Chalamalasetti, Sai

Sai.Chalamalasetti@dell.com

My Movie Plan Project

Online Booking Movie Tickets Responsive Web Application

**PROJECT OVERVIEW**

The main aim of the project is any user they can access the system through website at any time. As a new user also, he can collect all information like as a movies Available, list of theatres

1. **Administrator module**
2. **User module**
3. **Ticket booking module**

**1.Administrator module**:

The Administrator is maintaining the user Details, Movie details, Theatre details, and Check the No of seats available.

**2.User module**

The user can first Registration in enter the Personal details, and User login and If you want to update personal Details and you and update. The user collects all information like Movies details, Theatre Details.

3. **Booking module:**

The Users will search for the movie and then go for theatre then booking the tickets

online

GIT HUB LINK

<https://github.com/sudeepth003/MyMoviePlan>

**Use Case Diagrams**

**Admin:**

The Administrator is the user of the system. He is the responsible person to define the Employee details User Account Details.

**Admin**

🚹

**Administrator Login Sequence**

**Login screen**

**Enter log name**

**Administrator login master**

**Administrator login master**

**Validate**

**Log name ()**

**Admin Master**

**Check for required privileges ()**

**Validate**

**Password ()**

**Creation of**

**User registration,**

**Change password**

**Users**

**registration,**

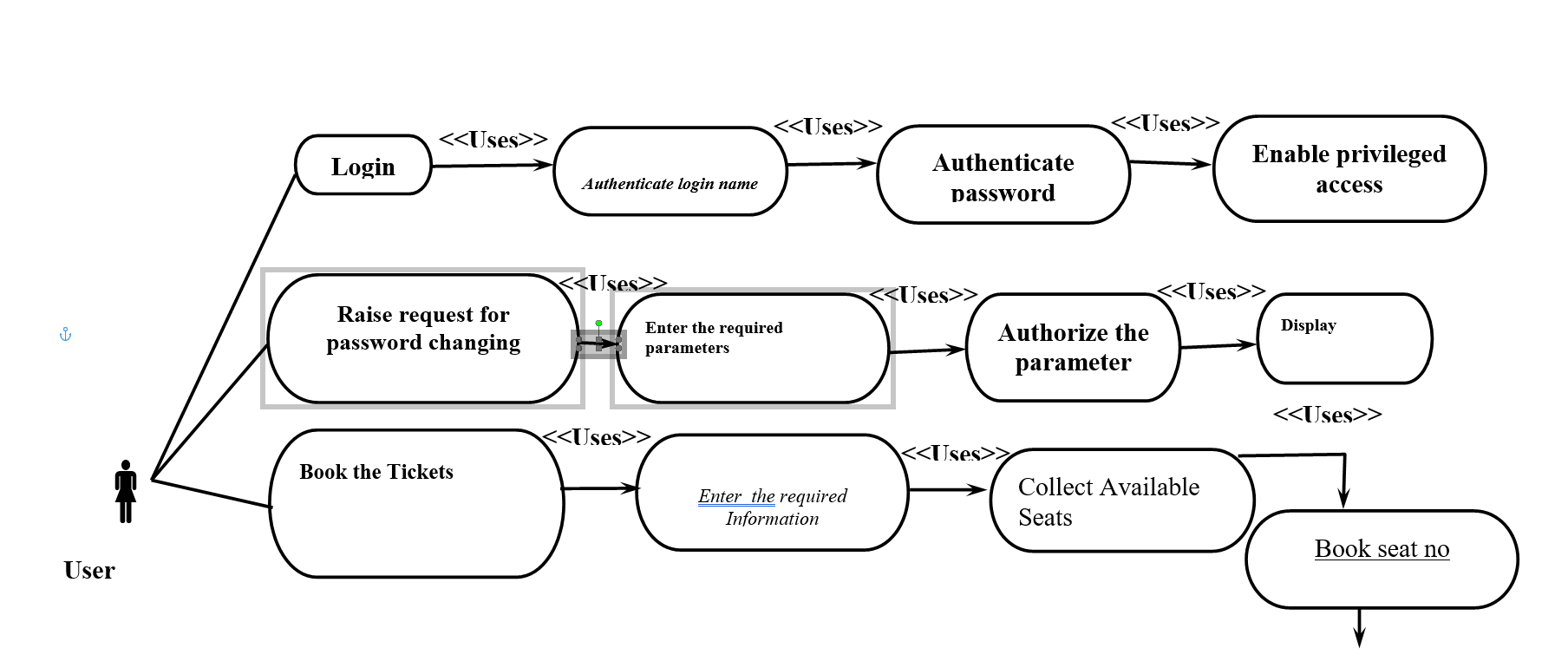
**User:**

He is responsible for the registration of the User Account, The user can Change password, view No seats Available and booking the tickets.

