**ONLINE MOVIE BOOKING**

**USING JAVA, JSP AND JDBC**

**TEAM MEMBERS:**

|  |
| --- |
| **SRIMATHI R(2018506125)**  **SANDHIYA R(2018506106)**  **JEYALAKSHMI S (2018506041)** |

**Aim:**

To implement a movie booking system using java jsp and jdbc concepts

**Modules involved:**

1.user

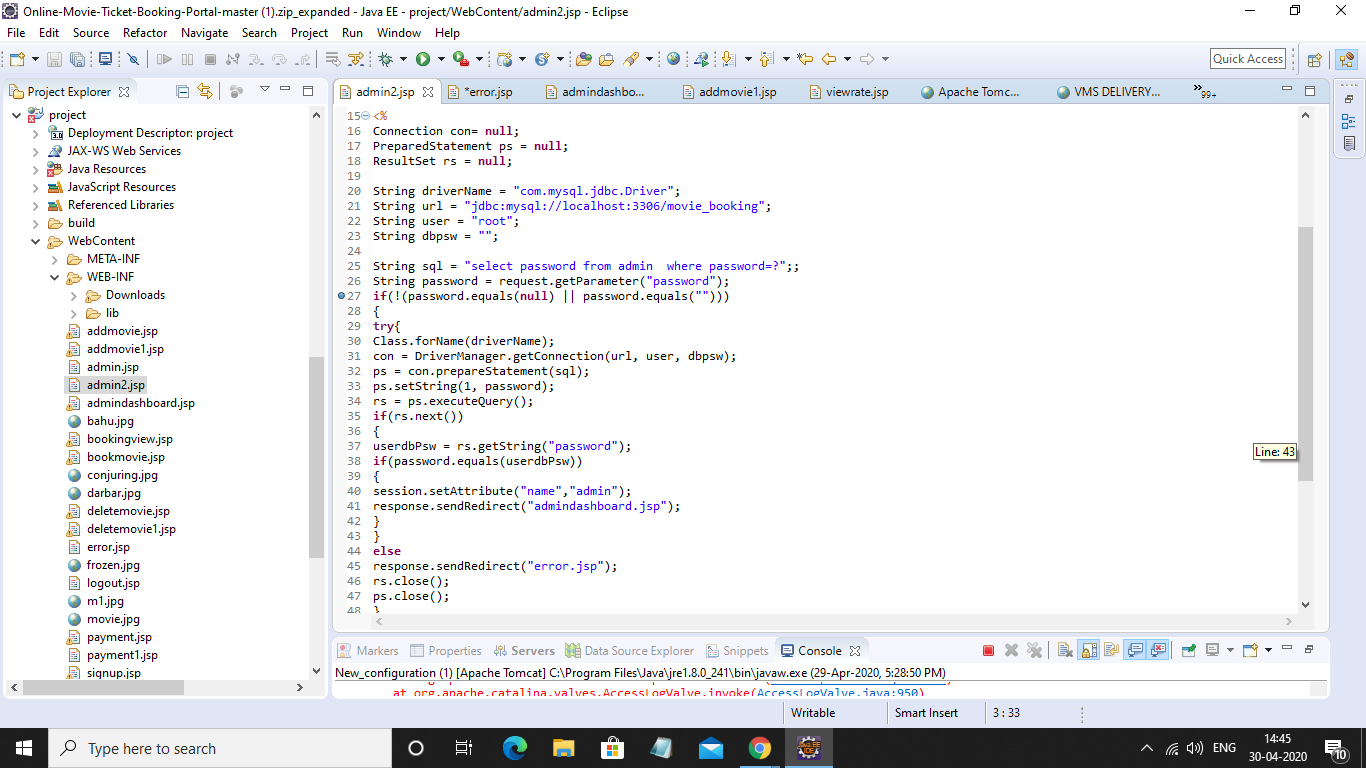
2.admin

**Concepts used:**

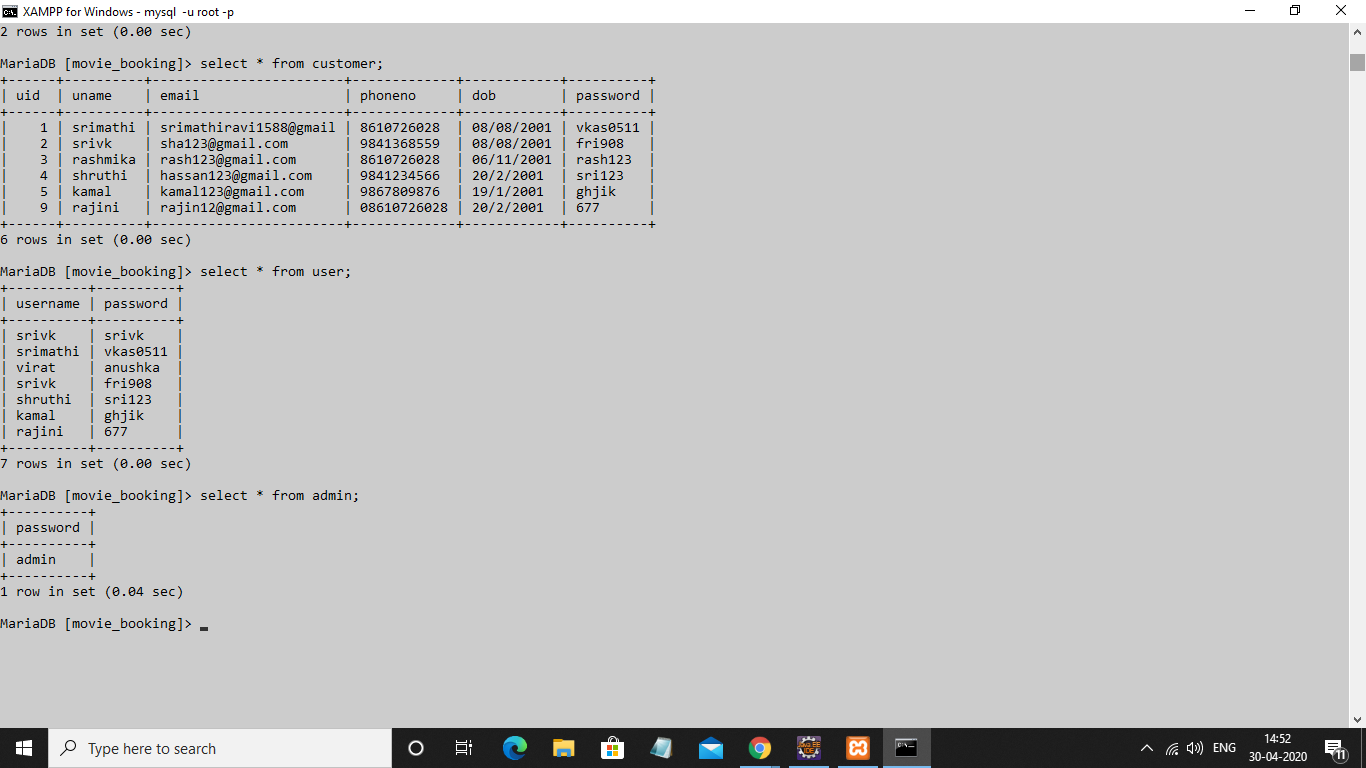
**1.JDBC connection:**

Java database connectivity with MySQL

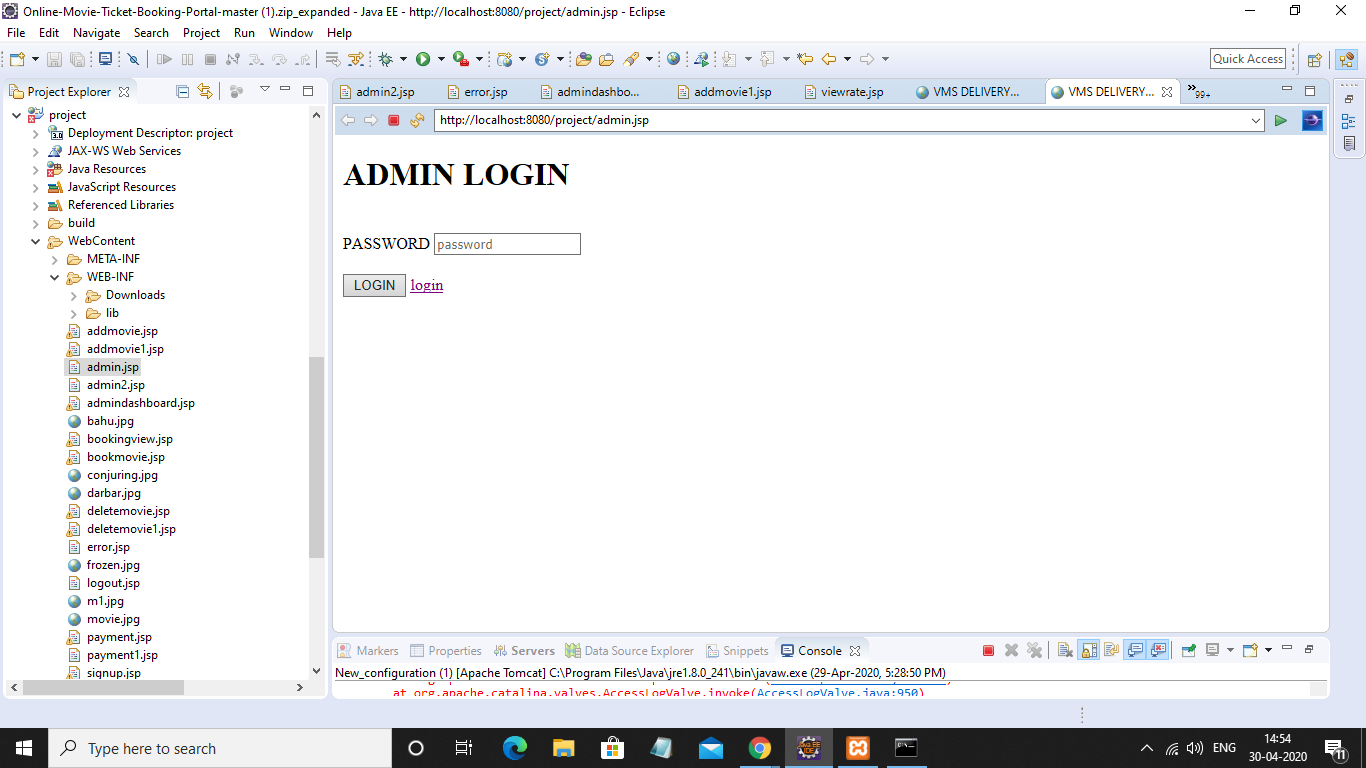
In the database **movie\_booking.**

****

* The username,password of admin,various users are stored.
* Details of the movies running in the theatre.
* Details of the customers
* Booking details



