Content:

1. Normal Chart Generation
2. With Array Data Table
3. With DataTable
4. With JSON
5. With Colors and Annotations
6. With Percentages

Loading the Libraries

**1. Loading GoogleChart API**

script type="text/javascript" src="https://www.google.com/jsapi"></script>  
<script type="text/javascript">

**2. Loading Google Visualization API**

// Load the Visualization API and the piechart package.  
 google.load('visualization', '1.0', {'packages':['corechart']});

**3. Create function in that add Columns, Add data, Create Chart, Pass the data in that**

function drawChart() {  
  
      // Create the data table.  
      var data = new google.visualization.DataTable();  
      data.addColumn('string', 'Topping');  
      data.addColumn('number', 'Slices');  
      data.addRows([  
        ['Mushrooms', 3],  
        ['Onions', 1],  
        ['Olives', 1],   
        ['Zucchini', 1],  
        ['Pepperoni', 2]  
      ]);  
  
      // Set chart options  
      var options = {'title':'How Much Pizza I Ate Last Night',  
                     'width':400,  
                     'height':300};  
  
      // Instantiate and draw our chart, passing in some options.  
      var chart = new google.visualization.PieChart(document.getElementById('chart\_div'));  
      chart.draw(data, options);  
    }

**4.Call that function**

google.setOnLoadCallback(drawChart);

**5.Place a Div for display data**

div id="chart\_div" style="width:400; height:300"></div>

**1.Normal Chart : arrayToDataTable**

<html>  
  <head>  
    <script type="text/javascript" src="https://www.google.com/jsapi"></script>  
    <script type="text/javascript">  
      google.load("visualization", "1", {packages:["corechart"]});  
      google.setOnLoadCallback(drawChart);  
      function drawChart() {  
        var data = google.visualization.arrayToDataTable([  
          ['Year', 'Sales', 'Expenses'],  
          ['2013',  1000,      400],  
          ['2014',  1170,      460],  
          ['2015',  660,       1120],  
          ['2016',  1030,      540]  
        ]);  
  
        var options = {  
          title: 'Company Performance',  
          hAxis: {title: 'Year',  titleTextStyle: {color: '#333'}},  
          vAxis: {minValue: 0}  
        };  
  
        var chart = new google.visualization.AreaChart(document.getElementById('chart\_div'));  
        chart.draw(data, options);  
      }  
    </script>  
  </head>  
  <body>  
    <div id="chart\_div" style="width: 900px; height: 500px;"></div>  
  </body>  
</html>

**2.With Datatable and Coloumns**

<html>  
  <head>  
    <!--Load the AJAX API-->  
    <script type="text/javascript" src="https://www.google.com/jsapi"></script>  
    <script type="text/javascript">  
      
      // Load the Visualization API and the piechart package.  
      google.load('visualization', '1.0', {'packages':['corechart']});  
        
      // Set a callback to run when the Google Visualization API is loaded.  
      google.setOnLoadCallback(drawChart);  
  
      // Callback that creates and populates a data table,   
      // instantiates the pie chart, passes in the data and  
      // draws it.  
      function drawChart() {  
  
      // Create the data table.  
      var data = new google.visualization.DataTable();  
      data.addColumn('string', 'Topping');  
      data.addColumn('number', 'Slices');  
      data.addRows([  
        ['Mushrooms', 3],  
        ['Onions', 1],  
        ['Olives', 1],   
        ['Zucchini', 1],  
        ['Pepperoni', 2]  
      ]);  
  
      // Set chart options  
      var options = {'title':'How Much Pizza I Ate Last Night',  
                     'width':400,  
                     'height':300};  
  
      // Instantiate and draw our chart, passing in some options.  
      var chart = new google.visualization.PieChart(document.getElementById('chart\_div'));  
      chart.draw(data, options);  
    }  
    </script>  
  </head>  
  
  <body>  
<!--Div that will hold the pie chart-->  
    <div id="chart\_div" style="width:400; height:300"></div>  
  </body>  
</html>

**3. Columns with Styles**

**function** drawVisualization14() {

// Some raw data (not necessarily accurate)

**var** data14 = **new** google.visualization.DataTable();

data14.addColumn('string', 'Q.No');

data14.addColumn('number', 'Time(in Seconds)');

data14.addColumn({type:'string', role:'style'});

**var** foo = '['+$('#dataLineChart').val()+']';

data14.addRows(JSON.parse(foo));

/\* google.visualization.arrayToDataTable([

[ 'Result', 'Time', {role : "style"} ],

['1', 20, "#679d1b" ],['2', 20, "#679d1b" ],

[ '3', 4, "#d12610" ],

[ '4', 4, "#d12610" ],

['5', 20, "#679d1b" ],['6', 20, "#679d1b" ],

[ '7', 4, "#d12610" ], ['8', 20, "#679d1b" ],

[ '9', 4, "#d12610" ],

[ '10', 4, "#d12610" ]

]); \*/

**var** options = {

title : 'Your Answers With Result and Time',

width : 800,

height : 250,

vAxis : {

title : "Avg. time per Qsn."

},

hAxis : {

title : "Question No."

},

seriesType : "bars",

legend : {

position : "none"

},

min : 1

};

**var** chart14 = **new** google.visualization.ComboChart(document

.getElementById('chart\_div14'));

chart14.draw(data14, options);

}

**5. Column with Percentage as Annotation**

**function** drawVisualization19() {

// Some raw data (not necessarily accurate)

**var** data19 = **new** google.visualization.DataTable();

data19.addColumn('string', 'Q.Type');

data19.addColumn('number', 'Wrong Count');

data19.addColumn({type:'string', role:'annotation'});

data19.addColumn('number', 'Correct Count');

data19.addColumn({type:'string', role:'annotation'});

**var** foo = '['+$('#dataQuestionType').val()+']';

data19.addRows(JSON.parse(foo));

**var** options = {

title : 'Percentage of Correct/Incorrect Answers by Question Type',

width: 800,

height: 300,

vAxis : {

title : ""

},

hAxis : {

title : "Test No"

},

bar : {

groupWidth : "75%"

},

colors:['#ed4e2a','#35aa47'],

min : 1,

max : 50

};

**var** chart19 = **new** google.visualization.BarChart(document

.getElementById('chart\_div19'));

chart19.draw(data19, options);

