

Using jQuery in ASP.NET 4.0

One of the most important feature of using ASP.NET is the integration of jQuery (its now official). When an ASP.NET 4.0 Web Application or Web Site is created using VS2010, you will get 'Scripts' folder where all jQuery files are stored. In this lab we will see various implementation of using jQuery in ASP.NET 4.0. jQuery allows us to develop ASP.NET 4.0 web applications which are really browser independent.

STEP 1

For this Lab Open VS2010 and create an ASP.NET 4.0 Project, name it as 'WebApp_jQuery'. In this Project add a new folder; name it as 'ASPNET_jQuery'.

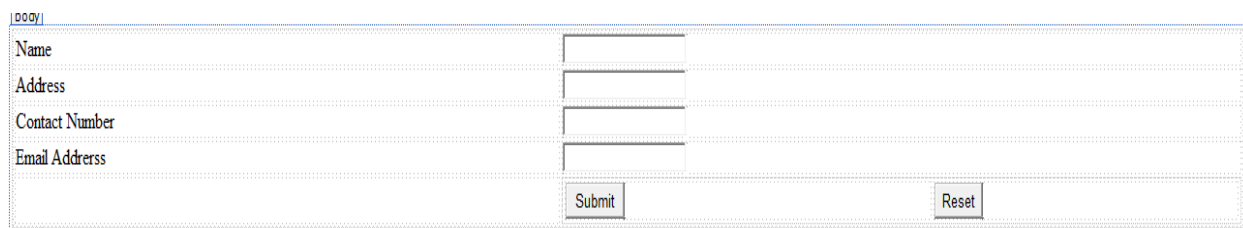
Working with ASP.NET 4.0 Controls using jQuery

Exercise 1: Setting the TAB for the first TextBox in ASP.NET Page.

This is one of the frequent requirements while developing a Web Application for the user who is more comfortable of using ENTER key to navigate (TAB) through the Textboxes and by default the first Textbox must be having Focus set to it.

Task 1: In the 'ASPNET_jQuery' folder add a new Web Form, name it as 'WF_Auto_Fcous_TextBox_Enter_Key_Tab.aspx'.

Task 2: Design the WebForm as below:

The image shows a screenshot of a web form in design view. It contains four text input fields stacked vertically, labeled 'Name', 'Address', 'Contact Number', and 'Email Address'. Below these fields are two buttons: 'Submit' and 'Reset'.

Task 3: In the Source of the WebForm, in the <head> tag add the below Script references (Note: In my case I have used jquery-1.4.1.js, in your case the version may change.)

```
<script src="../../Scripts/jquery-1.4.1.js" type="text/javascript"></script>
<script src="../../Scripts/jquery-1.4.1.min.js" type="text/javascript"></script>
```

The file "jquery-1.4.1.js" will provide the jQuery intellisense for VS2010 and the file "jquery-1.4.1.min.js" is the file which will be available with deployment package.

(Note: Task 3, needs to be repeated for every exercise of jQuery henceforth.)

Task 4: Now add the below jQuery code in the <script> tag.

```
<script language="javascript" type="text/javascript">
    $(document).ready(function () {
        //use the focus method to set the focus for the First TextBox
```

```

$('input:text:first').focus();
//Bind the Keydown event to every textbox.
$('input:text').bind('keydown', function (e) {
    //Check for the enter key
    if (e.which == 13) {
        //Prevent the Default form submission
        e.preventDefault();
        //increment the textbox index by one
        var nextTextBoxIndex = $('input:text').index(this) + 1;
        //Check if textbox index is completed
        if (nextTextBoxIndex == 4) {
            return;
        }
        else {
            //focus the next textbox
            $('input:text')[nextTextBoxIndex].focus();
        }
    }
});
//The Reset button Code
$('#btnReset').click(function () {
    $('form')[0].reset();
});
});
</script>

```

The 'document.ready()' method will be called when the document is loaded and the code written under it will get executed. '\$' sign represents the jQuery syntax which is used to refer the Elements. \$('input:text:first') represents the first input textbox. Using bind() method an event is defined for the control. In the above code 'keydown' event is bind with all textboxes. In this event the keydown of the 'ENTER' key is detected using the value 13. In the above code you will find that, the click event for the RESET button is added using \$('#btnReset'). Here '#' is used to refer an element with its name.

Task 5: View the WebForm in browser, you will find the first TextBox is focused and then using 'ENTER' key you can navigate through the Textboxes.

Exercise 2: Displaying values of the Selected CheckBoxes from the CheckBoxList

In this exercise, we will see how to display values of the selected checkboxes from the CheckBoxList.