**Test case**:

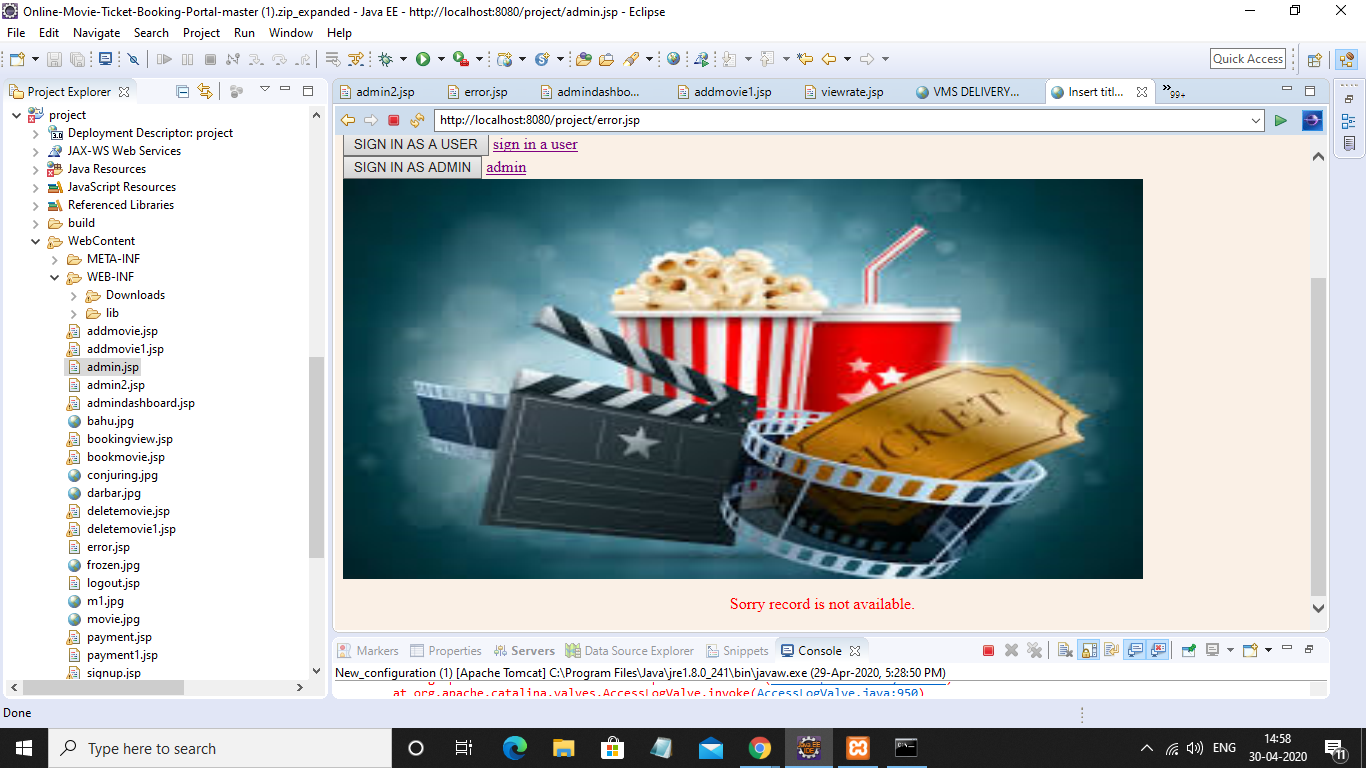


* Password of admin stored in database is admin.

When this form is submitted admin2.jsp works

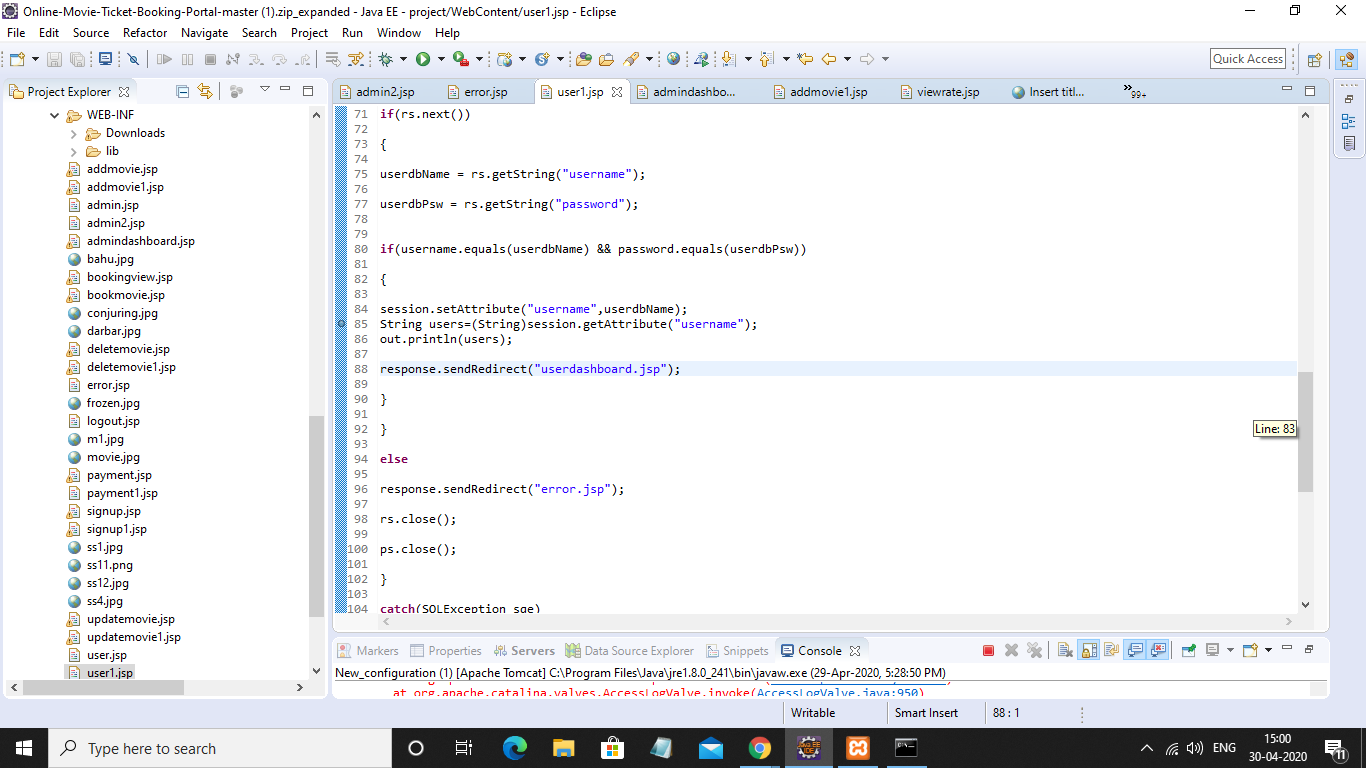
There the password from the form is checked against existing value in the database

If it does not match,redirected to home page



**2.Sessions**

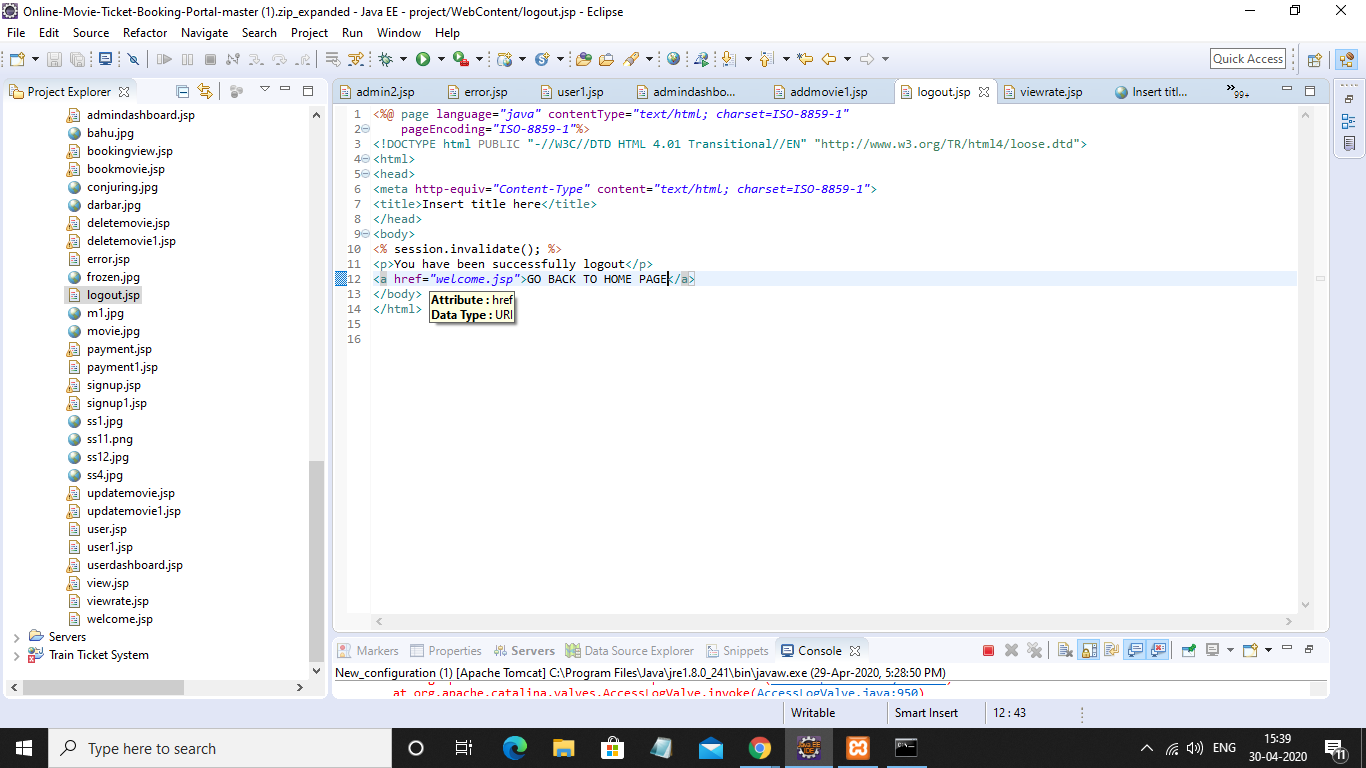
While different users login to the system,we associate them with an attribute called ‘username’ when the login password matches with the database password.