}

**Example with Database:**

**Resultpage.jsp**

<%@page import=*"com.gbuds.bosrev.action.BO.test.TestResultsBO"*%>

<%@page import=*"java.util.List"*%>

<%@page import=*"java.util.Iterator"*%>

<%@ taglib prefix=*"s"* uri=*"/struts-tags"*%>

<!DOCTYPE html>

<!--

Template Name: Metronic - Responsive Admin Dashboard Template build with Twitter Bootstrap 3.0.2

Version: 1.5.4

Author: KeenThemes

Website: http://www.keenthemes.com/

Purchase: http://themeforest.net/item/metronic-responsive-admin-dashboard-template/4021469?ref=keenthemes

-->

<!--[if IE 8]> <html lang="en" class="ie8 no-js"> <![endif]-->

<!--[if IE 9]> <html lang="en" class="ie9 no-js"> <![endif]-->

<!--[if !IE]><!-->

<html lang=*"en"* class=*"no-js"*>

<!--<![endif]-->

<!-- BEGIN HEAD -->

<head>

<!-- <meta charset="utf-8" />

<title>Metronic | UI Features - General</title>

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta content="width=device-width, initial-scale=1.0" name="viewport" />

<meta content="" name="description" />

<meta content="" name="author" />

<meta name="MobileOptimized" content="320">

BEGIN GLOBAL MANDATORY STYLES

<link href="assets/plugins/font-awesome/css/font-awesome.min.css"

rel="stylesheet" type="text/css" />

<link href="assets/plugins/bootstrap/css/bootstrap.min.css"

rel="stylesheet" type="text/css" />

<link href="assets/plugins/uniform/css/uniform.default.css"

rel="stylesheet" type="text/css" />

END GLOBAL MANDATORY STYLES

BEGIN THEME STYLES

<link href="assets/css/style-metronic.css" rel="stylesheet"

type="text/css" />

<link href="assets/css/style.css" rel="stylesheet" type="text/css" />

<link href="assets/css/style-responsive.css" rel="stylesheet"

type="text/css" />

<link href="assets/css/plugins.css" rel="stylesheet" type="text/css" />

<link href="assets/css/themes/default.css" rel="stylesheet"

type="text/css" id="style\_color" />

<link href="assets/css/custom.css" rel="stylesheet" type="text/css" />

END THEME STYLES

BEGIN PAGE LEVEL STYLES

<link href="assets/plugins/gritter/css/jquery.gritter.css"

rel="stylesheet" type="text/css" />

END PAGE LEVEL STYLES

<link rel="shortcut icon" href="favicon.ico" /> -->

<%--

<script type="text/javascript">

function flag() {

/\* les = document.getElementById("f").value;

document.fomm.action = "questionflag.action";

document.fomm.submit();

\*/

alert('DIV Strtred');

$.ajax({url:"design1.action #gb",

success:function(result){

var aa = $("#div1").load("design1.action #gb");

$("#div1").html(aa);

}});

}

</script> --%>

<%--

<script>

$(document).ready(function(){

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

alert('DIV Strtred');

$.ajax({url:"design1.action #gb",

success:function(result){

var aa = $("#div1").load("design1.action #gb");

$("#div1").html(aa);

}});

});

});

</script> --%>

<!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CHARTS @ SATYA AREA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<!--CHARTS SCRIPT \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->

<script type=*"text/javascript"* src=*"https://www.google.com/jsapi"*></script>

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

google.load('visualization', '1', {

packages : [ 'corechart' ]

});

google.load('visualization', '1', {

packages : [ 'gauge' ]

});

</script>

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

$(document).ready(**function**(){

// alert(" form data is :"+$('#dataHard').val());

// alert(" form data is :"+$('#chart\_div3\_data').val());

//alert(" form data is :"+$('#chart\_div4\_data').val());

});

//CHART 13 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Graph1\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**function** drawPie1() {

// Some raw data (not necessarily accurate)

/\* var dataPie1 = google.visualization.arrayToDataTable([

[ 'Result', 'Score', {role : "style"} ],

[ 'Correct', 70 , "#679d1b"],

[ 'In Correct', 30, "#d12610" ]

]);

\*/

**var** dataPie1 = **new** google.visualization.DataTable();

dataPie1.addColumn('string', 'Result');

dataPie1.addColumn('number', 'Score');

dataPie1.addColumn({type:'string', role:'style'});

**var** foo = '['+$('#dataPieChart').val()+']';

dataPie1.addRows(JSON.parse(foo));

/\* dataPie1.addRows([

[ 'Correct', 70 , "#679d1b"],

[ 'Wrong', 30, "#d12610"]

]);

\*/

**var** options = {

title : 'Quant Practice Tests ',

width : 500,

height : 400,

is3D : **true**,

};

**var** chartPie1 = **new** google.visualization.PieChart(document

.getElementById('chart\_div13'));

chartPie1.draw(dataPie1, options);

}

//CHART 13 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Graph1\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//CHART 14 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* DOUBTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**function** drawVisualization14() {

// Some raw data (not necessarily accurate)

**var** data14 = **new** google.visualization.DataTable();

data14.addColumn('string', 'Q.No');

data14.addColumn('number', 'Time(in Seconds)');

data14.addColumn({type:'string', role:'style'});

**var** foo = '['+$('#dataLineChart').val()+']';

data14.addRows(JSON.parse(foo));

/\* google.visualization.arrayToDataTable([

[ 'Result', 'Time', {role : "style"} ],

['1', 20, "#679d1b" ],['2', 20, "#679d1b" ],

[ '3', 4, "#d12610" ],

[ '4', 4, "#d12610" ],

['5', 20, "#679d1b" ],['6', 20, "#679d1b" ],

[ '7', 4, "#d12610" ], ['8', 20, "#679d1b" ],

[ '9', 4, "#d12610" ],

[ '10', 4, "#d12610" ]

]); \*/

**var** options = {

title : 'Your Answers With Result and Time',

width : 800,

height : 250,

vAxis : {

title : "Avg. time per Qsn."

},

hAxis : {

title : "Question No."

