System.out.println("Review successfully inserted");
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return s;
    }
    public List<Review> searchReview(String title){
        Statement stmt;
        List<Review> reviews = new ArrayList<>();
        try {

```



```

        int id;
        stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery("select movie_id from movie where title='"+title+"'");
        rs.next(); id = rs.getInt(1);
        rs = stmt.executeQuery("select user_id,reviewgiven from review where movie_id="+id);
        while(rs.next()){
            reviews.add(new Review(rs.getInt(1),rs.getString(2)));
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return reviews;
}
}

```

DAO.User

```

package DAO;
public class User {
    int user_id;
    public String username;
    String name;
    User() {
    }
    public User(String username, String name) {
        this.username = username;
        this.name = name;
    }
    public User(int user_id, String username, String name) {
        this.user_id = user_id;
        this.username = username;
        this.name = name;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String toString() {
        return "User [username=" + username + ", name=" + name + "]";
    }
}

```

DAO.UserDAO

```

package DAO;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import java.util.stream.Collectors;

public class UserDAO {
    Connection con;
    public UserDAO(Connection con) {

```

```

        this.con = con;
    }
    public int insertUser(User u) {
        int s = 0;
        try {
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("select count(*) from user");
            rs.next();
            u.user_id = rs.getInt(1)+1;
            s = stmt.executeUpdate("insert into user values (" + u.user_id + ", " + u.username + ", " + u.name +
"");
            rs.close();
            System.out.println("User inserted successfully");
        } catch (Exception e) {
            System.out.println("error in userdao insertUser function");
        }
        return s;
    }
    public List<User> listUser() {
        List<User> users = new ArrayList<>();
        ResultSet rs;
        try {
            Statement stmt = con.createStatement();
            rs = stmt.executeQuery("select * from user");
            while (rs.next())
                users.add(new User(rs.getInt(1), rs.getString(2), rs
                    .getString(3)));
            rs.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return users;
    }
    public boolean authenticate(String username){
        if(listUser().stream().map(u->u.username).collect(Collectors.toList()).contains(username)){
            return true;
        }
        return false;
    }
}

```

MainPkg Package

MainPkg.MenuDrivenProgram

```

package MainPkg;
import java.sql.Connection;
import java.sql.DriverManager;
import java.util.Arrays;
import java.util.List;
import java.util.Scanner;
import java.util.stream.Collectors;

import DAO.*;
public class MenuDrivenProgram {
    Connection con;
    MovieDAO moviedao;
    UserDAO userdao;
    ReviewDAO reviewdao;
    RatingDAO ratingdao;
    MenuDrivenProgram(String driver, String URL, String user, String pass){

```

```

        con=null;
        try {
            Class.forName(driver);
            con = DriverManager.getConnection(URL, user, pass);
            moviedao = new MovieDAO(con);
            userdao = new UserDAO(con);
            reviewdao = new ReviewDAO(con);
            ratingdao = new RatingDAO(con);
        }
        catch(Exception e)
        {}
    }
    public void takeInput(String s, Scanner sc){
        int key;
        while(true){
            System.out.println("*****");
            System.out.println("1.Go Back\n2.View detials of a movie\n3.View movies from a genre\n4.View top
10 movies");

            if(!s.equals(""))
                System.out.println("5.Add new movie\n6.Edit movie\n7.Add review\n8.Rate movie");
            key = Integer.parseInt(sc.nextLine().trim());
            if(key==1)
                break;
            if(key==2||key==3||key==4)
                guest(key,sc);
            else if(!s.equals("")){ // i.e. if user is registered
                if(key==5||key==6)
                    user(key,sc);
                else if(key==7||key==8)
                    user(key,s,sc);
            }
            else{
                System.out.println("Please enter one of the choices given to you");
                takeInput(s,sc);
            }
        }
    }
    public void guest(int key,Scanner sc){
        if(key==2){
            String title;
            System.out.println("Enter movie name");
            title = sc.nextLine().trim();
            Movie m = moviedao.searchMovieByName(title);
            if(m!=null){
                System.out.println(m);
                System.out.println("if you want to see reviews, press 1 else 0");
                if(Integer.parseInt(sc.nextLine().trim())==1){
                    reviewdao.searchReview(title).stream().forEach(System.out::println);
                }
                System.out.println("if you want to see ratings given by users press 1 else 0");
                if(Integer.parseInt(sc.nextLine().trim())==1){
                    ratingdao.searchRating(title).stream().forEach(System.out::println);
                }
            }
            else{
                System.out.println("Movie doesn't exist in database");
            }
        }
        else if(key==3){
            String genre;

```

