# MTS9207 Major assignment report

## Task 1 & Task 2

In the progress of completing the assignment, the first thing to do is to study the examples that lecture has provide us. It is a simple program that using recorded account and password combination that can be used to get access to student marks info. Based on that program we can develop our own program.

The asp components are similar with the html, only slight different. Thus, it will not be discuss here but will be attached to the appendix.

Thanks to Ado, to get access to database we only need to create connection to database. For convenient purpose we set alias to connection, command and datareader.The code of connecting to database is like the code:

OleDbConnection conn = null;

OleDbCommand cmd = null;

OleDbDataReader reader = null;

string sqlCommand;

conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data Source="

+ Server.MapPath("pcparts.accdb"));

Whereas we are using access database and the name of the database is pcparts.

Before using database we need to open the connection:

conn.Open();

and followed by sql command which is used to do queries in database, in this case the login program is looking for wither the username and the password combination fit the record of the database the command would like:

sqlCommand = "SELECT \* FROM customers WHERE username='";

sqlCommand = sqlCommand + txtUserName.Text;

sqlCommand = sqlCommand + "' AND psw='" + txtPassword.Text+"'";

this is multi-lined command which can be put in one line as:

sqlCommand = ”SELECT \* from customers where username=’”+txtUserName.Text+”’ And psw=’”+txtPassword.Text+”’”;

The “’” are used as these content is string as content in SQL.

After defines sql command we need to execute it:

cmd = new OleDbCommand(sqlCommand, conn);

reader = cmd.ExecuteReader();

reader helps us to get the content from database.

It is good idea to use catch to receive system feedback in case there are some errors in out program:

catch (Exception ex)

{

lblStatus.Text ="An error occured: "+ ex.Message;

}

Finally, always remember to free resources and close the connection:

finally

{

if (reader!=null)

reader.Close();

if (conn!=null)

conn.Close();

}

The logic here is that each user should choose either log in to his account or register as new customer before accessing the database. Thus, all page except register and log in page are redirected to log in page based on the session variables which record the log in status of the user.

We are going to create a signup page for new customers, the method here is to change sql command from select to insert some record to the existing database. And as for new customer the essential two keys are username and password both of these keys are stored in the customer tables. Thus, using ‘ Insert into customer (username,password)’ command and change the values of these keys. The whole command would be “Insert into customers (username,password) values(‘newusername’,’newpasswrod’)”. “newusername” and “newpassword” here is get by the users input which are transferred from the textbox to get these values use ID.text to receive values from html commpont. ID in the circumstances are the ID of these two textboxs.

However, while this command always returns an error as SQL sentence “insert into” syntax error, which has been fixed by changing the “password” key into “psw” and the error disappear and the website functions well. I have not found out why this error keeps coming as I may guess it could be cause by access conflict.

Register part is like this:

protected void Button1\_Click(object sender, EventArgs e)

{

OleDbConnection conn = null;

OleDbCommand cmd = null;

OleDbDataReader reader = null;

string sqlCommand;

try

{

conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data Source="

+ Server.MapPath("pcparts.accdb"));

conn.Open();

sqlCommand = "Insert into customers (username,psw) Values('"+newAccount.Text+"','"+newPass.Text+"')";

cmd = new OleDbCommand(sqlCommand, conn);

cmd.ExecuteNonQuery();

//check if the user has been exist

reader = cmd.ExecuteReader();

}

Different from log in, we need to insert a new valid user into the customers table.

The logic of my register is to direct the user to the login page as long as the registration is complete. Thus, in the finally part it will redirect to default.apsx.

finally

{

if (reader != null)

reader.Close();

if (conn != null)

conn.Close();

Response.Redirect("Default.aspx");

}

“Object reference not set to an instance of an object.” Is one of the problem that I have facing when I was trying to input data into the database, and still I have not found out what happens as the error disappear by itself.

## Task 3

The 4 features that I am using is slider, draggable, autocomplete and tooltips. Draggable can be used to drag the result that returns from the server, and for future uses, it can combine with drop to make a convenient data deleting gesture control. As for the autocomplete function, it can be used to aid customer to input record fast and correct. A list of items and words which are related to the orders are saved in availableTags array, each time while user is input the record, the computer is going to compare it with this array to find the most likely word. Tooltips, as it literal means when you point you mouse to an element in your website, the title attribute will display under your mouse, which can be used to guide user to input correct info into correct spaces.

