|  |  |  |  |
| --- | --- | --- | --- |
| **Ex. No. 10** | **DESIGNING INNOVATIVE ENTERPRISE APPLICATION – LIVE APTITUDE TEST** | | |
| Date of Exercise |  | Date of Output Verification |  |

**Aim**

To develop and deploy the enterprise innovative application for live aptitude test using J2EE programming.

**Description and Procedure**

* This is an Online Web Application which uses MVC pattern and very dynamic content for the user interface display which handles everything in just one page.
* This Application allows Audience to check the top ‘n’ scorers
* Users can register and can participate in various tests
* He will be displayed with the companies available and the history of the tests he has taken
* Admin can add questions
* Users can take tests and the scores will be updated automatically
* The top 3 scorers are live
* The test will provide ‘n’ number of random questions from the selected company
* Server side error Checking and Client side error checking
* Restriction of access to admin pages through URL

**Technologies Used**

* Jsp, HTML, CSS3, Bootstrap
* Angularjs, jQuery, JavaScript
* Servlet, Session Bean, Session bean for entity class, Entity Class from database, jdbc:derby

**Program**

Login.jsp

<script>

$(document).ready(function () {

$("#signupform").hide();

$("#errormessage").hide();

$("#signinclick").click(function () {

$("#signupform").slideUp();

$("#signinform").slideDown();

});

$("#signupclick").click(function () {

$("#signinform").slideUp();

$("#signupform").slideDown();

});

});

</script>

<div class='container alert alert-danger col-sm-12 ws-hide' id="errormessage">Error, Try Again!....</div>

<div id="loginbody">

<ul id="tabs">

<li id="signinclick">SignIn</li>

<li id="signupclick">SignUp</li>

</ul>

<form id="signinform" class="form-horizontal" ng-submit="signin()">

<p class="heading">Login</p>

<div class="form-group">

<label for="siuname" class="control-label col-sm-3">Username</label>

<div class="col-sm-8">

<input type="text" id="siuname" ng-model="signinData.username" placeholder="Enter Username" class="form-control" required>

</div>

</div>

<div class="form-group">

<label for="sipassword" class="control-label col-sm-3">Password</label>

<div class="col-sm-8">

<input type="password" id="sipassword" ng-model="signinData.password" placeholder="Enter Password" class="form-control" required>

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-8 col-xs-offset-1">

<input type="submit" value="SignIn" class="btn btn-default" />

</div>

</div>

</form>

<form id="signupform" class="form-horizontal ws-hide" ng-submit="signup()">

<p class="heading">Sign Up</p>

<div class="form-group">

<label for="fullname" class="control-label col-sm-3">Full Name</label>

<div class="col-sm-8">

<input type="text" id="fullname" ng-model="signupData.fullname" placeholder="Enter Full Name" class="form-control" required>

</div>

</div>

<div class="form-group">

<label for="username" class="control-label col-sm-3">Username</label>

<div class="col-sm-8">

<input type="text" id="username" ng-model="signupData.username" placeholder="Enter Username" class="form-control" required>

</div>

</div>

<div class="form-group">

<label for="pemail" class="control-label col-sm-3">Email</label>

<div class="col-sm-8">

<input type="email" id="pemail" ng-model="signupData.email" placeholder="Enter Email" class="form-control" required>

</div>

</div>

<div class="form-group">

<label for="dob" class="control-label col-sm-3">Date of Birth</label>

<div class="col-sm-8">

<input type="date" id="dob" ng-model="signupData.dob" id="signup\_dob" class="form-control">

</div>

</div>

<div class="form-group">

<label for="password" class="control-label col-sm-3">Password</label>

<div class="col-sm-8">

<input type="password" id="password" ng-model="signupData.password" placeholder="Enter Password" class="form-control" required>

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-8">

<input type="submit" value="SignUp" class="btn btn-default" />

</div>

</div>

</form>

</div>

<script>

var app = angular.module('myApp', []);

app.controller('loginControl', function ($scope, $http) {

$scope.signin = function () {

$("#errormessage").slideUp();

$http({

method: 'POST',

url: 'JCAction?Signin=true',

headers: {'Content-Type': 'application/json'},

data: $scope.signinData

}).success(function (data) {

if (data == "false")

$("#errormessage").slideDown();

else

window.location.href = 'index.jsp';

});

};

$scope.signup = function () {

$("#errormessage").slideUp();

$http({

method: 'POST',

url: 'JCAction?Signup=true',

headers: {'Content-Type': 'application/json'},

data: $scope.signupData

}).success(function (data) {

if (data == "false")

$("#errormessage").slideDown();

else

window.location.href = 'index.jsp';

});

};

});

</script>

Index.jsp

<%

if (session.getAttribute("name") == null) {

%>

<script>

var app = angular.module('myApp', []);

app.controller('indexControl', function ($scope, $http, $interval) {

$interval(function () {

$scope.gettop();

}, 1000);

$scope.gettop = function () {

$http.get("JCAction?GetTop=true")

.success(function (response) {

$scope.topscorers = response;

});

};

$scope.gettop();

});

</script>

<div class='container alert alert-info col-sm-6 col-sm-offset-3'>Kindly Login to Participate</div>

<%

} else {

%>

<script>

var app = angular.module('myApp', []);

app.controller('indexControl', function ($scope, $http, $interval) {

$scope.questions = [{}];

$interval(function () {

$scope.gettop();

}, 10000);

$scope.attempt = function () {

$http({

method: 'POST',

url: 'JCAction?AttemptTest=true',

headers: {'Content-Type': 'application/json'},

data: $scope.selectcompay

}).success(function (response) {

$scope.questions = response;