```

        System.out.println("Enter movie genre");
        genre = sc.nextLine().trim();
        List<Movie>m = moviedao.searchMovieByGenre(genre);
        if(m.isEmpty()){
            System.out.println("No movies of this genre in database");
        }
        else{
            m.stream().forEach(System.out::println);
        }
    }
    else if(key==4){
        System.out.println("Movies are sorted in descending order of the avg rating they got -->\n");
        moviedao.searchTop10Movies().stream().forEach(System.out::println);
    }
}

public void user(int key,Scanner sc){
    Movie m = new Movie();
    if(key==5){
        System.out.println("Enter movie title");
        m.setTitle(sc.nextLine().trim());
        System.out.println("Enter movie genre one of"
            + "\n-action\n-comedy\n-drama\n-fantasy\n-horror\n-romance\n-thriller");
        m.setGenre(sc.nextLine().trim());
        System.out.println("Enter plot");
        m.setPlot(sc.nextLine().trim());
        System.out.println("Enter movie cast, please separate each member of cast with a ,(comma)");
        m.setCast(Arrays.asList(sc.nextLine().split(",")).stream().map(x->x.trim()).collect(Collectors.toList()));
        moviedao.insertMovie(m);
    }
    else if(key==6){
        System.out.println("Enter movie title, whose details to be updated");
        m.setTitle(sc.nextLine().trim());
        m = moviedao.searchMovieByName(m.getTitle());
        System.out.println("if you want to update the movie name type 1 else 0");
        if(Integer.parseInt(sc.nextLine().trim())==1){
            System.out.println("Enter updated title");
            m.setTitle(sc.nextLine().trim());
        }
        System.out.println("if you want to update the movie genre type 1 else 0");
        if(Integer.parseInt(sc.nextLine().trim())==1){
            System.out.println("Enter updated genre");
            m.setGenre(sc.nextLine().trim());
        }
        System.out.println("if you want to update the movie plot type 1 else 0");
        if(Integer.parseInt(sc.nextLine().trim())==1){
            System.out.println("Enter updated plot");
            m.setPlot(sc.nextLine().trim());
        }
        moviedao.updateMovie(m);
    }
}

public void user(int key, String username,Scanner sc){
    if(key==7){
        System.out.println("Enter movie title for which you want to write review");
        String title = sc.nextLine().trim();
        System.out.println("Enter the review");
        String review = sc.nextLine().trim();
        reviewdao.insertReview(title, username, review);
    }
    else if(key==8){
        System.out.println("Enter movie title for which you want to give rating");
    }
}

```

```

        String title = sc.nextLine().trim();
        System.out.println("Enter the rating - a whole number between 0 and 5");
        int rating = Integer.parseInt(sc.nextLine().trim());
        ratingdao.insertRating(title,username,rating);
    }
}

MainPkg.Main
package MainPkg;
import java.util.Scanner;

import DAO.User;
public class Main {
    public static void main(String[] args) {
        MenuDrivenProgram mdp =new MenuDrivenProgram("com.mysql.jdbc.Driver",
        "jdbc:mysql://localhost:3306/moviesreview", "root", "password");
        int keyauth;
        Scanner sc = new Scanner(System.in);
        while(true){
            System.out.println("-----");
            System.out.println("
                                1.Registered User"
                                + "
                                2.Guest"
                                + "
                                3.Register yourself"
                                + "
                                4.Exit");
            System.out.println("-----");
            keyauth = Integer.parseInt(sc.nextLine().trim());
            if(keyauth==1){
                System.out.println("Enter username");
                String username = sc.nextLine().trim();
                if(mdp.userdao.authenticate(username)){
                    mdp.takeInput(username,sc);
                }
            }
            else{
                System.out.println("You are not a registered user");
                continue;
            }
        }
        else if(keyauth==2){
            mdp.takeInput("",sc);
        }
        else if(keyauth==3){
            System.out.println("Enter name");
            String name = sc.nextLine().trim();
            System.out.println("Enter username");
            String username = sc.nextLine().trim();
            User u = new User(username,name);
            mdp.userdao.insertUser(u);
        }
        else if(keyauth==4){
            break;
        }
        else{
            System.out.println("Please enter one of 1,2,3,4");
        }
    }
}
sc.close();
}
}

```

Output for different functional requirements -

This is the main menu everyone is greeted with-

```
-----  
1.Registered User  
2.Guest  
3.Register yourself  
4.Exit  
-----
```

Now if you are a registered user, you choose option 1(additional functionalities),

And if you are a guest and want to do guest browsing you choose option 2.

You can also register by choosing the option 3.

You can exit by choosing option 4.

Options for guest -

```
|*****  
1.Go Back  
2.View details of a movie  
3.View movies from a genre  
4.View top 10 movies
```

Options for registered user -

You enter your username, and once authenticated that indeed you are a registered user, then the following options are shown to you -

4 options that are - add new movie, edit movie, add review and rate a movie are only for registered users, rest all are same.

