## horizontal line



Movie Review System

1.1.1

**─**

Ashish Singh

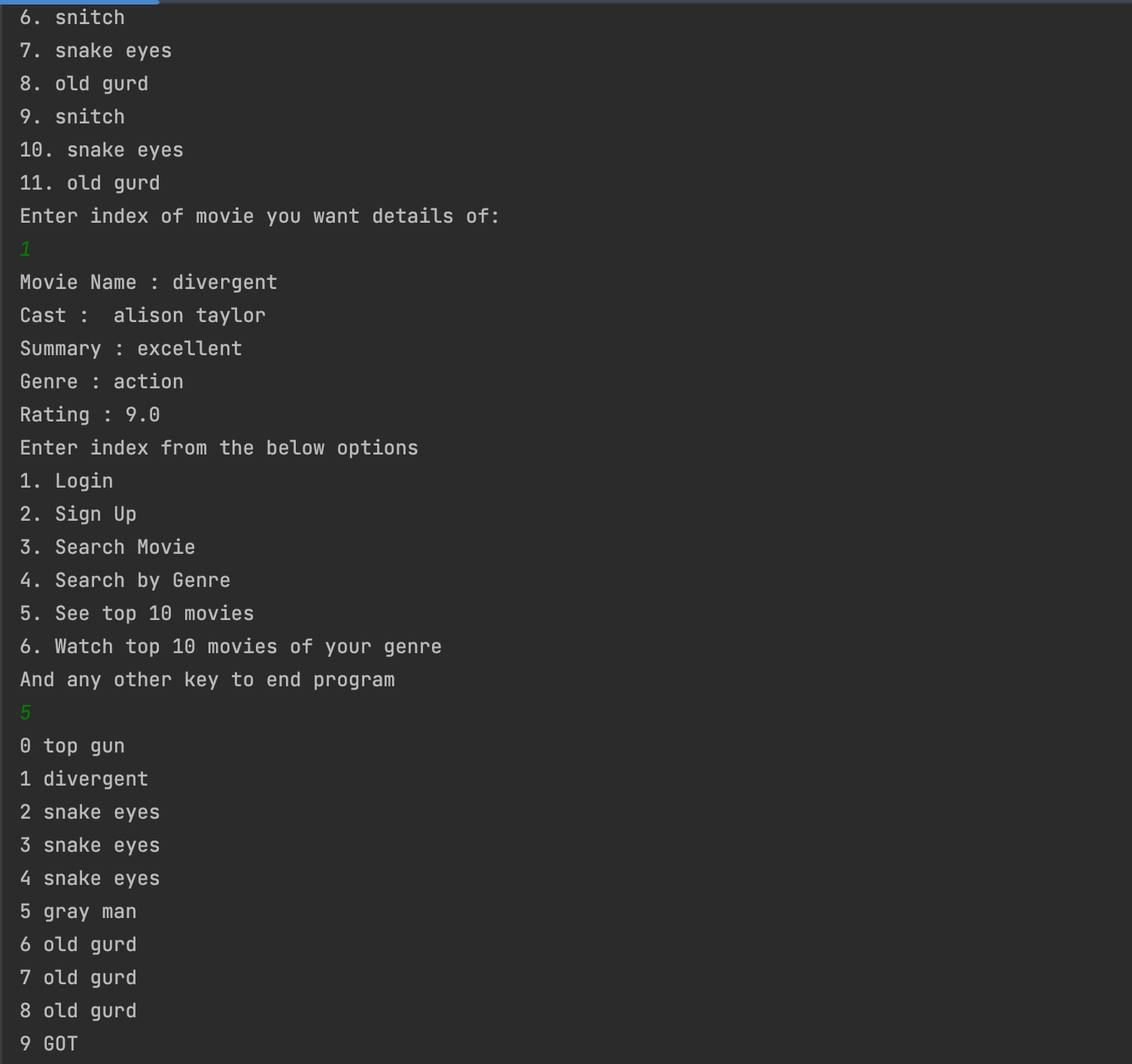
