package com.extremityindia.defectmanagement.apps.servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.extremityindia.defectmanagement.apps.model.Registration;

import com.extremityindia.defectmanagement.apps.service.ILoginUserService;

import com.extremityindia.defectmanagement.apps.serviceImpl.LoginUserServiceImpl;

import com.extremityindia.defectmanagement.apps.utility.Constants;

public class LoginUserServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

@SuppressWarnings("unused")

private RequestDispatcher rd = null;

@Override

protected void service(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException {

Registration user = new Registration();

ILoginUserService userService = new LoginUserServiceImpl();

user.setUsername(request.getParameter("username"));

user.setPassword(request.getParameter("password"));

user = userService.loginUser(user);

if (user.getUsertype() != null && user.getUsername() != null) {

HttpSession session = request.getSession();

session.setAttribute("userId", user.getRegistrationId());

session.setAttribute("user", user.getUsername());

session.setAttribute("userType", user.getUsertype().getUsertype());

String name = user.getFirstname() + " " + user.getLastname();

session.setAttribute("name", name);

if (user.getUsertype().getUsertype().equals(Constants.ADMIN)) {

System.out.println("Admin view Controller");

response.sendRedirect("contain/adminHome");

} else if (user.getUsertype().getUsertype()

.equals(Constants.DEVELOPER)) {

/\*request.getRequestDispatcher("/contain/devloperHome").forward(

request, response);\*/

response.sendRedirect("contain/developerHome");

} else if (user.getUsertype().getUsertype()

.equals(Constants.MANAGER)) {

/\*request.getRequestDispatcher("manager/manager\_homepage.jsp")

.forward(request, response);\*/

response.sendRedirect("contain/managerHome");

} else if (user.getUsertype().getUsertype()

.equals(Constants.TESTER)) {

/\*request.getRequestDispatcher("tester/tester\_homepage.jsp")

.forward(request, response);\*/

response.sendRedirect("contain/testerHome");

} else {

request.setAttribute("msg",

"<font color=red><h6>Please, Email Address Not Exist</h6></font>");

request.getRequestDispatcher("/index.jsp").forward(request,

response);

}

} else {

request.setAttribute("msg",

"<font color=red><h6>Please, Enter Correct User Name and Password.</h6></font>");

request.getRequestDispatcher("index.jsp")

.forward(request, response);

}

}

}

**package** com.extremityindia.defectmanagement.apps.service;

**import** com.extremityindia.defectmanagement.apps.model.Registration;

**public** **interface** ILoginUserService {

**public** Registration loginUser(Registration user);

}

package com.extremityindia.defectmanagement.apps.serviceImpl;

import com.extremityindia.defectmanagement.apps.dao.ILoginUserDao;

import com.extremityindia.defectmanagement.apps.daoImpl.LoginUserDaoImpl;

import com.extremityindia.defectmanagement.apps.model.Registration;

import com.extremityindia.defectmanagement.apps.service.ILoginUserService;

public class LoginUserServiceImpl implements ILoginUserService {

ILoginUserDao loginUserDao=new LoginUserDaoImpl();

@Override

public Registration loginUser(Registration user) {

return loginUserDao.authenticateUser(user);

}

}

**package** com.extremityindia.defectmanagement.apps.dao;

**import** com.extremityindia.defectmanagement.apps.model.Registration;;

**public** **interface** ILoginUserDao {

**public** Registration authenticateUser(Registration user);

}

package com.extremityindia.defectmanagement.apps.daoImpl;

import java.sql.ResultSet;

import com.extremityindia.defectmanagement.apps.dao.ILoginUserDao;

import com.extremityindia.defectmanagement.apps.model.Registration;

import com.extremityindia.defectmanagement.apps.model.Usertype;

import com.extremityindia.defectmanagement.apps.utility.JdbcUtility;

public class LoginUserDaoImpl implements ILoginUserDao {

/\*

\* check user authenticate or not if user authorized then its login

\* successfully else user unauthorized its give an error

\*/

@Override

public Registration authenticateUser(Registration user) {

try {

String status = "Active";

Usertype usertype = new Usertype();

String sql = "SELECT r.REGISTRATION\_ID,r.USERNAME, r.PASSWORD, u.USERTYPE, r.FIRSTNAME, r.LASTNAME "