And when the user books a ticket,it is inserted into the booking table along with the username obtained from the session attribute.

**LOGOUT-**

Session is destroyed.



**Admin dashboard:**

****

1.When admin enters correct password:

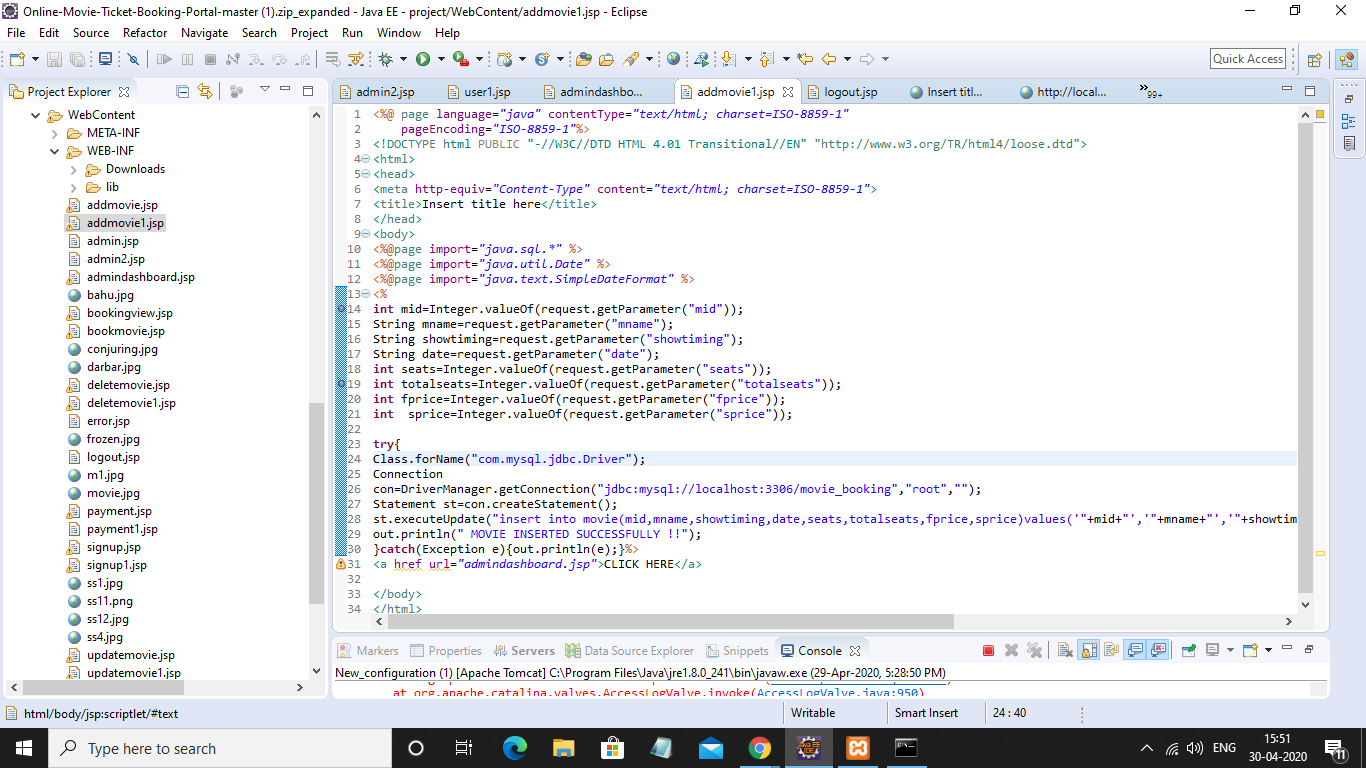
He has following options

|  |
| --- |
| 1.Add movie  2.Delete movie  3,Update movie  4.View booking details of various users  5.logout |

***1.Add movie***

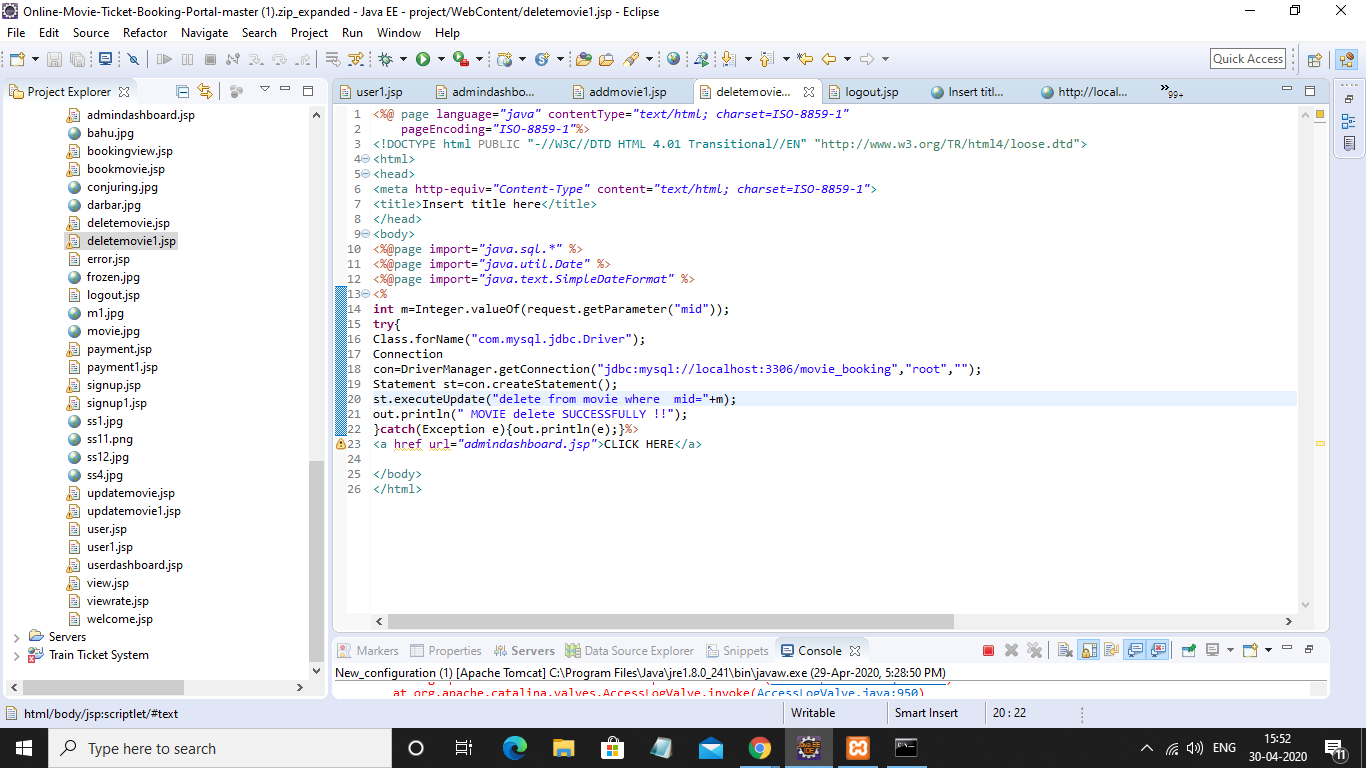
New entry of movie in the database.