Task 1: In the folder 'ASPNET_jQuery' created in STEP 1 add a new WebForm, name it as 'WF_CheckListBox_SelectedItems.aspx'. On this add a CheckBoxList of name 'lstName' and also add a <div> of name 'selectedData' as below:

WF_CheckListBox_SelectedItems.aspx X

body

☐ Tejas

☐ Mahesh

☐ Ramesh

☐ Ram

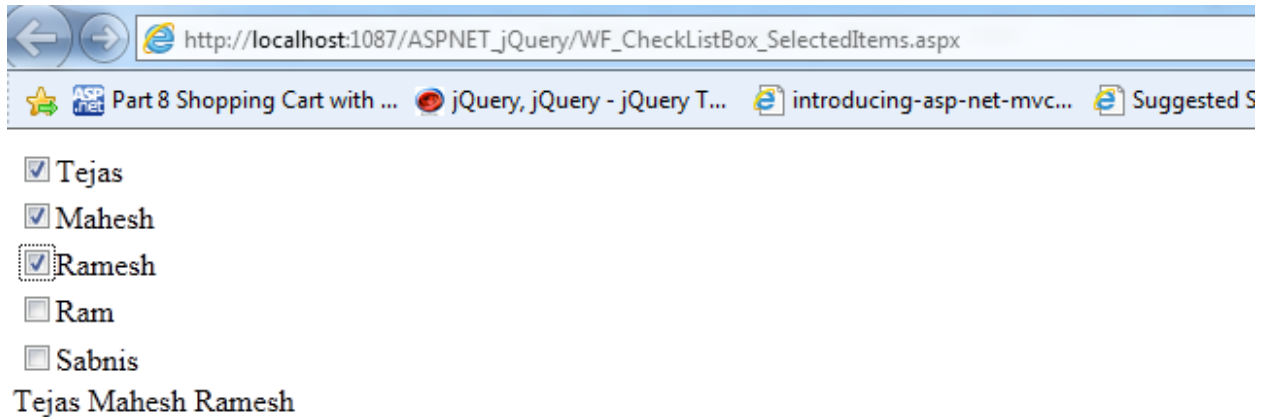
☐ Sabnis

Task 2: Add the below jQuery code in the Source:

```
<script language="javascript" type="text/javascript">
    $(document).ready(function () {
        $('#<%=lstNames.ClientID%>').click(function () {
            var data = "";
            //Use the jQuery Selector to retrieve the selected CheckBox. Here you need to make use of
            <input type="checkbox">
            //Use 'each' function to return each element
            $('#<%= lstNames.ClientID%> input[type=checkbox]:checked').each(function () {
                data = data + " " + $(this).next().text(); //Here the 'text()' is used to read the text of the
                CheckBox
            });
            $('#selectedData').text(data);
        });
    });
</script>
```

In the above case the CheckBoxList, is rendered as <table> and all check boxes gets rendered as individual <input type="checkbox">, so here the CheckBoxList Web control is referred using \$('#<%=lstNames.ClientID%>'), this is the static client ID feature of ASP.NET 4.0. Event check box is selected using \$('#<%=lstNames.ClientID%>input[type=checkbox]:checked') so that when this check box is checked its text can be read and can be displayed in 'selectedData' <div> tag.

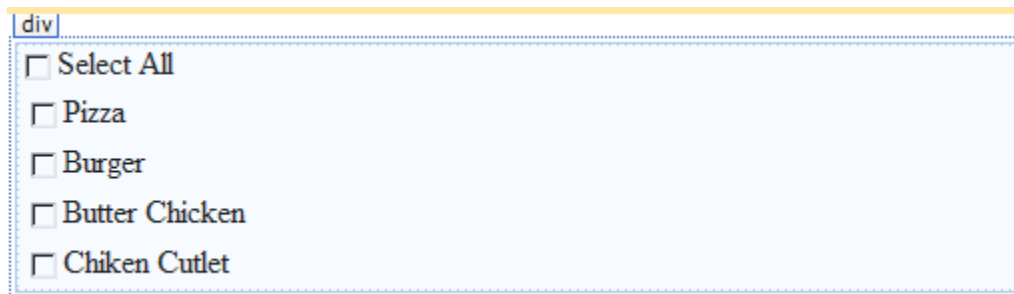
Task 3: View the page in browser and select checkbox the result will be as below:



Exercise 3: Selecting and De-Selecting CheckBoxes in CheckBoxLayout

In this exercise, we will see how to perform all Check Box selection and de-selection using one single check box.