**User**

🚺

**Elaborated Diagrams**



**User** **Login** **Sequence**

**Login screen**

**Enter log name**

**User login master**

**User login master**

**Validate**

**Log name ()**

**User Master**

**Validate**

**Password ()**

**Check for required privileges ()**

**Creation**

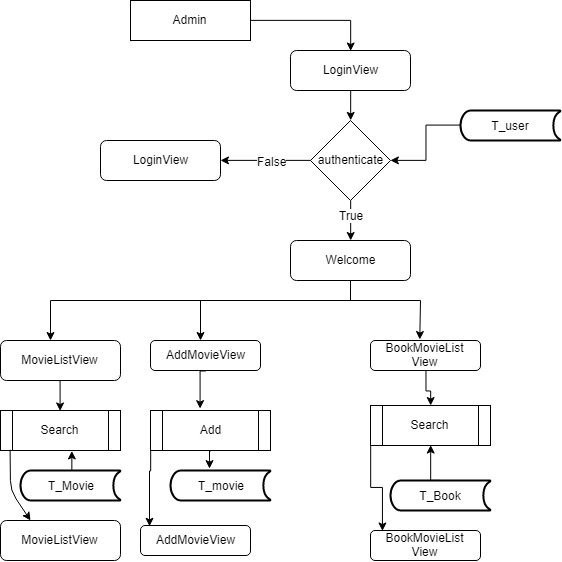
**User profile, booking tickets**

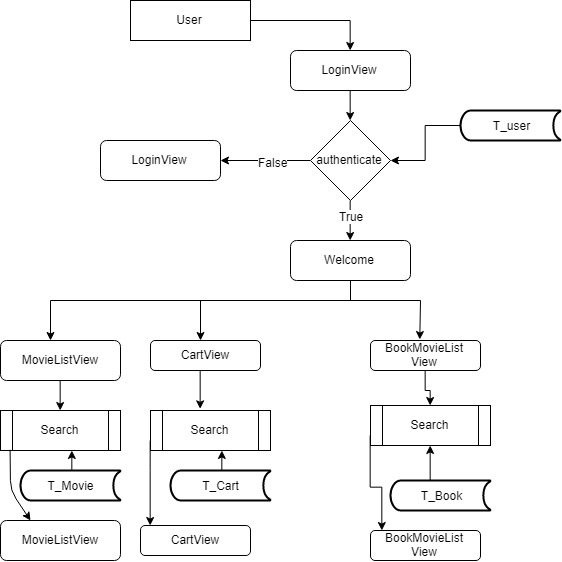
**Profile,**

**Add User Account,**

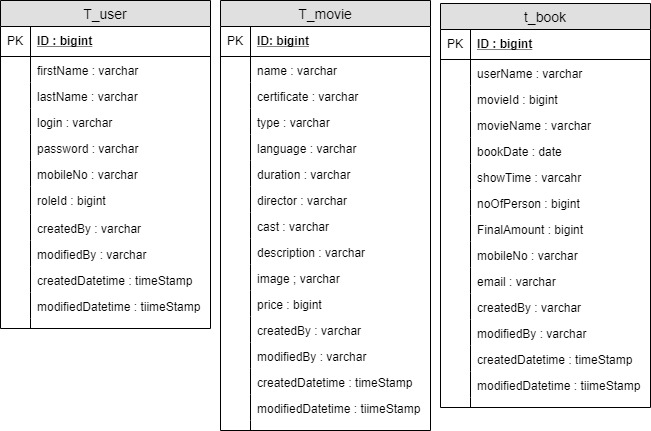
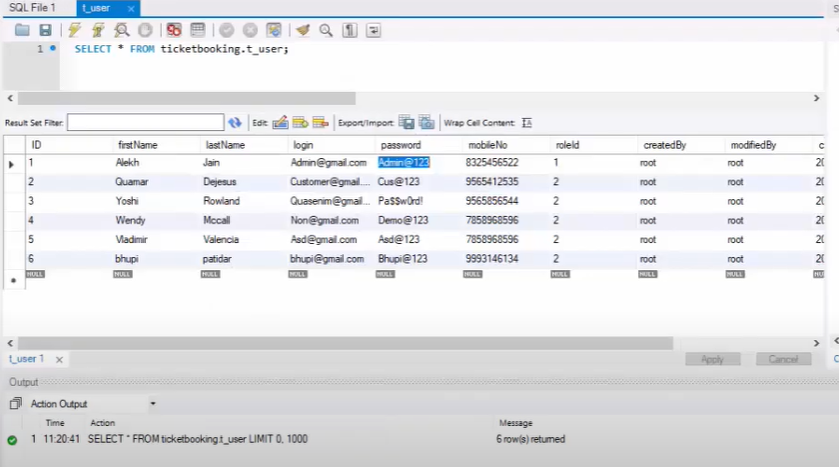
**View user Status**