Times Internet

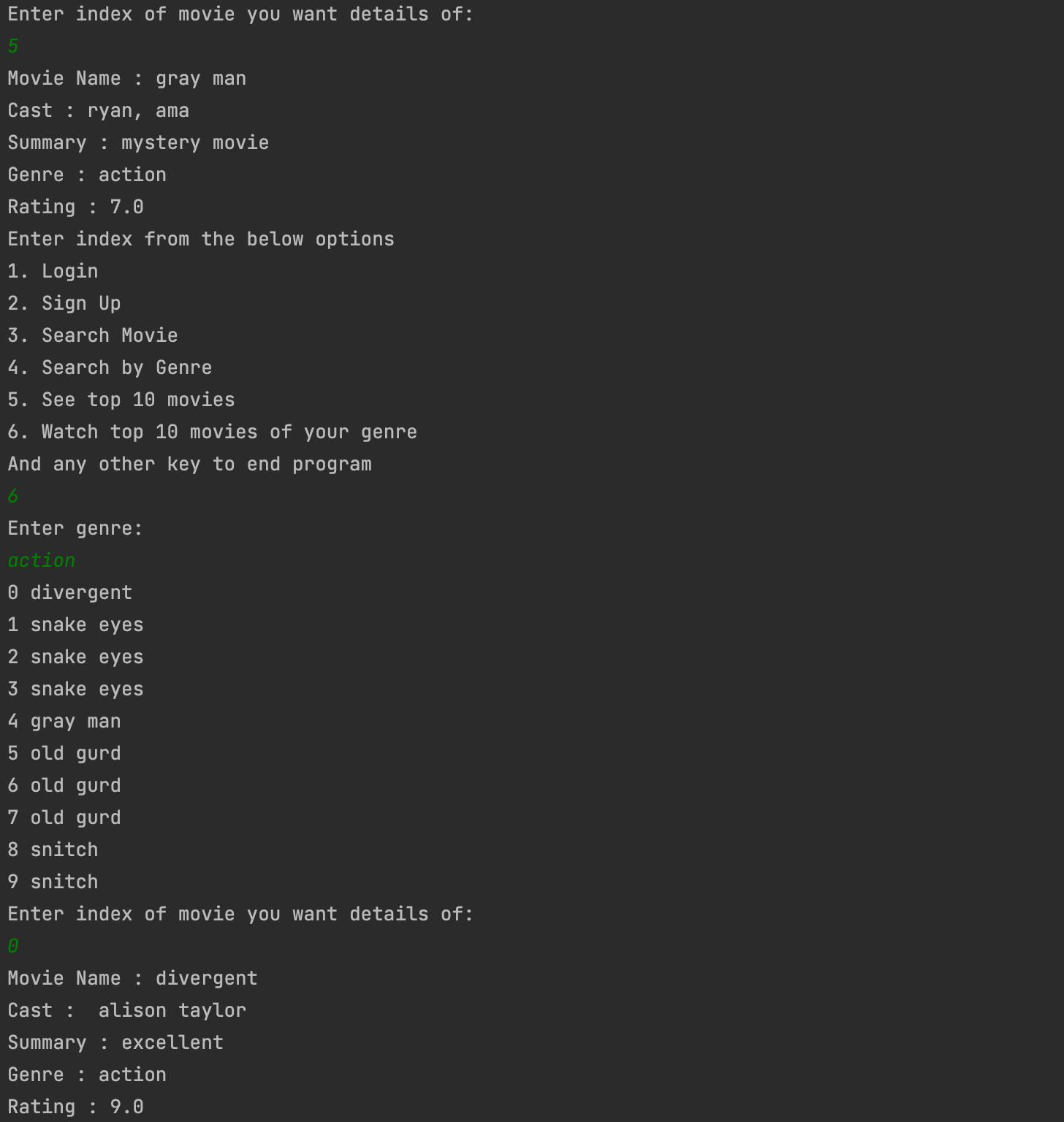
# 

# Outputs for normal User

# 

# 