Before we using jqury and jqury-ui it is necessary to import these libraries:

<script src="https://code.jquery.com/jquery-1.12.4.js"></script>

<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

Draggeable is a feature of jqury-ui, with few predefine it is easy to apply it into our website component.

<script>

$(function () {

$("#GridView1").draggable();

</script>

“#GridView1” is the element that I want it to have the functionality, it can be change to any element you want.

Slider is a bar that can be towed from in left and right direction. To applies it:

$(function () {

var handle = $("#custom-handle");

$("#slider").slider({

create: function () {

handle.text($(this).slider("value"));

},

slide: function (event, ui) {

handle.text(ui.value);

}

});

});

Autocomplete can be used to aiding user input, in case that user cannot recall the keywords. The mechanism in autocomplete is to build an array to store all the keywords and make comparison each time user input some letters:

$(function () {

var availableTags = [

"brand",

"amount",

"specs",

"price"

];

$("#SearchID").autocomplete({

source: availableTags

});

$("#key").autocomplete({

source: availableTags

});  
Tooltips is another reminder for users every time when user put their mouse over an element the title of that element show up under the mouse. To apply this function is similar as draggable just like:

$(function () {

$(document).tooltip();

});

## Task 4

a) The back-forth program is using jquery animation to move a block from left to right, then right to left it is using the callback method of animation function to fulfil this purpose. To make it up and down moving just change the position ‘left’ to ‘top’ which means the distance between the block and the top borders. And to control two boxes to make same movement just need to write another moving program in and make a new block class name is ‘block1’, while slide function was called make two boxes get same command, which makes two boxes moves the same way. And use stop() function to stop boxes moving.

To make block moves, just set the animate first argument of the direction, for example Left:200px, means move to the place that are the 200px of the screen. Thus, to make it up and down move change it to:

$("#block").animate({top: "200px"}, 2000, function() {slideUp();

});

2000 is the time that block need to move to the destination. While the function is callback function that will going to be functional after the animation finished.

To control two blocks the most easy way is to give the moving function two selected blocks as below:

var slideDir = "d";

function slideDown() {

slideDir = "d";

// slide to right

$("#block").animate({

top: "200px"

}, 2000, function() {

slideUp();

});

$("#block1").animate({

top: "200px"

}, 2000, function() {

slideUp();

});

}

function slideUp() {

slideDir = "u";

// slide to right

$("#block").animate({

top: "0px"

}, 2000, function() {

slideDown();

});

$("#block1").animate({

top: "0px"

}, 2000, function() {

slideDown();

});

}

These two defined functions make loop to running all the time. But to make it works, we need to call the function first.

$(document).ready(function() {

$("#start").on("click", function() {

if (slideDir == "d") {

slideDown();

} else {

slideUp();

}

});

The function was triggered by a button, once it was clicked the slideDown and slideUp function will going to work dependably.

Also, we are required to use a button to stop the moving of the two buttons. To accomplish this goal, stop() function has been used.

$("#stop").on("click", function() {

$("#block").stop(true);

$("#block1").stop(true);

});

And since we need to control two blocks, remember to add two selected blocks.

b) as we can see from the source code and the page runs by the browser that there could be some delay for block to move to another side, this is caused by the element that was chosen to move was random. Thus, when an element is in the position that we need it to move to, the program will wait for Time of Interval that we set for the element to move to the destination. The way to change this is simple, we add a pending process for each time when we need the block to move. If the block is on the left half of the distance that block need to move, we make it move to right and vice versa. This can help to get rid of the delay between blocks movement and also can be achieved by one function.

The lagging of this program was caused by the method it used to make the blocks moves, once your generate a random block, the position of such block can be anywhere. And the animate function can only move to certain location, so to fix this problem. The easiest way is to add a pending code like if(pos<500) than move to right and vice versa. The pos stand for the position of the block which can be acquired by position() method. The complete code should be like this: var Stopped=false;

var TimeInterval=1000; //milli-second

var b=["#a", "#b", "#c", "#d" ,"#e" ,"#f"]

function forward(){

i=Math.floor(Math.random() \* 6);

var pos=$(b[i]).position().left;

if(pos<500)

{

$(b[i]).animate({

left:'1000px',

},TimeInterval,forward);

}

else {

$(b[i]).animate({

left:'0px',

},TimeInterval,forward);

}}

$(document).ready(function(){

$("#m").click(forward);

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

TimeInterval=TimeInterval\*0.9

});

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

TimeInterval=TimeInterval/0.9

});

});

## Task 5

A plugin in jquery is similar with prototype. It can use prototype`s own property and inherited property that has been used in prototype. We may consider jquery plugin is just a new method we used to extend jquery`s prototype object. Basically, plugins can be created by programmers and like the roles prototype in java. It broadens the extension of prototype.