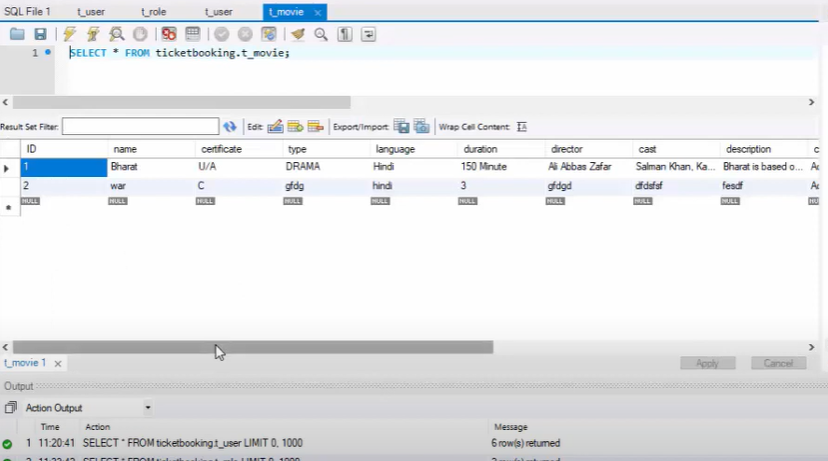
**Data Flow Diagram**

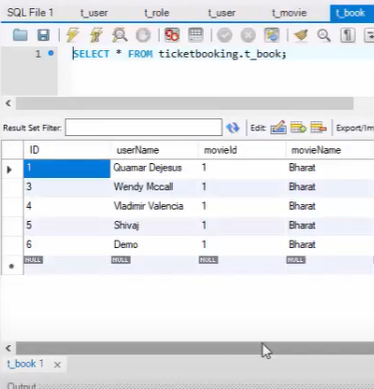


****

**Entity Relationship Diagram**

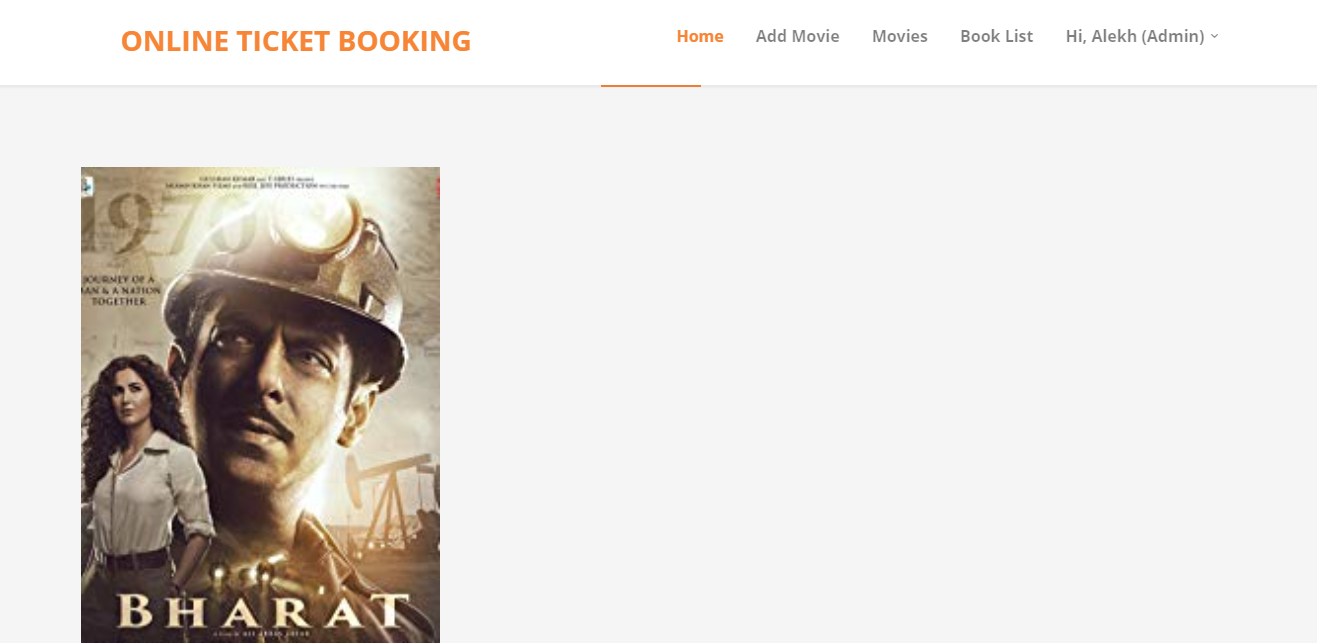
****



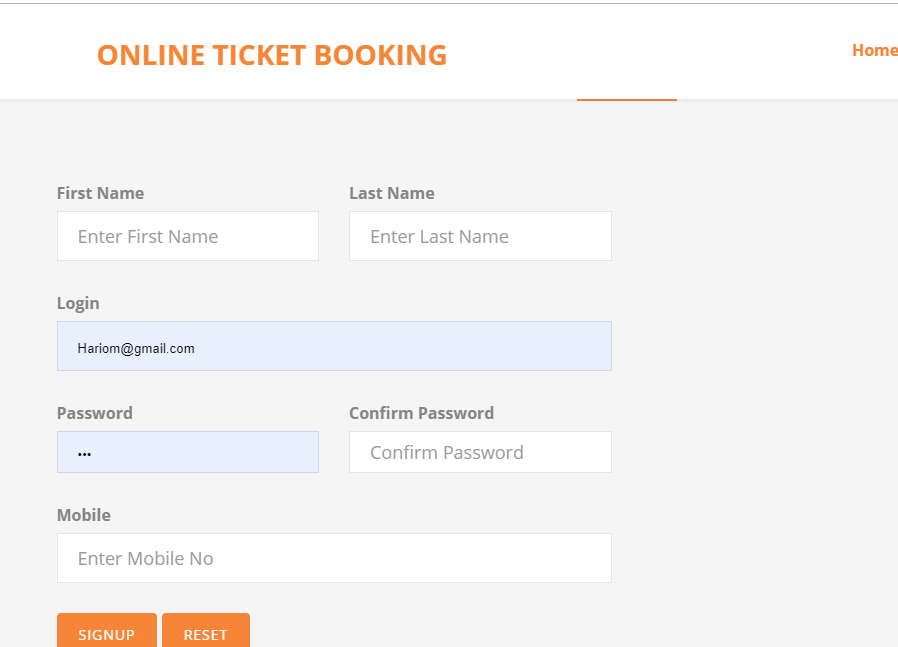


**Web Application**

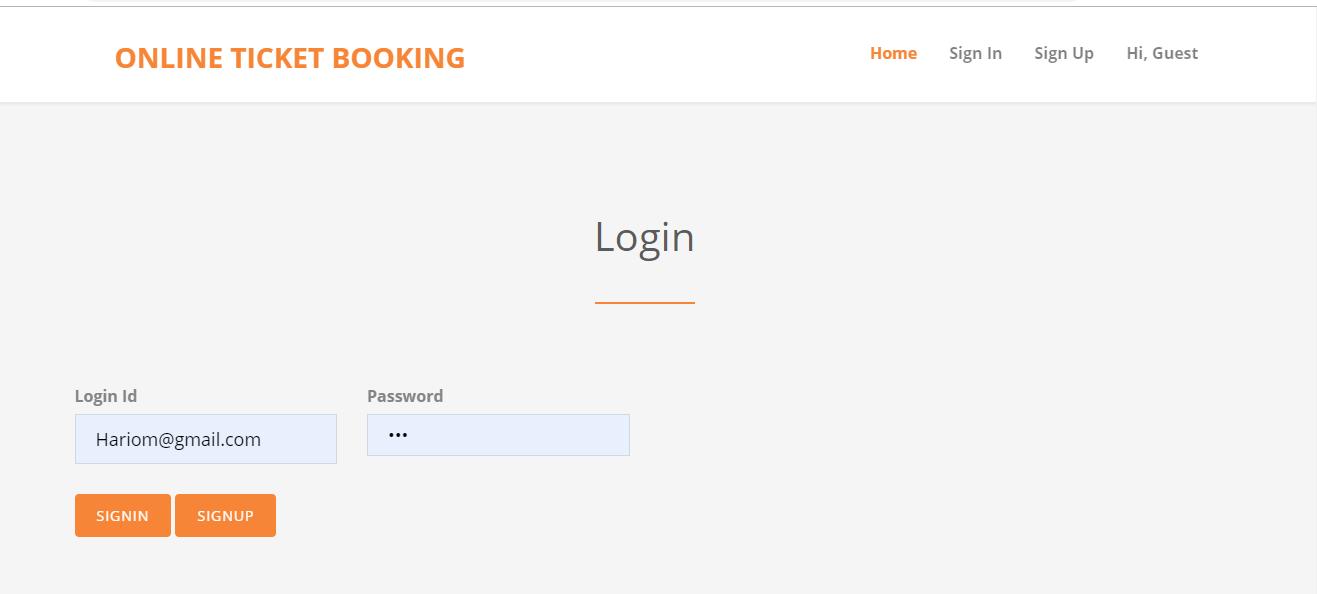
**Home Page :-**



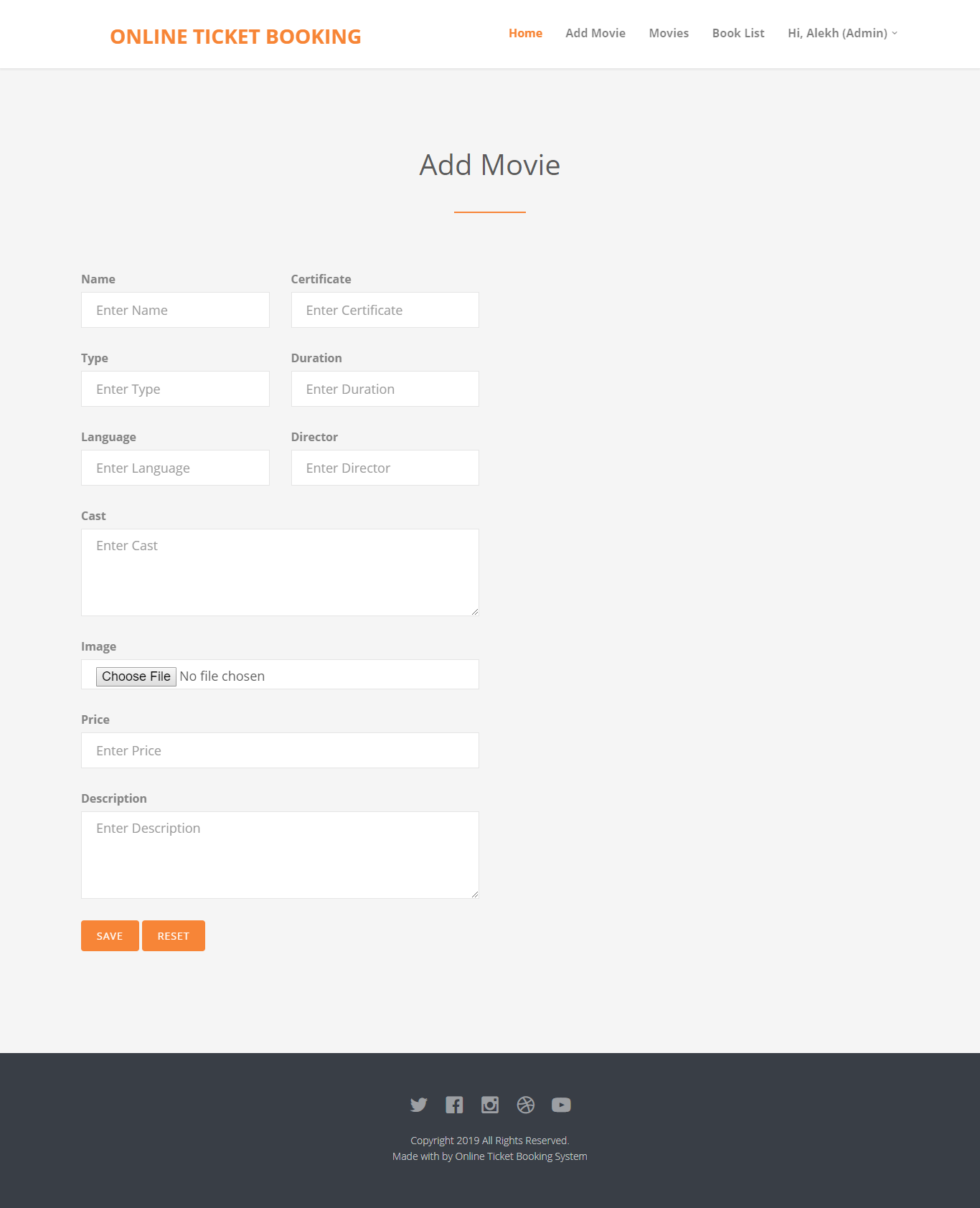
User Registration Page



**User Login Page:-**



**ADMIN Movie Add Page:-**

****

**Source code**

**package** in.co.online.ticket.bean;

**import** java.sql.Timestamp;

**public** **abstract** **class** BaseBean **implements** DropdownListBean, Comparable<BaseBean> {

/\*\*

\* Generic Attribute Id For All Child

\*/

**protected** **long** id;

/\*\*

\* Generic Attribute Created BY For All Child

\*/

**protected** String createdBy;

/\*\*

\* Generic Attribute Modified BY For All Child

\*/

**protected** String modifiedBy;

/\*\*

\* Generic Attribute Created Date and TIme For All Child

\*/

**protected** Timestamp createdDatetime;

/\*\*

\* Generic Attribute Modified Date And TIme For All Child

\*/

**protected** Timestamp modifiedDatetime;

/\*\*

\* accessor

\*/

/\*\*

\* **@return** ID of Child

\*/

**public** **long** getId() {

**return** id;

}

/\*\*

\* **@param** Id

\* To set Child ID

\*/

**public** **void** setId(**long** id) {

**this**.id = id;

}

