Enterprise Application Development (EAD)

Object Relational Persistence

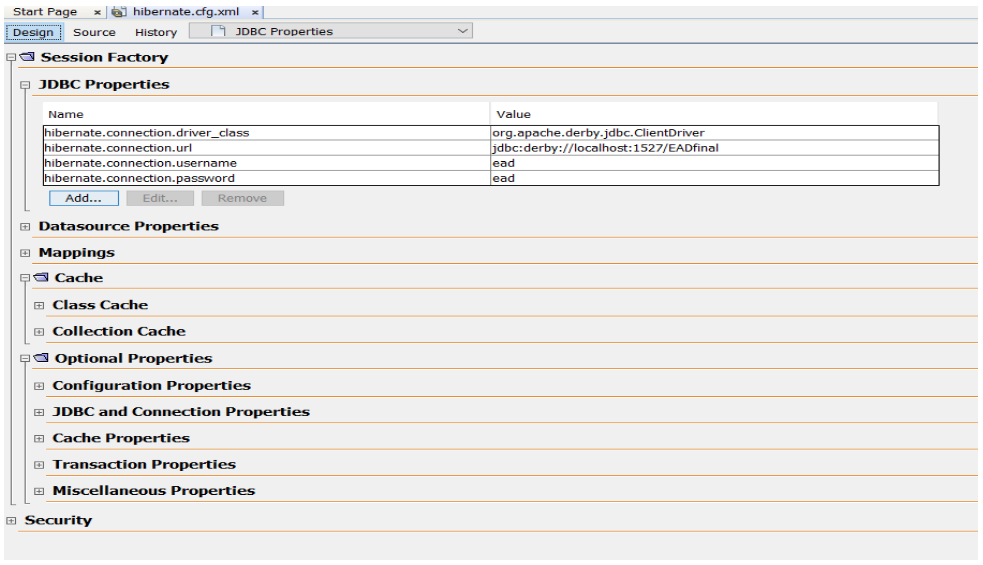
Introduction

GMDB(Global Movie Database) is a popular website in people. GMDB management decided to invest a new java application. I created a java web application with hibernate as the Object Relational Mapping technology. All the requirements works in this java web application.

Use java server pages to this web application. Netbeans IDE use to develop the application

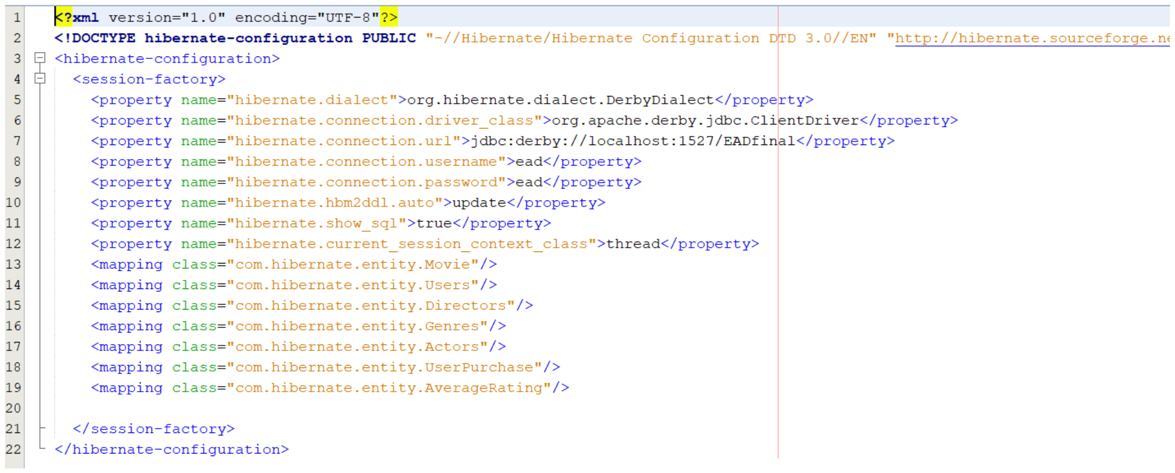
Description of Code

- create hibernate.cfg.xml and adding properties.



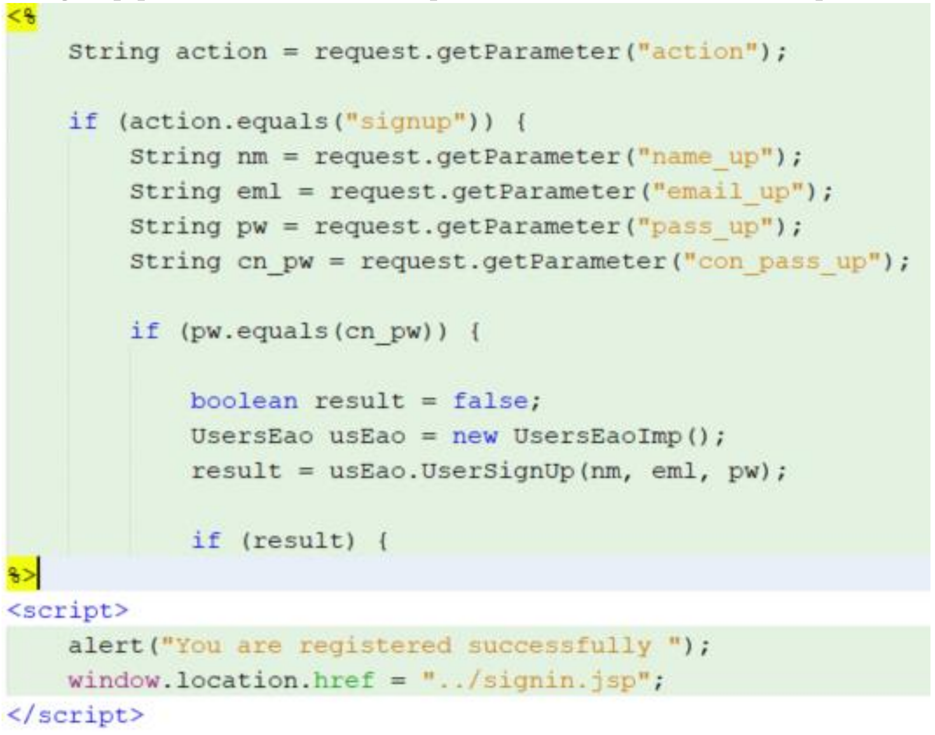
Name	Value
hibernate.connection.driver_class	org.apache.derby.jdbc.ClientDriver
hibernate.connection.url	jdbc:derby://localhost:1527/EADfinal
hibernate.connection.username	ead
hibernate.connection.password	ead

Created database url, username and password adding to this code. This is the code of hibernate.cfg.xml. use update in hibernate.hbm2ddl.auto property. Using create then the start the project always drop the tables and create tables again. It is not suitable to this application. adding entity java files name adding to this mapping. Create all the entity adding to this. Not adding the error showing in output glassfish server console.



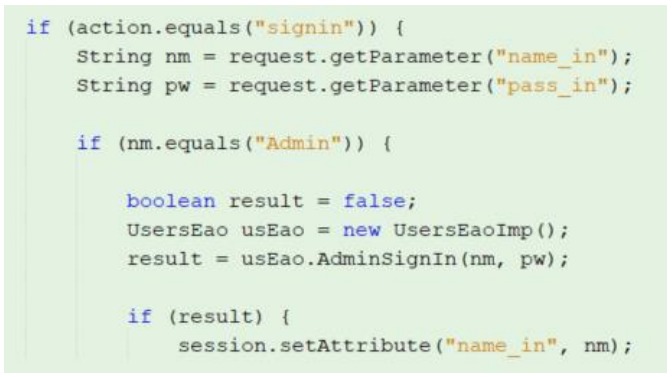
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN" "http://hibernate.sourceforge.ne
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.dialect">org.hibernate.dialect.DerbyDialect</property>
    <property name="hibernate.connection.driver_class">org.apache.derby.jdbc.ClientDriver</property>
    <property name="hibernate.connection.url">jdbc:derby://localhost:1527/EADfinal</property>
    <property name="hibernate.connection.username">ead</property>
    <property name="hibernate.connection.password">ead</property>
    <property name="hibernate.hbm2ddl.auto">update</property>
    <property name="hibernate.show_sql">true</property>
    <property name="hibernate.current_session_context_class">thread</property>
    <mapping class="com.hibernate.entity.Movie"/>
    <mapping class="com.hibernate.entity.Users"/>
    <mapping class="com.hibernate.entity.Directors"/>
    <mapping class="com.hibernate.entity.Genres"/>
    <mapping class="com.hibernate.entity.Actors"/>
    <mapping class="com.hibernate.entity.UserPurchase"/>
    <mapping class="com.hibernate.entity.AverageRating"/>