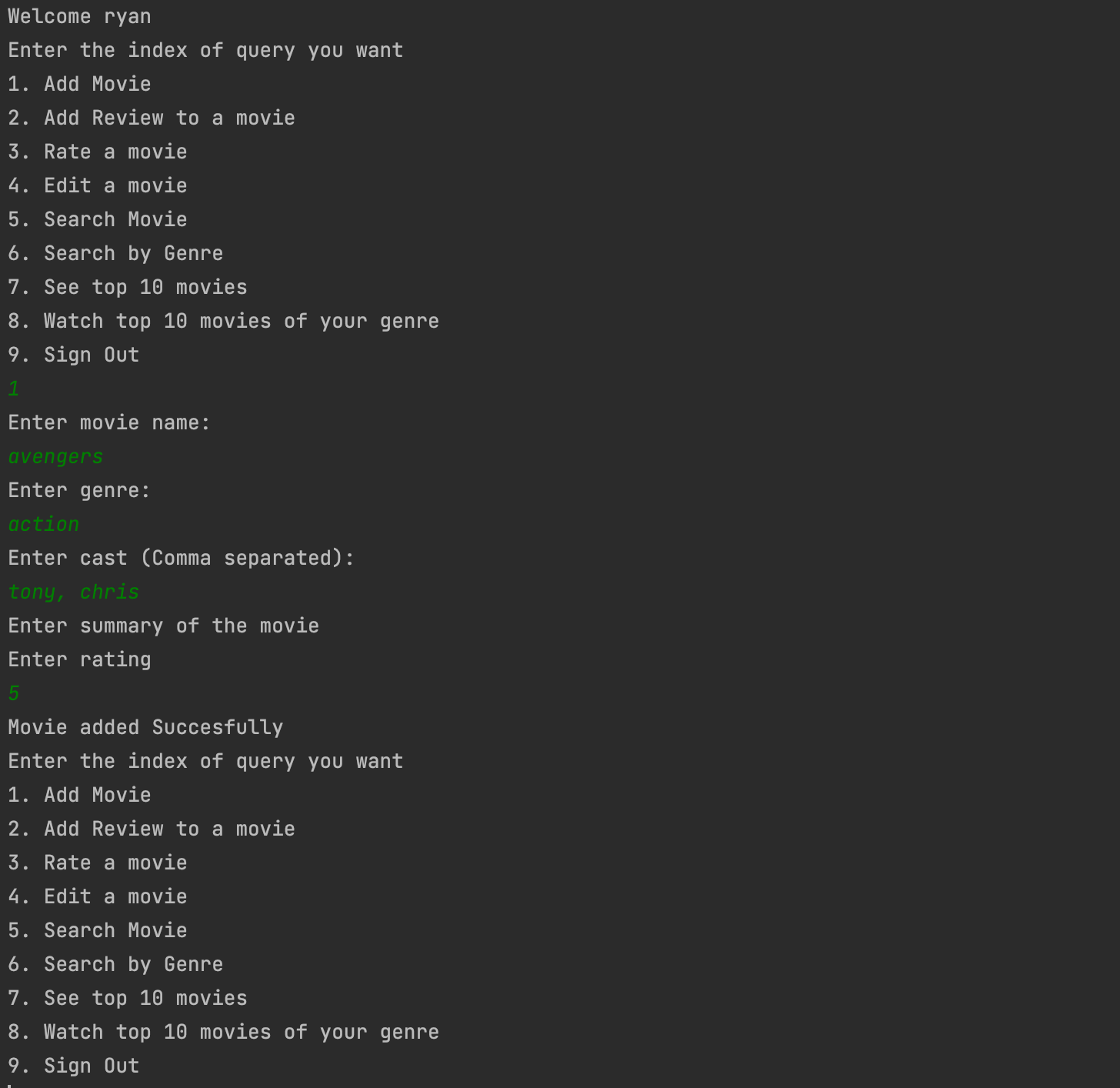


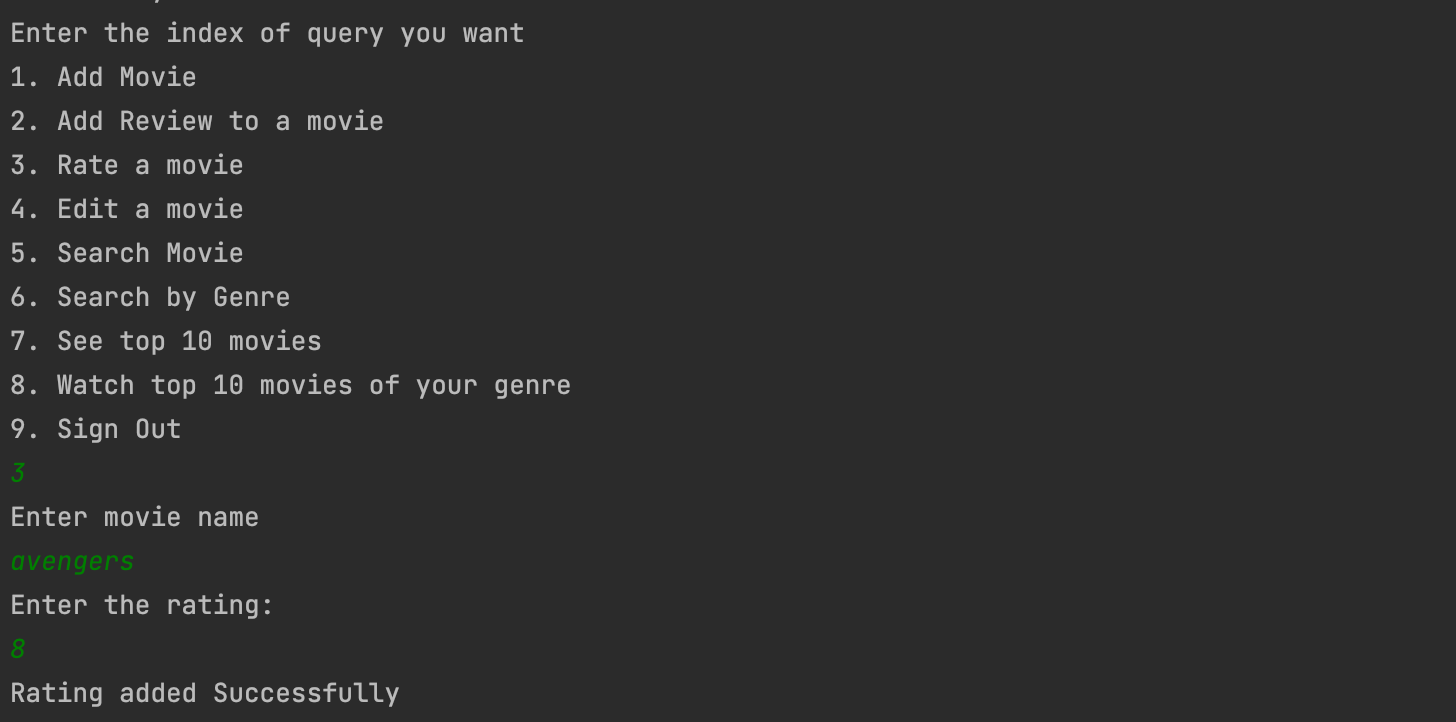
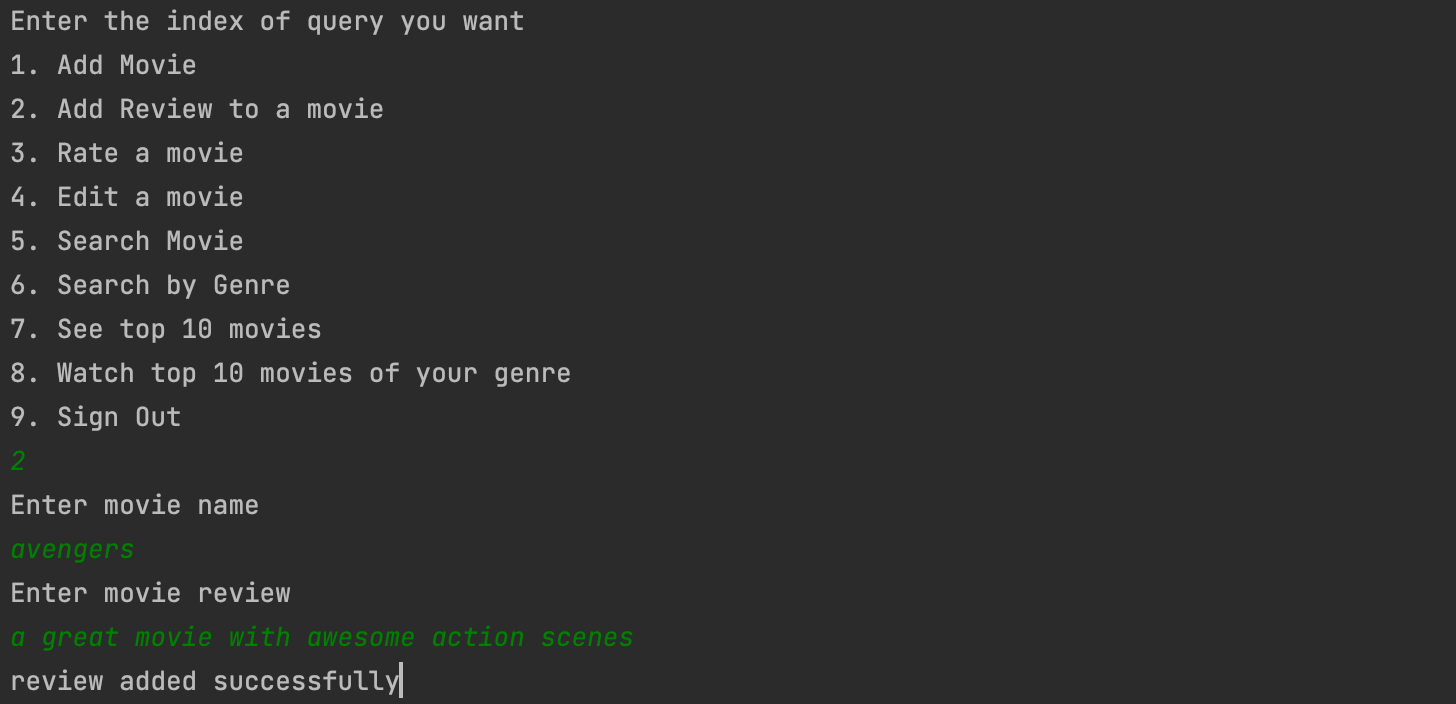