/\*\*

\* **@return** Created By Child

\*/

**public** String getCreatedBy() {

**return** createdBy;

}

/\*\*

\* **@param** CreatedBy

\* To set Child Creatd By

\*/

**public** **void** setCreatedBy(String createdBy) {

**this**.createdBy = createdBy;

}

/\*\*

\* **@return** Modified By Child

\*/

**public** String getModifiedBy() {

**return** modifiedBy;

}

/\*\*

\* **@param** Modified

\* by To set Child ModifiedBY

\*/

**public** **void** setModifiedBy(String modifiedBy) {

**this**.modifiedBy = modifiedBy;

}

/\*\*

\* **@return** Created Date And TIme of Child

\*/

**public** Timestamp getCreatedDatetime() {

**return** createdDatetime;

}

/\*\*

\* **@param** Created

\* Date and Time To set Child Created dATE aND tiME

\*/

**public** **void** setCreatedDatetime(Timestamp createdDatetime) {

**this**.createdDatetime = createdDatetime;

}

/\*\*

\* **@return** Modified Date And Time of Child

\*/

**public** Timestamp getModifiedDatetime() {

**return** modifiedDatetime;

}

/\*\*

\* **@param** Modified

\* By To set Child ModiFied By

\*/

**public** **void** setModifiedDatetime(Timestamp modifiedDatetime) {

**this**.modifiedDatetime = modifiedDatetime;

}

**public** **int** compareTo(BaseBean next) {

// **TODO** Auto-generated method stub

**return** getValue().compareTo(next.getValue());

}