**Implementation:**

****

***2.Delete movie***

**Implementation:**

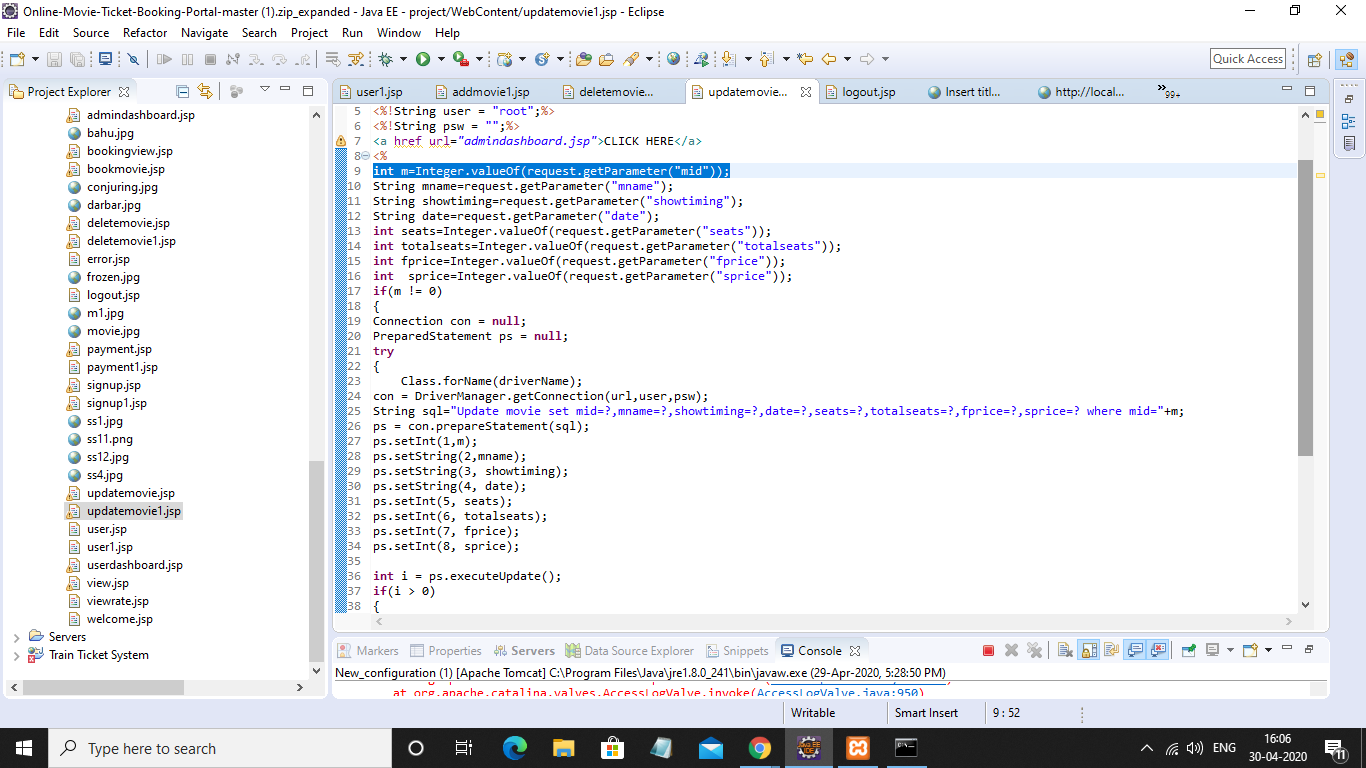


Delete a particular movie with movie\_id

***3.Update Movie***

**Implementation:**

Update the details of movie

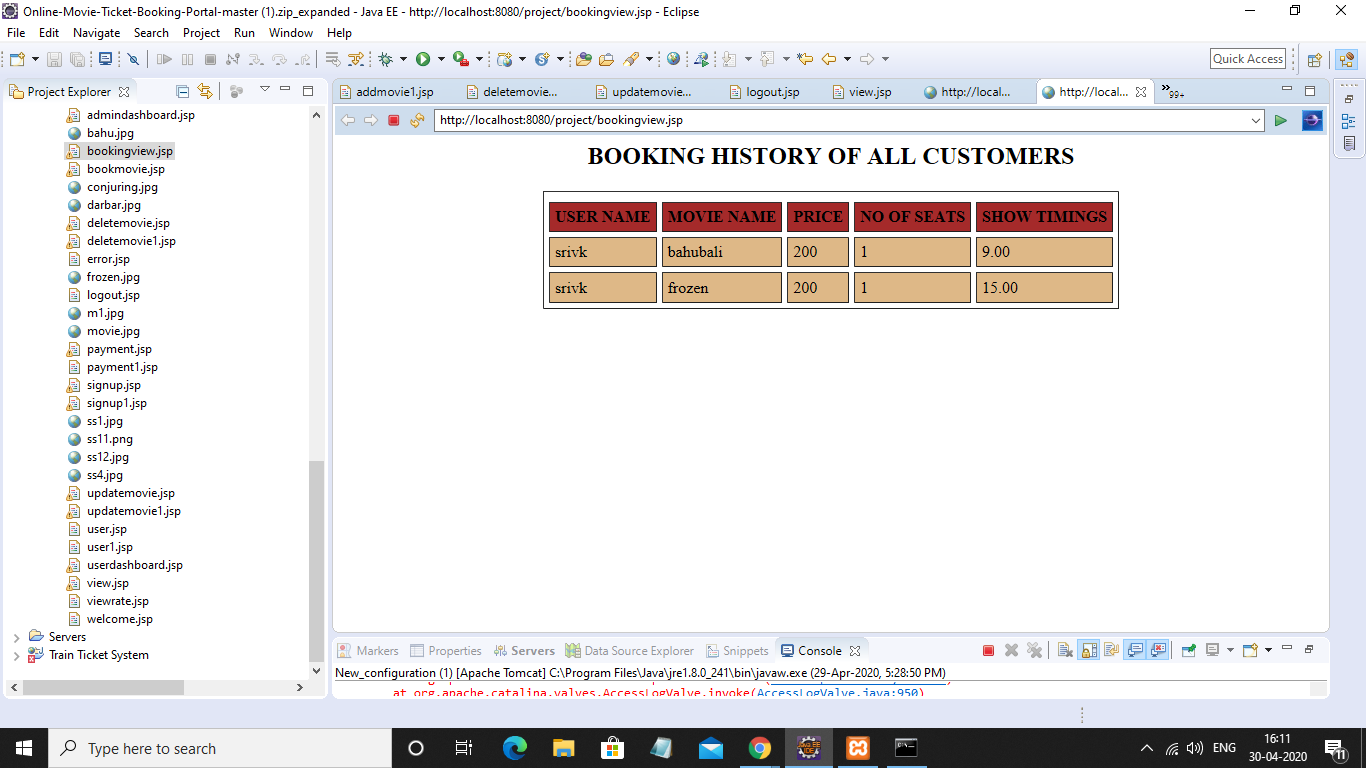


***4.View bookings***

**Implementation:**

The booking details from the ‘bookings3’ table are displayed.





***2.user:***

|  |
| --- |
| 1.book ticket  2.view booking history  3.Logout: |

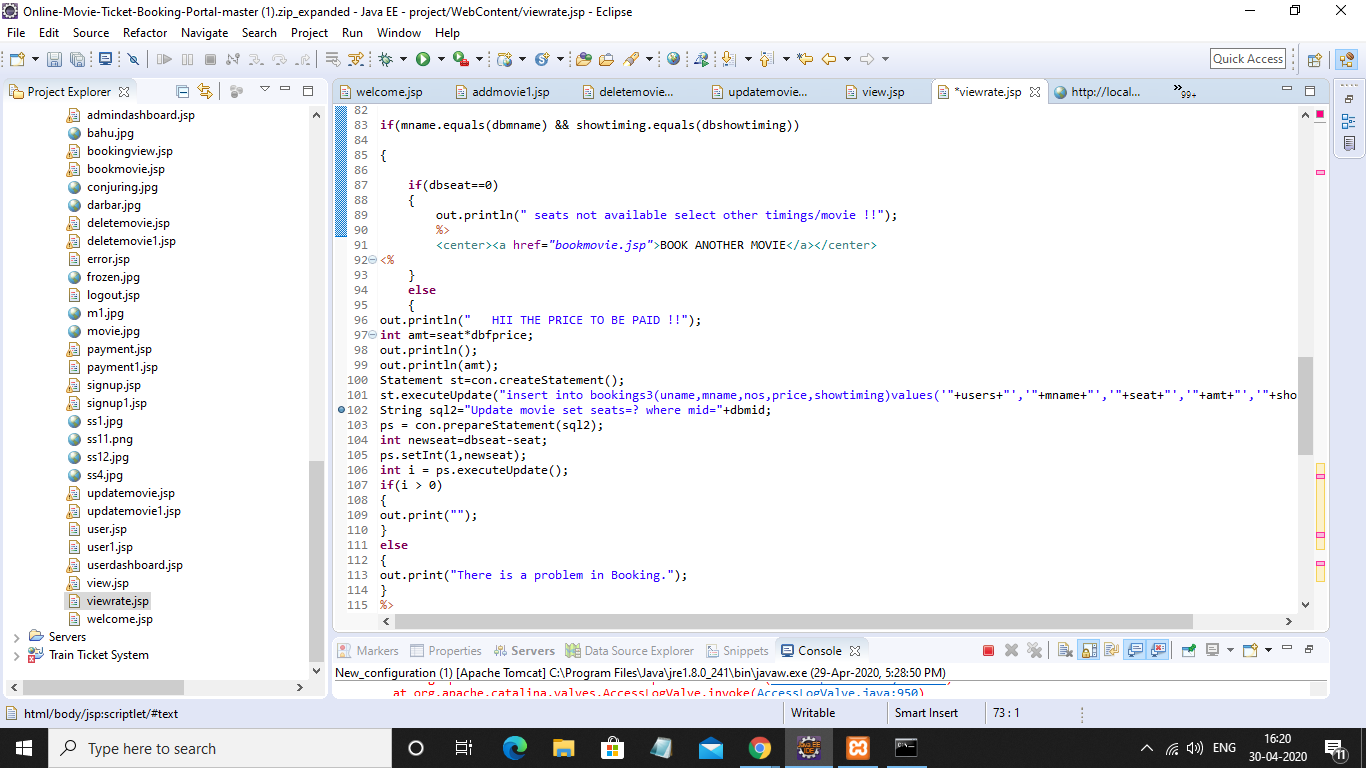
1.**book ticket:**

The movie name,show timing and no of seats is got from the user.