```
Enter username  
abhi Jain  
|*****  
1.Go Back  
2.View details of a movie  
3.View movies from a genre  
4.View top 10 movies  
5.Add new movie  
6.Edit movie  
7.Add review  
8.Rate movie
```

Now we will test all options one by one -

2) View details of a movie

```
*****
1.Go Back
2.View details of a movie
3.View movies from a genre
4.View top 10 movies
2
Enter movie name
drishyam
Movie [title=drishyam, genre=thriller,
      plot=a man tries to save his family from the dark side of law after they commit an unexpected crime,
      cast=[ajay devgn, tabu, shriya sharan]]
if you want to see reviews, press 1 else 0
1
Review [user_id=1, reviewgiven=these type of movies can change the indian cinema]
Review [user_id=2, reviewgiven=quite hypothetical, never happens in the real world]
Review [user_id=5, reviewgiven=just go and watch, dont wait]
Review [user_id=6, reviewgiven=this is what you say a movie is]
if you want to see ratings given by users press 1 else 0
1
Rating [user_id=1, ratinggiven=5]
Rating [user_id=2, ratinggiven=5]
Rating [user_id=5, ratinggiven=5]
*****
1.Go Back
2.View details of a movie
3.View movies from a genre
4.View top 10 movies
```

We are asked to enter the movie name, if it exists in the database it shows the details of the movie, after that the user is asked if he wants to see the reviews of the movie, if opted they are shown. Similarly the user is asked if he wants to see the ratings given by different users, if opted they are shown.

3) View movies from a genre

```
*****
1.Go Back
2.View details of a movie
3.View movies from a genre
4.View top 10 movies
3
Enter movie genre
comedy
Movie [title=go goa gone, genre=comedy,
      plot=group of friends in a party on a remote island in Goa find that island is infested with Zombies,
      cast=[saif ali khan, kunal khemu, vir das]]
Movie [title=happy new year, genre=comedy,
      plot=A gang of amateur dancers takes part in a dancing competition to avenge their personal vendetta,
      cast=[shahrukh khan, deepika padukone, boman irani]]
Movie [title=jolly llb, genre=comedy,
      plot=A small time lawyer files a PIL in court against his own idols,
      cast=[arshad warsi, amrita rao, boman irani]]
Movie [title=action replay, genre=comedy,
      plot=young man tries to revive his parents wilting marriage by travelling to the past in 1970s,
      cast=[akshay kumar, aishwarya rai, om puri]]
*****
```

The user is asked to enter the genre, relevant movies from the database are shown. In the above example we input comedy genre, and hence all comedy movies are shown. If no movies of any genre are present “No movies of this genre” is printed

4) View top 10 movies -

```
*****
1.Go Back
2.View details of a movie
3.View movies from a genre
4.View top 10 movies
4
Movies are sorted in descending order of the avg rating they got -->

Movie [title=drishyam, genre=thriller,
       plot=a man tries to save his family from the dark side of law after they commit an unexpected crime,
       cast=[ajay devgn, tabu, shriya sharan]]
Movie [title=barfi, genre=drama,
       plot=Three young people learn that love can not be defined by societys definition of normal and abnormal.,
       cast=[ranbir kapoor, priyanka chopra, ileana d cruz]]
Movie [title=raazi, genre=thriller,
       plot=kashmiri woman agrees to marry a Pakistani army officer in order to spy on Pakistan Indo-Pakistan War of 1971,
       cast=[alia bhatt, vicky kaushal, rajit kapoor]]
Movie [title=rab ne bana di jodi, genre=romance,
       plot=an awe inspiring love journey of an ordinary man Suri and his total opposite love Taani.,
       cast=[shahrukh khan, anushka sharma, vinay pathak]]
Movie [title=gully boy, genre=drama,
       plot=A coming-of-age story based on the lives of street rappers in Mumbai,
       cast=[ranveer singh, alia bhatt, siddhant chaturvedi]]
Movie [title=oh my god, genre=fantasy,
       plot=a shopkeeper takes God to court when his shop is destroyed by an earthquake.,
       cast=[paresh rawal, akshay kumar, mithun chakraborty]]
Movie [title=uri the surgical strike, genre=action,
       plot=Indian army execute a operation avenging the killing of fellow army men at their base by a terrorist group,
       cast=[vicky kaushal, yami gautam, kirti kulhari]]
Movie [title=main hoon na, genre=drama,
       plot=an army major goes undercover as college student to accomplish both personal and professional pursuits,
       cast=[shahrukh khan, sushmita sen, suniel shetty]]
Movie [title=love aa j kal, genre=romance,
       plot=An old-fashioned romantic tells her love story to a modern man who has just broken up with his girlfriend,
       cast=[saif ali khan, deepika padukone, rishi kapoor]]
Movie [title=phillauri, genre=fantasy,
       plot=A man must marry a tree to ward off threats to his love-life,
       cast=[anushka sharma, diljit dosanjh, nidhi bisht]]
*****
```

All movies are sorted in descending order based on the average rating they have received and the top 10 movies are displayed.

5) Add new movie -