}

Model

**package** in.co.online.ticket.bean;

**import** java.sql.Timestamp;

/\*\*

\* BaseBean JavaBean encapsulates Generic attributes

\*

\* **@author** Navigable Set

\* **@version** 1.0

\* **@Copyright** (c) Navigable Set

\*

\*/

**public** **abstract** **class** BaseBean **implements** DropdownListBean, Comparable<BaseBean> {

/\*\*

\* Generic Attribute Id For All Child

\*/

**protected** **long** id;

/\*\*

\* Generic Attribute Created BY For All Child

\*/

**protected** String createdBy;

/\*\*

\* Generic Attribute Modified BY For All Child

\*/

**protected** String modifiedBy;

/\*\*

\* Generic Attribute Created Date and TIme For All Child

\*/

**protected** Timestamp createdDatetime;

/\*\*

\* Generic Attribute Modified Date And TIme For All Child

\*/

**protected** Timestamp modifiedDatetime;

/\*\*

\* accessor

\*/

/\*\*

\* **@return** ID of Child

\*/

**public** **long** getId() {

**return** id;

}

/\*\*

\* **@param** Id

\* To set Child ID

\*/

**public** **void** setId(**long** id) {

**this**.id = id;

}

/\*\*

\* **@return** Created By Child

\*/

**public** String getCreatedBy() {

**return** createdBy;

}

/\*\*

\* **@param** CreatedBy

\* To set Child Creatd By

\*/

**public** **void** setCreatedBy(String createdBy) {

**this**.createdBy = createdBy;

}

/\*\*