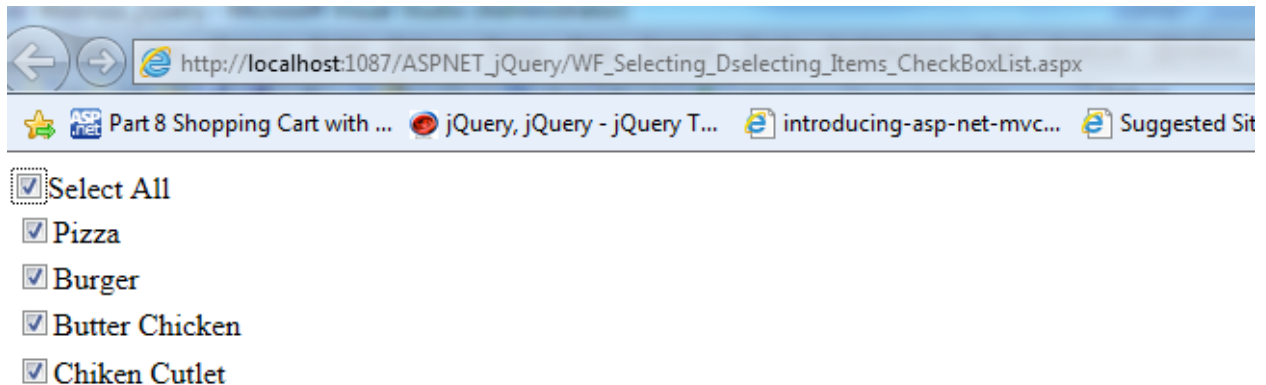
Task 1: In the folder created in STEP 1, add a new WebForm, name it as 'WF_Selecting_Dselecting_Items_CheckBoxLayout.aspx'. Add the check box of name 'chkIstSelectAll' and CheckBoxLayout of name 'IstOptionTypes' as below:



Task 2: Write the below jQuery code on the Click event of 'chkIstSelectAll'. This code will add the checked attribute for all checkboxes of the 'IstOptionTypes'.

```
<script language="javascript" type="text/javascript">
    $(document).ready(function () {
        $('#chkIstSelectAll').click(function () {
            //Here the jQuery selector is used to retrieve all checkboxed from the List and the selection
            //is toggled based upon the 'Checked' status of the SeelctAll CheckBox
            $("#<%=IstOptionTypes.ClientID%>
input[type='checkbox']").attr('checked',$('#chkIstSelectAll').is(':checked'));
        });
    });
</script>
```

Task 3: View the WebForm in browser and check the 'Select All' checkbox ypu will find all check boxes in the list gets checked.



Working with ASP.NET Controls Validation using jQuery.

In web application validation plays a very important role. But since ASP.NET server controls gets executed on Server side we need to write the validation logic on Server side using C# or VB.NET languages. If for every control if the page is posted back then this will take huge time to process the entire page and check the dirty inputs. Thankfully in ASP.NET we have been provided with Validation controls using which most of the validation operation can be performed easily. But still in ASP.NET there are several needs for performing validation e.g. number of items selected in ListBox or validating the users profile etc. To perform such operation on client side we need to write huge and complex JavaScript. Using jQuery this operation can be performed easily.

STEP 2:

In the project created in STEP 1, add a new folder, name it as 'ASPNET_Jquery_Control_Validation'.

Exercise 4: Validating number of CheckBoxes selected in CheckBoxList.

Task 1: In the folder created in STEP 2, add a new WebForm, name it as 'WF_Validations_CheckBoxList.aspx'.

Task 2: Add a CheckBoxList, Button and CustomValidator of name 'chklstCities', 'btnSubmit' and 'CustomValidator1' respectively.

WF_Validations_CheckBoxList.aspx X

form#form1

Select at least 3 Cities:

☐ Pune

☐ Mumbai

☐ Bangalore

☐ Hyderabad

☐ Channai

Submit

Please select at least 3 Cities

Task 3: Add the jQuery.validate.js in the WebForm as below:

```
<script src="../../Scripts/jquery.validate.js" type="text/javascript"></script>
```

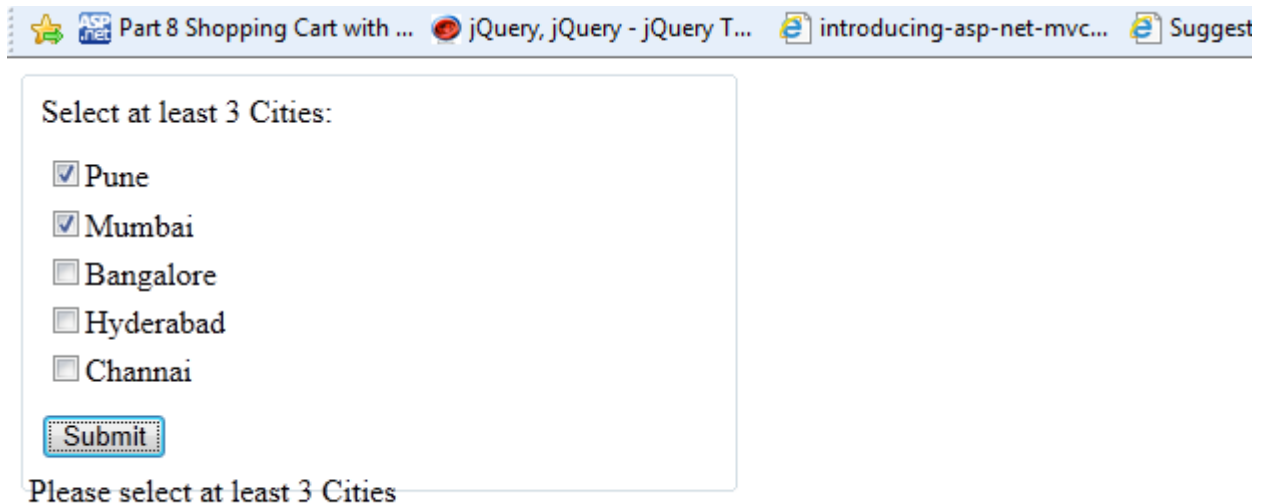
(Note: Henceforth you need to make use of the above script for all exercises of validations.)