$("#test").slideDown();

});

};

$scope.gettop = function () {

$http.get("JCAction?GetTop=true")

.success(function (response) {

$scope.topscorers = response;

});

};

$scope.getscore = function () {

$http.get("JCAction?GetScore=true")

.success(function (response) {

$scope.score = response;

});

};

$scope.gethistory = function () {

$http.get("JCAction?GetHistory=true")

.success(function (response) {

$scope.history = response;

});

};

$http.get("JCAction?GetCompanies=true")

.success(function (response) {

$scope.company = response;

});

$scope.gettop();

$scope.gethistory();

$scope.getscore();

});

function submit() {

$("#errormessage").slideUp();

$("#successmessage").slideUp();

$.ajax({

type: "post",

url: "JCAction?SubmitTest=true",

data: $("#submittest").serialize(),

success: function (data) {

if (data == "true") {

$("#test").slideUp();

$("#successmessage").slideDown();

angular.element(document.getElementById('index')).scope().gethistory();

angular.element(document.getElementById('index')).scope().getscore();

angular.element(document.getElementById('index')).scope().$apply();

}

else

$("#errormessage").slideDown();

}

});

}

$(document).ready(function () {

$("#test").slideUp();

});

</script>

<%

if (session.getAttribute("name").toString().equals("William Scott")) {

%>

<a href="admin\_modify.jsp" class="btn btn-success col-sm-offset-10"> Modify Database </a>

<%

}

}

%>

<div class='container alert alert-danger col-sm-6 col-sm-offset-3 ws-hide' id="errormessage">Error, Please Contact William!....</div>

<div class='container alert alert-success col-sm-6 col-sm-offset-3 ws-hide' id="successmessage">Operation Successfully!....</div>

<div class="panel col-sm-6 col-sm-offset-3 panel-default">

<div class="panel-heading panel-success">Top Scorers</div>

<div class="panel-body">

<table class="table table-striped">

<thead>

<tr>

<th>Name</th>

<th>Email</th>

<th>Score</th>

</tr>

</thead>

<tbody>

<tr ng-repeat="x in topscorers| filter:query | orderBy:orderByField:reverseSort">

<td>{{ x.name}}</td>

<td>{{ x.email}}</td>

<td>{{ x.score}}</td>

</tr>

</tbody>

</table>

</div>

</div>

<%

if (session.getAttribute("name") != null) {

%>

<div class="panel col-sm-10 col-sm-offset-1 panel-default ws-" id="test">

<div class="panel-heading panel-success">Test</div>

<div class="panel-body">

<form class="form-horizontal" role="form" style="overflow: hidden;" id="submittest">

<table class="table table-striped">

<thead>

<tr>

<th>Question</th>

<th>Option 1</th>

<th>Option 2</th>

<th>Option 3</th>

<th>Option 4</th>

</tr>

</thead>

<tbody>

<tr ng-repeat="x in questions| filter:query | orderBy:orderByField:reverseSort">

<td>{{ x.question}}</td>

<td><input type="radio" name="{{x.createdon}}" value="1" required=""/>{{ x.option1}}</td>

<td><input type="radio" name="{{x.createdon}}" value="2" required=""/>{{ x.option2}}</td>

<td><input type="radio" name="{{x.createdon}}" value="3" required=""/>{{ x.option3}}</td>

<td><input type="radio" name="{{x.createdon}}" value="4" required=""/>{{ x.option4}}</td>

</tr>

<tr>

<td></td>

<td></td>

<td></td>

<td></td>

<td><a class="btn btn-primary" onclick="submit()">Submit</a></td>

</tr>

</tbody>

</table>

</form>

</div>

</div>

<div class="panel col-sm-6 col-sm-offset-3 panel-default">

<div class="panel-heading panel-success">Test History</div>

<div class="panel-body">

<form class="form-horizontal" role="form" style="overflow: hidden;">

<table class="table table-striped">

<thead>

<tr>

<th>Attended on</th>

<th>Company Name</th>

<th>Score (%)</th>

</tr>

</thead>

<tbody>

<tr ng-repeat="x in history| filter:query | orderBy:orderByField:reverseSort">

<td>{{ x.createdon}}</td>

<td>{{ x.companyname}}</td>

<td>{{ x.score}}</td>

</tr>

</tbody>

</table>

</form>

</div>

</div>

<div class="panel col-sm-6 col-sm-offset-3 panel-default">

<div class="panel-heading panel-success">Attempt Test</div>

<div class="panel-body">

<form class="form-horizontal" role="form" style="overflow: hidden;" ng-submit="attempt()">

<table class="table table-striped">

<thead>

<tr>

<th>Company Name</th>

<th>Select</th>

</tr>

</thead>

<tbody>

<tr ng-repeat="x in company| filter:query | orderBy:orderByField:reverseSort">

<td>{{ x.company}}</td>

<td><input type="radio" name="select" ng-model="$parent.selectcompay.data" value="{{x.company}}" required=""/></td>

</tr>

<tr>

<td></td>

<%

if (session.getAttribute("name") != null) {

%>

<td><input type="submit" class="btn btn-success" value="Attempt Test"/></td>

<%

} else {

%>

<td><a href="login.jsp" class="btn btn-primary">Signin</a></td>

<%

}

%>

</tr>

</tbody>

</table>

</form>

</div>

</div>

<%

}

%>

Admin.jsp

<%@page import="java.sql.DriverManager"%>

<%@page import="java.sql.Statement"%>

<%@page import="java.sql.Connection"%>

<%

if (session.getAttribute("name") == null) {

%>

<script>

window.location.href = "index.jsp";