+ "FROM registration r, usertype u "

+ "WHERE r.USERNAME='"

+ user.getUsername()

+ "' AND r.PASSWORD='"

+ user.getPassword()

+ "' AND r.USERID = u.USERID AND r.STATUS='"

+ status

+ "';";

System.out.println("sql : " + sql);

ResultSet resultSet = JdbcUtility.getResultSet(sql);

while (resultSet.next()) {

user = new Registration();

usertype.setUsertype(resultSet.getString(4));

user.setRegistrationId(resultSet.getInt(1));

user.setUsername(resultSet.getString(2));

user.setUsertype(usertype);

user.setFirstname(resultSet.getString(5));

user.setLastname(resultSet.getString(6));

}

} catch (Exception e) {

e.printStackTrace();

} finally {

JdbcUtility.closeConnection();

}

return user;

}

}package com.extremityindia.defectmanagement.apps.filter;

import java.io.IOException;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import javax.servlet.annotation.WebFilter;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

/\*\*

\* Authentication filter for login purpose

\*

\*/

@WebFilter("/AuthenticationFilter")

public class AuthenticationFilter implements Filter {

@Override

public void destroy() {

System.out.println("User Destroy");

}

@Override

public void doFilter(ServletRequest request, ServletResponse response,

FilterChain chain) throws IOException, ServletException {

HttpServletRequest req = (HttpServletRequest) request;

HttpSession session = req.getSession(false);

System.out.println("---------------SESSION-----------------" + session);

if (session == null) {

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

rd.forward(request, response);

} else {

chain.doFilter(request, response);

}

}

@Override

public void init(FilterConfig arg0) throws ServletException {

System.out.println("Check User Authenticate Or Not");

}

}package com.extremityindia.defectmanagement.apps.filter;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import javax.servlet.annotation.WebFilter;

import javax.servlet.http.HttpServletRequest;

import org.apache.log4j.Logger;

import org.apache.log4j.PropertyConfigurator;

/\*\*

\* Servlet Filter implementation class RequestLoggingFilter

\*/

@WebFilter("/RequestLoggingFilter")

public class RequestLoggingFilter implements Filter {

private FilterConfig config;

private Logger log;

public void init(FilterConfig filterConfig) throws ServletException {

this.config = filterConfig;

//load the configuration for this application's loggers using the

// servletLog.properties file

PropertyConfigurator.configure(config.getServletContext().getRealPath(

"/")

+ "/WEB-INF/log.properties");

log = Logger.getLogger(RequestLoggingFilter.class);

log.info("Logger instantiated in " + getClass().getName());

}

public void doFilter(ServletRequest request, ServletResponse response,

FilterChain chain) throws java.io.IOException, ServletException {

HttpServletRequest req = null;

long startTime;

long endTime;

String id = config.getInitParameter("log-id");

if (id == null)

id = "unknown";

if (log != null && (request instanceof HttpServletRequest)) {

req = (HttpServletRequest) request;

startTime = System.currentTimeMillis();

chain.doFilter(request, response);

endTime = System.currentTimeMillis();

log.info("Log id:" + id + ": Request received from: "

+System.currentTimeMillis()+ " for " + req.getRequestURL());

//Log the servlet path and time taken

log.info("Log id:" + id + ": Request END: "

+(endTime - startTime) + " for " + req.getRequestURL());

}

else

chain.doFilter(request, response);

}// doFilter

public void destroy() {

/\*

\* called before the Filter instance is removed from service by the web

\* container

\*/

log = null;

}

}package com.extremityindia.defectmanagement.apps.model;

import java.util.HashSet;

import java.util.Set;

public class Registration implements java.io.Serializable {

// Fields

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private Integer registrationId;

private Usertype usertype;

private String firstname;

private String lastname;

private String contact;

private String email;

private String username;

private String password;

private String status;

private Set projects = new HashSet(0);

private Set bugassigneds = new HashSet(0);

private Set bugses = new HashSet(0);

private Set systemlogins = new HashSet(0);

private Set projectMemberses = new HashSet(0);