  </session-factory>
</hibernate-configuration>
```

- In sign up password and confirm password checked and values passed to table.



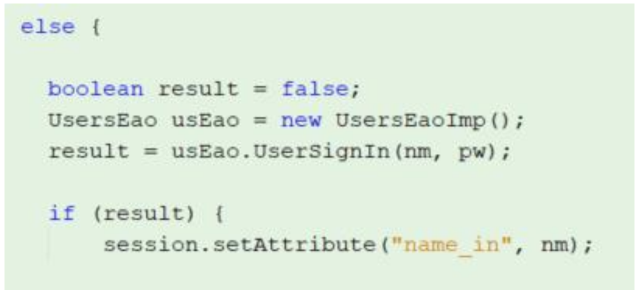
```
<%
    String action = request.getParameter("action");

    if (action.equals("signup")) {
        String nm = request.getParameter("name_up");
        String eml = request.getParameter("email_up");
        String pw = request.getParameter("pass_up");
        String cn_pw = request.getParameter("con_pass_up");

        if (pw.equals(cn_pw)) {

            boolean result = false;
            UsersEao usEao = new UsersEaoImp();
            result = usEao.UserSignUp(nm, eml, pw);

            if (result) {
%>
<script>
    alert("You are registered successfully ");
    window.location.href = "../signin.jsp";
</script>
```

- User is name as Admin and password include in signUp form. In signIn form check "Admin" and admin can go to adminHome.page. Use AdminSignIn method to validate admin password in users table.



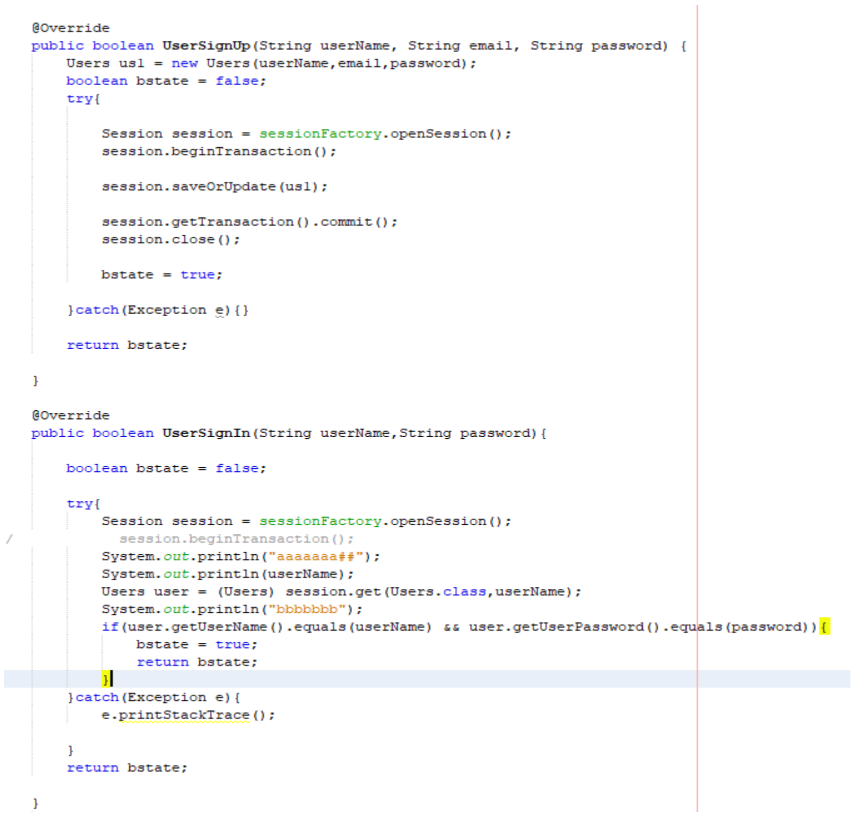
```
if (action.equals("signin")) {
    String nm = request.getParameter("name_in");
    String pw = request.getParameter("pass_in");

    if (nm.equals("Admin")) {

        boolean result = false;
        UsersEao usEao = new UsersEaoImp();
        result = usEao.AdminSignIn(nm, pw);

        if (result) {
            session.setAttribute("name_in", nm);
```

- . Users use UserSignIn method to validate the user.



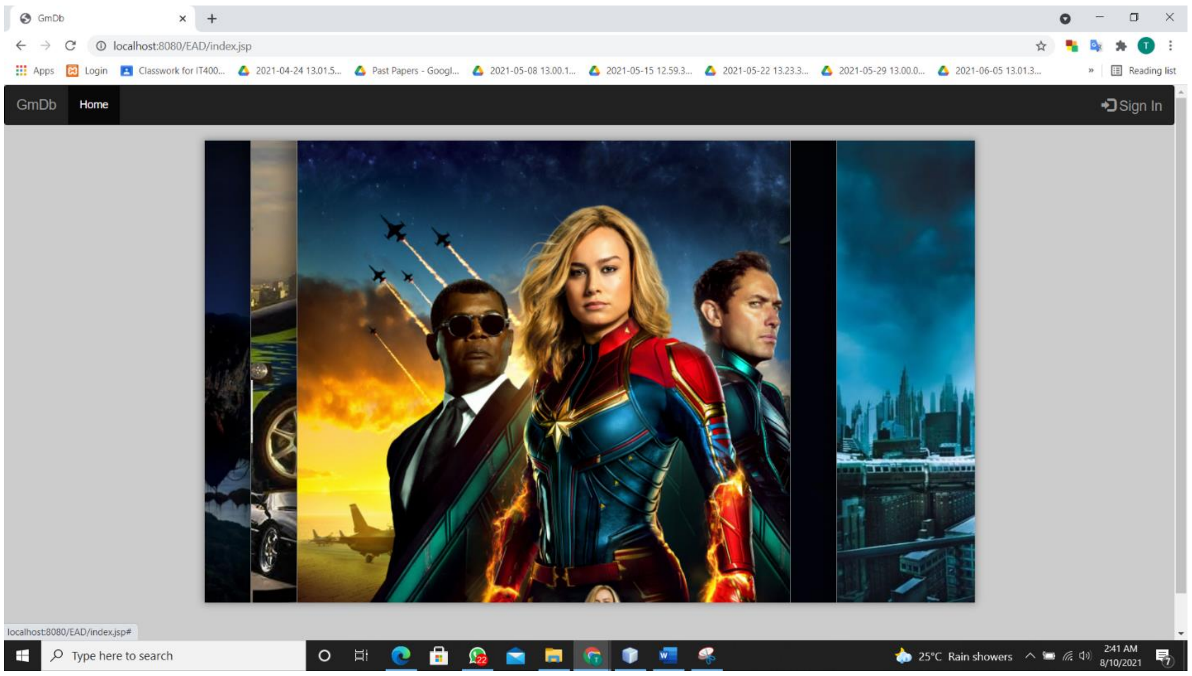
```
else {

  boolean result = false;
  UsersEao usEao = new UsersEaoImp();
  result = usEao.UserSignIn(nm, pw);

  if (result) {
      session.setAttribute("name_in", nm);
```

- Implement java file of UserEao(UserEaoImp.java)

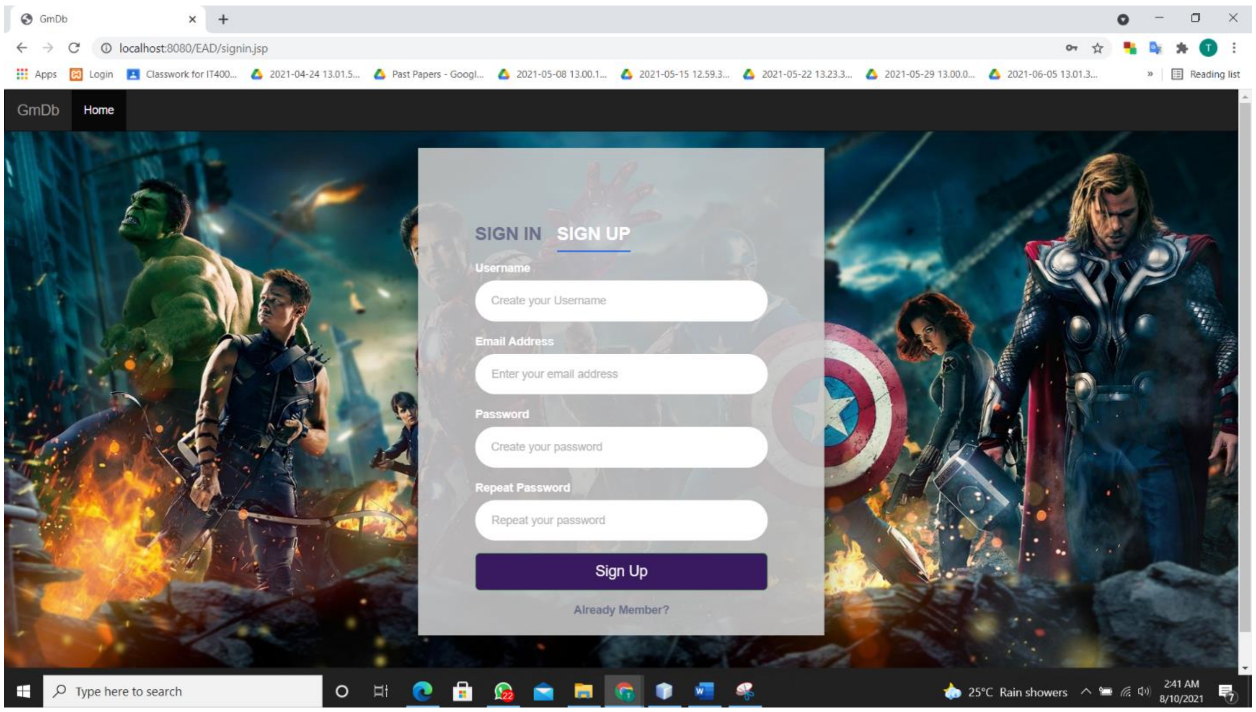


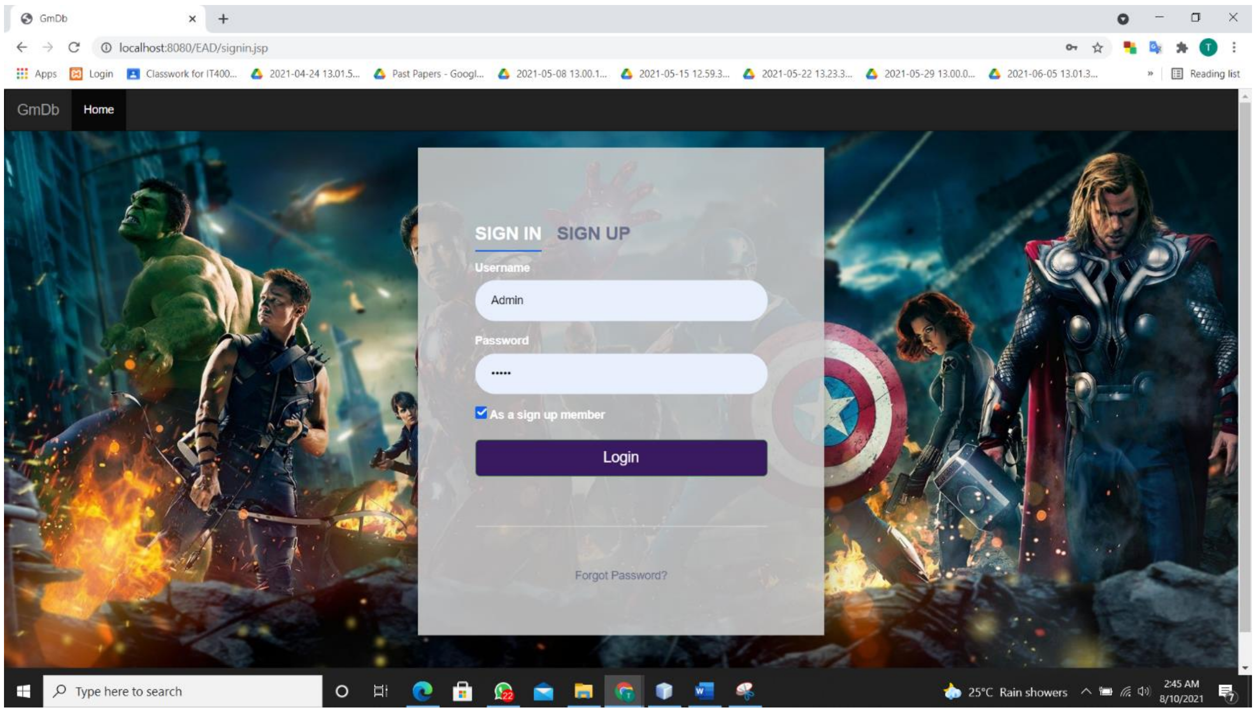
```
@Override
public boolean UserSignUp(String userName, String email, String password) {
    Users usl = new Users(userName,email,password);
    boolean bstate = false;
    try{