\* **@return** Modified By Child

\*/

**public** String getModifiedBy() {

**return** modifiedBy;

}

/\*\*

\* **@param** Modified

\* by To set Child ModifiedBY

\*/

**public** **void** setModifiedBy(String modifiedBy) {

**this**.modifiedBy = modifiedBy;

}

/\*\*

\* **@return** Created Date And TIme of Child

\*/

**public** Timestamp getCreatedDatetime() {

**return** createdDatetime;

}

/\*\*

\* **@param** Created

\* Date and Time To set Child Created dATE aND tiME

\*/

**public** **void** setCreatedDatetime(Timestamp createdDatetime) {

**this**.createdDatetime = createdDatetime;

}

/\*\*

\* **@return** Modified Date And Time of Child

\*/

**public** Timestamp getModifiedDatetime() {

**return** modifiedDatetime;

}

/\*\*

\* **@param** Modified

\* By To set Child ModiFied By

\*/

**public** **void** setModifiedDatetime(Timestamp modifiedDatetime) {

**this**.modifiedDatetime = modifiedDatetime;

}

**public** **int** compareTo(BaseBean next) {

// **TODO** Auto-generated method stub

**return** getValue().compareTo(next.getValue());

}

}

Controller

package in.co.online.ticket.controller;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Logger;

import in.co.online.ticket.bean.BaseBean;

import in.co.online.ticket.bean.RoleBean;

import in.co.online.ticket.bean.UserBean;

import in.co.online.ticket.exception.ApplicationException;