</script>

<%

} else if (!session.getAttribute("name").toString().equals("William Scott")) {

%>

<script>

window.location.href = "index.jsp";

</script>

<%

}

%>

<div class='container alert alert-danger col-sm-6 col-sm-offset-3 ws-hide' id="errormessage">Error, Please Contact William!....</div>

<div class='container alert alert-success col-sm-6 col-sm-offset-3 ws-hide' id="successmessage">Data Added Successfully!....</div>

<div class="panel col-sm-6 col-sm-offset-3 panel-default">

<div class="panel-heading panel-success">Add / Update Trains</div>

<div class="panel-body">

<form class="form-horizontal" role="form" style="overflow: hidden;" ng-submit="add()">

<div class="form-group">

<label class="control-label col-sm-3" for="q">Question</label>

<div class="col-sm-7">

<input type="text" class="form-control" ng-model="qData.question" id="q" placeholder="Enter Question">

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-3" for="o1">Option 1</label>

<div class="col-sm-7">

<input type="text" class="form-control" ng-model="qData.option1" id="o1" placeholder="Enter Option 1">

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-3" for="o2">Option 2</label>

<div class="col-sm-7">

<input type="text" class="form-control" ng-model="qData.option2" id="o2" placeholder="Enter Option 2">

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-3" for="o3">Option 3</label>

<div class="col-sm-7">

<input type="text" class="form-control" ng-model="qData.option3" id="o3" placeholder="Enter Option 3">

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-3" for="o4">Option 4</label>

<div class="col-sm-7">

<input type="text" class="form-control" ng-model="qData.option4" id="o4" placeholder="Enter Option 4">

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-3" for="answer">Answer</label>

<div class="col-sm-7">

<select class="form-control" id="answer" ng-model="qData.answer" >

<option>1</option>

<option>2</option>

<option>3</option>

<option>4</option>

</select>

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-3" for="company">Company</label>

<div class="col-sm-7">

<input type="text" class="form-control" ng-model="qData.company" id="company" placeholder="Enter Company">

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-8 col-sm-10">

<input type="submit" value="Add Question" class="btn btn-default" />

</div>

</div>

</form>

</div>

</div>

<script>

var app = angular.module('myApp', []);

app.controller('adminControl', function ($scope, $http) {

$scope.add = function () {

$("#errormessage").slideUp();

$("#successmessage").slideUp();

$http({

method: 'POST',

url: 'JCAction?Add=true',

headers: {'Content-Type': 'application/json'},

data: $scope.qData

}).success(function (data) {

if (data == "false")

$("#errormessage").slideDown();

else

$("#successmessage").slideDown();

});

};

});

</script>

JCAction.java

package com;

import java.io.IOException;

import java.io.PrintWriter;

import javax.ejb.EJB;

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 org.json.simple.JSONObject;

public class JCAction extends HttpServlet {

@EJB

private SBBusinessLogicLocal sBBusinessLogic;

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

response.setCharacterEncoding("utf-8");

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

HttpSession session = request.getSession();

int questionscount=5, topcount=3;

try {

String returndata;

if (request.getParameter("Signin") != null) {

JSONObject json = WSConvert.readerToJSON(request.getReader());

System.out.println("Signin: " + json);

if (json != null) {

returndata = sBBusinessLogic.signIn((String) json.get("username"), (String) json.get("password"));

if (!"false".equals(returndata)) {

session.setAttribute("name", returndata);

session.setAttribute("username", json.get("username"));

} else {

out.write("false");

}

}

}

if (request.getParameter("Signup") != null) {

JSONObject json = WSConvert.readerToJSON(request.getReader());

System.out.println("Signup: " + json);

if (json != null) {

returndata = sBBusinessLogic.signUp((String) json.get("fullname"), (String) json.get("username"), (String) json.get("email"), (String) json.get("dob"), (String) json.get("password"));

if (!"false".equals(returndata)) {

session.setAttribute("name", returndata);

session.setAttribute("username", json.get("username"));

} else {

out.write("false");

}

}

}

if (request.getParameter("Add") != null) {

JSONObject json = WSConvert.readerToJSON(request.getReader());

System.out.println("Add: " + json);

if (json != null && session.getAttribute("username").equals("williamscott701")) {

returndata = sBBusinessLogic.addQuestion((String) json.get("question"), (String) json.get("option1"), (String) json.get("option2"), (String) json.get("option3"), (String) json.get("option4"), (String) json.get("answer"), (String) json.get("company"));

out.write(returndata);

} else {

out.write("false");

}

}

if (request.getParameter("AttemptTest") != null) {

JSONObject json = WSConvert.readerToJSON(request.getReader());

System.out.println("AttemptTest: " + json);

if (json != null && session.getAttribute("username") != null) {

out.print(sBBusinessLogic.getQuestions((String) json.get("data"), questionscount));

} else {

out.write("false");

}

}

if (request.getParameter("SubmitTest") != null) {

if (session.getAttribute("username") != null) {

out.print(sBBusinessLogic.submitTest(request.getParameterMap(), questionscount, (String) session.getAttribute("username")));

} else {

out.write("false");

}

}

if (request.getParameter("GetScore") != null) {

out.write(sBBusinessLogic.getScore((String) session.getAttribute("username")));

}

if (request.getParameter("GetHistory") != null) {

out.print(sBBusinessLogic.getHistory((String) session.getAttribute("username")));

}

if (request.getParameter("GetTop") != null) {

out.print(sBBusinessLogic.getTop(topcount));