        Session session = sessionFactory.openSession();
        session.beginTransaction();

        session.saveOrUpdate(usl);

        session.getTransaction().commit();
        session.close();

        bstate = true;

    }catch(Exception e){}

    return bstate;

}

@Override
public boolean UserSignIn(String userName,String password){

    boolean bstate = false;

    try{
        Session session = sessionFactory.openSession();
/         session.beginTransaction();
        System.out.println("aaaaaaa##");
        System.out.println(userName);
        Users user = (Users) session.get(Users.class,userName);
        System.out.println("bbbbbbb");
        if(user.getUserName().equals(userName) && user.getUserPassword().equals(password)){
            bstate = true;
            return bstate;
        }
    }catch(Exception e){
        e.printStackTrace();

    }
    return bstate;

}
```

Project homepage. Click Sign In button and go to signin.jsp page.



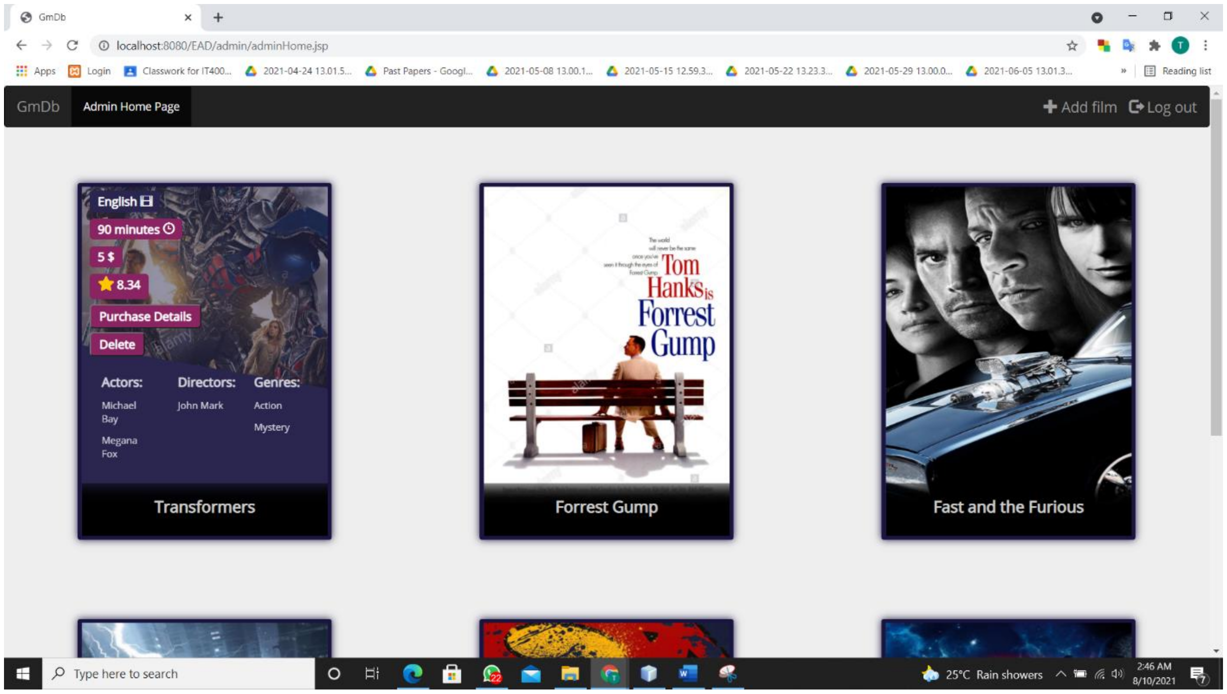
SignUp and SignIn UI





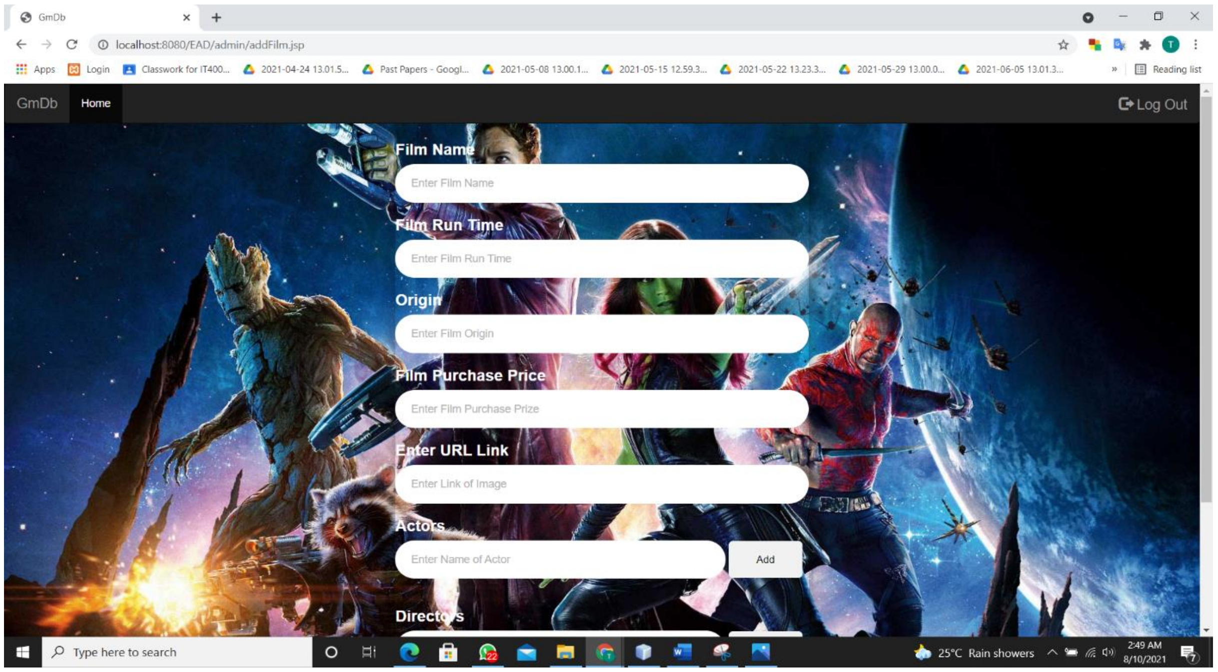
Admin Home page

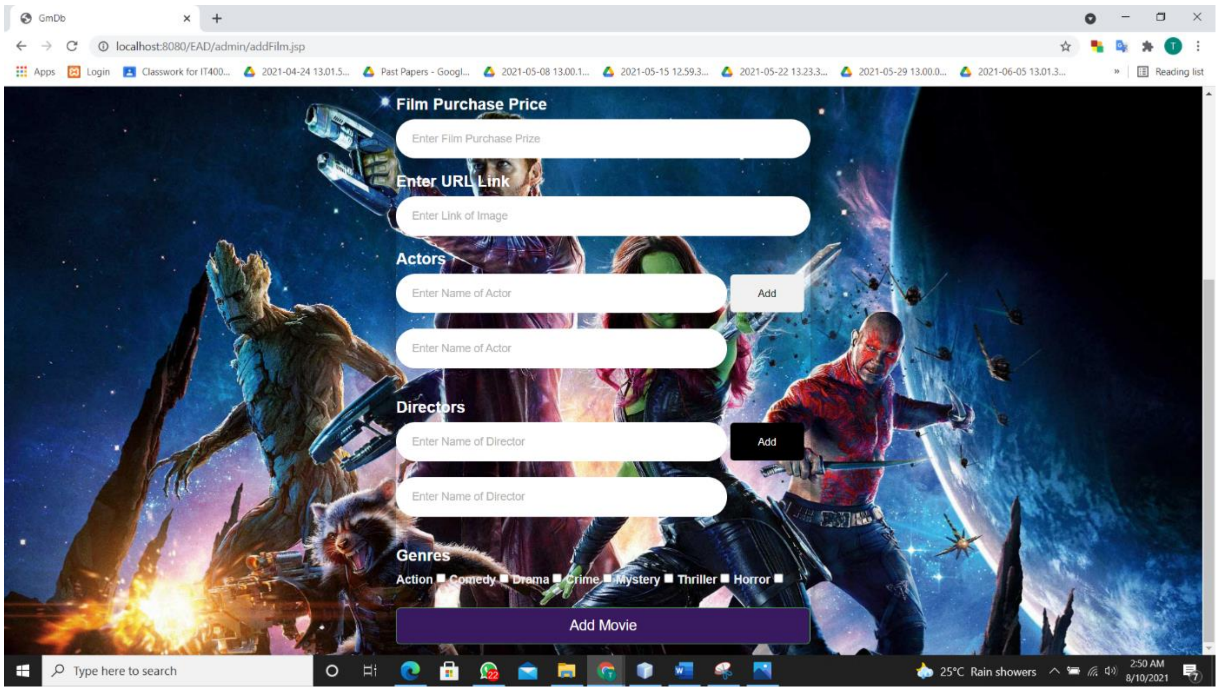
- In adminHome page, +add Film in navbar used to add new film. show the movies included in movie cards. Animation added to that Ui card.



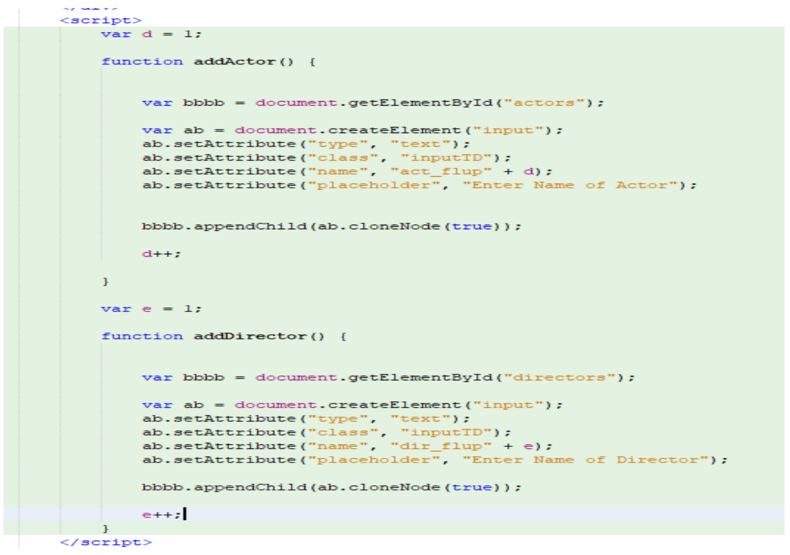
Add Film Page

- Click +Add film then go to this form. In this form click Add button infront of Actors, directors. Then add other text area. We could add actors and directors.





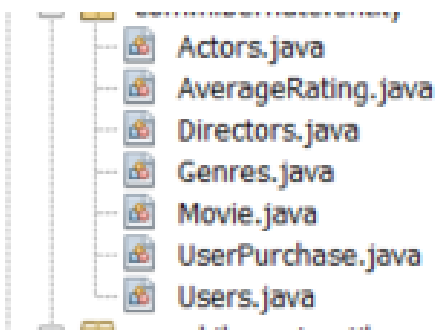
- Using this Script to create new text area. The created new text area has **unique** name. **It is important to add data to table.**