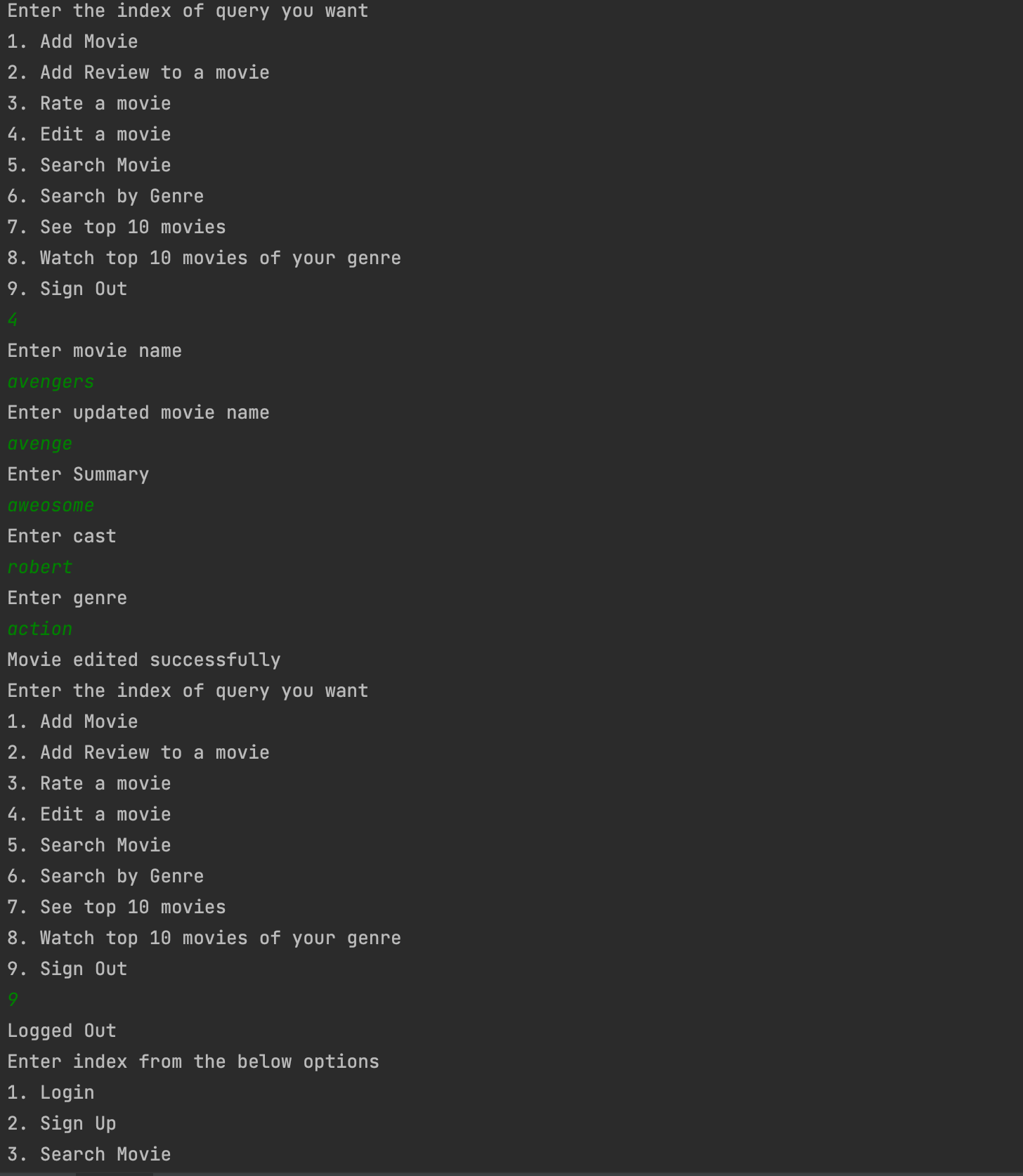
# 

# 

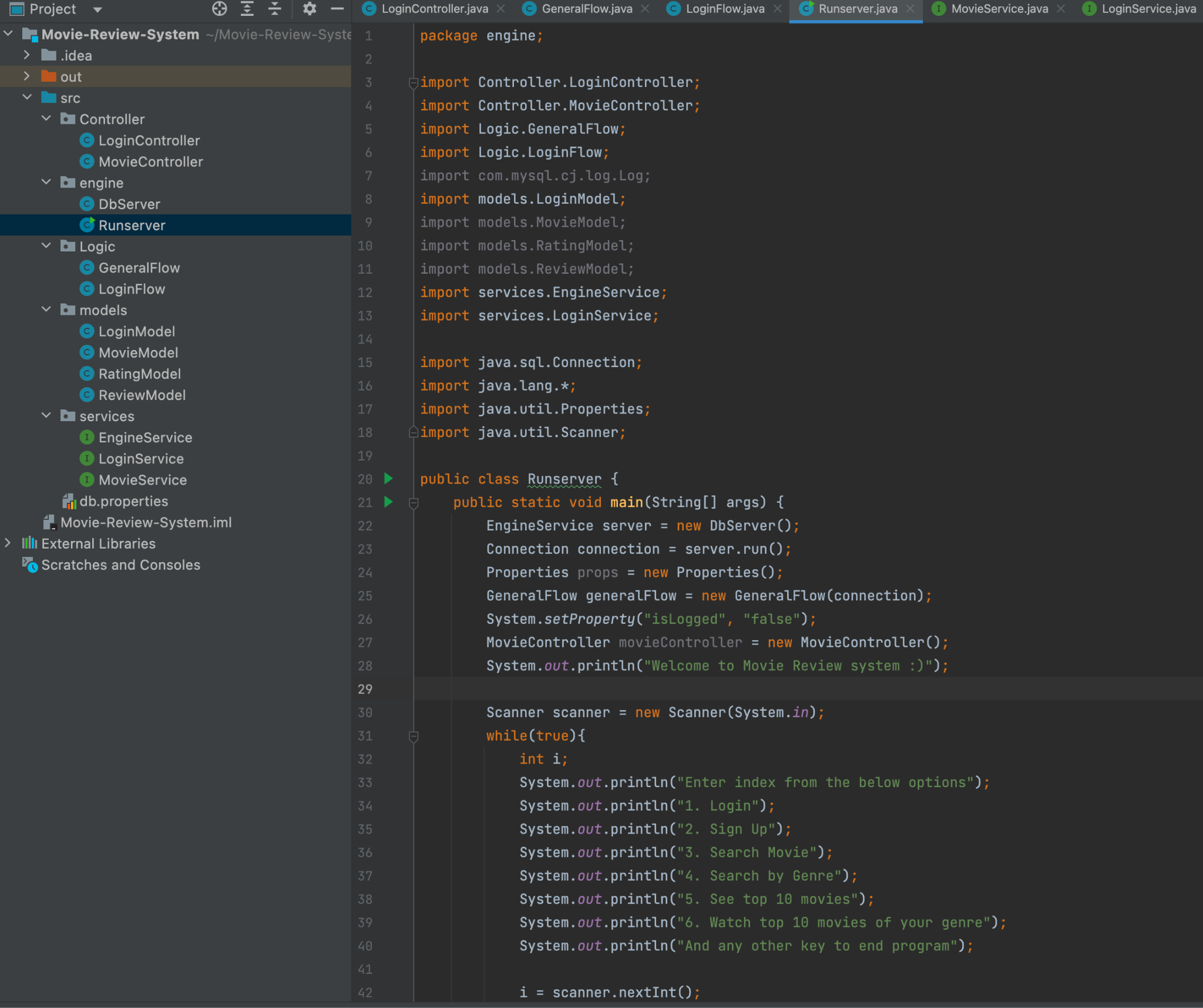
# Output for Logged in User







Source Code

—- Project Structure

Runserver.java

package engine;