import in.co.online.ticket.exception.DuplicateRecordException;

import in.co.online.ticket.model.UserModel;

import in.co.online.ticket.util.DataUtility;

import in.co.online.ticket.util.DataValidator;

import in.co.online.ticket.util.PropertyReader;

import in.co.online.ticket.util.ServletUtility;

@WebServlet(name = "UserRegistrationCtl", urlPatterns = { "/UserRegistrationCtl" })

public class UserRegistrationCtl extends BaseCtl {

public static final String OP\_SIGN\_UP = "SignUp";

private static Logger log = Logger.getLogger(UserRegistrationCtl.class);

/\*\*

\* Validate input Data Entered By User

\*

\* @param request

\* @return

\*/

@Override

protected boolean validate(HttpServletRequest request) {

log.debug("UserRegistrationCtl Method validate Started");

boolean pass = true;

String login = request.getParameter("login");

if (DataValidator.isNull(request.getParameter("firstName"))) {

request.setAttribute("firstName",

PropertyReader.getValue("error.require", "First Name"));

pass = false;

}

if (DataValidator.isNull(request.getParameter("lastName"))) {

request.setAttribute("lastName",

PropertyReader.getValue("error.require", "Last Name"));

pass = false;

}

if (DataValidator.isNull(login)) {

request.setAttribute("login",

PropertyReader.getValue("error.require", "Login Id"));

pass = false;