```
<script>
    var d = 1;

    function addActor() {


        var bbbb = document.getElementById("actors");

        var ab = document.createElement("input");
        ab.setAttribute("type", "text");
        ab.setAttribute("class", "inputTD");
        ab.setAttribute("name", "act_flup" + d);
        ab.setAttribute("placeholder", "Enter Name of Actor");


        bbbb.appendChild(ab.cloneNode(true));

        d++;

    }

    var e = 1;

    function addDirector() {


        var bbbb = document.getElementById("directors");

        var ab = document.createElement("input");
        ab.setAttribute("type", "text");
        ab.setAttribute("class", "inputTD");
        ab.setAttribute("name", "dir_flup" + e);
        ab.setAttribute("placeholder", "Enter Name of Director");

        bbbb.appendChild(ab.cloneNode(true));

        e++;
    }
</script>
```

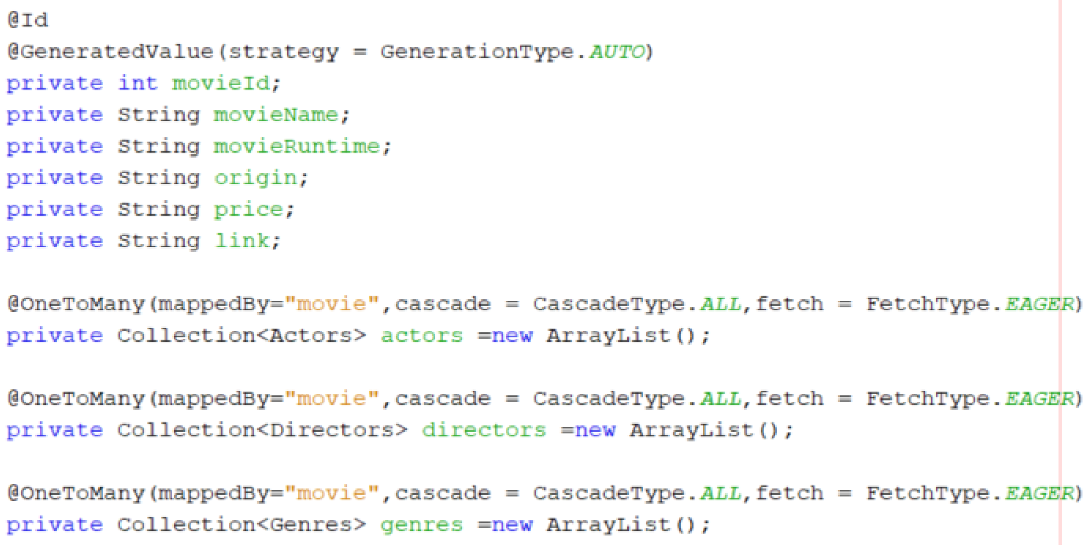
Table Create with annotation

- Create 3 tables for directors, actors and genres. They have multiple values in one movie. So use that java files to create database tables. Using movieId as a **foreign key** in Actors, Directors, Genres table.



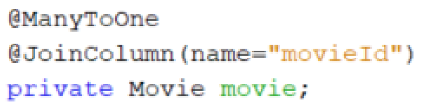
Actors.java
AverageRating.java
Directors.java
Genres.java
Movie.java
UserPurchase.java
Users.java

- In movie.java file using autogenerate movieId. It is primary key. OneToMany annotation for actors, directors, and genres. Collection is a data type using to put data in arraylist.
- Actors, directors, and genres tables are mappedBy movie. Using cascade and fetch.



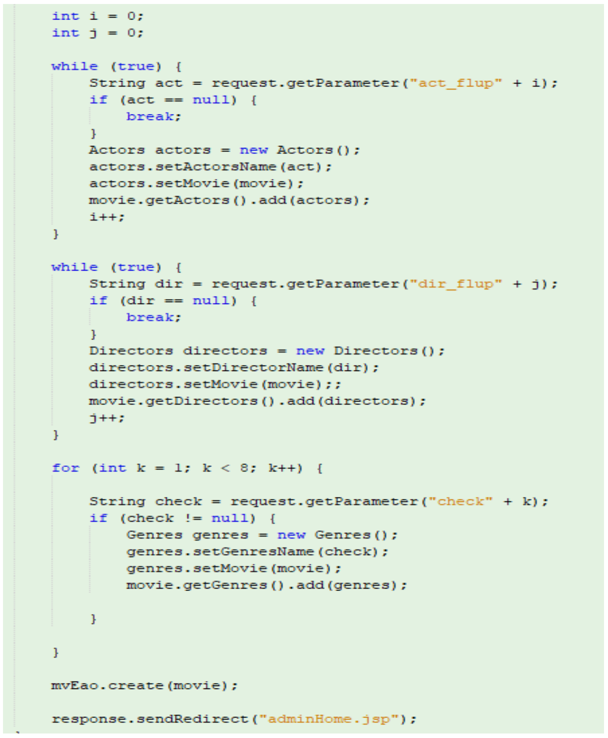
```
@Id
@GeneratedValue(strategy = GenerationType.AUTO)
private int movieId;
private String movieName;
private String movieRuntime;
private String origin;
private String price;
private String link;

@OneToMany(mappedBy="movie",cascade = CascadeType.ALL,fetch = FetchType.EAGER)
private Collection<Actors> actors =new ArrayList();

@OneToMany(mappedBy="movie",cascade = CascadeType.ALL,fetch = FetchType.EAGER)
private Collection<Directors> directors =new ArrayList();

@OneToMany(mappedBy="movie",cascade = CascadeType.ALL,fetch = FetchType.EAGER)
private Collection<Genres> genres =new ArrayList();
```

- Using ManyToOne annotation in Genres, Directors and Actors tables.

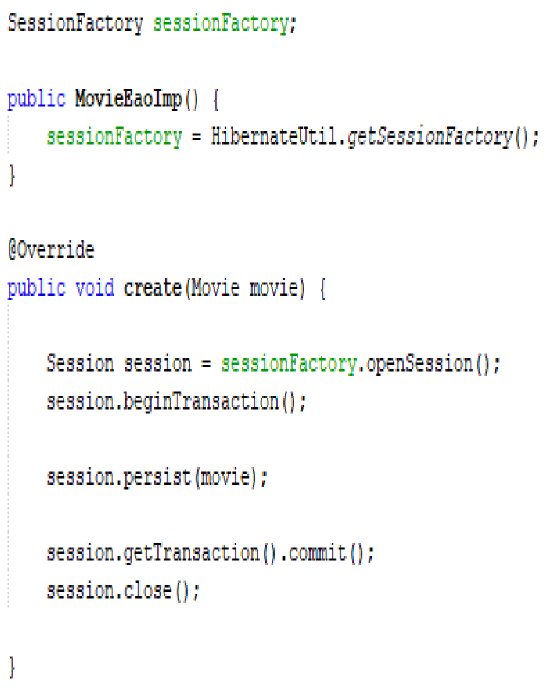