Task 3: Set the 'ClientValidationFunction' property for the Custom validator to 'chklatCities_Validation' and add the below code.

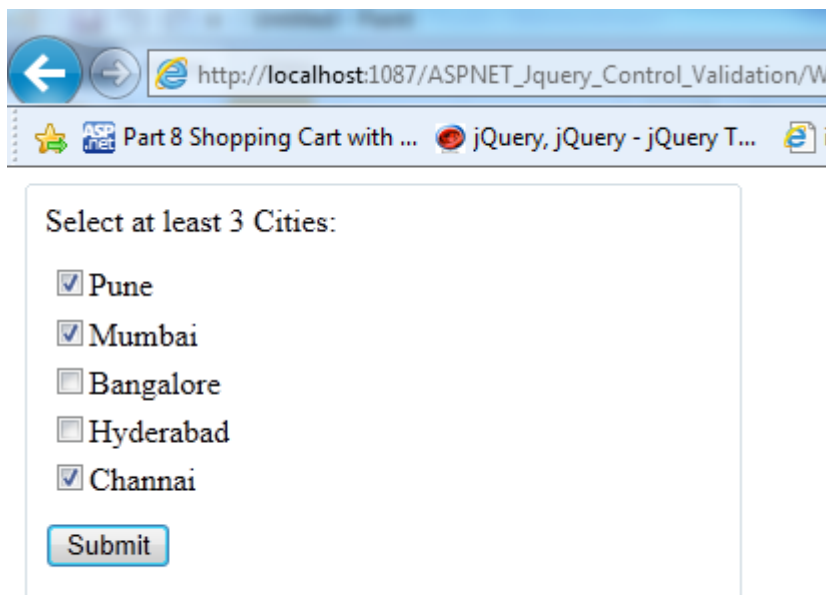
```
<script language="javascript" type="text/javascript">
function chklatCities_Validation(sender, args) {
    args.IsValid = false;
    var selectCount = $("#chklatCities input:checked").length;
    if(selectCount >= 3) {
        args.IsValid = true;
    }
}
</script>
```

The '\$(#chklatCities input:checked).length' will return the number of Check Box selected from the List.

Task 4: View the WebForm in browser, and select check boxes if the count is more than 3 the page will be displayed otherwise on the click of the error message will be displayed.



After selecting the 3 checkboxes the error message will be disappeared and the page will be submitted.



Exercise 5: Limiting Number of Characters in the TextBox.

In this exercise we will define validations on the number of characters entered in the TextBox.

Task 1: Add a WebForm in the folder created in STEP 2, name it as 'WF_Character_Limitations_Multiline_TextBox.aspx'. Add a Multiple TextBox and TextBox on the WebForm. Name them as 'txtData' and 'txtNumverOfCharacters' respectively. Also add a <div> of name 'errmsg' on the WebForm.