As said above, plugin likes an extension of prototype and it can be used as prototype as well. First we defined an extension of testPlugin, it can be treated as a new method called testPlugin, the functionality of this plugin is to move items left and right.

(function(){

$.fn.testPlugin=function(){

return this.each(function(){

var side=$(this);

var slideDir = "d";

function slideRight() {

slideDir = "r";

// slide to right

side.animate({

left: "1000px"

}, 1000, function() {

slideLeft();

});

};

function slideLeft() {

slideDir = "l";

// slide to right

side.animate({

left: "0px"

}, 1000, function() {

slideRight();

});

};

if (slideDir == "r") {

slideRight();

} else {

slideLeft();

};

})}

}());

To move certain items it just let these items to use testPlugin() method, the code is listed below: $(document).ready(function() {

var laptop=["#img1","#img2","#img3","#img4","#img5"];

for(var i=0;i<laptop.length;i++){

$(laptop[i]).testPlugin();}}

);

Laptop is an array that store the images that need to be moves.

Task 6

1. first we need to gather the info from xml the format like: {“table\_name”:[‘{“key1”:”values1”}’,’{“key2”:”values2”}’]},then we need to change the text into java object by using JSON.parse() function we can finish this process. Notice that we need to put all these data into a HTML table, thus import a new variable to generate <table> tag for us and we need </table> tag after all data was input into the table. The JSON object can be acquired by using obj.employees method to get the data. Using for loop to get all data and remember to add <tr><td> tag to make the table complete.

First we make a JSON text as :

Text='{"employees":['+

'{"firstName":"John", "lastName":"Doe"},'+

'{"firstName":"Anna","lastName":"Smith"},'+

'{"firstName":"Peter", "lastName":"Jones"}]}';

1. This question has been explained by above question. Using JSON.parse() to transfer text elements into JSON object and using the features of object to shows the result in a HTML table.

Then use JSON.parse(Text) to transfer it to a JSON object.

var obj=JSON.parse(text);

To output these items into a table we can use a variable to create a table :

structure+="<table border='solid'>";

for (var i=0;i<obj.employees.length;i++){

structure+="<tr><td>"+obj.employees[i].firstName+"."+obj.employees[i].lastName+"</td></tr>";

};

structure+="</table>";

document.getElementById("p1").innerHTML=structure;

p1 is a paragraph element that can be used to put the table.

1. Creating 400 boxes, it is easy to think of using loop to manipulate generate and control such amount of element. To make boxes overlapping by each box is 1inch above another, just alters next box`s top attribute. And the moving speed can be controlled by changing middle arguments of Jquery`s animate function. Moreover, to make boxes moving, use jquery selector to select each box and a Moving function to make it easier to be use. As there is no control requirement simple make the Moving function to callback itself all the time. One thing should be keep in mind is that a Moving function should be used in the loop where you create and manipulate these boxes to keep the consistency of the box moving. Each time a new box is generate, it should be moving right away, so that lagging between boxes can be reduced.

To generate 400 boxes, use a for loop and jquery command to make these blocks. And using ccs() method to change it`s top attribute to make them overlapping:

for(var i=0;i<size;i++){

var div={

element:$('<div style="height:100px;width:100px;position:absolute;"></div>')

}

div.element.css('top',i+'px');

To choose these blocks, use an array divs to record them:

var div=divs[i].element;

To make the block has different moving speed, make the time animate need to be a changing varieables.

div.animate({left:'1000px'},time+50\*i,function() {

div.animate({left:'0px'},time+50\*i);

And to make these block can be distinguish from others, make them different colours.

function randomColor(args){

r=Math.floor(Math.random()\*255);

g=Math.floor(Math.random()\*255);

b=Math.floor(Math.random()\*255);

rgb="rgb("+r+","+g+","+b+")";

args.css("background-color",rgb);

}

## Appendix

Default.aspx

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="\_Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

<script src="https://code.jquery.com/jquery-1.12.4.js"></script>

<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

<script>

$(function () {

$("#menu").menu();