```
@ManyToOne
@JoinColumn(name="movieId")
private Movie movie;
```

- Add actors, directors, genres to movie. Ex – **movie.getActors().add(actors)** . adding details include to **movie** table.



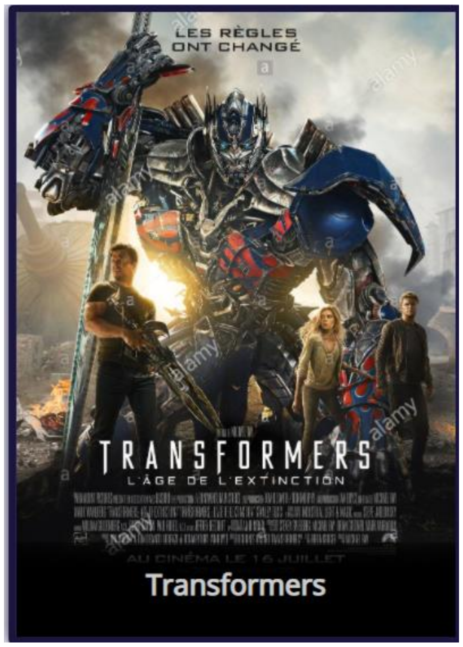
```
int i = 0;
int j = 0;

while (true) {
    String act = request.getParameter("act_flup" + i);
    if (act == null) {
        break;
    }
    Actors actors = new Actors();
    actors.setActorsName(act);
    actors.setMovie(movie);
    movie.getActors().add(actors);
    i++;
}

while (true) {
    String dir = request.getParameter("dir_flup" + j);
    if (dir == null) {
        break;
    }
    Directors directors = new Directors();
    directors.setDirectorName(dir);
    directors.setMovie(movie);;
    movie.getDirectors().add(directors);
    j++;
}

for (int k = 1; k < 8; k++) {

    String check = request.getParameter("check" + k);
    if (check != null) {
        Genres genres = new Genres();
        genres.setGenresName(check);
        genres.setMovie(movie);
        movie.getGenres().add(genres);

    }

}

mvEao.create(movie);

response.sendRedirect("adminHome.jsp");
```

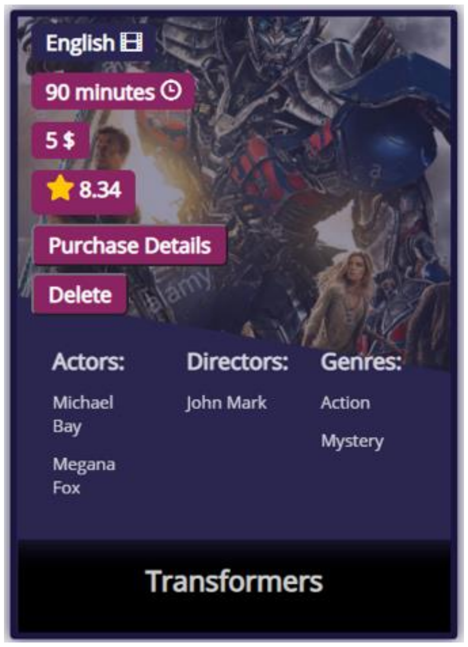


```
SessionFactory sessionFactory;

public MovieEaoImp() {
    sessionFactory = HibernateUtil.getSessionFactory();
}

@Override
public void create(Movie movie) {

    Session session = sessionFactory.openSession();
    session.beginTransaction();

    session.persist(movie);

    session.getTransaction().commit();
    session.close();

}
```

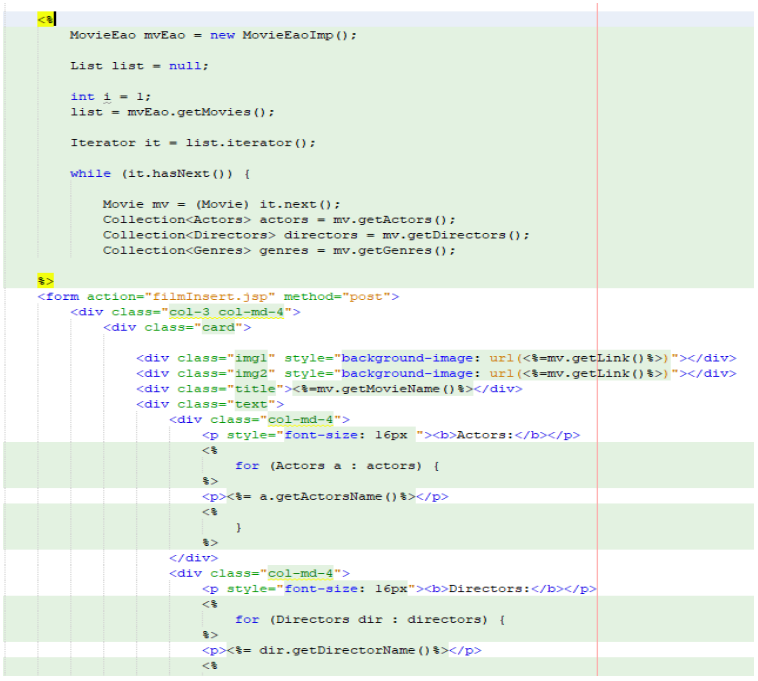
Movie Card UI in Admin Home page

- Use while function to repeat the card. Film Name, run time, price, rating average, Origin, Actors, directors and genres including in card. Animation added to the cards. The mouse put on the card then the animation works. In adminHome page screenshot include above.



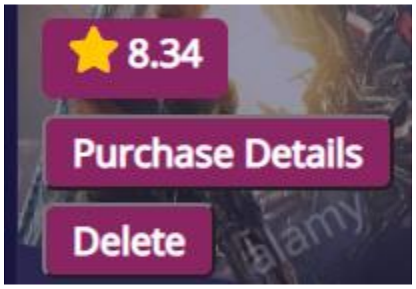


- Get movie details from getMovies() method. The method is a List type. Use Iterator and the all list data include to **it** named variable. it.hasNext() using to checked Are the data including in list. In whlile loop data include to **mv**. All the actors, directors and genres values are include in **mv** variable. All the values get from like mv.getMovieName()



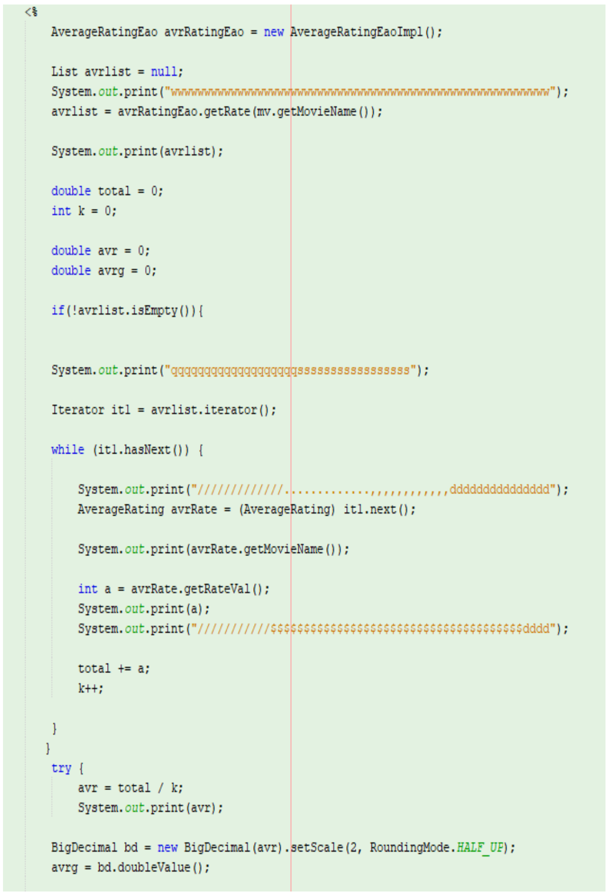
```
<%
    MovieEao mvEao = new MovieEaoImp();

    List list = null;

    int i = 1;
    list = mvEao.getMovies();

    Iterator it = list.iterator();

    while (it.hasNext()) {

        Movie mv = (Movie) it.next();
        Collection<Actors> actors = mv.getActors();
        Collection<Directors> directors = mv.getDirectors();
        Collection<Genres> genres = mv.getGenres();

%>
<form action="filmInsert.jsp" method="post">
    <div class="col-3 col-md-4">
        <div class="card">