// Constructors

/\*\* default constructor \*/

public Registration() {

}

/\*\* minimal constructor \*/

public Registration(Integer registrationId, Usertype usertype, String firstname, String lastname, String contact, String email, String username, String password, String status) {

this.registrationId = registrationId;

this.usertype = usertype;

this.firstname = firstname;

this.lastname = lastname;

this.contact = contact;

this.email = email;

this.username = username;

this.password = password;

this.status = status;

}

/\*\* full constructor \*/

public Registration(Integer registrationId, Usertype usertype, String firstname, String lastname, String contact, String email, String username, String password, String status, Set projects, Set bugassigneds, Set bugses, Set systemlogins, Set projectMemberses) {

this.registrationId = registrationId;

this.usertype = usertype;

this.firstname = firstname;

this.lastname = lastname;

this.contact = contact;

this.email = email;

this.username = username;

this.password = password;

this.status = status;

this.projects = projects;

this.bugassigneds = bugassigneds;

this.bugses = bugses;

this.systemlogins = systemlogins;

this.projectMemberses = projectMemberses;

}

// Property accessors

public Integer getRegistrationId() {

return this.registrationId;

}

public void setRegistrationId(Integer registrationId) {

this.registrationId = registrationId;

}

public Usertype getUsertype() {

return this.usertype;

}

public void setUsertype(Usertype usertype) {

this.usertype = usertype;

}

public String getFirstname() {

return this.firstname;

}

public void setFirstname(String firstname) {

this.firstname = firstname;

}

public String getLastname() {

return this.lastname;

}

public void setLastname(String lastname) {

this.lastname = lastname;

}

public String getContact() {

return this.contact;

}

public void setContact(String contact) {

this.contact = contact;

}

public String getEmail() {

return this.email;

}

public void setEmail(String email) {

this.email = email;

}

public String getUsername() {

return this.username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return this.password;

}

public void setPassword(String password) {

this.password = password;

}

public String getStatus() {

return this.status;

}

public void setStatus(String status) {

this.status = status;

}

public Set getProjects() {

return this.projects;

}

public void setProjects(Set projects) {

this.projects = projects;

}

public Set getBugassigneds() {

return this.bugassigneds;

}

public void setBugassigneds(Set bugassigneds) {

this.bugassigneds = bugassigneds;

}

public Set getBugses() {

return this.bugses;

}

public void setBugses(Set bugses) {

this.bugses = bugses;

}

public Set getSystemlogins() {

return this.systemlogins;

}

public void setSystemlogins(Set systemlogins) {

this.systemlogins = systemlogins;

}

public Set getProjectMemberses() {

return this.projectMemberses;

}

public void setProjectMemberses(Set projectMemberses) {

this.projectMemberses = projectMemberses;

}

}

package com.extremityindia.defectmanagement.apps.utility;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class JdbcUtility {

private static Connection con;

private static final String Driver = "com.mysql.jdbc.Driver";

private static final String URL = "jdbc:mysql://localhost:3306/defecttracking";

private static final String user = "root";

private static final String pwd = "root";

/\*\*

\* create Database object

\*/

public JdbcUtility() {

}

/\*\*

\* to load the database base driver

\*

\* @return a database connection

\* @throws SQLException

\* throws an exception if an error occurs

\*/

public static Connection loadDriver() throws SQLException {

try {

Class.forName(Driver);

} catch (ClassNotFoundException ex) {

System.out.println(ex.getMessage());

}

con = DriverManager.getConnection(URL, user, pwd);

return con;

}

/\*\*

\* to get a result set of a query

\*

\* @param query

\* custom query

\* @return a result set of custom query

\* @throws SQLException

\* throws an exception if an error occurs

\*/

public static ResultSet getResultSet(String query) throws SQLException {

Connection con = loadDriver();

ResultSet rs;

PreparedStatement st = con.prepareStatement(query);

rs = st.executeQuery();

return rs;

}

/\*\*

\* to run an update query such as update, delete

\*

\* @param query

\* custom query

\* @return

\* @throws SQLException

\* throws an exception if an error occurs

\*/

public static Integer runQuery(String query) throws SQLException {

Connection con = loadDriver();

PreparedStatement st = con.prepareStatement(query);

Integer id = st.executeUpdate();

return id;

}

public static void closeConnection() {

try {

con.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}**package** com.extremityindia.defectmanagement.apps.utility;