}

if (request.getParameter("GetCompanies") != null) {

out.print(sBBusinessLogic.getCompany());

}

} catch (Exception e) {

System.out.println(e);

out.write("false");

}

}

@Override

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

processRequest(request, response);

}

@Override

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

processRequest(request, response);

}

}

SBBusinessLogic.java

package com;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.Date;

import java.util.Map;

import javax.ejb.EJB;

import javax.ejb.Stateless;

import org.json.simple.JSONArray;

@Stateless

public class SBBusinessLogic implements SBBusinessLogicLocal {

@EJB

private UsersFacadeLocal usersFacade;

@EJB

private TesthistoryFacadeLocal testhistoryFacade;

@EJB

private QuestionsFacadeLocal questionsFacade;

public String comp;

@Override

public String signIn(String username, String password) {

if (usersFacade.find(username) == null) {

return "false";

} else {

Users r = usersFacade.find(username);

if (r.getPassword().equals(password.hashCode() + "")) {

return r.getName();

} else {

return "false";

}

}

}

@Override

public String signUp(String name, String username, String email, String dob, String password) {

try {

Users u = new Users(username, name, email, dob, password.hashCode() + "");

u.setScore(10);

usersFacade.create(u);

} catch (Exception e) {

return "false";

}

return name;

}

@Override

public String addQuestion(String question, String option1, String option2, String option3, String option4, String answer, String company) {

try {

Questions q = new Questions(new Date().toString(), question, option1, option2, option3, option4, answer, company);

questionsFacade.create(q);

} catch (Exception e) {

return "false";

}

return "true";

}

@Override

public String getScore(String username) {

try {

return usersFacade.find(username).getScore() + "";

} catch (Exception e) {

return "false";

}

}

@Override

public JSONArray getTop(int n) {

try {

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/j2eeexp", "ws", "6477");

String query = "SELECT \"NAME\", EMAIL, SCORE FROM WS.USERS ORDER BY SCORE DESC";

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(query);

return WSConvert.convertResultSetIntoJSON(rs, n);

} catch (Exception e) {

return null;

}

}

@Override

public JSONArray getHistory(String username) {

try {

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/j2eeexp", "ws", "6477");

String query = "SELECT \"CREATEDON\", COMPANYNAME, SCORE FROM WS.TESTHISTORY where username = '" + username + "'";

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(query);

return WSConvert.convertResultSetIntoJSON(rs);

} catch (Exception e) {

return null;

}

}

@Override

public JSONArray getCompany() {

try {

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/j2eeexp", "ws", "6477");

String query = "select distinct COMPANY from WS.QUESTIONS";

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(query);

return WSConvert.convertResultSetIntoJSON(rs);

} catch (Exception e) {

return null;

}

}

@Override

public JSONArray getQuestions(String company, int count) {

try {

comp = company;

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/j2eeexp", "ws", "6477");

String query = "SELECT \* FROM QUESTIONS WHERE COMPANY='" + company + "' ORDER BY RANDOM() OFFSET 0 ROWS FETCH NEXT " + count + " ROW ONLY";

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(query);

return WSConvert.convertResultSetIntoJSON(rs);

} catch (Exception e) {

return null;

}

}

@Override

public String submitTest(Map<String, String[]> parameterMap, int n, String username) {

try {

int count = 0;

for (String key : parameterMap.keySet()) {

String value = ((String[]) parameterMap.get(key))[0];

if (!key.equals("SubmitTest")) {

if (questionsFacade.find(key).getAnswer().equals(value)) {

count++;

}

}

}

int percent = count \* 100 / n;

testhistoryFacade.create(new Testhistory(new Date().toString(), username, comp, percent));

Users u = usersFacade.find(username);

u.setScore(u.getScore() + percent);

usersFacade.edit(u);

return "true";

} catch (Exception e) {

System.out.println(e);

return "false";

}

}

}

SBBusinessLogicLocal.java

package com;

import java.util.Map;

import javax.ejb.Local;

import org.json.simple.JSONArray;

@Local

public interface SBBusinessLogicLocal {

String signIn(String username, String password);

String signUp(String name, String username, String email, String dob, String password);

String addQuestion(String question, String option1, String option2, String option3, String option4, String answer, String company);

String getScore(String username);

JSONArray getTop(int n);

JSONArray getHistory(String username);

JSONArray getCompany();

JSONArray getQuestions(String company, int count);

String submitTest(Map<String, String[]> parameterMap, int n, String username);

}

WSConvert.java

package com;

import java.io.BufferedReader;

import java.sql.ResultSet;

import org.json.simple.JSONArray;

import org.json.simple.JSONObject;

import org.json.simple.parser.JSONParser;

public class WSConvert {

public static JSONObject readerToJSON(BufferedReader br) {

try {

StringBuilder sb = new StringBuilder();

String line;

while ((line = br.readLine()) != null) {

sb.append(line);

}

JSONParser parser = new JSONParser();

return (JSONObject) parser.parse(sb.toString());

} catch (Exception e) {

System.out.println(e);

return null;

}

}

static JSONArray convertResultSetIntoJSON(ResultSet resultSet, int count) {

int n = 1;

try {

JSONArray jsonArray = new JSONArray();

while (resultSet.next() && n++ <= count) {

int total\_rows = resultSet.getMetaData().getColumnCount();

JSONObject obj = new JSONObject();

for (int i = 0; i < total\_rows; i++) {

String columnName = resultSet.getMetaData().getColumnLabel(i + 1).toLowerCase();

Object columnValue = resultSet.getObject(i + 1);

if (columnValue == null) {

columnValue = "null";