import Controller.LoginController;

import Controller.MovieController;

import Logic.GeneralFlow;

import Logic.LoginFlow;

import com.mysql.cj.log.Log;

import models.LoginModel;

import models.MovieModel;

import models.RatingModel;

import models.ReviewModel;

import services.EngineService;

import services.LoginService;

import java.sql.Connection;

import java.lang.\*;

import java.util.Properties;

import java.util.Scanner;

public class Runserver {

public static void main(String[] args) {

EngineService server = new DbServer();

Connection connection = server.run();

Properties props = new Properties();

GeneralFlow generalFlow = new GeneralFlow(connection);

System.*setProperty*("isLogged", "false");

MovieController movieController = new MovieController();

System.*out*.println("Welcome to Movie Review system :)");

Scanner scanner = new Scanner(System.*in*);

while(true){

int i;

System.*out*.println("Enter index from the below options");

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

System.*out*.println("2. Sign Up");

System.*out*.println("3. Search Movie");

System.*out*.println("4. Search by Genre");

System.*out*.println("5. See top 10 movies");

System.*out*.println("6. Watch top 10 movies of your genre");

System.*out*.println("And any other key to end program");

i = scanner.nextInt();

if (i == 1) {

System.*out*.println("Enter your email:");

String email = scanner.next();

LoginService loginService = new LoginController();

LoginModel user = loginService.login(connection, email);

if(user.getComment() == null){

System.*setProperty*("Username", user.getUsername());

System.*setProperty*("UserID", String.*valueOf*(user.getUser\_id()));

System.*setProperty*("isLogged", "true");

System.*out*.println("Welcome " + user.getUsername());

new LoginFlow().loginOptions(connection);

}

else {

System.*out*.println("Please choose valid options");

}

} else if (i == 2) {

System.*out*.println("Enter your name:");

String name = scanner.next();

System.*out*.println("Enter your email:");

String email = scanner.next();

LoginService loginService = new LoginController();

if(!loginService.registerUser(connection, email,name)){

System.*out*.println("User Registered Successfully");

}

else{

System.*out*.println("Please choose valid option and fill details again");

}

} else if (i == 3) {

generalFlow.searchByMovie();

} else if (i == 4) {

generalFlow.searchByGenre();

} else if (i == 5) {

generalFlow.topTenMovies();

} else if(i == 6) {

generalFlow.topTenGenreMovies();

}

else{

System.*out*.println("Thank you for visiting :)");

break;

}

}

}

}

DBserver.java

package engine;

import services.EngineService;

import java.io.FileReader;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.sql.Connection;

import java.sql.DriverManager;

import java.util.Properties;

class DbServer implements EngineService {

private Connection connection;

private Properties properties;

private void loadProperties(Path path) {

try {

FileReader fileReader = new FileReader(String.*valueOf*(path));

this.properties = new Properties();

this.properties.load(fileReader);

} catch (Exception e) {

throw new RuntimeException(e);

}

}

public Connection run() {

loadProperties(Paths.*get*("/Users/Ashish.Singh/Movie-Review-System/src/db.properties"));

try {

Class.*forName*("com.mysql.jdbc.Driver");

connection = DriverManager.*getConnection*("jdbc:mysql://localhost:"+properties.getProperty("databasePort")+"/" + properties.getProperty("databaseName"), properties.getProperty("user"), properties.getProperty("password"));

return connection;

} catch (Exception E) {

System.*out*.println("Exception occurred " + E);

}

return null;

}

}

MovieController.java

package Controller;

import models.MovieModel;

import models.RatingModel;

import models.ReviewModel;

import services.MovieService;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

public class MovieController implements MovieService {

public void insertMovie(Connection connection, MovieModel movieModel) {

try {

PreparedStatement preparedStatement = connection.prepareStatement("insert into movie (movie\_name, Summary,cast, Genre, Average\_rating, user\_id) values (?,?,?,?,?,?)");

preparedStatement.setString(1, movieModel.getMovie\_name());

preparedStatement.setString(2, movieModel.getSummary());

preparedStatement.setString(3, movieModel.getCast());

preparedStatement.setString(4, movieModel.getGenre());

preparedStatement.setDouble(5, movieModel.getAverage\_rating());

preparedStatement.setInt(6, movieModel.getUser\_id());

preparedStatement.executeUpdate();

System.*out*.println("Movie added Succesfully");

MovieModel movieModel1 = searchMovie(connection, movieModel.getMovie\_name());

RatingModel ratingModel = new RatingModel(movieModel1.getMovie\_id(), movieModel.getAverage\_rating(), Integer.*parseInt*(System.*getProperty*("UserID")));

addRating(connection, ratingModel);

} catch (Exception e) {

System.*out*.println(e);

System.*out*.println("Please try again");

}

}

public void insertReview(Connection connection, ReviewModel reviewModel) {

try {

PreparedStatement preparedStatement = connection.prepareStatement("insert into review ( movie\_id, Review, user\_id) values (?,?,?)");

preparedStatement.setInt(1, reviewModel.getMovie\_id());

preparedStatement.setString(2, reviewModel.getReview());

preparedStatement.setInt(3, reviewModel.getUser\_id());

boolean response = preparedStatement.execute();

System.*out*.println("review added successfully");

} catch (Exception e) {

System.*out*.println(e);

}

}

public void addRating(Connection connection, RatingModel ratingModel) {

PreparedStatement preparedStatement = null;

try {

preparedStatement = connection.prepareStatement("insert into ratings (movie\_id, Rating, user\_id) values (?,?,?)");

preparedStatement.setInt(1, ratingModel.getMovie\_id());

preparedStatement.setDouble(2, ratingModel.getRating());

preparedStatement.setInt(3, ratingModel.getUser\_id());

boolean response = preparedStatement.execute();