},

seriesType : "bars",

legend : {

position : "none"

},

min : 1

};

**var** chart14 = **new** google.visualization.ComboChart(document

.getElementById('chart\_div14'));

chart14.draw(data14, options);

}

//CHART 15 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Rigt or wrong with\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//CHART 19 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* PERFORMENCE VERBAL\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**function** drawVisualization19() {

// Some raw data (not necessarily accurate)

**var** data19 = **new** google.visualization.DataTable();

data19.addColumn('string', 'Q.Type');

data19.addColumn('number', 'Wrong Count');

data19.addColumn({type:'string', role:'annotation'});

data19.addColumn('number', 'Correct Count');

data19.addColumn({type:'string', role:'annotation'});

**var** foo = '['+$('#dataQuestionType').val()+']';

data19.addRows(JSON.parse(foo));

/\* google.visualization.arrayToDataTable([

['Test No', 'Wrong', 'Correct'],

['1', 32, 24],

['2', 126, 22],

['3', 28, 19],

]);

\*/

**var** options = {

title : 'Percentage of Correct/Incorrect Answers by Question Type',

width: 800,

height: 300,

vAxis : {

title : ""

},

hAxis : {

title : "Test No"

},

bar : {

groupWidth : "75%"

},

colors:['#ed4e2a','#35aa47'],

min : 1,

max : 50

};

**var** chart19 = **new** google.visualization.BarChart(document

.getElementById('chart\_div19'));

chart19.draw(data19, options);

}

//CHART 19 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* VERBAL\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//CHART 3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* DOUBTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* NO NEED\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**function** drawVisualization3() {

// Some raw data (not necessarily accurate)

**var** data3 = **new** google.visualization.DataTable();

data3.addColumn('string', 'Section');

data3.addColumn('number', 'Seconds');

/\* data3.addRows([

[ 'RC', 12 ],

[ 'SE', 8],

[ 'RC', 12 ],

[ 'SE', 8],

[ 'MC', 4 ]

]); \*/

**var** foo = '['+$('#dataTimeQuestionType').val()+']';

data3.addRows(JSON.parse(foo));

**var** options = {

title : 'Average time in Seconds by Question Type',

width : 800,

height : 300,

vAxis : {

title : "Time In Seconds"

},

seriesType : "bars",

bar : {

groupWidth : "25%"

},

legend : {

position : "none"

},

min : 1

};

**var** chart3 = **new** google.visualization.ComboChart(document

.getElementById('chart\_div3'));

chart3.draw(data3, options);

}

//CHART 4 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Time Spent\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* NO NEED\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//CHART 5 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Time Spent\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* NO NEED\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**function** drawVisualization29() {

// Some raw data (not necessarily accurate)

**var** data29 = **new** google.visualization.DataTable();

data29.addColumn('string', 'Section');

data29.addColumn('number', 'Wrong');

data29.addColumn({type:'string', role:'annotation'});

data29.addColumn('number', 'Correct');

data29.addColumn({type:'string', role:'annotation'});

**var** foo = '['+$('#dataHard').val()+']';

data29.addRows(JSON.parse(foo));

/\* data29.addRows([

["Very Easy", {v: 0, f: "0.0%" }, {v: 3, f: "100.0%" }],

["Easy", {v: 0, f: "0.0%" }, {v: 1, f: "100.0%" }],

["Medium", {v: 0, f: "0.0%" }, {v: 0, f: "0.0%" }],

["Very Easy", {v: 0, f: "0.0%" }, {v: 3, f: "100.0%" }],

["Easy", {v: 0, f: "0.0%" }, {v: 1, f: "100.0%" }],

["Medium", {v: 0, f: "0.0%" }, {v: 0, f: "0.0%" }]

]);

\*/

/\*

var data29 = google.visualization.arrayToDataTable([

['Test No', 'Wrong', 'Correct'],

['Very Easy', {v: 12, f: '10.0%'}, {v: 45, f: '40.0%'}],

['Very Easy', {v: 12, f: '10.0%'}, {v: 45, f: '40.0%'}],

['Very Easy', {v: 12, f: '10.0%'}, {v: 45, f: '40.0%'}],

['Very Easy', {v: 12, f: '10.0%'}, {v: 45, f: '40.0%'}],

['Very Easy', {v: 12, f: '10.0%'}, {v: 45, f: '40.0%'}],

['Very Easy', {v: 12, f: '10.0%'}, {v: 45, f: '40.0%'}],

['Very Easy', {v: 12, f: '10.0%'}, {v: 45, f: '40.0%'}],

['Very Easy', {v: 12, f: '10.0%'}, {v: 45, f: '40.0%'}],

["Very Easy", {v: 0, f: '0.0%'}, {v: 3, f: '100.0%'}],

["Easy", {v: 0, f: '0.0%'}, {v: 1, f: '100.0%'}],

["Medium", {v: 0, f: '0.0%'}, {v: 1, f: '100.0%'}]

]); \*/

**var** options = {

title : 'Your Performance with Difficulty Level',

width: 800,

height: 400,

vAxis : {

title : "No.of Questions" ,minValue: 1

},

hAxis : {

title : "Difficulty Levels"

},

bar : {

groupWidth : "30%"

},

colors:['#ed4e2a','#35aa47'],

isStacked: **true**,

min : 2

};

**var** chart29 = **new** google.visualization.ColumnChart(document

.getElementById('chart\_div29'));

chart29.draw(data29, options);

}

//Calling chart Functions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

google.setOnLoadCallback(drawPie1);

google.setOnLoadCallback(drawVisualization14);

google.setOnLoadCallback(drawVisualization19);

google.setOnLoadCallback(drawVisualization3);

google.setOnLoadCallback(drawVisualization29);

</script>

<!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CHARTS @ SATYA AREA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<script>

**function** flagButtonClicked(button) {

**var** test = document.getElementById('ltest').innerHTML;

**var** testdate = document.getElementById('tdate').innerHTML;

**var** userid = document.getElementById('uid').innerHTML;

**var** quesid = document.getElementById('ltest').innerHTML;

**var** quesno = document.getElementById('qno').innerHTML;

**var** testno = document.getElementById('qid').innerHTML;

**var** status = document.getElementById("flaglist").checked;

$.getJSON("flagaction.action", {

'testv' : test,

'testdatev' : testdate,

'useridv' : userid,

'quesidv' : quesid,

'quesnov' : quesno,

'tnov' : testno,

'statusv' : status,

},

**function**(input) {

**if** (input.status) {

**if** (input.status == 'success') {

**if** (name == 'Like')

button.attr("value", " Un Like");

**else**

button.attr("value", "Like");

}

}

});

}

</script>

<script>

**function** questionNavigation(button) {

//alert('NO NOS');

//QuestionBO bo,int test,String testdate,int userid,String status

// alert('BEfERRRRRRRRRRRRRRRRRRRRRR');

**var** test = document.getElementById('ltest').innerHTML;

**var** testdate = document.getElementById('tdate').innerHTML;

**var** userid = document.getElementById('uid').innerHTML;

**var** status = document.getElementById("flag").checked;

// alert('AFTERRRRRRRRRRRRRRRRRRRRRR');

alert("TEST " + test + "TEST DATE : " + testdate + "USER ID : "