}

if (obj.containsKey(columnName)) {

columnName += "1";

}

obj.put(columnName, columnValue);

}

jsonArray.add(obj);

}

return jsonArray;

} catch (Exception e) {

return null;

}

}

public static JSONArray convertResultSetIntoJSON(ResultSet resultSet) throws Exception {

JSONArray jsonArray = new JSONArray();

while (resultSet.next()) {

int total\_rows = resultSet.getMetaData().getColumnCount();

JSONObject obj = new JSONObject();

for (int i = 0; i < total\_rows; i++) {

String columnName = resultSet.getMetaData().getColumnLabel(i + 1).toLowerCase();

Object columnValue = resultSet.getObject(i + 1);

if (columnValue == null) {

columnValue = "null";

}

if (obj.containsKey(columnName)) {

columnName += "1";

}

obj.put(columnName, columnValue);

}

jsonArray.add(obj);

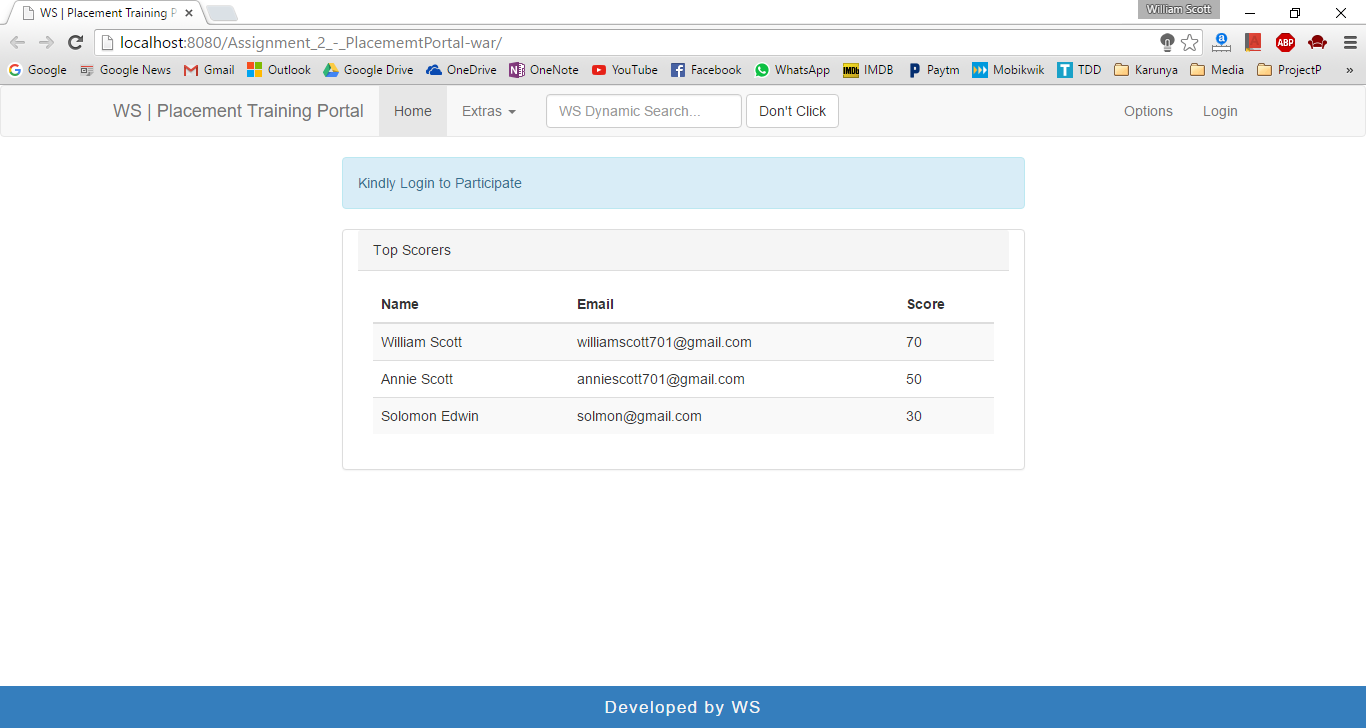
}

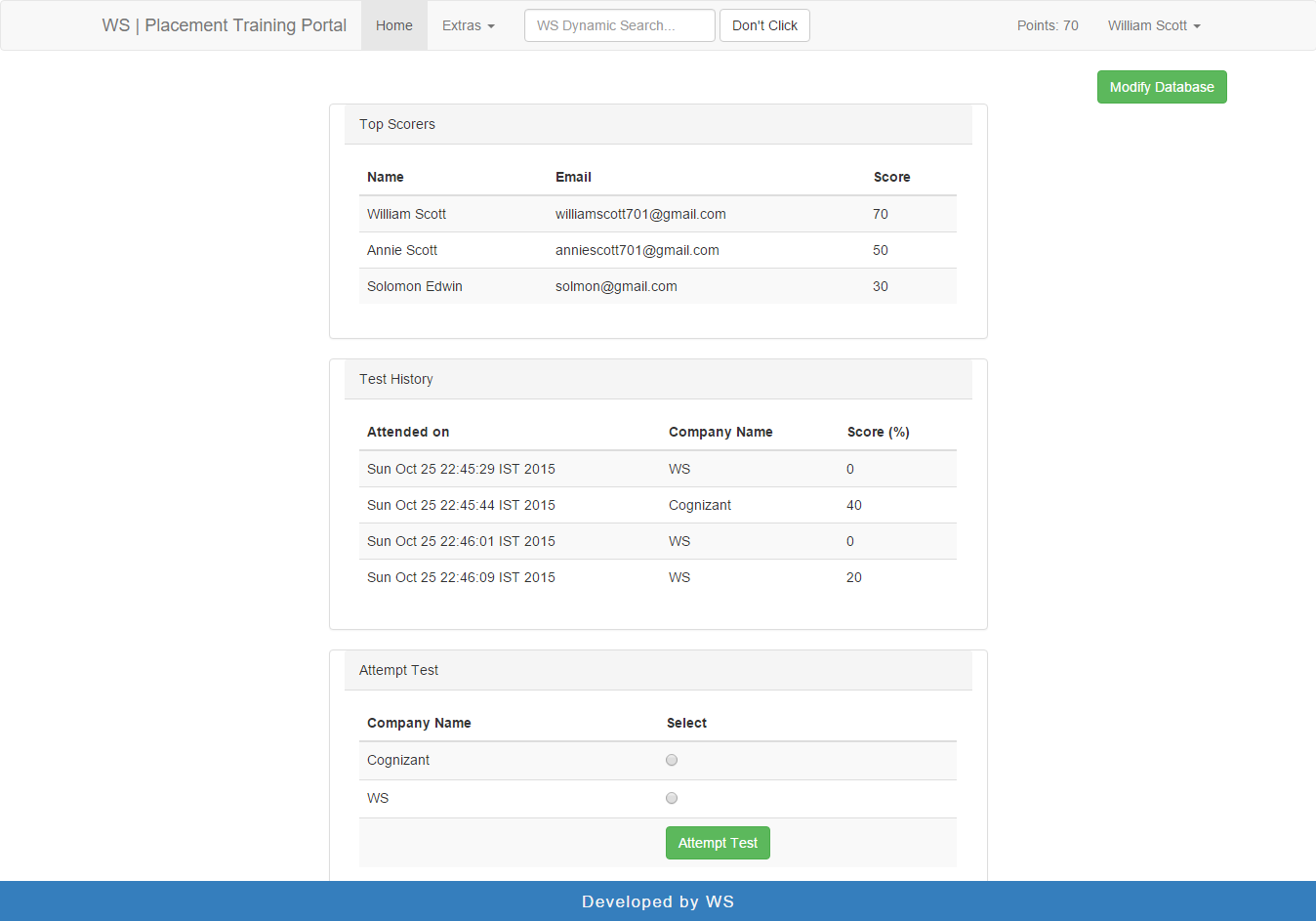
return jsonArray;