Task 2: Write the below jQuery code in Source:

```

<script language="javascript" type="text/javascript">
    $(document).ready(function () {
        var minCount = 0;
        var maxCount = 200;

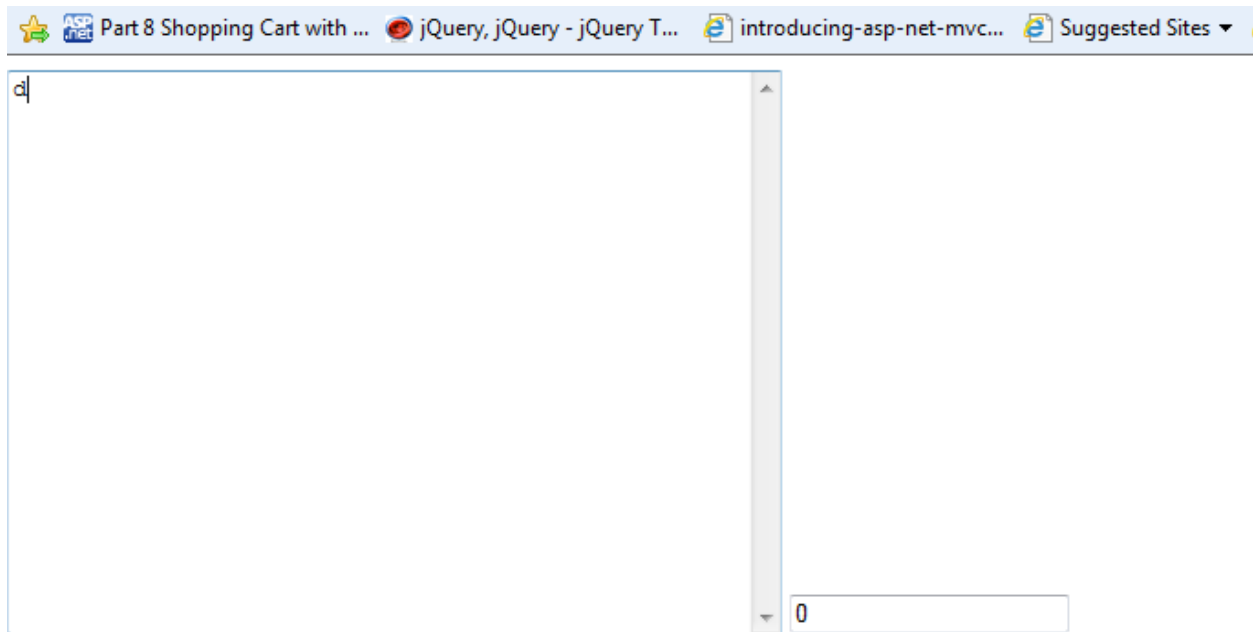
        $("#txtData").bind("cut copy paste", function (e) {
            e.preventDefault();
        });
        $("#txtData").keypress(function () {
            var textCount = $("#txtData").val().length;
            $("#txtNumberOfCaharacters").val(textCount);
            if (textCount < 5 || textCount > 200) {
                $("#errmsg").text("No. of Characters in the TextBox should be in between 5 to 200");
            }
            else {
                $("#errmsg").text(" ");
            }
        });
    });
</script>

```

The above code will check the number of characters typed in the TextBox 'txtData' using keypress event, the length of the characters entered in the textbox will be returned using '\$("#txtData").val().length'. The 'txtNumberOfCaharacters' will display the characters count in it using '\$("#txtNumberOfCharacters").val(textCount)'. If the characters entered in the textbox are not in min to max limit the the error message will be displayed.

Task 3: View the WebForm in browser and enter some text if the condition is invalid the error message will be displayed:

Error Message when the number of characters are below 5:



No. of Characters in the TextBox should be in between 5 to 200

The same error message will be displayed when the max count is reached and you still continue to enter data in TextBox.

Exercise 6: Validating the Date Range.

This is one of the most frequently required functionality while developing Web Applications. Here jQuery providers has introduced jQuery UI elements, which you can download from <http://jqueryui.com/download>. The page will be as below:



From here you can download the UI effects elements for an App development which are present in jquery-ui-1.8.16.custom.zip. Unzip it and in the 'jquery-ui-1.8.16.custom\js' path you will find 'jquery-ui-1.8.16.custom.min.js' and 'jquery-1.6.2.min.js'. You need to add these files in the application we are creating in 'Scripts' folder.

Task 1: In the folder created in STEP 2 add a new WebForm, name it as 'WF_Validation_Date_Range.aspx'.

Task 2: Design the form as below:

Task 3: Refer the above added js files in the <script> as below:

```
<script src="..\Scripts/jquery-1.6.2.min.js" type="text/javascript"></script>
<script src="..\Scripts/jquery-ui-1.8.16.custom.min.js" type="text/javascript"></script>
```

Task 4: Add the below jQuery code:

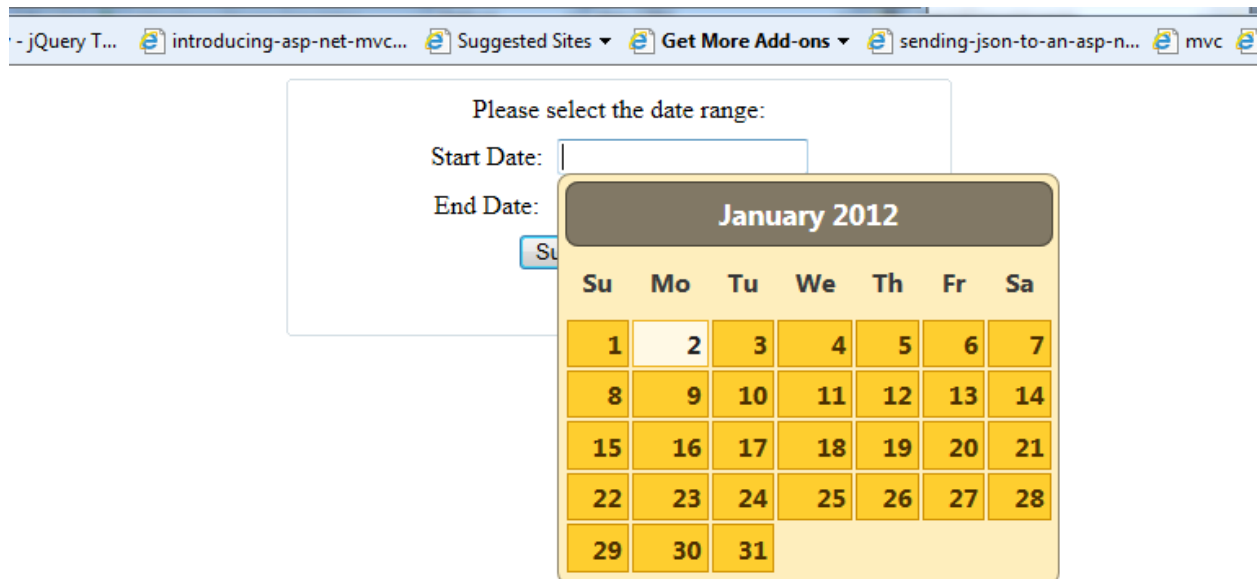
```
<script type="text/javascript" language="javascript">
$(document).ready(function () {
    $("#txtStartDate").datepicker();
    $("#txtEndDate").datepicker();
    $("#btnSubmit").click(function (e) {
        e.preventDefault();

        var startDate = new Date();
        startDate = Date.parse($("#txtStartDate").val());

        var endDate = new Date();
        endDate = Date.parse($("#txtEndDate").val());

        if (startDate > endDate) {
            alert("End Date must be greater than start date");
        }
    });
    $("#btnReset").click(function (e) {
        e.preventDefault();
        $("#txtStartDate").val("");
        $("#txtEndDate").val("");
    });
});
</script>
```