//whenever we add rating of movie, we have to calculate avg rating in movie;

PreparedStatement preparedStatement1 = null;

preparedStatement1 = connection.prepareStatement("update movie set Average\_rating=( select avg(rating) from ratings where movie\_id = ?)");

preparedStatement1.setInt(1, ratingModel.getMovie\_id());

boolean response1 = preparedStatement1.execute();

System.*out*.println("Rating added Successfully");

} catch (Exception e) {

System.*out*.println(e);

}

}

public MovieModel findById(Connection connection, int id) {

try {

PreparedStatement preparedStatement = connection.prepareStatement("select movie\_name, Summary,cast, Genre, Average\_rating, user\_id from movie where movie\_id = ?");

preparedStatement.setInt(1, id);

ResultSet resultSet = preparedStatement.executeQuery();

MovieModel movieModel = new MovieModel();

if(resultSet.next()){

movieModel.setMovie\_id(id);

movieModel.setMovie\_name(resultSet.getString(1));

movieModel.setSummary(resultSet.getString(2));

movieModel.setCast(resultSet.getString(3));

movieModel.setGenre(resultSet.getString(4));

movieModel.setAverage\_rating(resultSet.getDouble(5));

movieModel.setUser\_id(resultSet.getInt(6));

return movieModel;

}

else{

return null;

}

} catch (Exception e) {

System.*out*.println(e);

return null;

}

}

public List<MovieModel> findByGenre(Connection connection, String genre, Boolean overall) {

try {

PreparedStatement preparedStatement;

if (overall)

preparedStatement = connection.prepareStatement("select movie\_name, Summary,cast, Genre, Average\_rating, movie\_id from movie where genre = ? order by average\_rating desc limit 10");

else

preparedStatement = connection.prepareStatement("select movie\_name, Summary,cast, Genre, Average\_rating, movie\_id from movie where genre = ? ");

preparedStatement.setString(1, genre);

ResultSet resultSet = preparedStatement.executeQuery();

List<MovieModel> movieList = new ArrayList<>();

while (resultSet.next()) {

MovieModel movieModel = new MovieModel();

movieModel.setMovie\_name(resultSet.getString(1));

movieModel.setSummary(resultSet.getString(2));

movieModel.setCast(resultSet.getString(3));

movieModel.setGenre(resultSet.getString(4));

movieModel.setAverage\_rating(resultSet.getDouble(5));

movieModel.setMovie\_id(resultSet.getInt(6));

movieList.add(movieModel);

}

if (movieList.isEmpty())

System.*out*.println("No Movies available for given genre :(");

return movieList;

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

public List<MovieModel> findMovies(Connection connection, Boolean overall) {

try {

PreparedStatement preparedStatement;

if (overall)

preparedStatement = connection.prepareStatement("select movie\_name, Summary,cast, Genre, Average\_rating, movie\_id from movie order by average\_rating desc limit 10");

else

preparedStatement = connection.prepareStatement("select movie\_name, Summary,cast, Genre, Average\_rating, movie\_id from movie ");

ResultSet resultSet = preparedStatement.executeQuery();

List<MovieModel> movieList = new ArrayList<>();

while (resultSet.next()) {

MovieModel movieModel = new MovieModel();

movieModel.setMovie\_name(resultSet.getString(1));

movieModel.setSummary(resultSet.getString(2));

movieModel.setCast(resultSet.getString(3));

movieModel.setGenre(resultSet.getString(4));

movieModel.setAverage\_rating(resultSet.getDouble(5));

movieModel.setMovie\_id(resultSet.getInt(6));

movieList.add(movieModel);

}

if (movieList.isEmpty())

System.*out*.println("No Movies available for given genre :(");

return movieList;

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

public MovieModel searchMovie(Connection connection, String movieName) {

try {

PreparedStatement preparedStatement = connection.prepareStatement("select movie\_name, Summary,cast, Genre, Average\_rating, movie\_id from movie where movie\_name = ?");

preparedStatement.setString(1, movieName);

ResultSet resultSet = preparedStatement.executeQuery();

if (resultSet.next()) {

MovieModel movieModel = new MovieModel();

movieModel.setMovie\_name(resultSet.getString(1));

movieModel.setSummary(resultSet.getString(2));

movieModel.setCast(resultSet.getString(3));

movieModel.setGenre(resultSet.getString(4));

movieModel.setAverage\_rating(resultSet.getDouble(5));

movieModel.setMovie\_id(resultSet.getInt(6));

movieModel.setStatusCode(200);

return movieModel;

} else {

MovieModel movieModel = new MovieModel();

movieModel.setStatusCode(404);

System.*out*.println("No Movie available for the given name :(");

return null;

}

} catch (Exception e) {

System.*out*.println(e);

}

return null;

}

public List<ReviewModel> findReviews(Connection connection, int movieId){

PreparedStatement preparedStatement = null;

try {

preparedStatement = connection.prepareStatement("select Review from review where movie\_id = ?");

preparedStatement.setInt(1, movieId);

List<ReviewModel> reviewModelList = new ArrayList<>();

ResultSet resultSet = preparedStatement.executeQuery();

while(resultSet.next()){

ReviewModel reviewModel = new ReviewModel();

reviewModel.setReview(resultSet.getString(1));

reviewModelList.add(reviewModel);

}

return reviewModelList;

} catch (Exception e) {

System.*out*.println(e);

}

return null;

}

public void updateMovie(Connection connection, MovieModel movieModel){

PreparedStatement preparedStatement = null;

try {

preparedStatement = connection.prepareStatement("update movie set movie\_name = ?, summary = ?, cast=?, genre=?, user\_id=? where movie\_id = ?");

preparedStatement.setString(1, movieModel.getMovie\_name());

preparedStatement.setString(2, movieModel.getSummary());

preparedStatement.setString(3, movieModel.getCast());

preparedStatement.setString(4, movieModel.getGenre());

preparedStatement.setInt(5, movieModel.getUser\_id());

preparedStatement.setInt(6, movieModel.getMovie\_id());

int response = preparedStatement.executeUpdate();

if(response>=1)

{

System.*out*.println("Movie edited successfully");

}

else{

System.*out*.println("Please try again");

}

}catch (Exception exception){

System.*out*.println(exception);

