

















import javax.servlet.ServletContext; 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 javax.servlet.http.HttpSession;

import com.ata.bean.CredentialsBean; import com.ata.dao.CredentialDAOImpl;

@WebServlet("/ChangePasswordServlet")

public class ChangePasswordServlet extends HttpServlet {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

/\*\*

* @see HttpServlet#HttpServlet()

\*/

public ChangePasswordServlet() { super();

// TODO Auto-generated constructor stub

}

/\*\*

* @param request Function Parameter
* @param response Function Parameter
* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
* @throws ServletException Throws any Servlet Exception
* @throws IOException Throws any IO Exception

\*/

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

{

HttpSession session=request.getSession();

String userId = (String) session.getAttribute("userid");

String password=request.getParameter("newpass");

CredentialDAOImpl credentialDAOImpl = new CredentialDAOImpl();

CredentialsBean credentialsBean = new CredentialsBean(); ServletContext sc = request.getServletContext(); credentialsBean.setUserID(userId);

System.out.println(password.length());

if(password.length()>=4)

{

String status = credentialDAOImpl.changePassword(credentialsBean, password);

if("SUCCESS".equals(status))

{

session.invalidate();

RequestDispatcher rd = sc.getRequestDispatcher("/index.jsp"); rd.forward(request, response);

}

else

{

session.setAttribute("UpdateStatus", "Password Could Not be

Changed !!!");

RequestDispatcher rd = sc.getRequestDispatcher("/index.jsp");

rd.forward(request, response);

}

}

else

{

session.setAttribute("UpdateStatus", "Password Must Be Atleast of 4 Characters !!!");

session.invalidate();

RequestDispatcher rd = sc.getRequestDispatcher("/beforelogin.jsp"); rd.forward(request, response);

}

}

} package com.ata.servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher; import javax.servlet.ServletContext; 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 javax.servlet.http.HttpSession;

import com.ata.bean.DriverBean;

import com.ata.service.AdministratorImpl;

@WebServlet("/CreateDriverServlet")

public class CreateDriverServlet extends HttpServlet {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

/\*\*

* @see HttpServlet#HttpServlet()

\*/

public CreateDriverServlet() { super();

// TODO Auto-generated constructor stub }

/\*\*

* @param request Function Parameter
* @param response Function Parameter
* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
* @throws ServletException Throws any Servlet Exception \* @throws IOException Throws any IO Exception \*/

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException

{

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

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

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

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

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

String pinCode=request.getParameter("pincode");

String mobileNo=request.getParameter("mno");

String licenseNo=request.getParameter("lno");

DriverBean dbean = new DriverBean();

dbean.setDriverID("temp"); dbean.setName(name); dbean.setStreet(street); dbean.setLocation(location); dbean.setCity(city); dbean.setState(state); dbean.setPincode(pinCode); dbean.setMobileNo(mobileNo);

dbean.setLicenseNumber(licenseNo);

AdministratorImpl adminImplement = new AdministratorImpl();

String finalStatus =adminImplement.addDriver(dbean);

HttpSession session = request.getSession();

ServletContext sc = request.getServletContext();

if("SUCCESS".equals(finalStatus))

{

session.setAttribute("UpdateStatus", "Driver Registered Successfully

!!!");

RequestDispatcher rd = sc.getRequestDispatcher("/admin.jsp");

rd.forward(request, response);

}

else if ("FAIL".equals(finalStatus))

{

session.setAttribute("UpdateStatus", "Driver Could Not be Registered !!!");

RequestDispatcher rd = sc.getRequestDispatcher("/admin.jsp");

rd.forward(request, response);

}

else

{

session.setAttribute("UpdateStatus", "Some Error Occured, Please Try again !!!");

RequestDispatcher rd = sc.getRequestDispatcher("/admin.jsp");

rd.forward(request, response);

}

}

} package com.ata.servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher; import javax.servlet.ServletContext; 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 javax.servlet.http.HttpSession;

import com.ata.bean.RouteBean;

import com.ata.service.AdministratorImpl;

@WebServlet("/CreateRouteServlet")

public class CreateRouteServlet extends HttpServlet {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

/\*\*

* @see HttpServlet#HttpServlet()

\*/

public CreateRouteServlet() { super();

// TODO Auto-generated constructor stub

}

/\*\*

* @param request Function Parameter
* @param response Function Parameter
* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
* @throws ServletException Throws any Servlet Exception
* @throws IOException Throws any IO Exception