            <div class="img1" style="background-image: url(<%=mv.getLink()%>)"></div>
            <div class="img2" style="background-image: url(<%=mv.getLink()%>)"></div>
            <div class="title"><%=mv.getMovieName()%></div>
            <div class="text">
                <div class="col-md-4">
                    <p style="font-size: 16px "><b>Actors:</b></p>
                    <%
                        for (Actors a : actors) {
                    %>
                    <p><%= a.getActorsName()%></p>
                    <%
                        }
                    %>
                </div>
                <div class="col-md-4">
                    <p style="font-size: 16px"><b>Directors:</b></p>
                    <%
                        for (Directors dir : directors) {
                    %>
                    <p><%= dir.getDirectorName()%></p>
                    <%
```

Average Rating, Purchase Details and Delete in Movie Card

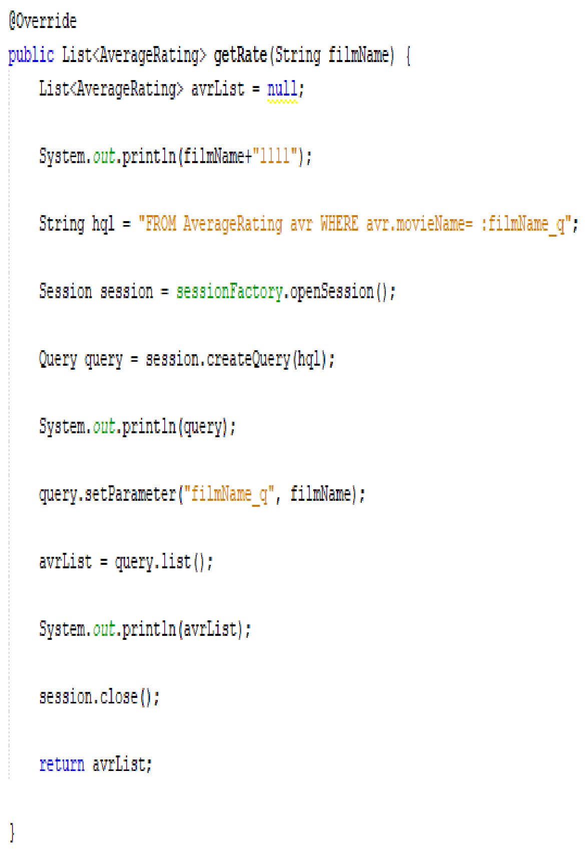


- ### Average Rating

Firstly check the avrlist is not Empty. Because that case use to in first time running program there haven't data in tables. So error in the code. The error solve using this function. If avrlist is not empty then the go inside the code. Use **getRateVal()** methode to get users including rate values of suitable movie. Use setScale(2,RoundingMode.HALF_UP) for round the decimal value to two decimal values.

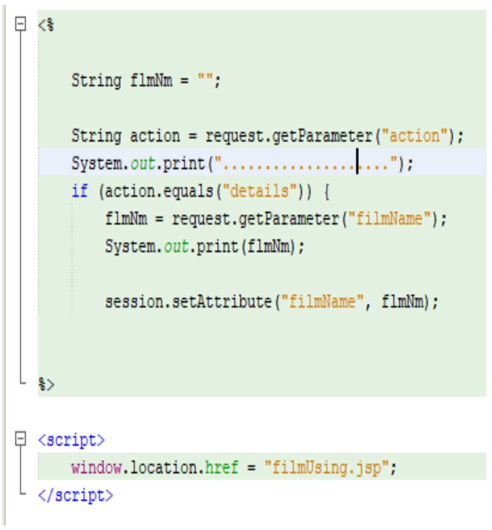


```
<%
    AverageRatingEao avrRatingEao = new AverageRatingEaoImpl();

    List avrlist = null;
    System.out.print("wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww");
    avrlist = avrRatingEao.getRate(mv.getMovieName());

    System.out.print(avrlist);

    double total = 0;
    int k = 0;

    double avr = 0;
    double avrg = 0;

    if(!avrlist.isEmpty()){


    System.out.print("qqqqqqqqqqqqqqqqqqqssssssssssssssss");

    Iterator itl = avrlist.iterator();

    while (itl.hasNext()) {

        System.out.print("/////////////.............,,,,,,,,,,,ddddddddddddddd");
        AverageRating avrRate = (AverageRating) itl.next();

        System.out.print(avrRate.getMovieName());

        int a = avrRate.getRateVal();
        System.out.print(a);
        System.out.print("///////////$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$dddd");

        total += a;
        k++;

    }
   }
    try {
        avr = total / k;
        System.out.print(avr);

    BigDecimal bd = new BigDecimal(avr).setScale(2, RoundingMode.HALF_UP);
    avrg = bd.doubleValue();
```

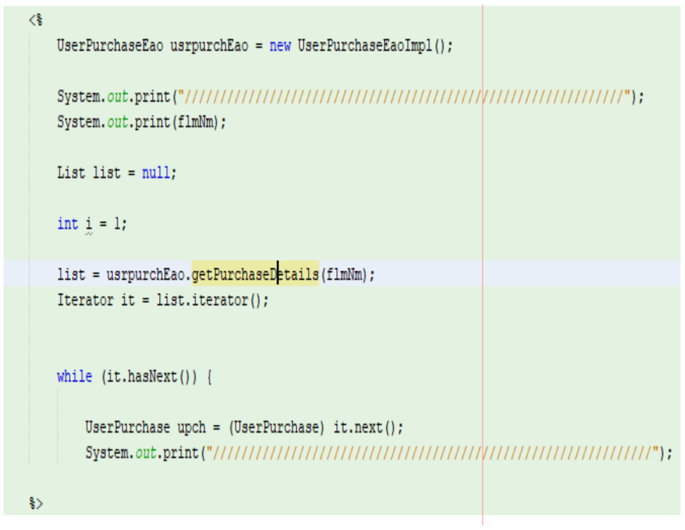


```
@Override
public List<AverageRating> getRate(String filmName) {
    List<AverageRating> avrList = null;

    System.out.println(filmName+"1111");

    String hql = "FROM AverageRating avr WHERE avr.movieName= :filmName_q";

    Session session = sessionFactory.openSession();

    Query query = session.createQuery(hql);

    System.out.println(query);

    query.setParameter("filmName_q", filmName);

    avrList = query.list();

    System.out.println(avrList);

    session.close();

    return avrList;

}
```

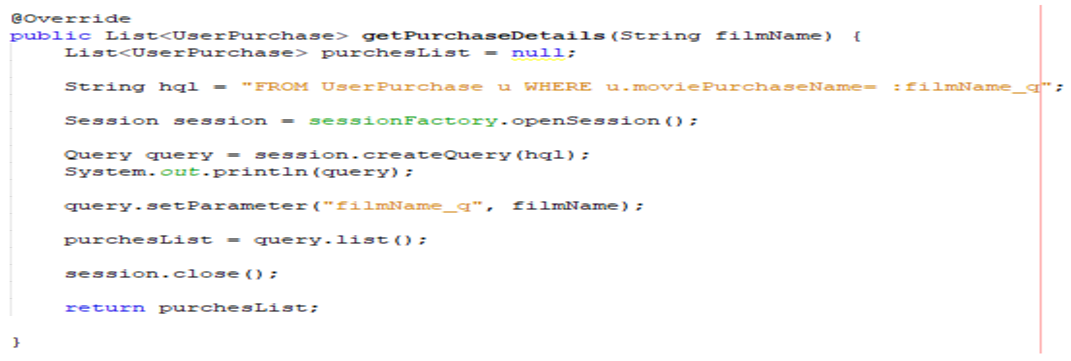
- **Purchase Details**
  This is a button to get the purchase use details.



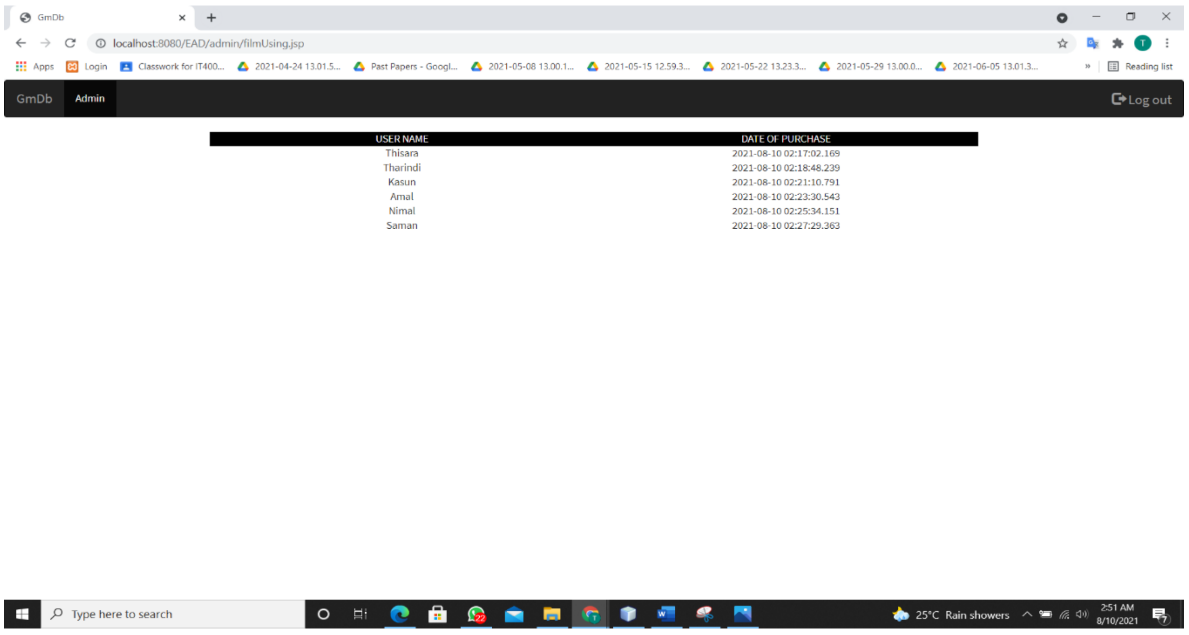
```
<%

    String flmNm = "";

    String action = request.getParameter("action");
    System.out.print("....................");
    if (action.equals("details")) {
        flmNm = request.getParameter("filmName");
        System.out.print(flmNm);

        session.setAttribute("filmName", flmNm);

%>

<script>
    window.location.href = "filmUsing.jsp";
</script>
```



```
<%
    UserPurchaseEao usrpurchEao = new UserPurchaseEaoImpl();

    System.out.print("//////////////////////////////////////////////////////////////");
    System.out.print(flmNm);

    List list = null;

    int i = 1;

    list = usrpurchEao.getPurchaseDetails(flmNm);
    Iterator it = list.iterator();


    while (it.hasNext()) {

        UserPurchase upch = (UserPurchase) it.next();
        System.out.print("//////////////////////////////////////////////////////////////");

%>
```



```
@Override
public List<UserPurchase> getPurchaseDetails(String filmName) {
    List<UserPurchase> purchesList = null;

    String hql = "FROM UserPurchase u WHERE u.moviePurchaseName= :filmName_q";

    Session session = sessionFactory.openSession();

    Query query = session.createQuery(hql);
    System.out.println(query);

    query.setParameter("filmName_q", filmName);

    purchesList = query.list();

    session.close();

    return purchesList;