} else if (!DataValidator.isEmail(request.getParameter("login"))) {

request.setAttribute("login",

PropertyReader.getValue("error.email", "Login"));

pass = false;

}

if (DataValidator.isNull(request.getParameter("confirmPassword"))) {

request.setAttribute("confirmPassword", PropertyReader.getValue(

"error.require", "Confirm Password"));

pass = false;

}

if (DataValidator.isNull(request.getParameter("password"))) {

request.setAttribute("password",

PropertyReader.getValue("error.require", "Password"));

pass = false;

} else if (!DataValidator.isPassword(request.getParameter("password"))) {

request.setAttribute("password",

PropertyReader.getValue("error.password", "Password"));

return false;

}else if (!DataValidator.isPassword(request.getParameter("password"))) {

request.setAttribute("password",

PropertyReader.getValue("error.password", "Password"));

return false;

}

if (!request.getParameter("password").equals(

request.getParameter("confirmPassword"))

&& !"".equals(request.getParameter("confirmPassword"))) {

/\*ServletUtility.setErrorMessage("Confirm Password did not match",

request);\*/

request.setAttribute("confirmPassword", PropertyReader.getValue("error.confirmPassword","Confirm Password"));

pass = false;

}

if (DataValidator.isNull(request.getParameter("mobile"))) {

request.setAttribute("mobile", PropertyReader.getValue("error.require","Mobile No"));

pass = false;

}else if(!DataValidator.isPhoneNo(request.getParameter("mobile"))){

request.setAttribute("mobile", PropertyReader.getValue("error.invalid","Mobile No"));

pass=false;

}

log.debug("UserRegistrationCtl Method validate Ended");

return pass;

}

\* Populates bean object from request parameters

\*

\* @param request

\* @return

\*/

@Override

protected BaseBean populateBean(HttpServletRequest request) {

log.debug("UserRegistrationCtl Method populatebean Started");

UserBean bean = new UserBean();

bean.setId(DataUtility.getLong(request.getParameter("id")));

bean.setRoleId(RoleBean.CUSTOMER);

bean.setFirstName(DataUtility.getString(request

.getParameter("firstName")));

bean.setLastName(DataUtility.getString(request.getParameter("lastName")));

bean.setLogin(DataUtility.getString(request.getParameter("login")));

bean.setPassword(DataUtility.getString(request.getParameter("password")));

bean.setConfirmPassword(DataUtility.getString(request

.getParameter("confirmPassword")));

bean.setMobileNo(DataUtility.getString(request.getParameter("mobile")));

populateDTO(bean, request);

log.debug("UserRegistrationCtl Method populatebean Ended");

return bean;

}

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public UserRegistrationCtl() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* Contains display logic

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

log.debug("UserRegistrationCtl Method doGet Started");

ServletUtility.forward(getView(), request, response);

}

/\*\*

\* Contains submit logic

\*/

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

System.out.println("in post method");

log.debug("UserRegistrationCtl Method doPost Started");

String op = DataUtility.getString(request.getParameter("operation"));

// get model

UserModel model = new UserModel();

long id = DataUtility.getLong(request.getParameter("id"));

if (OP\_SIGN\_UP.equalsIgnoreCase(op)) {

UserBean bean = (UserBean) populateBean(request);

try {

// System.out.println("in try sign up");

long pk = model.registerUser(bean);

//System.out.println("register");

bean.setId(pk);

request.getSession().setAttribute("UserBean", bean);

ServletUtility.setSuccessMessage("User Successfully Registered", request);

ServletUtility.forward(OTBView.USER\_REGISTRATION\_VIEW, request, response);

return;

} catch (DuplicateRecordException e) {

log.error(e);

ServletUtility.setBean(bean, request);

ServletUtility.setErrorMessage("Login id already exists",

request);

ServletUtility.forward(getView(), request, response);

} catch (ApplicationException e) {

ServletUtility.handleException(e, request, response);

e.printStackTrace();

return;

}

}else if (OP\_RESET.equalsIgnoreCase(op)) {

ServletUtility.redirect(OTBView.USER\_REGISTRATION\_CTL, request, response);

return;

}

log.debug("UserRegistrationCtl Method doPost Ended");

}

@Override

protected String getView() {

return OTBView.USER\_REGISTRATION\_VIEW;

}

}