\*/

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

{

String routeSource=request.getParameter("source");

String routeDestination=request.getParameter("destination");

String routeDistance=request.getParameter("distance");

String routeDuration=request.getParameter("duration");

Double rDistance=Double.parseDouble(routeDistance); Double rDuration=Double.parseDouble(routeDuration);

RouteBean rbean = new RouteBean();

//generating route id

rbean.setRouteID("temp"); rbean.setSource(routeSource); rbean.setDestination(routeDestination); rbean.setDistance(rDistance);; rbean.setTravelDuration(rDuration); if(routeSource.equals(routeDestination)) {

RequestDispatcher rd =request.getRequestDispatcher("/error.jsp"); rd.forward(request, response);

System.out.println("source and destination cannot be same");

}

else

{

AdministratorImpl adminImplement = new AdministratorImpl(); String finalStatus =adminImplement.addRoute(rbean);

if("SUCCESS".equals(finalStatus))

{

HttpSession session = request.getSession();

session.setAttribute("UpdateStatus", "Route Registered Successfully !!!");

ServletContext sc = request.getServletContext();

RequestDispatcher rd = sc.getRequestDispatcher("/admin.jsp");

rd.forward(request, response);

}

else if ("FAIL".equals(finalStatus))

{

HttpSession session = request.getSession();

session.setAttribute("UpdateStatus", "Route Could Not be Registered !!!");

ServletContext sc = request.getServletContext();

RequestDispatcher rd = sc.getRequestDispatcher("/admin.jsp");

rd.forward(request, response);

}

else

{

HttpSession session = request.getSession();

session.setAttribute("UpdateStatus", "Some Error Occured, Please Try again !!!");

ServletContext sc = request.getServletContext();

RequestDispatcher rd = sc.getRequestDispatcher("/admin.jsp");

rd.forward(request, response);

}

}

}

}

package com.ata.servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher; import javax.servlet.ServletContext; 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 javax.servlet.http.HttpSession;

import com.ata.bean.RouteBean;

import com.ata.service.AdministratorImpl;

@WebServlet("/RouteUpdateServlet")

public class RouteUpdateServlet extends HttpServlet {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

/\*\*

* @see HttpServlet#HttpServlet()

\*/

public RouteUpdateServlet() { super();

// TODO Auto-generated constructor stub

}

/\*\*

* @param request Function Parameter
* @param response Function Parameter
* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
* @throws ServletException Throws any Servlet Exception
* @throws IOException Throws any IO Exception

\*/

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

{

String rId =request.getParameter("rid");

String rSource=request.getParameter("source");

String rDestination=request.getParameter("destination");

String rDistance=request.getParameter("distance");

String rDuration=request.getParameter("duration");

Double routeDistance=Double.parseDouble(rDistance);

Double routeDuration=Double.parseDouble(rDuration);

RouteBean routeBean = new RouteBean();

routeBean.setRouteID(rId); routeBean.setSource(rSource); routeBean.setDestination(rDestination); routeBean.setDistance(routeDistance);

routeBean.setTravelDuration(routeDuration);

AdministratorImpl administratorImpl=new AdministratorImpl(); boolean updateStatus=false;

updateStatus=administratorImpl.modifyRoute(routeBean);

HttpSession session=request.getSession();

ServletContext scontext = request.getServletContext();

if(updateStatus==true)

{

session.setAttribute("UpdateStatus", "Route Updated Successfully !!!");

RequestDispatcher rd= scontext.getRequestDispatcher("/admin.jsp"); rd.forward(request, response);

}

else

{

session.setAttribute("UpdateStatus", "Route Could Not Be Updated !!!");

RequestDispatcher rd= scontext.getRequestDispatcher("/admin.jsp"); rd.forward(request, response); }

}

}

package com.ata.util;

import com.ata.bean.CredentialsBean;

public interface Authentication

{

/\*\*

* @param credentialsBean Function Parameter
* @return boolean

\*/

boolean authenticate(CredentialsBean credentialsBean);

/\*\*

* @param userID Function Parameter
* @return String

\*/

String authorize(String userID);

/\*\*

* @param credentialsBean Function Parameter
* @param loginStatus Function Parameter
* @return boolean

\*/

boolean changeLoginStatus(CredentialsBean credentialsBean, int loginStatus);