});

</script>

<style>

.ui-menu { width: 150px; }

</style>

</head>

<body>

<ul id="menu">

<li><a href="Default.aspx">home</a></li>

<li><a href="register.aspx">registration</a></li>

<li><a href="editing.aspx">edit orders</a></li>

<li><a href="showOrders.aspx">show orders</a></li>

</ul>

<form id="form1" runat="server">

<div>

<asp:Label ID="lblStatus" runat="server"></asp:Label>

<br />

<asp:Label ID="Label2" runat="server" Text="Username:"></asp:Label>

&nbsp;<asp:TextBox ID="txtUserName" runat="server"></asp:TextBox>

<br />

Password:&nbsp;

<asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>

<br />

<asp:Button ID="Button1" runat="server" OnClick="Button1\_Click" Text="Login" />

<asp:Button id="Button2" runat="server" OnClick="Button2\_Click" Text="Regist Now"/>

</div>

</form>

</body>

</html>

Editing.aspx

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="\_Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

<script src="https://code.jquery.com/jquery-1.12.4.js"></script>

<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

<script>

$(function () {

$("#menu").menu();

});

</script>

<style>

.ui-menu { width: 150px; }

</style>

</head>

<body>

<ul id="menu">

<li><a href="Default.aspx">home</a></li>

<li><a href="register.aspx">registration</a></li>

<li><a href="editing.aspx">edit orders</a></li>

<li><a href="showOrders.aspx">show orders</a></li>

</ul>

<form id="form1" runat="server">

<div>

<asp:Label ID="lblStatus" runat="server"></asp:Label>

<br />

<asp:Label ID="Label2" runat="server" Text="Username:"></asp:Label>

&nbsp;<asp:TextBox ID="txtUserName" runat="server"></asp:TextBox>

<br />

Password:&nbsp;

<asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>

<br />

<asp:Button ID="Button1" runat="server" OnClick="Button1\_Click" Text="Login" />

<asp:Button id="Button2" runat="server" OnClick="Button2\_Click" Text="Regist Now"/>

</div>

</form>

</body>

</html>

Register.aspx

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="\_Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

<script src="https://code.jquery.com/jquery-1.12.4.js"></script>

<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

<script>

$(function () {

$("#menu").menu();

});

</script>

<style>

.ui-menu { width: 150px; }

</style>

</head>

<body>

<ul id="menu">

<li><a href="Default.aspx">home</a></li>

<li><a href="register.aspx">registration</a></li>

<li><a href="editing.aspx">edit orders</a></li>

<li><a href="showOrders.aspx">show orders</a></li>

</ul>

<form id="form1" runat="server">

<div>

<asp:Label ID="lblStatus" runat="server"></asp:Label>

<br />

<asp:Label ID="Label2" runat="server" Text="Username:"></asp:Label>

&nbsp;<asp:TextBox ID="txtUserName" runat="server"></asp:TextBox>

<br />

Password:&nbsp;

<asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>

<br />

<asp:Button ID="Button1" runat="server" OnClick="Button1\_Click" Text="Login" />

<asp:Button id="Button2" runat="server" OnClick="Button2\_Click" Text="Regist Now"/>

</div>

</form>

</body>

</html>

Showorders.aspx

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="\_Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

<script src="https://code.jquery.com/jquery-1.12.4.js"></script>

<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

<script>

$(function () {

$("#menu").menu();

});

</script>

<style>

.ui-menu { width: 150px; }

</style>

</head>

<body>

<ul id="menu">

<li><a href="Default.aspx">home</a></li>

<li><a href="register.aspx">registration</a></li>

<li><a href="editing.aspx">edit orders</a></li>

<li><a href="showOrders.aspx">show orders</a></li>

</ul>

<form id="form1" runat="server">

<div>

<asp:Label ID="lblStatus" runat="server"></asp:Label>

<br />

<asp:Label ID="Label2" runat="server" Text="Username:"></asp:Label>

&nbsp;<asp:TextBox ID="txtUserName" runat="server"></asp:TextBox>

<br />

Password:&nbsp;

<asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>

<br />

<asp:Button ID="Button1" runat="server" OnClick="Button1\_Click" Text="Login" />

<asp:Button id="Button2" runat="server" OnClick="Button2\_Click" Text="Regist Now"/>

</div>

</form>

</body>

</html>