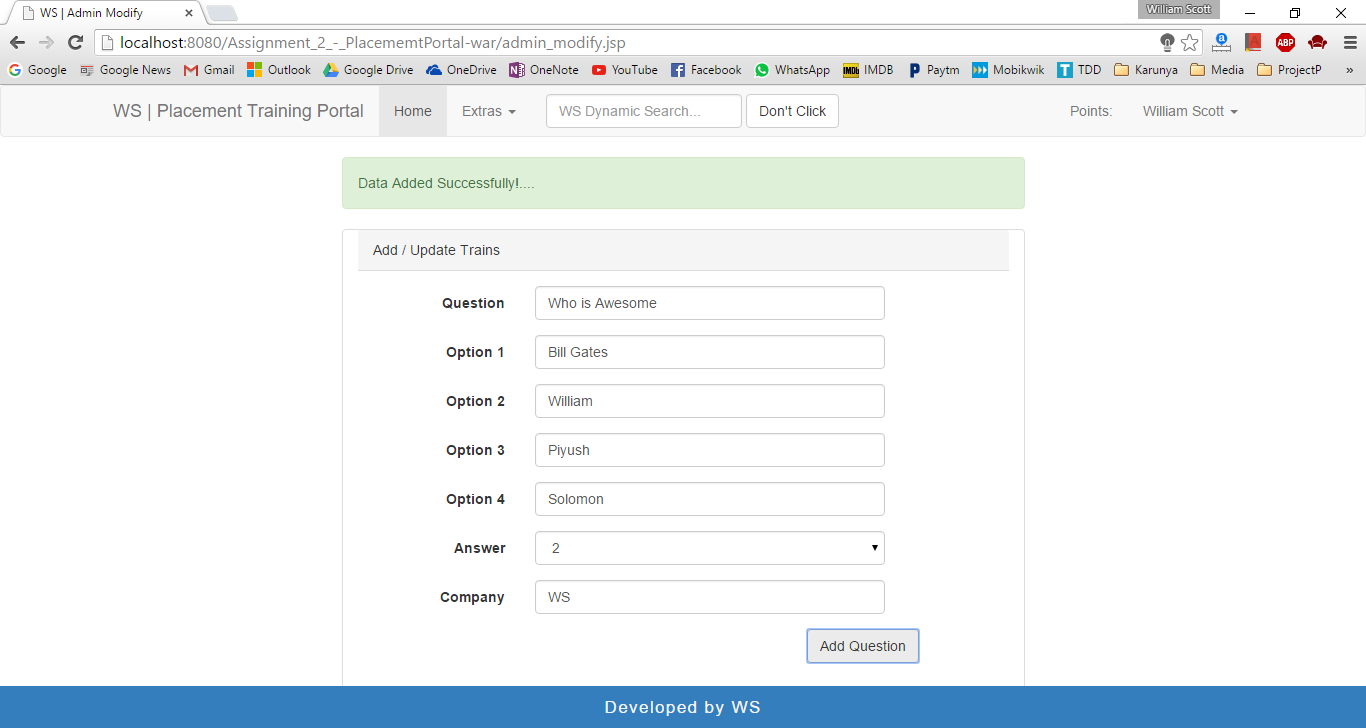
}

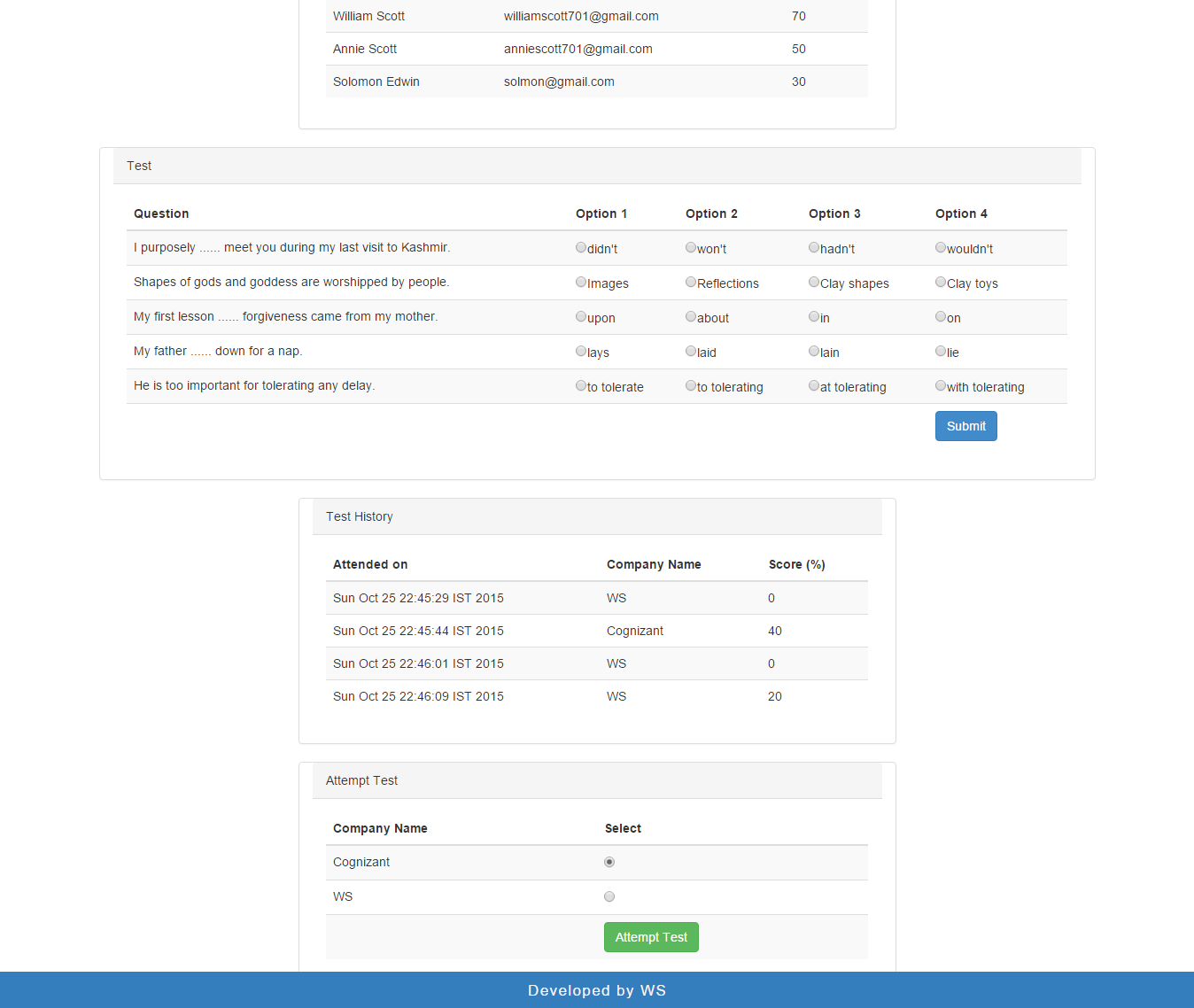
}

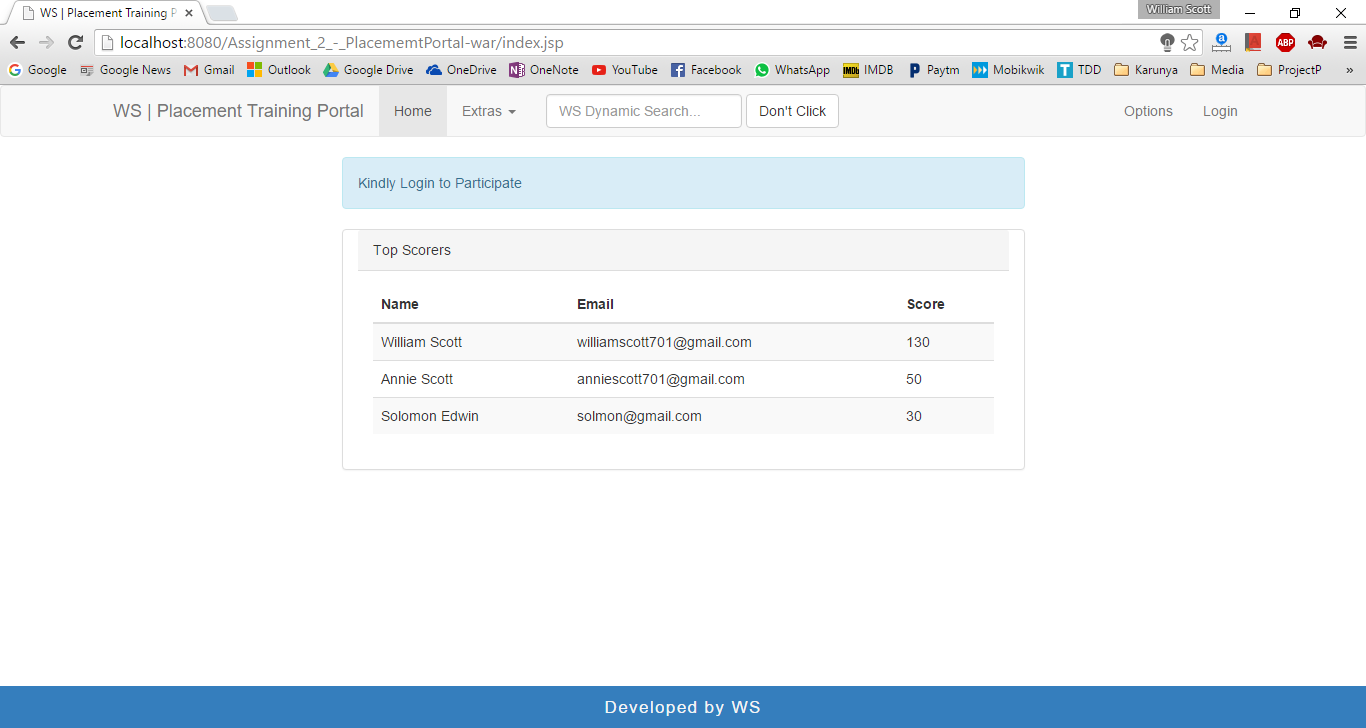