```
Enter username
amita
*****
1.Go Back
2.View details of a movie
3.View movies from a genre
4.View top 10 movies
5.Add new movie
6.Edit movie
7.Add review
8.Rate movie
5
Enter movie title
rang de basanti
Enter movie genre one of
-action
-comedy
-drama
-fantasy
-horror
-romance
-thriller
drama
Enter plot
story of six young Indians who assist an English woman to film a documentary on the freedom fighters from their past
Enter movie cast, please separate each member of cast with a ,(comma)
amir khan,sharman joshi,soha ali khan
Movie successfully inserted
*****
```

On successful authentication of a user, he can add a movie, user is asked to enter movie title, genre, plot and cast. "Movie successfully inserted" is printed on successful insertion in db.

6) Edit movie -

```
*****
1.Go Back
2.View details of a movie
3.View movies from a genre
4.View top 10 movies
5.Add new movie
6.Edit movie
7.Add review
8.Rate movie
6
Enter movie title, whose details to be updated
uri the surgical strike
if you want to update the movie name type 1 else 0
1
Enter updated title
uri: the surgical strike
if you want to update the movie genre type 1 else 0
1
Enter updated genre
drama
if you want to update the movie plot type 1 else 0
0
Movie successfully updated
*****
```

User is asked one by one the details which he wants to update, as per his choice he can update the necessary details.

```
2
Enter movie name
uri: the surgical strike
Movie [title=uri: the surgical strike, genre=drama,
      plot=Indian army execute a operation avenging the killing of fellow army men at their base by a terrorist group,
      cast=[vicky kaushal, yami gautam, kirti kulhari]]
if you want to see reviews, press 1 else 0
```

We can cross check by printing details of the movie, and yes they have indeed gotten updated.

7) Add review -

```
Enter username
abhijain
*****
1.Go Back
2.View details of a movie
3.View movies from a genre
4.View top 10 movies
5.Add new movie
6.Edit movie
7.Add review
8.Rate movie
7
Enter movie title for which you want to write review
rang de basanti
Enter the review
amir khan is a fine actor and proves himself again by producing another gem
Review successfully inserted
*****
```

After authentication is done, user is asked the title of the movie he wants to review, and then is asked to fill the review. On successful insertion of review "Review successfully inserted" is printed.

We can cross check by printing reviews of this movie -

```
rang de basanti
Movie [title=rang de basanti, genre=drama,
      plot=story of six young Indians who assist an English woman to film a documentary on the freedom fighters from their past,
      cast=[amir khan, sharman joshi, soha ali khan]]
if you want to see reviews, press 1 else 0
1
Review [user_id=1, reviewgiven=amir khan is a fine actor and proves himself again by producing another gem]
if you want to see ratings given by users press 1 else 0
```

And yes as expected review gets inserted.

8) Rate movie -

```
*****
1.Go Back
2.View details of a movie
3.View movies from a genre
4.View top 10 movies
5.Add new movie
6.Edit movie
7.Add review
8.Rate movie
8
Enter movie title for which you want to give rating
rang de basanti
Enter the rating - a whole number between 0 and 5
4
Rating successfully given
*****
```

User is asked for the title of the movie he wants to rate, and then is asked the rating(a whole number between 0 and 5).

On successful insertion “Rating successfully given” is printed.

We can crosscheck by printing the ratings of this movie -

```
Enter movie name
rang de basanti
Movie [title=rang de basanti, genre=drama,
       plot=story of six young Indians who assist an English woman to film a documentary on the freedom fighters from their past,
       cast=[amir khan, sharma joshi, soha ali khan]]
if you want to see reviews, press 1 else 0
0
if you want to see ratings given by users press 1 else 0
1
Rating [user_id=1, ratinggiven=4]
*****
```

And yes as expected rating gets inserted, with user_id whose account was used for adding the rating.

-----THE END-----

REPORT BY **ABHISHEK JAIN**, TIMES OF INDIA, TIMES INTERNET.
FOR TIL TRAINING AUG'22 BATCH PROJECT.