* It check whether that particular seats needed by the user
* If present it reduces the no of seats in the db by the seats booked by user
* And then inserts into booking table by the session name
* And then prints the amount to be paid by the user
* And redirects to the payment page
* If seats are not available ,it displays try another movie

***IMPLEMENTATION:***

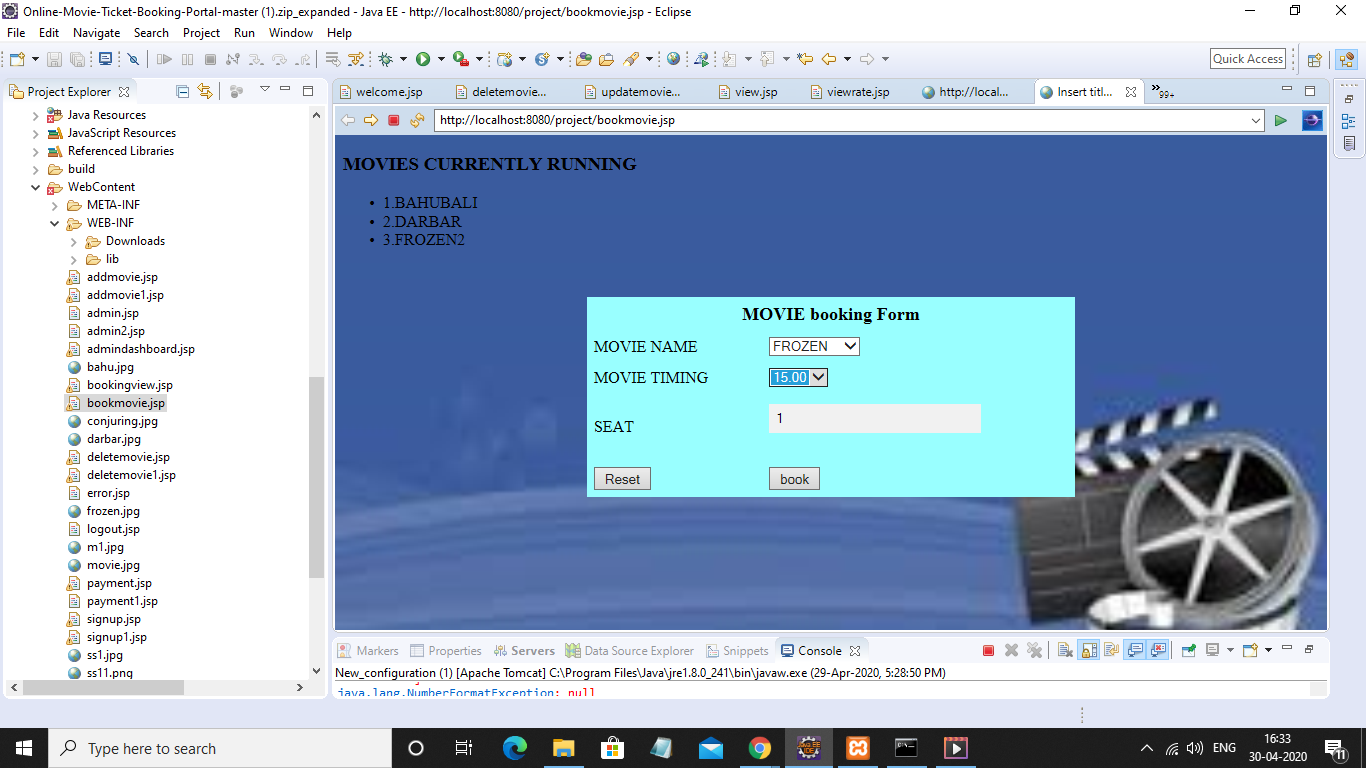