+ userid + "STATUS : " + status);

$.getJSON("previous.action", {

'testv' : test,

'testdatev' : testdate,

'useridv' : userid,

'statusv' : status,

'qid' : questionid

},

**function**(input) {

**if** (input.status) {

**if** (input.status == 'success') {

**if** (name == 'Like')

button.attr("value", " Un Like");

**else**

button.attr("value", "Like");

}

}

});

}

</script>

</head>

<!-- END HEAD -->

<!-- BEGIN BODY -->

<body class=*"page-header-fixed"*>

<!-- BEGIN HEADER -->

<!-- END HEADER -->

<!-- BEGIN CONTAINER -->

<div class=*"page-container"*>

<!-- BEGIN SIDEBAR -->

<!-- END SIDEBAR -->

<!-- BEGIN PAGE -->

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

<s:form method=*"post"* name=*"fomm"*>

<!-- BEGIN PAGE HEADER-->

<div class=*"row"*>

<div class=*"col-md-12"*>

<!-- BEGIN PAGE TITLE & BREADCRUMB-->

<ul class=*"page-breadcrumb breadcrumb"*>

<li><i class=*"icon-home"*></i> <a

href=*"*<s:url action=*"#"* />*"*>Home</a> <i

class=*"icon-angle-right"*></i></li>

<li><a href=*"#"*>Result Page</a></li>

</ul>

<!-- END PAGE TITLE & BREADCRUMB-->

</div>

</div>

<!-- END PAGE HEADER-->

<!-- BEGIN PAGE CONTENT-->

<div class=*"row"*>

<div class=*"col-md-12"*>

<!-- BEGIN ALERTS PORTLET-->

<div class=*"portlet yellow box"*>

<div class=*"portlet-title"*>

<div class=*"caption"*>

<i class=*"fa fa-cogs"*></i>Summary

</div>

<div class=*"tools"*>

<a href=*"javascript:;"* class=*"collapse"*></a> <a

href=*"javascript:;"* class=*"reload"*></a>

</div>

</div>

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

<%-- request.getAttribute("result")

List size is :<s:property value="%{#request.data.length}"/> --%>

<!-- value="%{#request.resultdata}" -->

<%-- <s:iterator value="resultsBO" status="userStatus"> --%>

<s:iterator value=*"resultsBO"* status=*"userStatus"*>

<s:if test=*"#userStatus.last == true "*>

<div class=*"row"*>

<div class=*"col-md-4"*>

<div class=*"note note-success"*>

<h4 class=*"block"* style="text-align: *center*">

<b>Your Percent Correct</b>

</h4>

<h2 style="text-align: *center*">

<s:property value=*"resultper"* /><%=application.getAttribute("resultper")%>%

</h2>

<p style="text-align: *center*">

(

<%-- <s:property value="correctques" /> --%>

<%=application.getAttribute("correct")%>

of

<%-- <s:property value="totalques" /> --%><%=application.getAttribute("totalcount")%>

)

</p>

</div>

</div>

<!-- Col-md-1 -->

<div class=*"col-md-4"*>

<div class=*"note note-info"*>

<h4 class=*"block"* style="text-align: *center*">

<b>Your Average Pace</b>

</h4>

<%

System.out.println("usertotal avg"

+ application.getAttribute("usertotalavg"));

%>

<h2 style="text-align: *center*"

value=*"%{application.usertotalavg}"*><%=application.getAttribute("usertotalavg")%>

</h2>

<%-- <h2 style="text-align: center"><s:property value="usertotal" /></h2> --%>

<p style="text-align: *center*">

<%-- <s:property value="usertotal" /> --%><%=application.getAttribute("usertotal")%>

(total)

</p>

</div>

</div>

<!-- Col-md-2 -->

<div class=*"col-md-4"*>

<div class=*"note note-warning"*>

<h4 class=*"block"* style="text-align: *center*">

<b>Others' Average Pace</b>

</h4>

<h2 style="text-align: *center*">

<%-- <s:property value="othertotal" /> --%>

</h2>

<h2 style="text-align: *center*">

<%-- <s:property value="othertotalavgpace" /> --%>

<%=application.getAttribute("othertotalavg")%>

</h2>

<p style="text-align: *center*">

<%-- <s:property value="othertotal" /> --%><%=application.getAttribute("othertotal")%>

(total)

</p>

</div>

</div>

<!-- Col-md-3 -->

</div>

<!--End ROW-->

</s:if>

</s:iterator>

</div>

<!-- END ALERTS PORTLET-->

</div>

</div>

<!-- Fisrt table table End-->

<!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Charts Area STart\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<div class=*"row"*> <div class= *"col-md-8 col-md-offset-3"* id=*"chart\_div13"*></div></div>

<div class=*"row"*> <div class= *"col-md-10 col-md-offset-1"* id=*"chart\_div14"*></div></div>

<div class=*"row"*> <div class= *"col-md-10 col-md-offset-1"* id=*"chart\_div19"*></div></div>

<div class=*"row"*> <div class= *"col-md-10 col-md-offset-1"* id=*"chart\_div3"*></div></div>

<div class=*"row"*> <div class= *"col-md-10 col-md-offset-1"* id=*"chart\_div29"*></div></div>

<!-- <s:label value="%{#request.ResultHard}" > Seee THIS HAS TO COme</s:label>-->

<s:hidden id=*"dataPieChart"* value=*"%{#request.ResultPieChart}"* ></s:hidden>

<s:hidden id=*"dataLineChart"* value=*"%{#request.ResultLineChart}"*></s:hidden>

<s:hidden id=*"dataQuestionType"* value=*"%{#request.ResultQuestionType}"*></s:hidden>

<s:hidden id=*"dataTimeQuestionType"* value=*"%{#request.ResultTimeQuestionType}"*></s:hidden>

<s:hidden id=*"dataHard"* value=*"%{#request.ResultHard}"*></s:hidden>

</div>

<!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Charts Area End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<div class=*"row"*>