**Output**

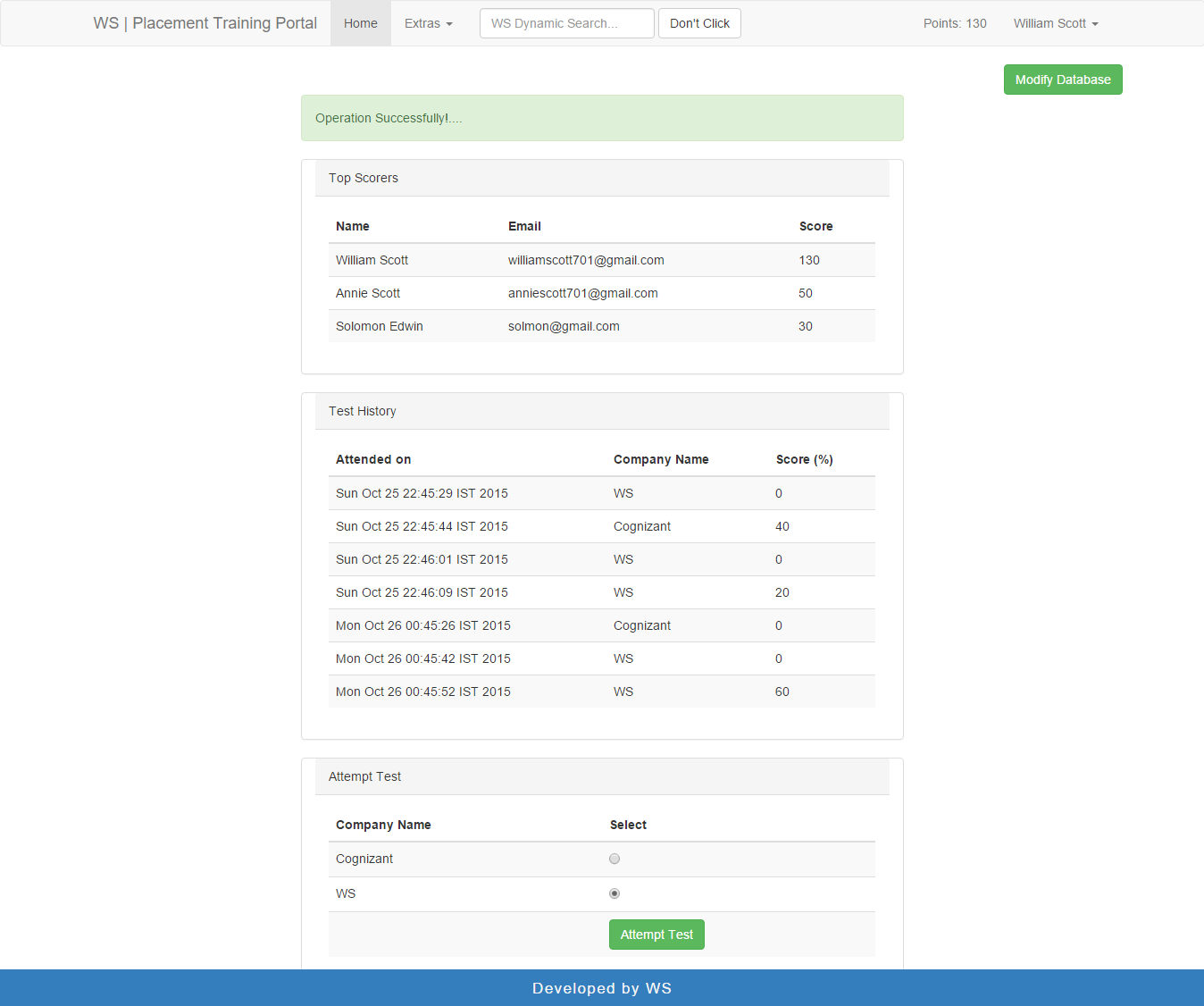












**Result**

Thus, the enterprise innovative application for live aptitude test was successfully developed and deployed using java programming.

[Signature of the Staff In-charge]

[Name of the Staff In – charge]