}
```

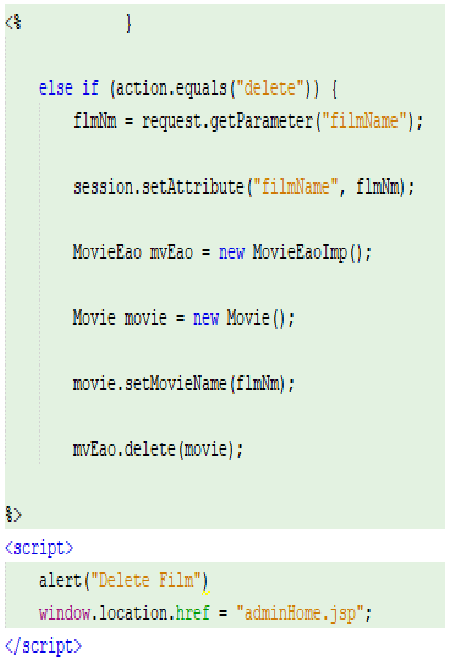


GmDb Admin Log out

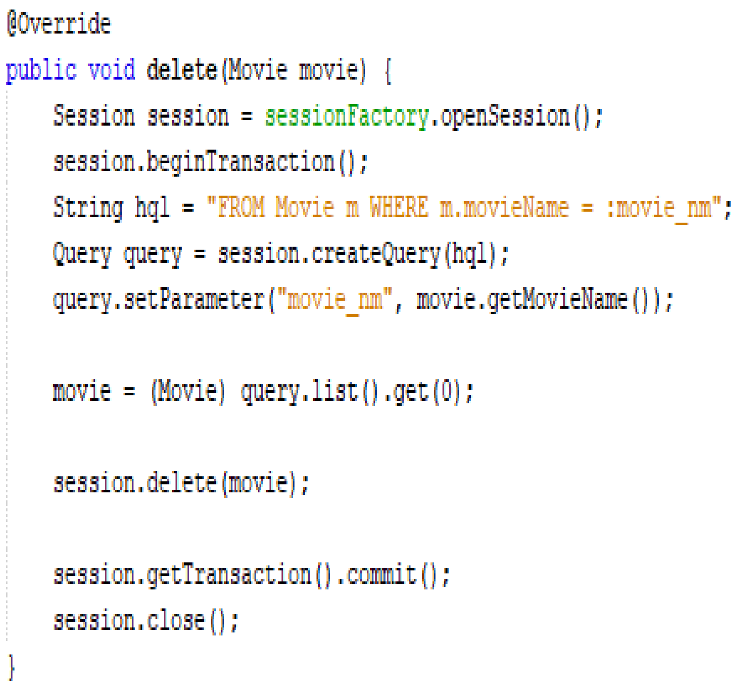
localhost:8080/EAD/admin/filmUsing.jsp

USER NAME	DATE OF PURCHASE
Thisara	2021-08-10 02:17:02.169
Tharindi	2021-08-10 02:18:48.239
Kasun	2021-08-10 02:21:10.791
Amal	2021-08-10 02:23:30.543
Nimal	2021-08-10 02:25:34.151
Saman	2021-08-10 02:27:29.363

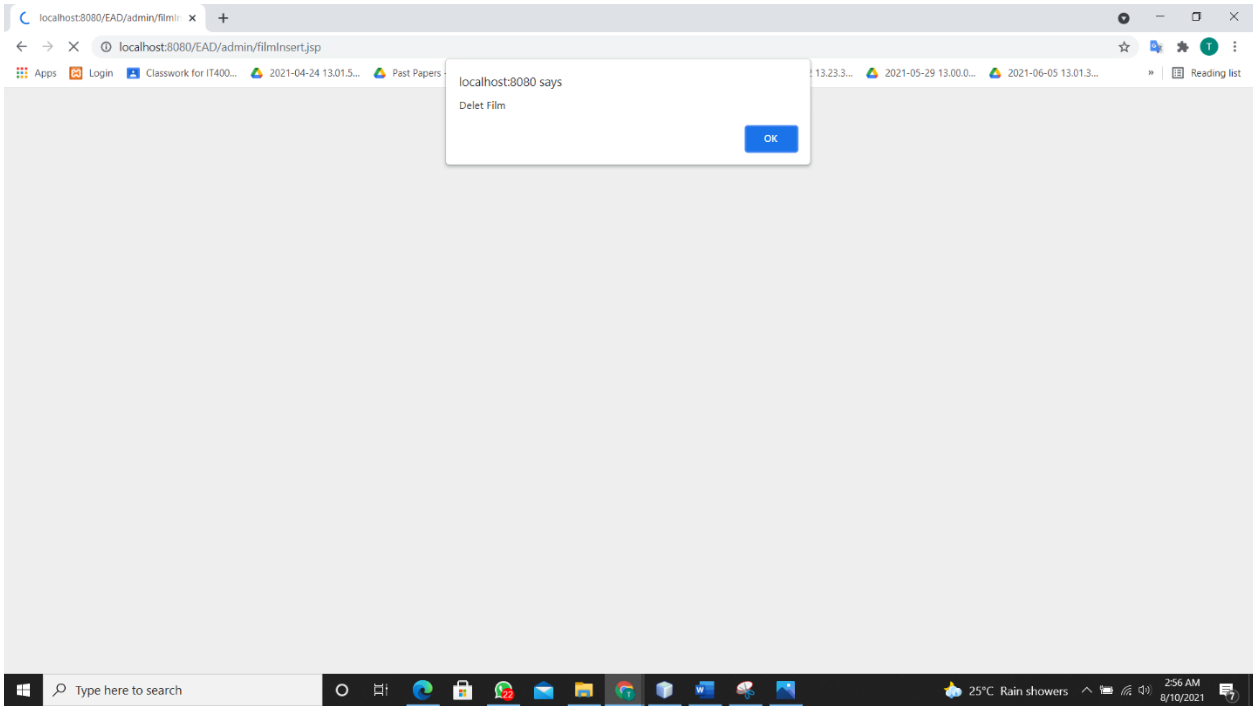
- **Delete Button**
  Click the button then the delete the include film.



```
<%          }

    else if (action.equals("delete")) {
        flmNm = request.getParameter("filmName");

        session.setAttribute("filmName", flmNm);

        MovieEao mvEao = new MovieEaoImp();

        Movie movie = new Movie();

        movie.setMovieName(flmNm);

        mvEao.delete(movie);

%>
<script>
    alert("Delete Film")
    window.location.href = "adminHome.jsp";
</script>
```



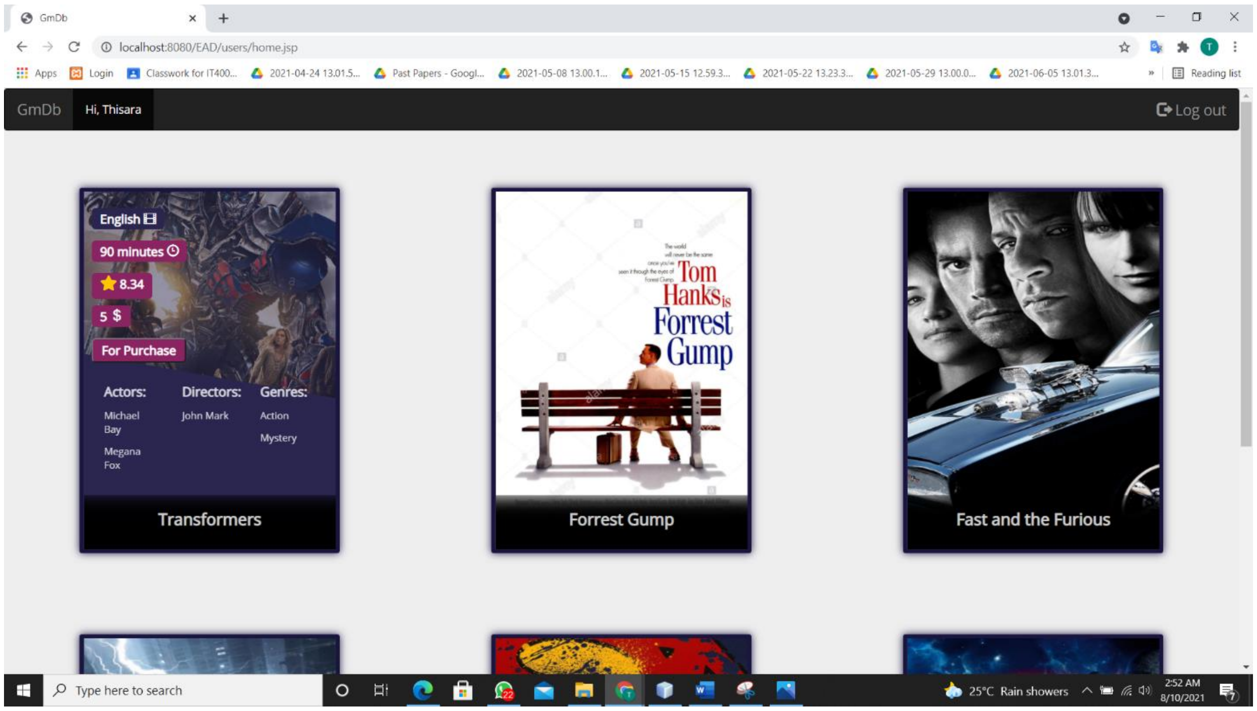
```
@Override
public void delete(Movie movie) {
    Session session = sessionFactory.openSession();
    session.beginTransaction();
    String hql = "FROM Movie m WHERE m.movieName = :movie_nm";
    Query query = session.createQuery(hql);
    query.setParameter("movie_nm", movie.getMovieName());

    movie = (Movie) query.list().get(0);

    session.delete(movie);