<div class=*"col-md-12"*>

<!-- BEGIN SAMPLE TABLE PORTLET-->

<div class=*"portlet"*>

<div class=*"portlet-title"*>

<div class=*"caption"*>

<i class=*"fa fa-bell-o"*></i>Review Table

</div>

<div class=*"tools"*>

<a class=*"collapse"* href=*"javascript:;"*></a> <a class=*"reload"*

href=*"javascript:;"*></a>

</div>

</div>

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

<div class=*"table-responsive"*>

<table

class=*"table table-striped table-bordered table-advance table-hover"*>

<thead>

<tr>

<th>Result</th>

<th>Question Title</th>

<th>Section</th>

<th>Subject</th>

<th>Difficulty</th>

<th>Your Pace</th>

<th>Other's Pace</th>

<th>Date</th>

<th>Note</th>

<th>Flag</th>

</tr>

</thead>

<tbody>

<%-- <%

List l = (List) request.getAttribute("resultdata");

System.out.println(" DATA " + l);

if (l != null) {

Iterator it = l.iterator();

%>

<%

while (it.hasNext()) {

TestResultsBO bo = (TestResultsBO) it.next();

%>

<tr>

<%

System.out.println("Result "+bo.getResult());

if(bo.getResult().equals("Y"))

{

%>

<td><i class="fa fa-check-circle-o" style="color:green"></i></td>

<%}

else {%>

<td><i class="fa fa-check-circle-o" style="color:red"></i></td>

<%} %>

<td><a href="#"><%=bo.getQuestion\_title()%> </a></td>

<td><%=bo.getSection\_name()%></td>

<td><%=bo.getLesson\_name()%></td>

<td><%=bo.getDifficulty\_name()%></td>

<td><%=bo.getUser\_time()%></td>

<td><%=bo.getAvg\_time()%></td>

<td><%=bo.getTest\_date()%></td>

<td>My Note</td>

<td><input type="checkbox" class="badge-danger"></td>

</tr>

<%

}

}

%> --%>

<%

List l = (List) request.getAttribute("resultdata");

//System.out.println(" DATA " + l);

//System.out.println("Result " + l.get(0));

**if** (l != **null**) {

Iterator it = l.iterator();

it.next();

%>

<%

**while** (it.hasNext()) {

TestResultsBO bo = (TestResultsBO) it.next();

System.out.println("sdvfvsdf" + bo.getTest\_id());

System.out.println("sdvfvsdf 111111 " + bo.getTest\_no());

System.out.println("sdvfvsdf 222222 " + bo.getQuestion\_id());

System.out.println("sdvfvsdf 33333 " + bo.getQuestion\_no());

%>

<tr>

<%

**if** (bo.getResult().equals("Y")

&& (!bo.getResult().equalsIgnoreCase(**null**))) {

%>

<!-- <td><i class="fa fa-check-circle-o" style="color:green">Correct</i></td> -->

<td>Correct</td>

<%

} **else** {

%>

<!-- <td><i class="fa fa-check-circle-o" style="color:red">Wrong</i></td> -->

<td>Wrong</td>

<%

}

%>

<%-- <s:hidden id ="quesid" value="<%=bo.getQuestion\_id()%>"/>

<s:hidden id ="quesno" value="<%=bo.getQuestion\_no()%>"/> --%>

<div>

<%-- <label id="ltest"><%=bo.getTest\_id()%></label> <label

id="uid"><%=bo.getUser\_id()%></label>

<label

id="qid"><%=bo.getQuestion\_id()%></label>

<label

id="qno"><%=bo.getQuestion\_no()%></label>

<label

id="tno"><%=bo.getTest\_no()%></label> --%>

<%-- <s:set value="%{#request.resultdata[0]}" var="ids"/>

<s:text name="%{ids.test\_id}"/>

<s:hidden id="ltest" name="ltest" value="<%=bo.getTest\_id()%>"/>

<input type=hidden id="uid" name="uid" value="<%=bo.getUser\_id()%>"/>

<s:hidden id="qid" name="qid" value="<%=bo.getQuestion\_id()%>"/>

<s:hidden id="qno" name="qno" value="<%=bo.getQuestion\_no()%>"/>

<s:hidden id="tno" name="tno" value="<%=bo.getTest\_no()%>"/>

<input type="hidden" id="ltest" value="<%=bo.getTest\_id()%>"/>

<input type="hidden" id="uid" value="<%=bo.getUser\_id()%>"/>

<input type="hidden" id="qid" value="<%=bo.getQuestion\_id()%>"/>

<input type="hidden" id="qno" value="<%=bo.getQuestion\_no()%>"/>

<input type="hidden" id="tno" name="tno" value="<%=bo.getTest\_no()%>"/>

--%>

</div>

<td><a href=*"*<s:url action=*"questionNavigation(this)"*/>*"* ><%=bo.getQuestion\_title()%>

</a></td>

<td><%=bo.getSection\_name()%></td>

<td><%=bo.getLesson\_name()%></td>

<td><%=bo.getDifficulty\_name()%></td>

<td><%=bo.getUser\_time()%></td>

<td><%=bo.getAvg\_time()%></td>

<td><label id=*"tdate"*><%=bo.getTest\_date()%></label></td>

<td>My Note</td>

<td><input type=*"checkbox"* class=*"badge-danger"*

onchange=*"flagButtonClicked(this)"* id=*"flaglist"* value=*"*<%=bo.getQuestion\_id()%>*"*></td>

<%-- <td><input type="checkbox" class="badge-danger" name="flaglist" id="flaglist" value="<%=bo.getQuestion\_id()%>"></td> --%>

</tr>

<%

}

}

%>

</tbody>

</table>

</div>

<!-- </form> -->

<%-- <a href="<s:url action="send\_report\_Email.action"></s:url>"></a> --%>

</div>

</div>

</div>

<!-- END SAMPLE TABLE PORTLET-->

</div>

</s:form>

<%-- <%

}

%> --%>

</div>

<div id=*"dum"*></div>

<!-- Second table End-->

</div>

<!-- END PAGE CONTENT-->

<!-- BEGIN PAGE -->

<!-- END CONTAINER -->

</body>

<!-- END BODY -->

</html>

REPORTDAO.java

package com.gbuds.bosrev.dao.reports;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.sql.Time;

import java.util.Random;

import javax.servlet.http.HttpSession;

import com.gbuds.bosrev.action.BO.reports.ReportDetails;

import com.gbuds.bosrev.data.BRDAO;

import com.gbuds.bosrev.exception.ConnectionException;

public class ReportsDAO extends BRDAO implements IReportsDAO{

ReportDetails details;

String ch1;

public int verbalCount(String userid) {

int count = 0;

ResultSet resultSet = null;

Connection connection = null;

CallableStatement cs = null;

Statement st=null;

java.sql.PreparedStatement preparedStatement=null;

//System.out.println("USer ID"+session.getAttribute("uid"));