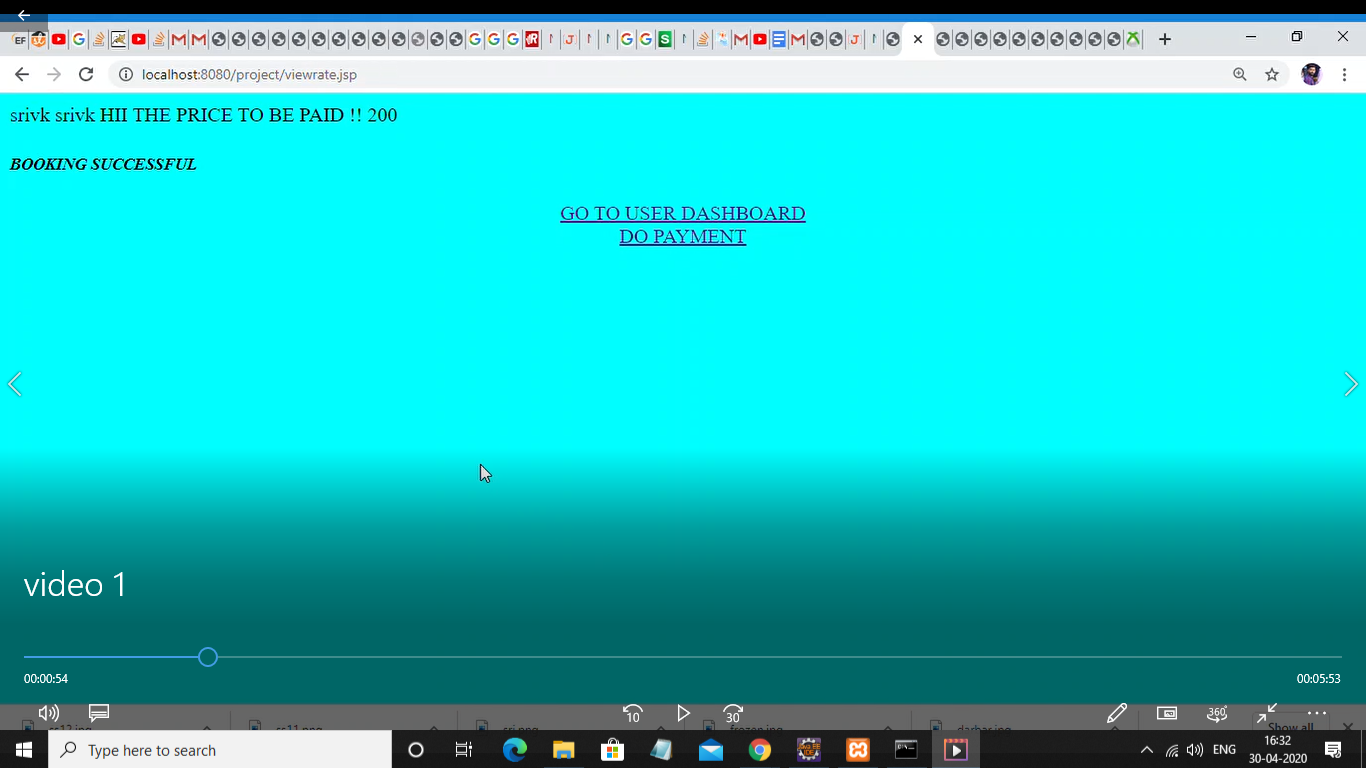


***Test case:***

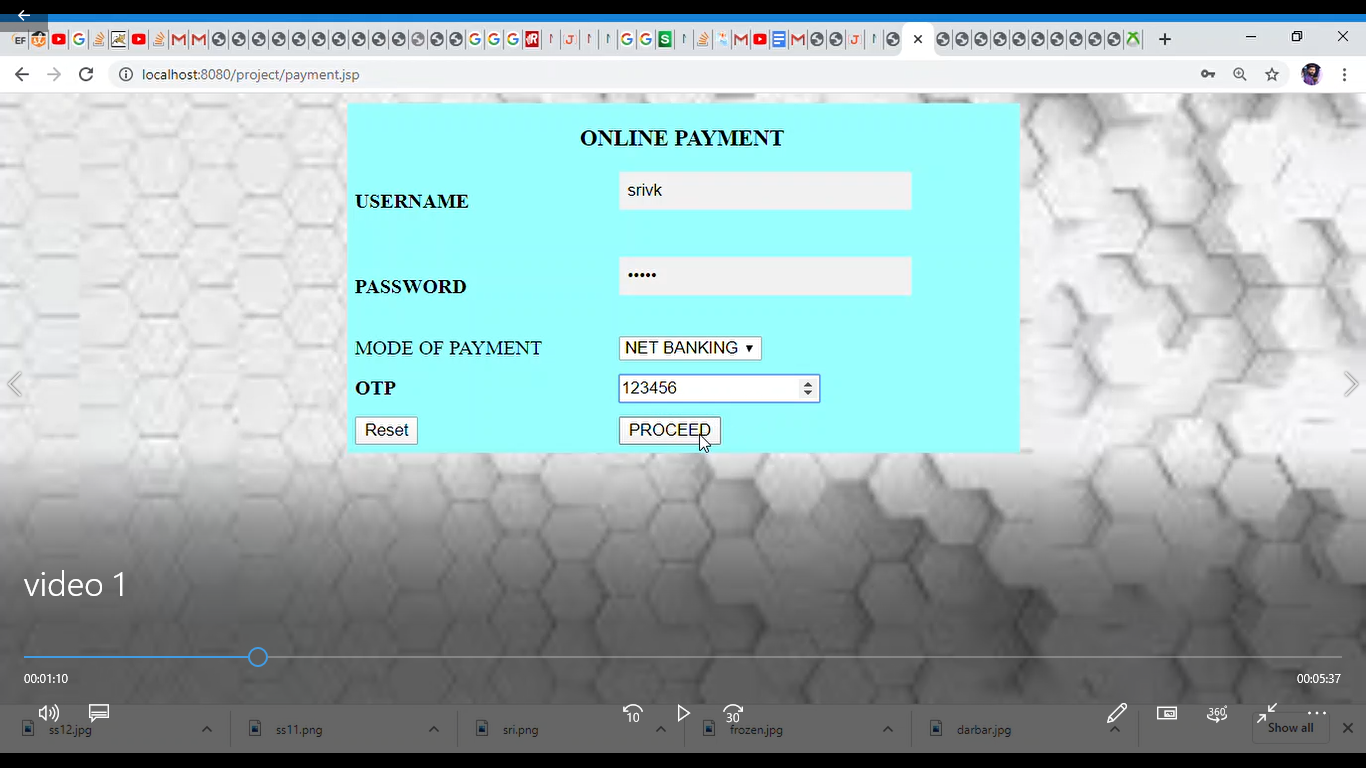
**Available movie:**

**Frozen 2 15.00 show:**

****

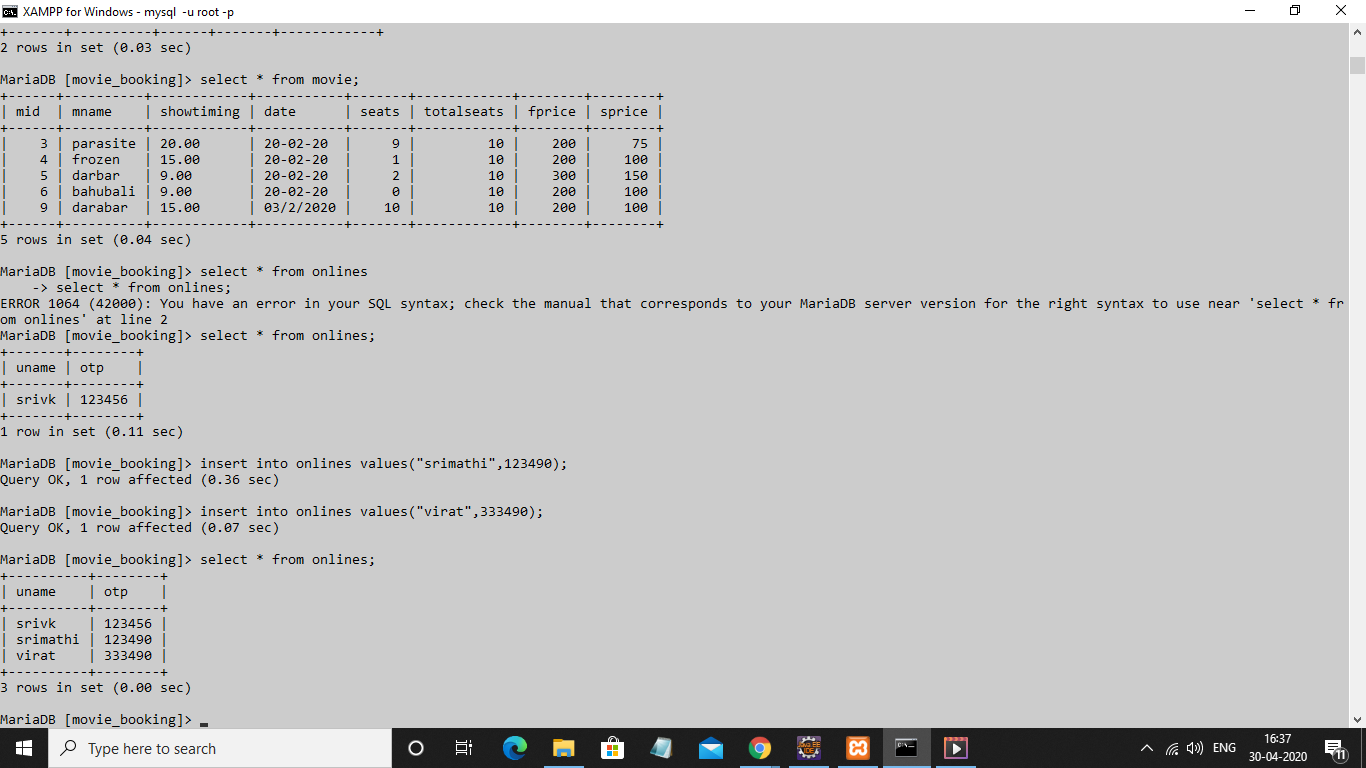
****

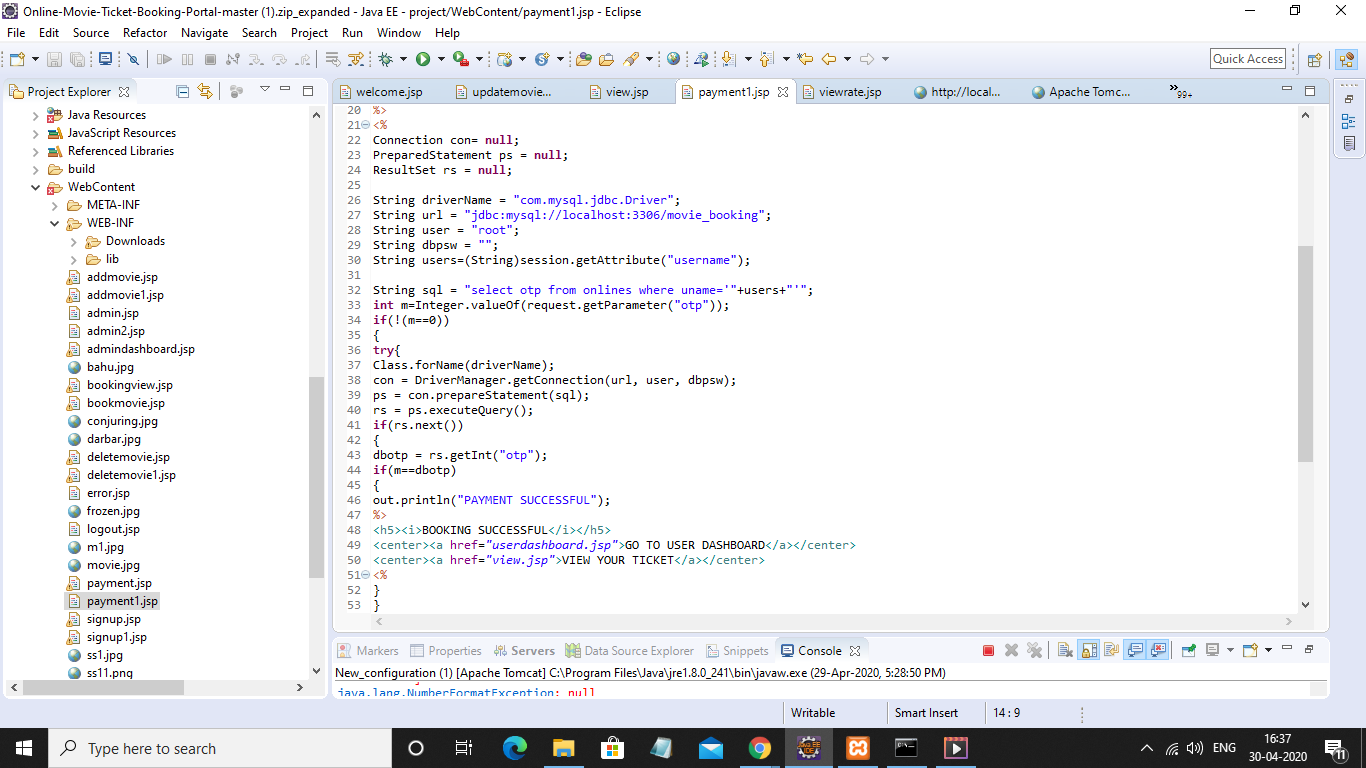
**Payment:**

****

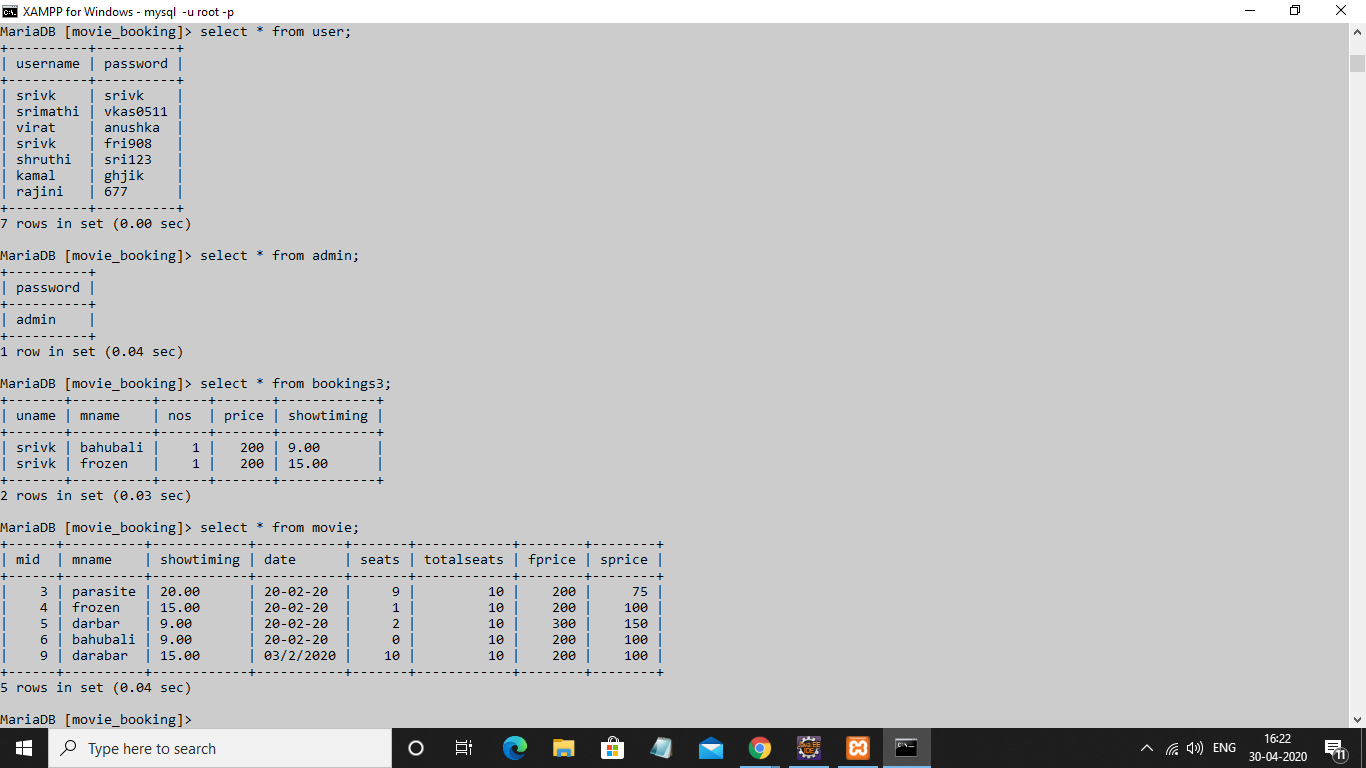
***It checks with online table whether it matches with the otp of the username in the database.***