    session.getTransaction().commit();
    session.close();
}
```



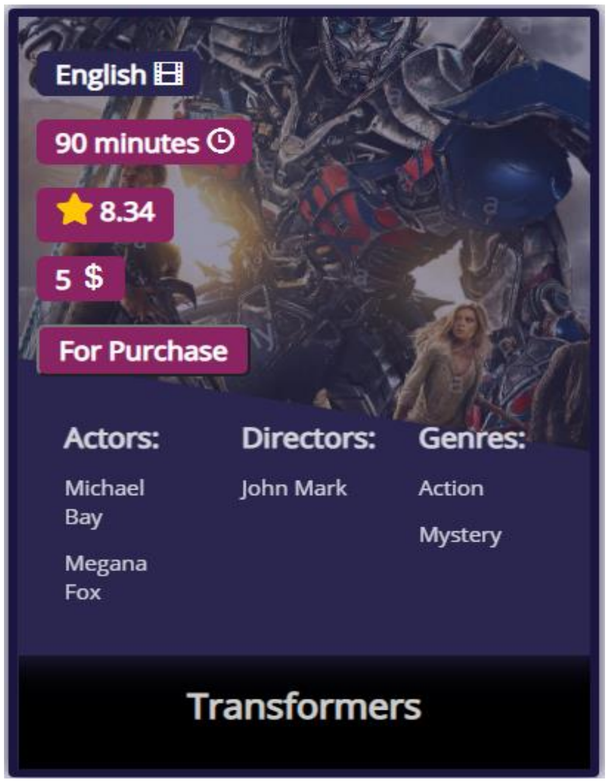
localhost:8080 says

Delet Film

User Home Page UI

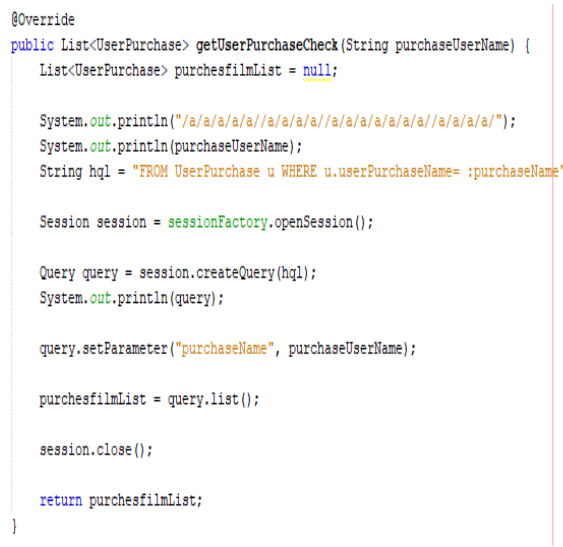


Movie Card in Users home page



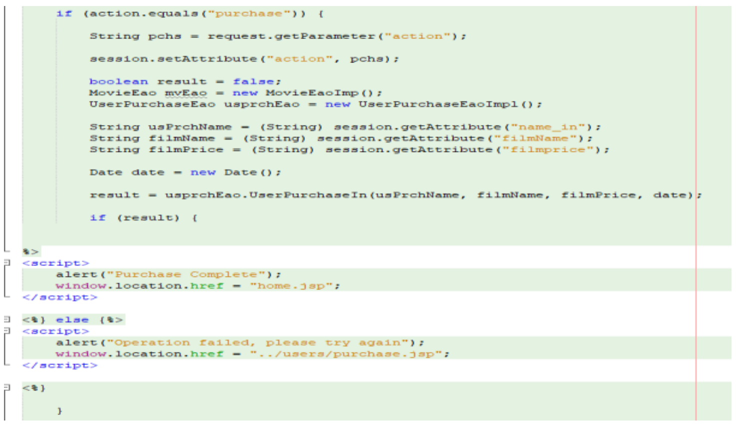
Click For Purchase Button and go to purchase.jsp page. Not purchase the movie couldn't rate the movie.

- Check the movie purchased (validate purchase)

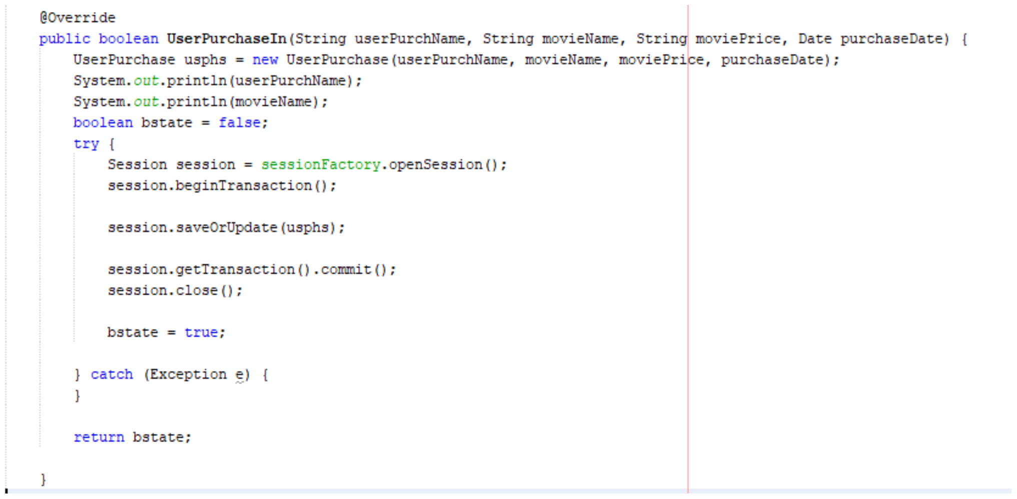


```java
@Override
public List<UserPurchase> getUserPurchaseCheck(String purchaseUserName) {
    List<UserPurchase> purchesfilmList = null;

    System.out.println("/a/a/a/a/a//a/a/a/a//a/a/a/a/a/a/a//a/a/a/a/");
    System.out.println(purchaseUserName);
    String hql = "FROM UserPurchase u WHERE u.userPurchaseName= :purchaseName";

    Session session = sessionFactory.openSession();

    Query query = session.createQuery(hql);
    System.out.println(query);

    query.setParameter("purchaseName", purchaseUserName);

    purchesfilmList = query.list();

    session.close();

    return purchesfilmList;
}
```

- Purchase the movie

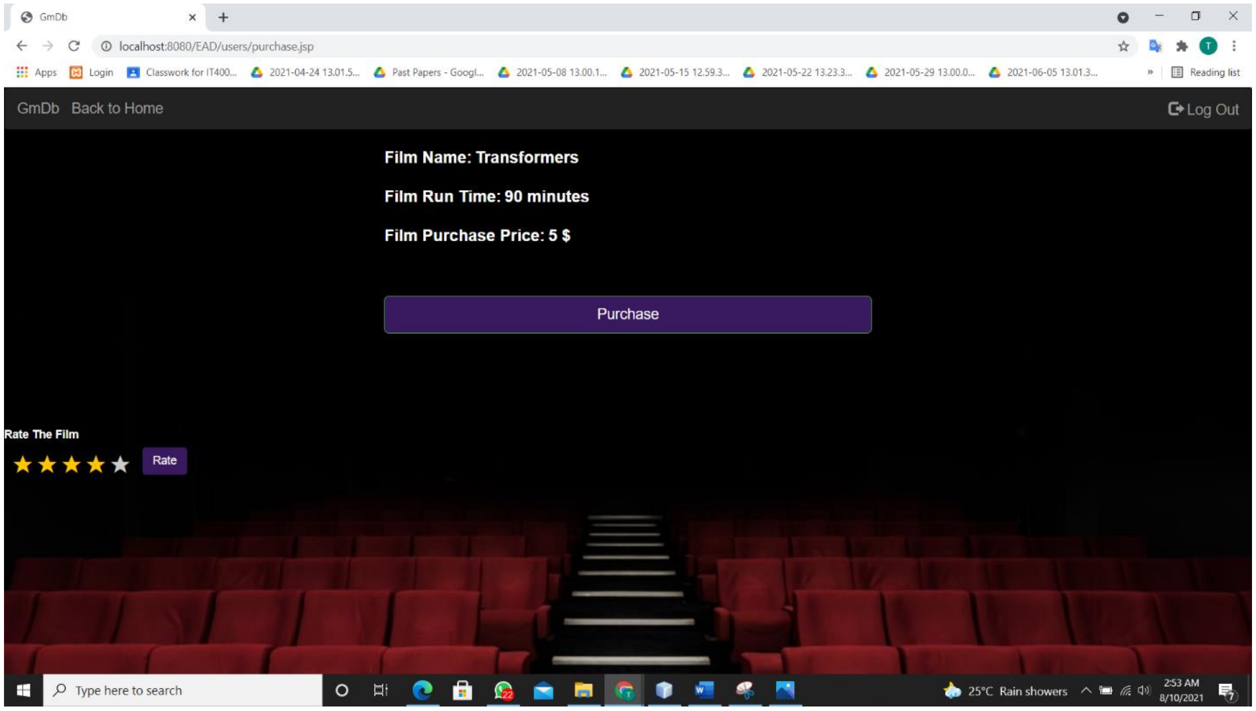


```jsp
    if (action.equals("purchase")) {

        String pchs = request.getParameter("action");

        session.setAttribute("action", pchs);

        boolean result = false;
        MovieEao mvEao = new MovieEaoImp();
        UserPurchaseEao usprchEao = new UserPurchaseEaoImpl();

        String usPrchName = (String) session.getAttribute("name_in");
        String filmName = (String) session.getAttribute("filmName");
        String filmPrice = (String) session.getAttribute("filmprice");

        Date date = new Date();

        result = usprchEao.UserPurchaseIn(usPrchName, filmName, filmPrice, date);

        if (result) {

%>
<script>
    alert("Purchase Complete");
    window.location.href = "home.jsp";
</script>

<%} else {%>
<script>
    alert("Operation failed, please try again");
    window.location.href = "../users/purchase.jsp";
</script>

<%}

    }
```



```java
@Override
public boolean UserPurchaseIn(String userPurchName, String movieName, String moviePrice, Date purchaseDate) {
    UserPurchase usphs = new UserPurchase(userPurchName, movieName, moviePrice, purchaseDate);
    System.out.println(userPurchName);
    System.out.println(movieName);
    boolean bstate = false;
    try {
        Session session = sessionFactory.openSession();
        session.beginTransaction();

        session.saveOrUpdate(usphs);

        session.getTransaction().commit();
        session.close();

        bstate = true;

    } catch (Exception e) {
    }

    return bstate;

}
```

Purchase Page UI



Film Name: Transformers

Film Run Time: 90 minutes

Film Purchase Price: 5 $