String qry = "SELECT COUNT(t.test\_name) from br\_test\_master t WHERE t.user\_id = "+userid;

System.out.println(qry);

try {

connection = getBRConnection();

preparedStatement=connection.prepareStatement(qry);

resultSet = preparedStatement.executeQuery();

if(resultSet.next())

{

count = resultSet.getInt(1);

System.out.println("Count : "+count);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (ConnectionException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

finally{

try {

connection.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

// TODO Auto-generated method stub

return count;

}

public ReportDetails getDetails(String userid) {

// TODO Auto-generated method stub

details = new ReportDetails();

ResultSet resultSet = null;

Connection connection = null;

CallableStatement cs = null;

Statement st=null;

java.sql.PreparedStatement preparedStatement=null;

//System.out.println("USer ID"+session.getAttribute("uid"));

String qry = "SELECT t.firstname,t.lastname,t.email,t.startdate,t.enddate,t.test\_date,t.target\_score,t.actual\_score,t.address,t.contactno,t.photo FROM br\_register t WHERE t.userid = "+userid;

System.out.println(qry);

try {

connection = getBRConnection();

preparedStatement=connection.prepareStatement(qry);

resultSet = preparedStatement.executeQuery();

if(resultSet.next())

{

details.setFirstname(resultSet.getString(1));

details.setLastname(resultSet.getString(2));

details.setEmailid(resultSet.getString(3));

details.setStartdate(resultSet.getString(4));

details.setEnddate(resultSet.getString(5));

details.setTestdate(resultSet.getString(6));

details.setTargetscore(resultSet.getString(7));

details.setActualscore(resultSet.getString(8));

details.setAddress(resultSet.getString(9));

details.setContactno(resultSet.getString(10));

details.setImg(resultSet.getString(11));

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (ConnectionException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return details;

}

//For usage Chart \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public String UsageChart(String userid) {

// TODO Auto-generated method stub

String ch1 = "";

String tnam="",cn="",pr="";

int i=0;

ResultSet resultSet = null;

Connection connection = null;

CallableStatement cs = null;

Statement st=null;

java.sql.PreparedStatement preparedStatement=null;

//System.out.println("USer ID"+session.getAttribute("uid"));

String qry = "select tscr.test\_master\_id,tmstr.test\_name,count(tscr.row\_id) CountNo from br\_test\_score tscr,br\_test\_master tmstr where tscr.test\_master\_id=tmstr.test\_master\_id AND tscr.userid = "+userid+" group by tscr.test\_master\_id";

System.out.println(qry);

try {

connection = getBRConnection();

preparedStatement=connection.prepareStatement(qry);

String cls[] = {"#FF0000","#00FF00","#0000FF","#FF00FF","#FF8C00","#8A2BE2"};

resultSet = preparedStatement.executeQuery();

while(resultSet.next())

{

String tname = resultSet.getString(2);

int cnt = resultSet.getInt(3);

double per = (double)(((double)10/(double)50)\*(double)100);

ch1 = ch1+"[\""+tname+"\","+cnt+",\""+cls[i]+"\"],";

i++;

}

qry = "SELECT COUNT(fs.user\_id) flashCount FROM br\_flash\_session fs WHERE fs.user\_id = "+userid;

preparedStatement=connection.prepareStatement(qry);

resultSet = preparedStatement.executeQuery();

if(resultSet.next())

{

int cnt = resultSet.getInt(1);

double per = (double)(((double)10/(double)50)\*(double)100);

String tname = "Verbal Cards";

ch1 = ch1+"[\""+tname+"\","+cnt+",\""+cls[i]+"\"],";

i++;

}

qry = "SELECT COUNT(ms.studentid) flashCount FROM br\_math\_session ms WHERE ms.studentid = "+userid;

preparedStatement=connection.prepareStatement(qry);

resultSet = preparedStatement.executeQuery();

if(resultSet.next())

{

int cnt = resultSet.getInt(1);

double per = (double)(((double)10/(double)50)\*(double)100);

String tname = "Math Cards";

ch1 = ch1+"[\""+tname+"\","+cnt+",\""+cls[i]+"\"],";

i++;

}

int limit=ch1.lastIndexOf(",");

ch1=ch1.substring(0,limit);

System.out.println("Value : "+ch1);

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (ConnectionException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return ch1;

}

//For usage Chart \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//For Doubts Chart \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public String DoubtsChart(String userid) {

String ch2 = "";

int count = 0;

ResultSet resultSet = null;

Connection connection = null;

CallableStatement cs = null;

Statement st=null;

java.sql.PreparedStatement preparedStatement=null;

//System.out.println("USer ID"+session.getAttribute("uid"));

String qry = "SELECT count(d.doubt\_id),status FROM br\_user\_doubt d WHERE d.userid ="+userid+" Group by status";

System.out.println(qry);

/\*

[ 'Asked', 12, "#0000ee" ], //blue

[ 'Answerd', 8, "#339900" ], //green

[ 'UnAnswerd', 4, "#d12610" ]//red

\*/

try {

connection = getBRConnection();

preparedStatement=connection.prepareStatement(qry);

resultSet = preparedStatement.executeQuery();

while(resultSet.next())

{

int cnt = resultSet.getInt(1);

count = count+cnt;

String tname = resultSet.getString(2);

if(tname.equalsIgnoreCase("answered"))

{

ch2 = ch2+"[ \"Answerd\", "+cnt+", \"#339900\" ],";

}

else{

ch2 = ch2+"[ \"UnAnswerd\", "+cnt+", \"#d12610\" ],";

}

}//End of while

ch2 = "[ \"Asked\", "+count+", \"#0000ee\" ],"+ch2;

int limit=ch2.lastIndexOf(",");

ch2=ch2.substring(0,limit);

System.out.println("Value : "+ch2);

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (ConnectionException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

// TODO Auto-generated method stub

return ch2;

}

public String TimeSpentChart(String userid) {

// TODO Auto-generated method stub

int count = 0;

String ch3 = "";

ResultSet resultSet = null;

Connection connection = null;

CallableStatement cs = null;

Statement st=null;

java.sql.PreparedStatement preparedStatement=null;

//System.out.println("USer ID"+session.getAttribute("uid"));

String qry = "select tm.test\_name, SUM(TIME\_TO\_SEC(t.user\_time)) as testtime , SEC\_TO\_TIME(SUM(TIME\_TO\_SEC(t.user\_time))) from br\_test\_master tm,br\_test\_results t , br\_flash\_session f, br\_math\_session m where t.test\_master\_id =tm.test\_master\_id and t.userid = "+userid+" group by t.test\_master\_id";