**public** **class** Constants {

**public** **static** **final** String ***ADMIN*** = "Admin";

**public** **static** **final** String ***DEVELOPER*** = "Developer";

**public** **static** **final** String ***MANAGER*** = "Manager";

**public** **static** **final** String ***TESTER*** = "Tester";

**public** **static** **final** String ***ACTIVE*** = "Active";

**public** **static** **final** String ***DEACTIVE*** = "Deactive";

}

<%@include file=*"common\_webpages/header\_webpages/front\_header.jsp"*%>

<link rel=*"stylesheet"*

href=*"//code.jquery.com/ui/1.11.4/themes/smoothness/jquery-ui.css"*>

<script src=*"//code.jquery.com/jquery-1.10.2.js"*></script>

<script src=*"//code.jquery.com/ui/1.11.4/jquery-ui.js"*></script>

<script type=*"text/javascript"*>

//for textboxes

$(**function**() {

$("#dailogbox").dialog({

autoOpen : **false**,

show : {

effect : "blind",

duration : 1000

},

hide : {

effect : "explode",

duration : 1000

}

});

});

<%String msg = (String) request.getAttribute("msg");

System.out.println("msg : "+msg);

**if** (msg != **null**) {%>

$("#dailogbox1").dialog("open");

<%}%>

</script>

<%

String message = (String)request.getAttribute("message");

System.out.println("message : "+message);

%>

<div class=*"modal small fade"* id=*"myModal"* tabindex=*"-1"* role=*"dialog"*

aria-labelledby=*"myModalLabel"* aria-hidden=*"true"*>

<div class=*"modal-dialog"* id=*"dailogbox"* title=*"Warning !!!!"*>

<div class=*"modal-content"*>

<div class=*"modal-body"*>

<p class=*"error-text"*>

<i class=*"fa fa-warning modal-icon"*></i><%=msg%>

</p>

</div>

</div>

</div>

</div>

<div class=*"dialog"*>

<div class=*"panel panel-default"*>

<p class=*"panel-heading no-collapse"*>Sign In</p>

<div class=*"panel-body"*>

<form action=*"*${pageContext.request.contextPath}*/login"* method=*"post"*>

<div class=*"form-group"*>

<label>Username</label> <input required type=*"text"*

class=*"form-control span12"* name=*"username"*>

</div>

<div class=*"form-group"*>

<label>Password</label> <input required type=*"password"*

class=*"form-control\_span12 form-control"* name=*"password"*>

</div>

<input type=*"submit"* class=*"btn btn-primary pull-right"*

value=*"Sign In"*> <label class=*"remember-me"*><input

type=*"checkbox"*> Remember me</label>

<div class=*"clearfix"*></div>

</form>

</div>

</div>

<p>

<a href=*"resetpassword"*>Forgot your password?</a>

</p>

</div>

<%@include file=*"common\_webpages/footer\_webpages/footer.jsp"*%><%@include file=*"/common\_webpages/header\_webpages/front\_header.jsp"*%>

<link rel=*"stylesheet"*

href=*"//code.jquery.com/ui/1.11.4/themes/smoothness/jquery-ui.css"*>

<script src=*"//code.jquery.com/jquery-1.10.2.js"*></script>

<script src=*"//code.jquery.com/ui/1.11.4/jquery-ui.js"*></script>

<script type=*"text/javascript"*>

//for textboxes

$(**function**() {

$("#dailogbox").dialog({

autoOpen : **false**,

show : {

effect : "blind",

duration : 1000

},

hide : {

effect : "explode",

duration : 1000

}

});

});

//for contact

$(**function**() {

$("#dailogbox1").dialog({

autoOpen : **false**,

show : {

effect : "blind",

duration : 1000

},

hide : {

effect : "explode",

duration : 1000

}

});

});

**function** validate() {

**var** firstname = document.register.firstname.value;

**var** lastname = document.register.lastname.value;

**var** contact = document.register.contact.value;

**var** email = document.register.email.value;

**var** username = document.register.username.value;

**var** password = document.register.password.value;

**var** usertype = document.register.usertype.value;

alert(usertype);

**if** (firstname == "") {

$("#dailogbox").dialog("open");

**return** **false**;

}

**if** (lastname == "") {

$("#dailogbox").dialog("open");