}

}

}

LoginController.java

package Controller;

import com.mysql.cj.xdevapi.PreparableStatement;

import models.LoginModel;

import services.LoginService;

import java.sql.\*;

public class LoginController implements LoginService {

public boolean registerUser(Connection connection, String email, String name){

try {

PreparedStatement statement = connection.prepareStatement("insert into users (user\_name, email\_id) values (?, ?);");

statement.setString(1, name);

statement.setString(2, email);

return statement.execute();

} catch (Exception e) {

System.*out*.println(e);

}

return true;

}

public LoginModel login(Connection connection, String email){

LoginModel loginModel = new LoginModel();

PreparedStatement statement = null;

if(!email.trim().isEmpty()){

try {

statement = connection.prepareStatement("select user\_id,user\_name,email\_id from users where email\_id=?");

statement.setString(1, email);

ResultSet res =statement.executeQuery();

res.next();

loginModel.setUser\_id(res.getInt(1));

loginModel.setUsername(res.getString(2));

loginModel.setEmail(res.getString(3));

return loginModel;

} catch (Exception e) {

loginModel.setComment("No user present for given email");

return loginModel;

}

}

else{

loginModel.setComment("enter valid email address");

return loginModel;

}

}

}

LoginFlow.java

package Logic;

import Controller.MovieController;

import models.MovieModel;

import models.RatingModel;

import models.ReviewModel;

import services.MovieService;

import java.sql.Connection;

import java.util.List;

import java.util.Scanner;

public class LoginFlow {

private Scanner scanner = new Scanner(System.*in*);

private MovieService movieService = new MovieController();

private GeneralFlow generalFlow;

private Connection connection;

public void loginOptions(Connection connection) {

int value;

this.connection = connection;

generalFlow = new GeneralFlow(connection);

while(System.*getProperty*("isLogged") == "true"){

System.*out*.println("Enter the index of query you want");

System.*out*.println("1. Add Movie");

System.*out*.println("2. Add Review to a movie");

System.*out*.println("3. Rate a movie");

System.*out*.println("4. Edit a movie");

System.*out*.println("5. Search Movie");

System.*out*.println("6. Search by Genre");

System.*out*.println("7. See top 10 movies");

System.*out*.println("8. Watch top 10 movies of your genre");

System.*out*.println("9. Sign Out");

value = scanner.nextInt();

if( value == 1){

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

MovieModel movieModel = new MovieModel() ;

movieModel.setMovie\_name(scanner.next());

movieModel.setUser\_id(Integer.*parseInt*(System.*getProperty*("UserID")));

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

movieModel.setGenre(scanner.next());

System.*out*.println("Enter cast (Comma separated): ");

movieModel.setCast(scanner.next());

System.*out*.println("Enter summary of the movie");

movieModel.setSummary(scanner.next());

System.*out*.println("Enter rating");

Double rating = scanner.nextDouble();

movieModel.setAverage\_rating(rating);

movieService.insertMovie(connection, movieModel);

}

else if (value == 2){

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

String movieName = scanner.next();

MovieModel movieModel = movieService.searchMovie(connection, movieName);

if(movieModel!=null){

System.*out*.println("Enter movie review");

ReviewModel reviewModel = new ReviewModel(Integer.*parseInt*(System.*getProperty*("UserID")), scanner.next(), movieModel.getMovie\_id() );

movieService.insertReview(connection, reviewModel);

}

}

else if( value == 3){

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

String movieName = scanner.next();

MovieModel movieModel = movieService.searchMovie(connection, movieName);

if(movieModel!=null){

System.*out*.println("Enter the rating: ");

Double rating = scanner.nextDouble();

movieService.addRating(connection, new RatingModel(movieModel.getMovie\_id(), rating, Integer.*parseInt*(System.*getProperty*("UserID"))));

}

} else if (value == 4) {

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

String movieName = scanner.next();

MovieModel movieModel = movieService.searchMovie(connection, movieName);

if(movieModel!=null){

System.*out*.println("Enter updated movie name");

String var = scanner.next();

movieModel.setMovie\_name(var.trim().isEmpty() ? movieModel.getMovie\_name() : var);

System.*out*.println("Enter Summary");

String var1 = scanner.next();

movieModel.setSummary(var.trim().isEmpty() ? movieModel.getSummary() : var1);

System.*out*.println("Enter cast");

String var2 = scanner.next();

movieModel.setCast(var.trim().isEmpty() ? movieModel.getCast() : var2);

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

String var3 = scanner.next();

movieModel.setGenre(var.trim().isEmpty() ? movieModel.getGenre() : var3);

movieModel.setUser\_id(Integer.*parseInt*(System.*getProperty*("UserID")));

movieService.updateMovie(connection, movieModel);

}

} else if (value == 5) {

generalFlow.searchByMovie();

} else if (value == 6) {

generalFlow.searchByGenre();

} else if (value == 7) {

generalFlow.topTenMovies();

} else if ( value == 8) {

generalFlow.topTenGenreMovies();

}

else {

System.*clearProperty*("Username");

System.*clearProperty*("UserID");

System.*setProperty*("isLogged", "false");

System.*out*.println("Logged Out");

break;

}

}

}

}

GeneralFlow.java

package Logic;

import Controller.MovieController;

import models.MovieModel;

import models.ReviewModel;

import services.MovieService;

import javax.sound.midi.Soundbank;

import java.sql.Connection;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class GeneralFlow {

public GeneralFlow(Connection connection) {

this.connection = connection;

}

Scanner scanner = new Scanner(System.*in*);

private MovieService movieService = new MovieController();

private Connection connection;

public void searchByGenre(){

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

String genre = scanner.next();

List<MovieModel> movieList = movieService.findByGenre(connection, genre, false);

for(int i=0;i<movieList.size();i++){

System.*out*.println(i + ". " + movieList.get(i).getMovie\_name());

}

System.*out*.println("Enter index of movie you want details of: ");

int index = scanner.nextInt();

showDetails(movieList.get(index));

}

public void searchByMovie(){

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

String movieName = scanner.next();

MovieModel movieModel = movieService.searchMovie(connection, movieName);

if(movieModel != null)

showDetails(movieModel);

}

public void topTenGenreMovies() {

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

String genre = scanner.next();

List<MovieModel> movieList = movieService.findByGenre(connection, genre, true);

for (int i = 0; i < movieList.size(); i++) {

System.*out*.println(i + " " + movieList.get(i).getMovie\_name());

}

System.*out*.println("Enter index of movie you want details of: ");

int index = scanner.nextInt();

showDetails(movieList.get(index));