System.out.println(qry);

try {

connection = getBRConnection();

preparedStatement=connection.prepareStatement(qry);

resultSet = preparedStatement.executeQuery();

while(resultSet.next())

{

String tname = resultSet.getString(1);

double tim = resultSet.getDouble(2);

//String tp = resultSet.getString(3);

double inhr = (double)(((double)tim/(double)60));

ch3= ch3+"[\""+tname+"\","+inhr+"],";

//ch3= ch3+"[\""+tname+"\","+inhr+", \""+tp+"\"],";

}

qry = "select SUM(TIME\_TO\_SEC(f.user\_pace)) as flashtime from br\_test\_master tm,br\_test\_results t , br\_flash\_session f where t.userid="+userid;

preparedStatement=connection.prepareStatement(qry);

System.out.println(qry);

resultSet = preparedStatement.executeQuery();

if(resultSet.next())

{

String tname = "Verbal Cards";

Double tim = resultSet.getDouble(1);

double inhr = (double)(((double)tim/(double)60));

ch3= ch3+"[\""+tname+"\","+inhr+"],";

}

preparedStatement=connection.prepareStatement(qry);

qry = "select SEC\_TO\_TIME(SUM(TIME\_TO\_SEC(m.student\_pace))) as mathtime from br\_test\_master tm,br\_test\_results t , br\_math\_session m where t.userid ="+userid;

System.out.println(qry);

resultSet = preparedStatement.executeQuery();

if(resultSet.next())

{

String tname = "Math Cards";

Double tim = resultSet.getDouble(1);

double inhr = (double)(((double)tim/(double)60));

ch3= ch3+"[\""+tname+"\","+inhr+"],";

}

int limit=ch3.lastIndexOf(",");

if(limit>0)

{

ch3=ch3.substring(0,limit);

}

System.out.print("DAo: "+ch3);

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (ConnectionException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

finally{

try {

connection.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

// TODO Auto-generated method stub

return ch3;

}

//\*\*\*\*\*\*\*\*\*\*\*\* For Result Page \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public String ResultPieChart(int userid, int testtype, int testno) {

// TODO Auto-generated method stub

String ResultPieChart = "";

int count = 0;

ResultSet resultSet = null;

Connection connection = null;

CallableStatement cs = null;

Statement st=null;

java.sql.PreparedStatement preparedStatement=null;

//System.out.println("USer ID"+session.getAttribute("uid"));

String qry = "select t.test\_no, (select count(\*) from br\_test\_results t where t.result like 'Y' and t.test\_no="+testno+" and t.userid="+userid+" ) as Correct,(select count(\*) from br\_test\_results t where t.result like 'N' and t.test\_no="+testno+" and t.userid="+userid+" ) as wrong from br\_test\_results t where t.test\_master\_id = "+testtype+" and t.userid="+userid+" and t.test\_no="+testno+" group by t.test\_no";

System.out.println(qry);

try {

connection = getBRConnection();

preparedStatement=connection.prepareStatement(qry);

resultSet = preparedStatement.executeQuery();

if(resultSet.next())

{

String crt = resultSet.getString(2);

String wrg = resultSet.getString(3);

ResultPieChart= ResultPieChart+"[ \"Correct\", "+crt+" , \"#679d1b\"],";

ResultPieChart= ResultPieChart+"[ \"Wrong\", "+wrg+" , \"#d12610\"],";

}

int limit=ResultPieChart.lastIndexOf(",");

if(limit>0)

{

ResultPieChart=ResultPieChart.substring(0,limit);

}

System.out.print("DAo ResultPieChart \*\*\*\*\*\*\*\*\*\* : "+ResultPieChart);

} catch (Exception e) {

// TODO: handle exception

e.printStackTrace();

}

return ResultPieChart;

}

public String ResultLineChart(int userid, int testtype, int testno) {

String ResultLineChart = "";

int count = 0;

ResultSet resultSet = null;

Connection connection = null;

CallableStatement cs = null;

Statement st=null;

java.sql.PreparedStatement preparedStatement=null;

//System.out.println("USer ID"+session.getAttribute("uid"));

String qry = "select t.question\_number,t.result,(TIME\_TO\_SEC(t.user\_time)) as Testtime from br\_test\_results t,br\_question\_type qt where t.test\_master\_id="+testtype+" and qt.question\_type\_id=t.type\_id and t.userid="+userid+" and t.test\_no="+testno+" group by t.question\_id order by t.question\_number";

System.out.println(qry);

try {

connection = getBRConnection();

preparedStatement=connection.prepareStatement(qry);

resultSet = preparedStatement.executeQuery();

while(resultSet.next())

{

String qno = resultSet.getString(1);

String res = resultSet.getString(2);

String tim = resultSet.getString(3);

if(res.equalsIgnoreCase("Y"))

{

ResultLineChart= ResultLineChart+"[\""+qno+"\", "+tim+", \"#679d1b\" ],";

}else{

ResultLineChart= ResultLineChart+"[\""+qno+"\", "+tim+", \"#d12610\" ],";

}

}

int limit=ResultLineChart.lastIndexOf(",");

if(limit>0)

{

ResultLineChart=ResultLineChart.substring(0,limit);

}

System.out.print("DAo ResultLineChart \*\*\*\*\*\*\*\*\*: "+ResultLineChart);

} catch (Exception e) {

// TODO:

e.printStackTrace();

}

return ResultLineChart;

}

public String ResultQuestionType(int userid, int testtype, int testno) {

String ResultQuestionType = "";

int count = 0;

ResultSet resultSet = null;

Connection connection = null;

CallableStatement cs = null;

Statement st=null;

java.sql.PreparedStatement preparedStatement=null;

//System.out.println("USer ID"+session.getAttribute("uid"));

String qry = "select c.category\_name,(select count(\*) from br\_test\_results t where t.result like 'Y' and t.test\_no="+testno+" and t.userid="+userid+" and t.category\_id=c.category\_id ) as Correct,(select count(\*) from br\_test\_results t where t.result like 'N' and t.test\_no="+testno+" and t.userid="+userid+" and t.category\_id=c.category\_id ) as wrong from br\_test\_results t,br\_question\_type qt,br\_question\_categories c where t.test\_no="+testno+" and t.category\_id=c.category\_id and t.userid="+userid+" and t.test\_master\_id="+testtype+" group by c.category\_name";