}

package com.ata.util;

import java.sql.SQLException;

import com.ata.bean.CredentialsBean; import com.ata.dao.CredentialDAOImpl;

public class AuthenticationImpl implements Authentication

{

@Override

public boolean authenticate(CredentialsBean credentialsBean)

{

boolean authenticateStatus=false;

CredentialDAOImpl credsDao = new CredentialDAOImpl(); try {

if(credentialsBean!=null)

{

authenticateStatus=credsDao.authenticate(credentialsBean);

}

else

{

authenticateStatus=false;

}

}

catch (SQLException e)

{

authenticateStatus=false; e.getMessage();

} catch (Exception e)

{

authenticateStatus=false; e.getMessage();

}

return authenticateStatus;

}

@Override

public String authorize(String userID)

{

boolean status=false;

String uType="FAIL";

CredentialDAOImpl credsDao = new CredentialDAOImpl();

try {

if(userID!=null)

{

uType=credsDao.authorize(userID);

status=true;

}

if(status==false)

{

uType="INVALID";

}

}

catch (SQLException e)

{

uType="ERROR";

e.getMessage();

}

catch (Exception e)

{

uType="ERROR";

e.getMessage();

}

return uType;

}

@Override

public boolean changeLoginStatus(CredentialsBean credentialsBean,int loginStatus)

{

boolean status=false;

CredentialDAOImpl credsDao = new CredentialDAOImpl();

try

{

if(credentialsBean!=null)

{

status=credsDao.changeLoginStatus(credentialsBean, loginStatus);

}

else

{

status=false;

}

}

catch (SQLException e)

{

status=false;

e.getMessage();

}

catch (Exception e)

{

status=false;

e.getMessage();

}

return status;

}

} package com.ata.util;

import java.sql.SQLException;

import com.ata.bean.CredentialsBean; import com.ata.bean.ProfileBean; import com.ata.dao.CredentialDAOImpl; import com.ata.dao.ProfileDAOImpl;

public class UserImpl implements User

{

@Override

public String login(CredentialsBean credentialsBean)

{

String aStatus="FAIL";

// Login Check

AuthenticationImpl authenticate = new AuthenticationImpl();

boolean authenticateStatus = authenticate.authenticate(credentialsBean);

if(credentialsBean!=null)

{

if(authenticateStatus==true)

{

aStatus=authenticate.authorize(credentialsBean.getUserID()); authenticate.changeLoginStatus(credentialsBean, 1);

}

else

{

aStatus="INVALID";

}

}

else

{

aStatus="FAIL";

}

return aStatus;

}

@Override

public boolean logout(String userId)

{

CredentialsBean cbean= new CredentialsBean(); cbean.setUserID(userId);

AuthenticationImpl authenticate = new AuthenticationImpl();

boolean status =authenticate.changeLoginStatus(cbean, 0); return status;

}

@Override

public String changePassword(CredentialsBean credentialsBean,String newPassword)

{

String status="FAIL";

CredentialDAOImpl credsDao = new CredentialDAOImpl();

try {

if(credentialsBean!=null)

{

status=credsDao.changePassword(credentialsBean, newPassword);

}

else

{

status="FAIL";

}

}

catch (Exception e)

{

status="ERROR"; e.getMessage();

}

return status;

}

@Override

public String register(ProfileBean profileBean)

{

String status="FAIL";

ProfileDAOImpl pDao = new ProfileDAOImpl();

try {

if(profileBean!=null)

{

status=pDao.register(profileBean);

}

else

{

status="FAIL";

}

}

catch (SQLException e)

{

status="ERROR"; e.getMessage();

}

catch (Exception e)

{

status="ERROR"; e.getMessage();

}

return status;

}

}

package com.ata.util;

import java.sql.SQLException;

import com.ata.bean.CredentialsBean; import com.ata.bean.ProfileBean; import com.ata.dao.CredentialDAOImpl; import com.ata.dao.ProfileDAOImpl;

public class UserImpl implements User

{

@Override

public String login(CredentialsBean credentialsBean)

{

String aStatus="FAIL";

// Login Check

AuthenticationImpl authenticate = new AuthenticationImpl();

boolean authenticateStatus = authenticate.authenticate(credentialsBean);

if(credentialsBean!=null)

{

if(authenticateStatus==true)

{

aStatus=authenticate.authorize(credentialsBean.getUserID()); authenticate.changeLoginStatus(credentialsBean, 1);