**return** **false**;

}

**if** (contact == "") {

$("#dailogbox").dialog("open");

**return** **false**;

}

**if** (email == "") {

$("#dailogbox").dialog("open");

**return** **false**;

}

**if** (password == "") {

$("#dailogbox").dialog("open");

**return** **false**;

}

**return** **true**;

}

**var** specialKeys = **new** Array();

specialKeys.push(8);

**function** numberValidate(e) {

**var** keyCode = e.which ? e.which : e.keyCode;

**var** ret = ((keyCode >= 48 && keyCode <= 57) || specialKeys

.indexOf(keyCode) != -1);

**var** isnumber = ret ? "true " : "false";

**if** (isnumber.indexOf("true") > -1) {

**return** **true**;

} **else** {

$("#dailogbox1").dialog("open");

e.preventDefault();

**return** **false**;

}

}

</script>

<!-- ALERT FOR CONTACT -->

<div class=*"modal small fade"* id=*"myModal"* tabindex=*"-1"* role=*"dialog"*

aria-labelledby=*"myModalLabel"* aria-hidden=*"true"*>

<div class=*"modal-dialog"* id=*"dailogbox1"* title=*"Warning !!!!"*>

<div class=*"modal-content"*>

<div class=*"modal-body"*>

<p class=*"error-text"*>

<i class=*"fa fa-warning modal-icon"*></i>This cannot be undone.

</p>

</div>

</div>

</div>

</div>

<!-- ALERT FOR TEXTBOXES -->

<div class=*"modal small fade"* id=*"myModal"* tabindex=*"-1"* role=*"dialog"*

aria-labelledby=*"myModalLabel"* aria-hidden=*"true"*>

<div class=*"modal-dialog"* id=*"dailogbox"* title=*"Warning !!!!"*>

<div class=*"modal-content"*>

<div class=*"modal-body"*>

<p class=*"error-text"*>

<i class=*"fa fa-warning modal-icon"*></i>Please fill all the fields

</p>

</div>

</div>

</div>

</div>

<div class=*"dialog"*>

<div class=*"panel panel-default"*>

<p class=*"panel-heading no-collapse"*>Sign Up</p>

<div class=*"panel-body"*>

<form name=*"register"* action=*"*${pageContext.request.contextPath}*/register"* method=*"post"*

onsubmit="return validate()">

<div class=*"form-group"*>

<label>First Name</label> <input type=*"text"*

class=*"form-control span12"* name=*"firstname"*>

</div>

<div class=*"form-group"*>

<label>Last Name</label> <input type=*"text"*

class=*"form-control span12"* name=*"lastname"*>

</div>

<div class=*"form-group"*>

<label>Contact</label> <input type=*"text"*

class=*"form-control span12"* name=*"contact"*

onkeypress="numberValidate(event)" pattern=*"[789][0-9]{9}"*

maxlength=*"10"*>

</div>

<div class=*"form-group"*>

<label>User Type</label> <select class=*"form-control span12"*

name=*"usertype"*>

<option value=*"0"*>Select</option>

<option value=*"2"*>Manager</option>

<option value=*"3"*>Developer</option>

<option value=*"4"*>Tester</option>

</select>

</div>

<div class=*"form-group"*>

<label>Email Address</label> <input type=*"email"*

class=*"form-control span12"* name=*"email"*>

</div>

<div class=*"form-group"*>

<label>Username</label> <input type=*"text"*

class=*"form-control span12"* name=*"username"*>

</div>

<div class=*"form-group"*>

<label>Password</label> <input type=*"password"*

class=*"form-control span12"* name=*"password"*>

</div>

<div class=*"form-group"*>

<label class=*"remember-me"*><input type=*"checkbox"*>

I agree with the <a href=*"terms-and-conditions.html"*>Terms and

Conditions</a></label> <input type=*"submit"*

class=*"btn btn-primary pull-right"* value=*"Sign Up"*>

</div>

<div class=*"clearfix"*></div>

</form>

</div>

</div>

<p>

<a href=*"privacy-policy.html"*

style="font-size: *.75em*; margin-top: *.25em*;">Privacy Policy</a>

</p>

</div>

<%@include file=*"/common\_webpages/footer\_webpages/footer.jsp"*%>