The 'datepicker()' function is defined for the TextBoxes, this will show the DatePicker when the user click on the TextBox as below:



In the code the Date() instance is created using

```
var startDate= new Date();
```

statement. Using Date.Parse() method the Date selected using the DatePicker is assigned to these date objects. If the End Date is less than the start date then the error message will be displayed.

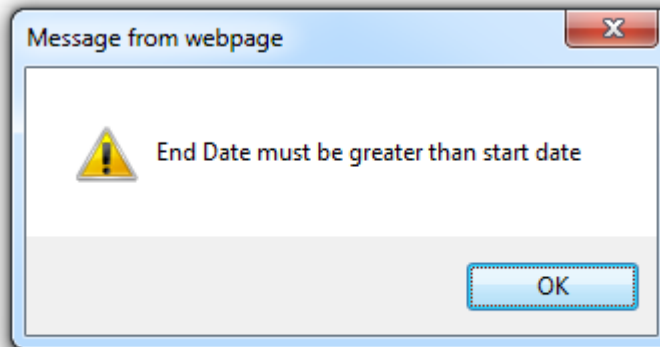
Task 5: View the WebForm in browser, select invalid date condition i.e. start date is greater than end date, and click on submit button, the below error message will be displayed:



Please select the date range:

Start Date:

End Date:



Exercise 7: Performing the User Profile Validations.

In this lab we will validate the basic user profile information e.g. Name, Email, Date of Birth etc using Forms validations. For this lab we need jQuery Form plugin which we need to download from <http://archive.plugins.jquery.com/>. From this we get jquery.form.js. The form validation a 'validate()' method. This allows us to define validation rules, error message etc.

Task 1: Add a new WebForm created in folder STEP 2. Name it as 'WF_Validate_Basic_User_Profile_Type.aspx'. Define the UI using the below Html and also add css as below:

```
<style media="screen" type="text/css">
    .header
    {
        background-color:Gray;
```

```

        font-weight:bold;
        color:White;
        text-align:center;
    }
    .errorContainer
    {
        display:none;
    }
    .alertMsg
    {
        margin-left:20px;
        color:#660000;
    }
    .mandatory
    {
        color:#FF0000;
    }
    .input-highlight
    {
        background-color:#CCCCCC;
    }
</style>

```

Add the UI controls as below:

```

<body>
    <form id="form1" runat="server">
        <div align="center" class="errorContainer">
            <fieldset style="width:550px;">
                <p align="left" class="alertMsg">
                    There was an error processing your request. Please correct the
                    following to proceed:
                </p>
            </fieldset>
        </div>
        <div>
            <fieldset style="width:350px;height:400px;">
                <table border="0" cellpadding="3" cellspacing="3">
                    <tr><td colspan="2" class="header">REGISTER NEW USER</td></tr>
                    <tr><td align="right">Name<span class="mandatory">*</span></td>
                        <td align="left">
                            <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
                        </td></tr>
                    <tr><td align="right">Email<span class="mandatory">*</span></td>
                        <td align="left">
                            <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>
                        </td></tr>
                    <tr><td align="right">Password<span class="mandatory">*</span></td>
                        <td align="left">
                            <asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>
                        </td></tr>
                </table>
            </fieldset>
        </div>
    </form>