}

else

{

aStatus="INVALID";

}

} else

{ aStatus="FAIL";

}

return aStatus;

}

@Override

public boolean logout(String userId)

{

CredentialsBean cbean= new CredentialsBean();

cbean.setUserID(userId);

AuthenticationImpl authenticate = new AuthenticationImpl();

boolean status =authenticate.changeLoginStatus(cbean, 0); return status;

}

@Override

public String changePassword(CredentialsBean credentialsBean,String newPassword)

{

String status="FAIL";

CredentialDAOImpl credsDao = new CredentialDAOImpl();

try {

if(credentialsBean!=null)

{

status=credsDao.changePassword(credentialsBean, newPassword);

}

else

{

status="FAIL";

}

}

catch (Exception e)

{

status="ERROR"; e.getMessage();

}

return status;

}

@Override

public String register(ProfileBean profileBean)

{

String status="FAIL";

ProfileDAOImpl pDao = new ProfileDAOImpl(); try {

if(profileBean!=null)

{

status=pDao.register(profileBean);

} else

{

status="FAIL";

}

}

catch (SQLException e)

{

status="ERROR"; e.getMessage();

}

catch (Exception e)

{

status="ERROR"; e.getMessage();

}

return status;

}

}

DATABASE: create table ATA\_TBL\_USER\_credentials

( userid varchar(6) primary key, password varchar(20) not null, usertype varchar(1), loginstatus int

);

create table ATA\_TBL\_USER\_PROFILE

(

userid varchar(6) primary key references ata\_tbl\_user\_credentials(userid), firstname varchar(15) not null, lastname varchar(15) not null, dateofbirth date not null, gender varchar(7) not null, street varchar(30) not null, location varchar(15) not null, city varchar(15) not null, state varchar(15) not null, pincode varchar(10) not null, mobileno varchar(10), emailid varchar(30)

); create table ATA\_TBL\_CREDITCARD

( creditcardnumber varchar(16) primary key, validfrom varchar(7) not null, validto varchar(7) not null, creditbalance double not null,

userid varchar(6) references ata\_tbl\_user\_credentials(userid) );

create table ATA\_TBL\_vehicle

( vehicleid varchar(6) primary key, name varchar(20) not null, type varchar(8) not null,

registrationnumber varchar(20) not null, seatingcapacity int not null, fareperkm double

);

create table ATA\_TBL\_route

( routeid varchar(8) primary key , source varchar(20) not null, destination varchar(20) not null, distance double not null, travelduration double not null

); create table ATA\_TBL\_RESERVATION

(

reservationid varchar(6) primary key ,

userid varchar(6) references ata\_tbl\_user\_credentials(userid), vehicleid varchar(6) references ATA\_TBL\_vehicle(vehicleid), routeid varchar(8) references ATA\_TBL\_route(routeid), bookingdate date not null, journeydate date not null,

driverid varchar(6) references ATA\_TBL\_driver(driverid), bookingstatus varchar(20) not null, totalfare double not null, boardingpoint varchar(30) not null, droppoint varchar(30) not null

);

alter table ATA\_TBL\_USER\_credentials add column qus()varchar(20));

insert into ATA\_TBL\_CREDITCARD

values('abc1234567890def','02/2001','02/2015',1000000,'de1002'); alter table ATA\_TBL\_USER\_credentials add column qus varchar(20); alter table ATA\_TBL\_USER\_credentials add column ans varchar(20);

insert into ATA\_TBL\_CREDITCARD

values('abcdef1234567890','02/2000','02/2015',1000000,'as1001');

insert into ATA\_TBL\_USER\_credentials values('abhi', 'abhi', 'A',0); insert into ATA\_TBL\_Vehicle values('AB2345','MARUTI','AC','123456',5,25);

insert into ATA\_TBL\_CREDITCARD values

('1020304050607080','06/2010','05/2015',20000,'Ab1000'); insert into ATA\_TBL\_CREDITCARD values

('2030405060708090','01/2010','12/2014',20000,'Ab1001'); insert into ATA\_TBL\_CREDITCARD values

('3040506070809010','03/2009','02/2014',20000,'Ja1002'); insert into ATA\_TBL\_CREDITCARD values

('4050607080901020','06/2012','05/2016',20000,'Ki1003');

insert into ATA\_TBL\_ROUTE values('AB786','ASD','',21,2);