***Implementation*:**

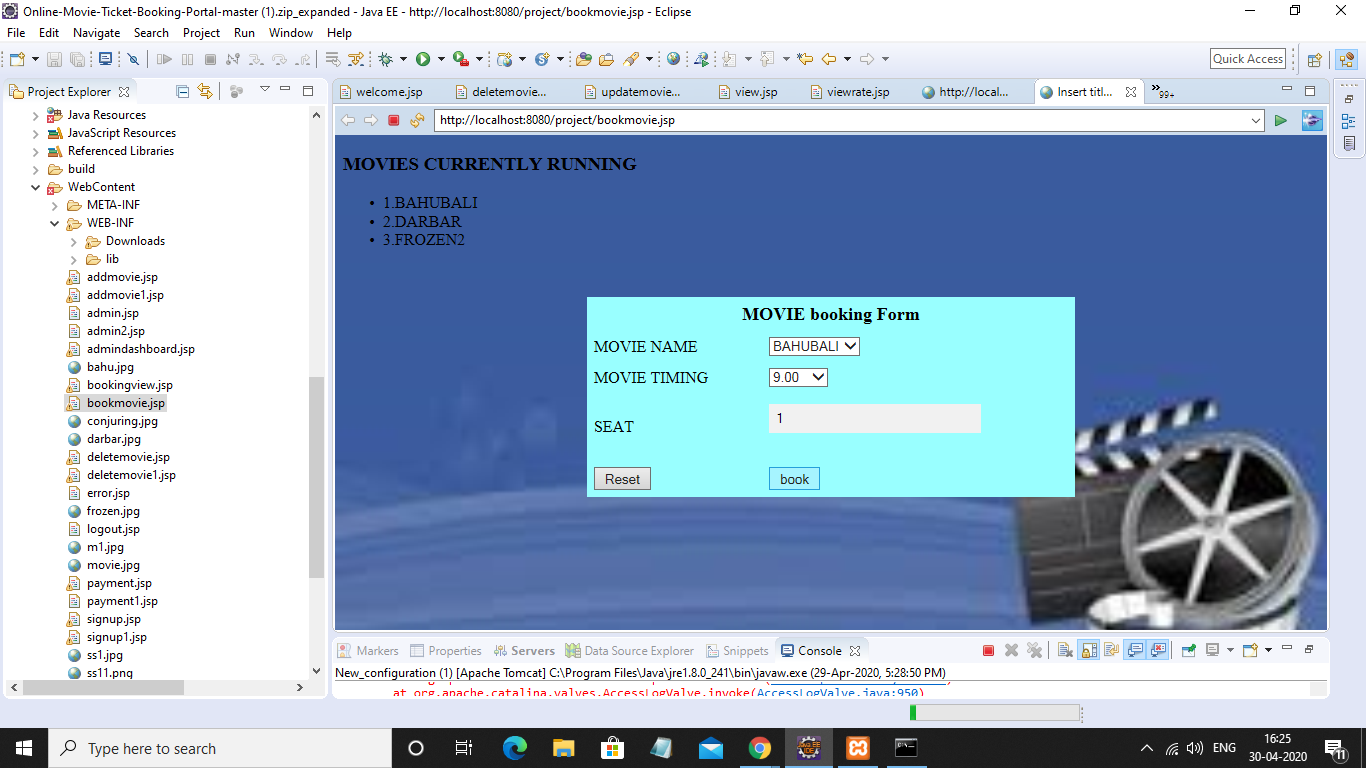
****

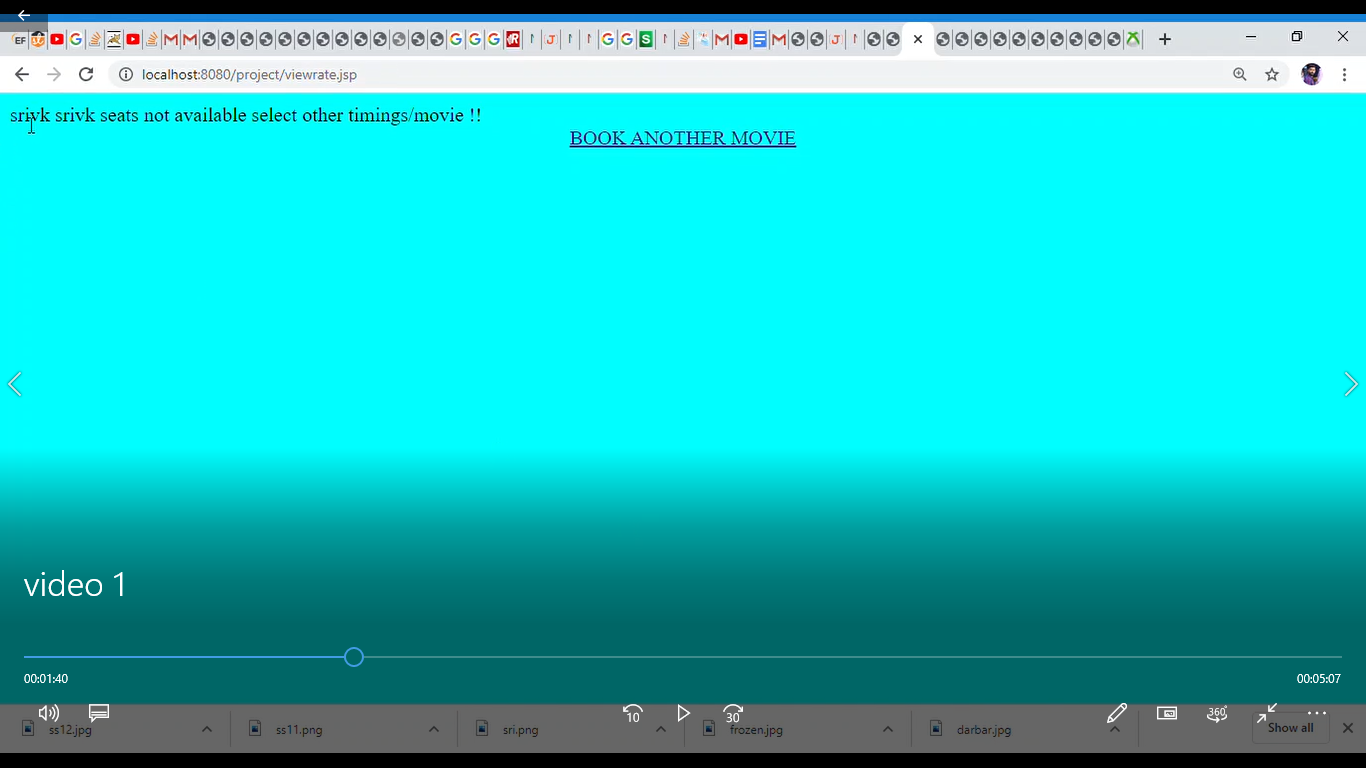
****

**Unavailable movie:**

****

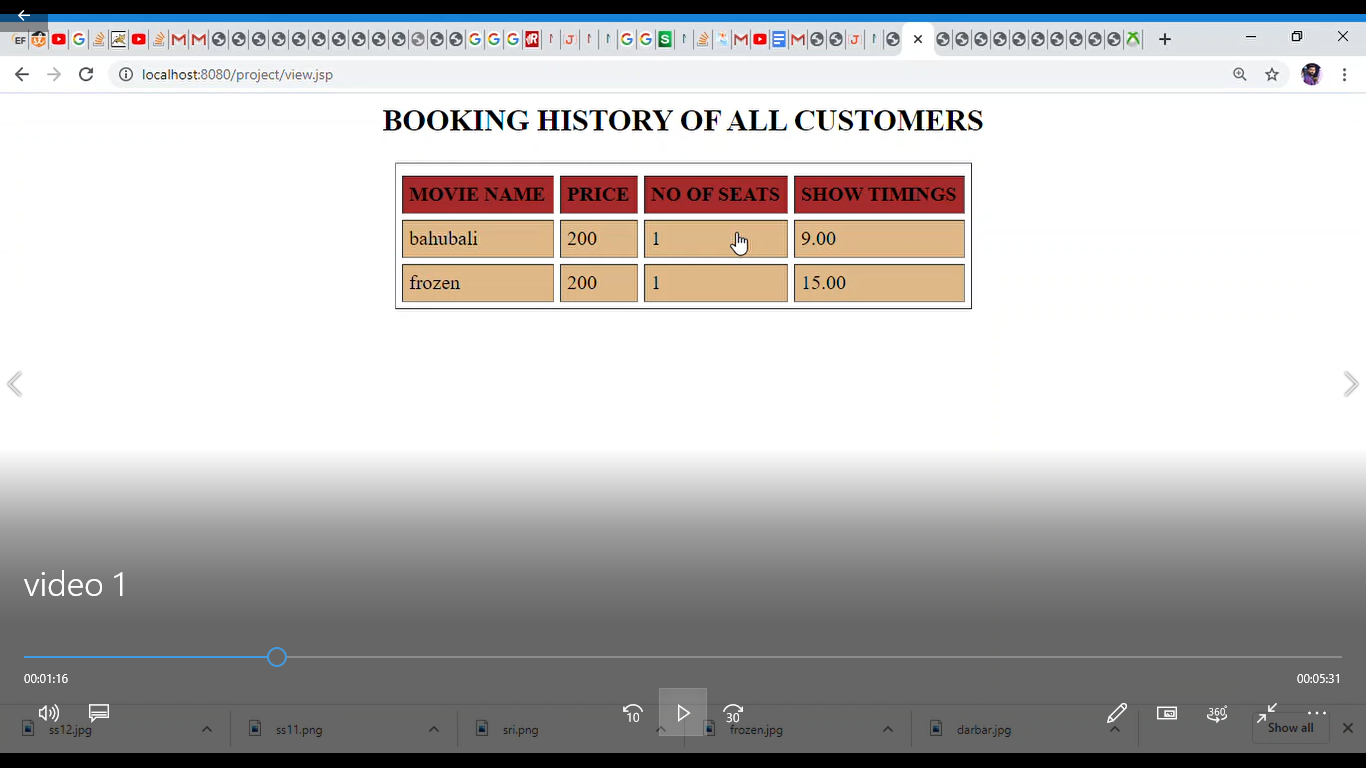
**Bahubali ‘9.00’ show**

****

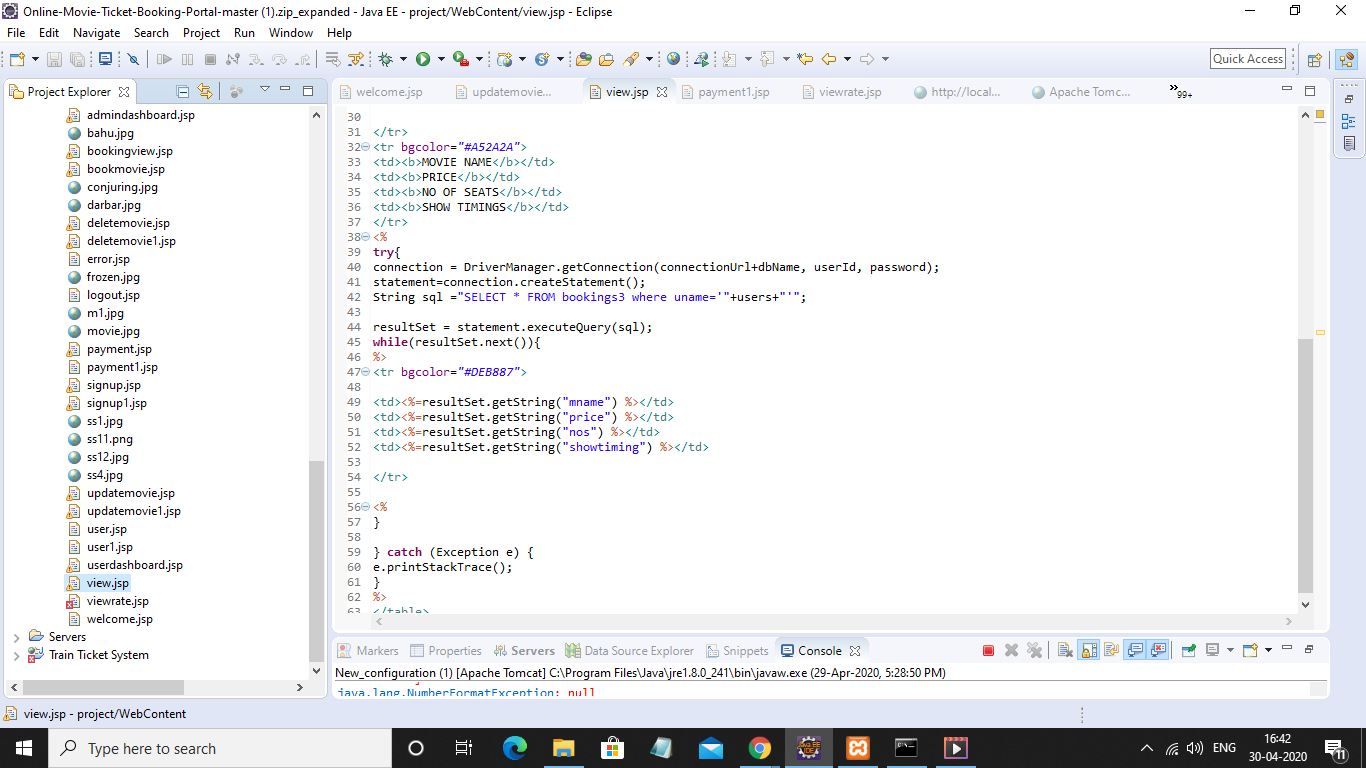


**3.view ticket;**

The ticket is displayed by obtaining values from the booking table by the session username.



Implementation:



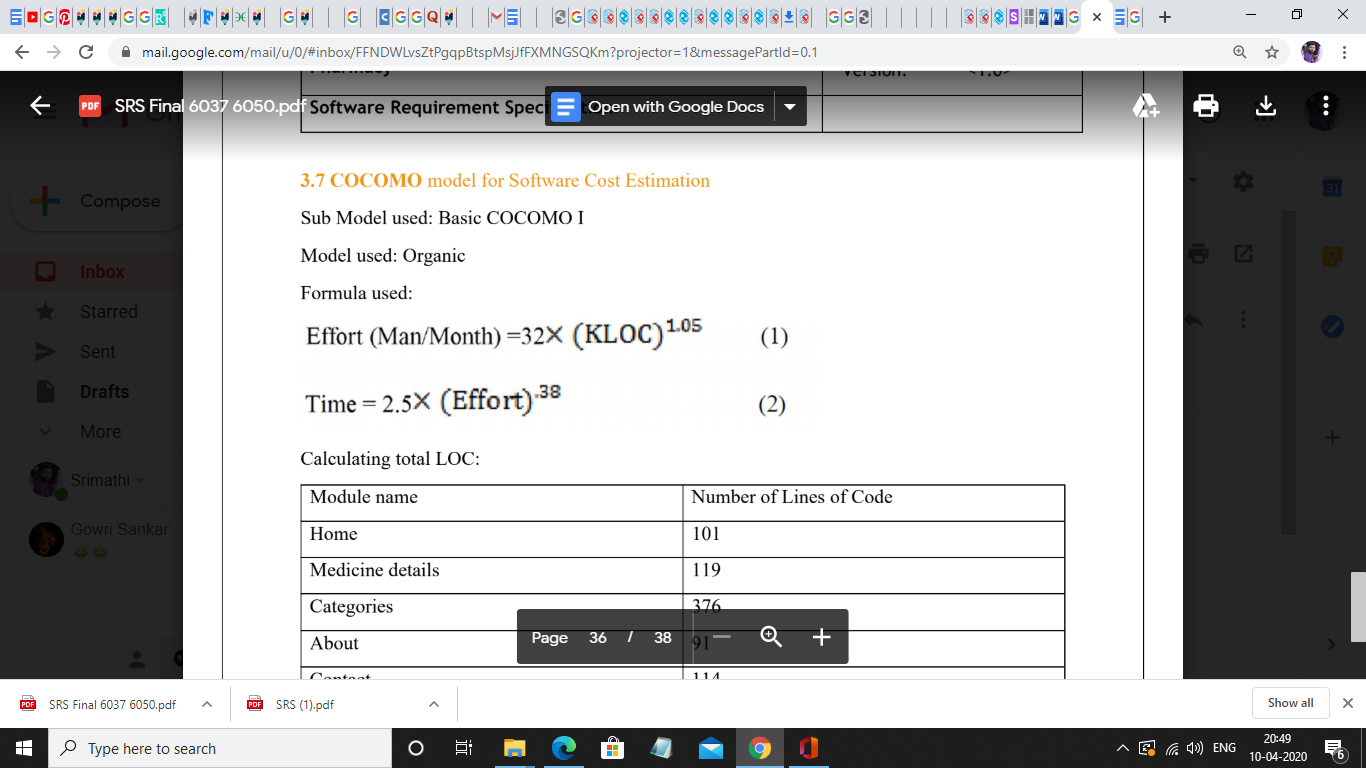
### **ONLINE MOVIE BOOKING**

### Cost Estimation

**Sub Model used:** Basic COCOMO I

**Model used:** Organic

**Formula used**:



**Calculating total LOC:**

Module name Number of Lines of Code

|  |  |
| --- | --- |
| **Module name** | **No of lines of code** |
| **USER** |  |
| User registration | 160 |
| Login | 180 |
| Home screen | 30 |
| Booking movies | 107 |
| Preview booking | 163 |
| Payment | 200 |
| logout | 14 |
| **ADMIN** |  |
| login | 85 |
| Dashboard | 150 |
| Booking management | 273 |
| **MOVIE MANAGEMENT** |  |
| Add/update movie | 240 |
| Delete movie | 102 |
| logout | 23 |
| sum | 1727=1.727 KLOC |

**Estimating effort:** effort=32 \* (1.727) ^1.05=5.679 M

**Estimating time:**

time=2.5 \*(5.679)^0.38=4.836 Months

**SUB MODEL USED:**

**Intermediate COCOMO 1**

**Mode used:**organic

**Formula used:**

****

Cost Drivers:

**Product Attributes:**

RELY – Required Software Reliability.