```

```

<tr><td align="right">Confirm Password<span class="mandatory">*</span></td>
<td align="left">
<asp:TextBox ID="txtConfirmPwd" runat="server" TextMode="Password"></asp:TextBox>
</td></tr>
<tr><td align="right">Date of Birth<span class="mandatory">*</span></td>
<td align="left">
<asp:TextBox ID="txtDOB" runat="server"></asp:TextBox>
</td></tr>
<tr><td align="right">Address 1<span class="mandatory">*</span></td>
<td align="left">
<asp:TextBox ID="txtAddress1" runat="server"></asp:TextBox>
</td></tr>
<tr><td align="right">Address 2</td>
<td align="left">
<asp:TextBox ID="txtAddress2" runat="server"></asp:TextBox>
</td></tr>
<tr><td align="right">Postal Code<span class="mandatory">*</span></td>
<td align="left">
<asp:TextBox ID="txtPostal" runat="server"></asp:TextBox>
</td></tr>
<tr><td align="right">Website</td>
<td align="left">
<asp:TextBox ID="txtWebsite" runat="server"></asp:TextBox>
</td></tr>
<tr><td colspan="2">
<asp:CheckBox ID="chkTandC" runat="server" />
I accept the Terms and Conditions</td></tr>
<tr><td colspan="2" align="center">
<asp:Button ID="btnSubmit" runat="server" Text="Submit" />
&nbsp;<asp:Button ID="btnReset" runat="server" Text="Reset" /></td></tr>
</table>
</fieldset>

</div>
</form>
</body>

```

The UI will be as below:

WF_Validate_Basic...r_Profile_Type.aspx X

body

REGISTER NEW USER

Name*

Email*

Password*

Confirm Password*

Date of Birth*

Address 1*

Address 2

Postal Code*

Website

☐ [chkTandC] I accept the Terms and Conditions

Submit Reset

Task 2: Add the Form validation js file in the <script> as below:

```
<script src="../Scripts/jquery.form.js" type="text/javascript"></script>
```

Task 3: Now add the jQuery code for validation as below:

```
<script type="text/javascript" language="javascript">
$(document).ready(function () {
    $.validator.defaults({
        highlight: function (input) {
            $(input).addClass("input-highlight");
        },
        unhighlight: function (input) {
            $(input).removeClass("input-highlight");
        }
    });

    var validator = $("#form1").validate({
        rules: {
            txtName: "required",
            txtEmail: { required: true, email: true },
            txtPassword: { required: true, minlength: 8 },
            txtConfirmPwd: { required: true, minlength: 8, equalTo: "#txtPassword" },
            txtDOB: { required: true, date: true },
            txtAddress1: { required: true, minlength: 100 },
            txtAddress2: { maxlength: 100 },
            txtPostal: { required: true, digits: true },
            txtWebSite: { url: true },
        }
    });
});
```

```

        chkTandC: "required"
    },
    messages: {
        txtName: "Please enter your Name",
        txtEmail: {
            required: "Please enter your Email",
            email: "Please enter a valid email address"
        },
        txtPassword: {
            required: "Please enter your Password",
            minlength: "Password should be at least 8 characters long"
        },
        txtConfirmPwd: {
            required: "Please reenter your Password to confirm",
            minlength: "The Confirm Password should be at least 8
characters long",
            equalTo: "The entered password and confirm password should
match"
        },
        txtDOB: {
            required: "Please enter your Date of Birth",
            date: "Please enter a valid date"
        },
        txtAddress1: {
            required: "Please enter your Mailing Address",
            maxlength: "Address can be upto maximum 100 characters"
        },
        txtAddress2: {
            maxlength: "Address can be upto maximum 100 characters"
        },
        txtPostal: {
            required: "Please enter the Postal Code",
            digits: "Please enter a valid postal code"
        },
        txtWebsite: {
            url: "Please enter a valid URL"
        },
        chkTandC: {
            required: "Please accept the Terms & Conditions to proceed"
        }
    },
    wrapper: 'li',
    errorContainer: $("#div.errorContainer"),
    errorLabelContainer: $("#form1 p.alertMsg")
});

$("#btnReset").click(function(e) {
    e.preventDefault();
    $("#form1").resetForm();
});
</script>

```


The validate() method on the form1 element defines rules and messages for the data validations. The form will be submitted only when the valid data is entered.

Task 4: View the WebForm in browser and click on 'Submit' button, the below result will be displayed:

The form is titled "REGISTER NEW USER" in a dark header. It contains the following fields and controls:

- Name* (text input)
- Email* (text input)
- Password* (text input)
- Confirm Password* (text input)
- Date of Birth* (text input)
- Address 1* (text input)
- Address 2 (text input)
- Postal Code* (text input)
- Website (text input)
- ☐ I accept the Terms and Conditions
- Submit button (blue)
- Reset button (grey)

There was an error processing your request. Please correct the following to proceed:

- Please enter your Name
- Please enter your Email
- Please enter your Password
- Please reenter your Password to confirm
- Please enter your Date of Birth
- Please enter your Mailing Address
- Please enter the Postal Code
- Please accept the Terms & Conditions to proceed

Now enter the valid data in every TextBox, according to the logic error message will be removed and finally you will be able to submit the form.