System.out.println(qry);

try {

connection = getBRConnection();

preparedStatement=connection.prepareStatement(qry);

resultSet = preparedStatement.executeQuery();

while(resultSet.next())

{

Double wrgper = 0.0;

Double crtper = 0.0;

String cat = resultSet.getString(1);

int crt = resultSet.getInt(2);

int wrg = resultSet.getInt(3);

int tot = wrg+crt;

System.out.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Wrong : "+wrg+" Correct : "+crt);

if(crt>0){

crtper= (double) (((double)crt/(double)tot)\*100);

System.out.println("crt %"+wrgper);

}

if(wrg>0){

wrgper= (double) (((double)wrg/(double)tot)\*100);

System.out.println("Wrg %"+wrgper);

}

//ResultQuestionType= ResultQuestionType+" [\""+cat+"\", {\"v\": "+wrg+" , \"f\": \""+wrgper+"%\" }, {\"v\": "+crt+" , \"f\": \""+crtper+"%\" }],";

ResultQuestionType= ResultQuestionType+" [\""+cat+"\", {\"v\": "+wrgper+" , \"f\": \""+wrg+"\" },\""+wrgper.intValue()+"%\", {\"v\": "+crtper+" , \"f\": \""+crt+"\" },\""+crtper.intValue()+"%\"],";

}

int limit=ResultQuestionType.lastIndexOf(",");

if(limit>0)

{

ResultQuestionType=ResultQuestionType.substring(0,limit);

}

System.out.print("DAo ResultQuestionType \*\*\*\*\*\*\*: "+ResultQuestionType);

} catch (Exception e) {

e.printStackTrace();// TODO: handle exception

}

return ResultQuestionType;

}

public String ResultTimeQuestionType(int userid, int testtype, int testno) {

String ResultTimeQuestionType = "";

int count = 0;

ResultSet resultSet = null;

Connection connection = null;

CallableStatement cs = null;

Statement st=null;

java.sql.PreparedStatement preparedStatement=null;

//System.out.println("USer ID"+session.getAttribute("uid"));

String qry = "select c.category\_name,(select count(t.question\_number) from br\_test\_results t where t.category\_id=c.category\_id and t.test\_no="+testno+" ) as questioncount, (select sum(SECOND(t.user\_time)) from br\_test\_results t where t.category\_id=c.category\_id and t.test\_no="+testno+")/(select count(t.question\_number) from br\_test\_results t where t.category\_id=c.category\_id and t.test\_no="+testno+") as avgtime from br\_test\_results t,br\_question\_type qt,br\_question\_categories c where t.test\_master\_id="+testtype+" and t.category\_id=c.category\_id and t.userid="+userid+" and t.test\_no="+testno+" group by t.category\_id order by t.question\_number ";

System.out.println(qry);

try {

connection = getBRConnection();

preparedStatement=connection.prepareStatement(qry);

resultSet = preparedStatement.executeQuery();

while(resultSet.next())

{

String qtype = resultSet.getString(1);

String avtime = resultSet.getString(3);

ResultTimeQuestionType= ResultTimeQuestionType+"[ \""+qtype+"\", "+avtime+" ],";

}

int limit=ResultTimeQuestionType.lastIndexOf(",");

if(limit>0)

{

ResultTimeQuestionType=ResultTimeQuestionType.substring(0,limit);

}

System.out.print("DAo ResultTimeQuestionType \*\*\*\*\*\*\*\*\*\*: "+ResultTimeQuestionType);

} catch (Exception e) {

e.printStackTrace();// TODO: handle exception

}

return ResultTimeQuestionType;

}

public String ResultHard(int userid, int testtype, int testno) {

String ResultHard = "";

int count = 0;

ResultSet resultSet = null;

Connection connection = null;

CallableStatement cs = null;

Statement st=null;

java.sql.PreparedStatement preparedStatement=null;

//System.out.println("USer ID"+session.getAttribute("uid"));

String qry = "select d.diff\_name,(select count(\*) from br\_test\_results t where t.result like 'N' and t.diff\_id=d.diff\_id and t.test\_no="+testno+" and t.userid="+userid+") as Wrong ,(select count(\*) from br\_test\_results t where t.result like 'Y' and t.diff\_id=d.diff\_id and t.test\_no="+testno+" and t.userid="+userid+" ) as Correct from br\_test\_results t,br\_difficulty\_level d where t.test\_master\_id="+testtype+" and t.test\_no="+ testno+ " and d.diff\_id=t.diff\_id and t.userid="+userid+" group by t.diff\_id ";

System.out.println(qry);

try {

connection = getBRConnection();

preparedStatement=connection.prepareStatement(qry);

resultSet = preparedStatement.executeQuery();

while(resultSet.next())

{

Double wrgper = 0.0;

Double crtper = 0.0;

String hard = resultSet.getString(1);

int crt = resultSet.getInt(2);

int wrg = resultSet.getInt(3);

int tot = wrg+crt;

System.out.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Wrong : "+wrg+" Correct : "+crt);

if(crt>0){

crtper= (double) (((double)crt/(double)tot)\*100);

System.out.println("crt %"+wrgper);

}

if(wrg>0){

wrgper= (double) (((double)wrg/(double)tot)\*100);

System.out.println("Wrg %"+wrgper);

}

//ResultQuestionType= ResultQuestionType+" [\""+cat+"\", {\"v\": "+wrg+" , \"f\": \""+wrgper+"%\" }, {\"v\": "+crt+" , \"f\": \""+crtper+"%\" }],";

ResultHard= ResultHard+" [\""+hard+"\", {\"v\": "+wrgper+" , \"f\": \""+wrg+"\" },\""+wrgper.intValue()+"%\", {\"v\": "+crtper+" , \"f\": \""+crt+"\" },\""+crtper.intValue()+"%\"],";

// ResultHard= ResultHard+" [\""+hard+"\", {\"v\": "+wrg+" , \"f\": \""+wrgper+"%\" }, {\"v\": "+crt+" , \"f\": \""+crtper+"%\" }],";

}

int limit=ResultHard.lastIndexOf(",");

if(limit>0)

{

ResultHard=ResultHard.substring(0,limit);

}

System.out.print("DAo ResultHard \*\*\*\*\*\*\*\*\*\*\*\* : "+ResultHard);

} catch (Exception e) {

e.printStackTrace();// TODO: handle exception

}

return ResultHard;

}

//\*\*\*\*\*\*\*\*\*\*\*\* For Result Page \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//For Doubts Chart \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}