}

public void topTenMovies(){

List<MovieModel> movieList = movieService.findMovies(connection, true);

for(int i=0;i<movieList.size();i++){

System.*out*.println(i + " " + movieList.get(i).getMovie\_name());

}

System.*out*.println("Enter index of movie you want details of: ");

int index = scanner.nextInt();

showDetails(movieList.get(index));

}

public void showDetails(MovieModel movieModel){

System.*out*.println("Movie Name : " + movieModel.getMovie\_name() + "\n"+

"Cast : " + movieModel.getCast() + "\n" +

"Summary : " + movieModel.getSummary() + "\n"+

"Genre : " + movieModel.getGenre() + "\n" +

"Rating : " + movieModel.getAverage\_rating());

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

list = movieService.findReviews(connection, movieModel.getMovie\_id());

String r = "";

if(!list.isEmpty() && list!=null){

for(int i=0;i<list.size();i++){

r += list.get(i).getReview() + " ";

}

System.*out*.println("Reviews : " + r);

}

}

}

LoginModel.java

package models;

public class LoginModel {

public LoginModel() {

}

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 getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public LoginModel(int user\_id, String username, String email) {

this.user\_id = user\_id;

this.username = username;

this.email = email;

}

@Override

public String toString() {

return "LoginModel{" +

"user\_id=" + user\_id +

", username='" + username + '\'' +

", email='" + email + '\'' +

'}';

}

private int user\_id;

private String username;

private String email;

private String Comment = null;

public String getComment() {

return Comment;

}

public void setComment(String comment) {

Comment = comment;

}

}

MovieModel.java

package models;

public class MovieModel {

@Override

public String toString() {

return "MovieModel{" +

"movie\_id=" + movie\_id +

", movie\_name='" + movie\_name + '\'' +

", Summary='" + Summary + '\'' +

", cast='" + cast + '\'' +

", Genre='" + Genre + '\'' +

", Average\_rating=" + Average\_rating +

", user\_id=" + user\_id +

'}';

}

public MovieModel() {

}

public int getMovie\_id() {

return movie\_id;

}

public void setMovie\_id(int movie\_id) {

this.movie\_id = movie\_id;

}

public String getMovie\_name() {

return movie\_name;

}

public void setMovie\_name(String movie\_name) {

this.movie\_name = movie\_name;

}

public String getSummary() {

return Summary;

}

public void setSummary(String summary) {

Summary = summary;

}

public String getCast() {

return cast;

}

public void setCast(String cast) {

this.cast = cast;

}

public String getGenre() {

return Genre;

}

public void setGenre(String genre) {

Genre = genre;

}

public double getAverage\_rating() {

return Average\_rating;

}

public MovieModel(String movie\_name, String summary, String cast, String genre, double average\_rating, int user\_id) {

this.movie\_name = movie\_name;

Summary = summary;

this.cast = cast;

Genre = genre;

Average\_rating = average\_rating;

this.user\_id = user\_id;

}

public void setAverage\_rating(double average\_rating) {

Average\_rating = average\_rating;

}

public int getUser\_id() {

return user\_id;

}

public void setUser\_id(int user\_id) {

this.user\_id = user\_id;

}

private int movie\_id;

private String movie\_name ;

private String Summary ;

private String cast ;

private String Genre ;

private double Average\_rating ;

private int user\_id ;

public int getStatusCode() {

return statusCode;

}

public void setStatusCode(int statusCode) {

this.statusCode = statusCode;

}

public String getStatusMessage() {

return statusMessage;

}

public void setStatusMessage(String statusMessage) {

this.statusMessage = statusMessage;

}

public String getComment() {

return Comment;

}

public void setComment(String comment) {

Comment = comment;

}

private int statusCode;

private String statusMessage;

private String Comment;

}

RatingModel.java

package models;

public class RatingModel {

public RatingModel(int movie\_id,Double rating, int user\_id) {

this.movie\_id = movie\_id;

this.rating = rating;

this.user\_id = user\_id;

}

public RatingModel() {

}

public int getMovie\_id() {

return movie\_id;

}

public void setMovie\_id(int movie\_id) {

this.movie\_id = movie\_id;

}

public Double getRating() {

return rating;

}

public void setRating(Double rating) {

this.rating = rating;

}

public int getUser\_id() {

return user\_id;

}

public void setUser\_id(int user\_id) {

this.user\_id = user\_id;

}

public int getRating\_id() {

return rating\_id;

}

public void setRating\_id(int rating\_id) {

this.rating\_id = rating\_id;

}

int movie\_id;

Double rating;

int user\_id;

int rating\_id;

}

ReviewModel.java

package models;

public class ReviewModel {

public ReviewModel() {

}

public int getUser\_id() {

return user\_id;

}

public void setUser\_id(int user\_id) {

this.user\_id = user\_id;

}

public int getReview\_id() {

return review\_id;

}

public void setReview\_id(int review\_id) {

this.review\_id = review\_id;

}

public String getReview() {

return review;

}

public void setReview(String review) {

this.review = review;

}

public int getMovie\_id() {

return movie\_id;

}

public void setMovie\_id(int movie\_id) {

this.movie\_id = movie\_id;

}

public ReviewModel(int user\_id, String review, int movie\_id) {

this.user\_id = user\_id;

this.review = review;

this.movie\_id = movie\_id;

}

int user\_id;

int review\_id;

String review;

int movie\_id;

}

EngineService.java

package services;

import java.sql.Connection;

public interface EngineService {

public Connection run();

}

LoginService.java

package services;

import models.LoginModel;

import java.sql.Connection;

public interface LoginService {

public boolean registerUser(Connection connection, String email, String name);

public LoginModel login(Connection connection, String email);

}

MovieService.java

package services;

import models.MovieModel;

import models.RatingModel;

import models.ReviewModel;

import java.sql.Connection;

import java.util.List;

public interface MovieService {

public void insertMovie(Connection connection, MovieModel movieModel);

public void addRating(Connection connection, RatingModel ratingModel);

public void insertReview(Connection connection, ReviewModel reviewModel);

public MovieModel searchMovie(Connection connection, String movieName);

public List<MovieModel> findByGenre(Connection connection, String genre, Boolean overall);

public List<ReviewModel> findReviews(Connection connection, int movieId);

public List<MovieModel> findMovies(Connection connection, Boolean overall) ;

public void updateMovie(Connection connection, MovieModel movieModel);

}

db.properties

user=root

password=password

databasePort= 3306

databaseName = MovieReview