Working with GridView using jQuery

As all you know that GridView is the most important control in ASP.NET which is mostly used to display Data for the processing purpose. But most of the time, end-user wants to perform some operation on it directly on the client side e.g. Highlighting Row or removing row etc. To perform these kind of operations as a developer you need to write huge JavaScript code. Now with jQuery these operations can be made more easy.

STEP 3:

In the project created in STEP 1 add a new folder, name it as 'ASPNET_GridView'.

Exercise 8: Highlighting the GridView Row on Mouse-Hover event

In this exercise we are going to code for highlighting GridView row. To do this we are going to use the static client id of the GridView. Since GridView is rendered as <table> we must make use of <tr> for 'Hover' event.

Task 1: In the folder created in STEP 3 add a new WebForm; name it as 'WebForm_GridView_Highlight.aspx'. IN this page add the below Html and Styles.

```
<style type="text/css">
.highlight
{
    background-color:Red;
}
td { cursor:pointer;}
</style>
```

The Style set the mouse cursor to the 'Pointer'.

```
<div align="center">
<fieldset style="width:400px;height:230px;">
    <table border="0" cellpadding="3" cellspacing="3">
        <tr>
            <td colspan="2" class="header">Employee Catalog</td>
        </tr>
        <tr>
            <td colspan="2">
                <asp:GridView ID="GridView1" SkinID="Professional"
                    runat="server">
                </asp:GridView>
            </td>
        </tr>
    </table>
</fieldset>
</div>
```

Task 2: Write the below code in the Loaded event in code-behind:

```
SqlConnection Conn;
SqlDataAdapter AdEmp;
protected void Page_Load(object sender, EventArgs e)
{
    Conn = new SqlConnection("Data Source=.;Initial Catalog=Company;Integrated
Security=SSPI");
    AdEmp = new SqlDataAdapter("Select * from Employee",Conn);
    DataSet Ds = new DataSet();
    AdEmp.Fill(Ds,"Employee");
    if (this.IsPostBack == false)
    {
        GridView1.DataSource = Ds.Tables["Employee"];
        GridView1.DataBind();
    }
}
```

Task 3: In the source of the WebForm add the below jQuery code:

```
<script language="javascript" type="text/javascript">
$(document).ready(function () {
    $("#<%=GridView1.ClientID%> tr").hover(
        function () {
            $(this).addClass("highlight");
        },
        function () {
            $(this).removeClass("highlight");
        }
    );
});
</script>
```

Task 4: View the WebForm in browser and move the mouse cursor on the Rows the result will be as below:

Employee Catalog			
EmpNo	EmpName	Salary	DeptNo
101	Natrajan	78000	30
102	Makrand P.	34500	10
103	Mahehs Sabnis	97000	20
104	Jayvant	40000	60
105	Abhay	50000	30
106	Leena Sabnis	46000	30
107	Anil	60000	30
108	Amit	89000	20
109	Maruti	20000	20
110	Vikram Pendse	96000	50
111	Saket Karnik	70000	20
112	Kiran P.	34000	10
113	Makarand D.	45000	10
114	Mahesh Sabnis	34000	30
115	Sanjau Godbole	45000	20
116	P.L. Deshpande	34000	50
118	Mahesh Pandit	23000	10
119	Emp_119	65888	10

Exercise 9: Removing GridView Row on Mouse Click.

In this exercise we will perform the GridView row removing operations using jQuery. Since the GridView is rendered as <table> , we need to filter the table header <th> and the table tag <table> so that it should not be removed.

Task 1: In the folder created in STEP 3 add a new WebForm, name it as 'WebForm_Removing_GridView_Row_Cells.aspx'. Add the below style in the header section.

```
<style type="text/css">
    .highlight
    {
        background-color:#9999FF;
    }
    td { cursor:pointer;}
</style>
```

Task 2: Add the below Html in the Form tag which contains GridView.

```
<div align="center">
    <fieldset style="width:400px;height:230px;">
        <table border="0" cellpadding="3" cellspacing="3">
            <tr>
                <td colspan="2" class="header">EMPLOYEE CATALOG</td>
            </tr>
            <tr>
                <td colspan="2">
                    <asp:GridView ID="GridView1" SkinID="Professional"
                        runat="server">
                    </asp:GridView>
                </td></tr>
        </table>
    </fieldset>
</div>
```

Task 3: Keep the same code behind as given in Exercise 8 Task 2.

Task 4: Write the below jQuery which read the GridView.

```
<script language="javascript" type="text/javascript">
    $(document).ready(function() {
        $("#<%=GridView1.ClientID%> tr").filter(":not(:has(table,th))").click(function ()
        {
            $(this).addClass("highlight");
            $(this).fadeOut(1000, function ()
            {
                $(this).remove();
            });
        });
    });
</script>
```

The above code set the filter on the rendered table row so that table header <th> should not be removed. The time for the removal is set to 1000 milliseconds (1 sec.)

Task 5: View the WebForm in browser, and click on the row, it will get removed after 1 second.