DATA – Database Size.

CPLX- Product Complexity.

**Computer Attributes:**

TIME- Execution Time.

STOR- Main Storage.

VIRT- Virtual Machine Volatility.

TURN- Computer Turn Around Time.

**Personal Attributes**

o ACAP-Analyst Capability.

o AEXP-Application Experience.

o PCAP- Programmer Capability.

o VEXP- Virtual Machine Experience.

o LEXP- Programming Language Experience.

Project Attributes

o MODP- Use of Modern Programming Practices.

o TOOL- Use of Software Tool.

o SCED- Required Development Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | low | normal | high | Very high |
| RELY | 0.90 |  |  |  |
| DATA |  |  |  | 1.20 |
| STOR |  |  |  | 1.20 |
| AXEP |  | 1.0 |  |  |
| DCAP | 0.80 |  |  |  |
| VIRT |  |  | 1.40 |  |
| TURN |  | 1.05 |  |  |
| COMPLEX |  |  |  | 1.45 |
| TIME | 0.90 |  |  |  |
| SCED | 0.65 |  |  |  |
| TOOL | 0.70 |  |  |  |
| MODP |  | 1.10 |  |  |
| LEXP |  | 1.0 |  |  |
| VEXP |  |  | 1.2 |  |
| ACAP |  | 1.0 |  |  |

Cost driver values:

EAF:product of these

0.90\*1.20\*1.70\*1.0\*0.80\*1.20\*1.0\*1.45\*0.90\*0.6\*0.70\*1.2\*1\*1.10=1.521

THUS

EFFORT=1.5\*32\*(1.727)^1.05

effort= 8.52 MM

time=2.5\*( 6.89 )^0.38

time= 5.64 months

**RESULT:**

User and admin modules has been implemented and various test cases are discussed and cost estimation for this project is attached.