jQuery Programming for ASP.NET using VS2010

In this lab we will perform various exercise for the frequently used programming methodology while programming with jQuery. Following are the types of exercise we are going to solve:

- Selectors.
- Filters
    - Basic.
    - Child.
    - Form.
    - Hierarchy.
- Events.

Exercise 1: Working with "All Selector (*)".

This selects all elements for applying effects on it.

Task 1: Open VS2010 and create a Web Site, name it as 'ASPNET_jQueryApp'. Add a new folder in it, name it as 'Selector_Filters'. In this folder add a new Web_Form, name it as 'WF_Selector_All_Elements.aspx'.

Task 2: Design the aspx as below:

```
<table style="width: 100%;">
    <tr>
      <td>
        Selected Color Type</td>
      <td>
        <asp:DropDownList ID="lstStyle" runat="server">
          <asp:ListItem>solid yellow</asp:ListItem>
          <asp:ListItem>solid red</asp:ListItem>
          <asp:ListItem>solid black</asp:ListItem>
        </asp:DropDownList>
      </td>
      <td>
         </td>
    </tr>
    <tr>
      <td>
         
        <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
      </td>
      <td>
         
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
      </td>
      <td>
         
        <asp:RadioButton ID="RadioButton1" runat="server" />
```

```
      </td>
    </tr>
    <tr>
      <td>
         
        <asp:FileUpload ID="FileUpload1" runat="server" />
      </td>
      <td>
         
        <asp:Button ID="Button1" runat="server" Text="Button" />
      </td>
      <td>
         
        <asp:CheckBox ID="CheckBox1" runat="server" />
      </td>
    </tr>
    <tr>
      <td>
         
        <asp:DropDownList ID="DropDownList1" runat="server">
        </asp:DropDownList>
      </td>
      <td>
         
        <asp:ListBox ID="ListBox1" runat="server"></asp:ListBox>
      </td>
      <td>
         
        <asp:Image ID="Image1" runat="server" />
      </td>
    </tr>
  </table>
```

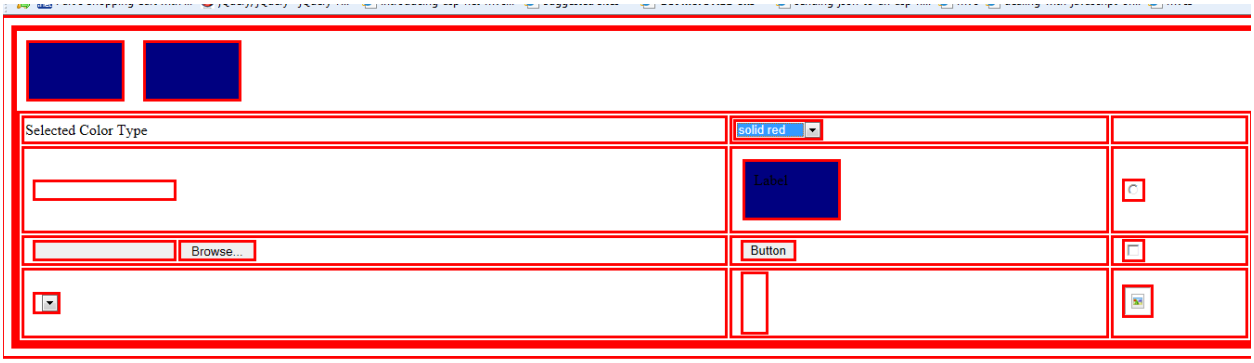
Task 3: On the page write the below javascript,:

```
$('select#lstStyle').bind('keyup change', function () {
                    var txt = $(this).find(":selected").text();
                    alert(txt);
                    var eleCount = $("*").css("border", "3px " + txt).length;
                    alert("No. of . Elements " + eleCount);
                });
```

The above code performs the below operations:

- Bind the ‘keyup change’ event to the ‘select’ element of name ‘lstStyle’. Using the ‘:selected’ filter, stored the selected value from the ‘select’ element in ‘txt’. Now here ‘$(“*”)’ selector represent all elements of the page. To all these elements the ‘border’ is set using ‘.css’ method for the selected color from the ‘select’ element.

Task 4: Run the page an select the color from the List, the result will be as below:



Note: You can use other Basic selector as below:

Class Selector (".class")

```
$(".sampleClass").css("border","3px solid red");
```

Element Selector ("element")

Selects all elements those match with the specific tag in HTML

```
<script>$("div").css("border","9px solid red");</script>
```

ID Selector ("#id")

Select a single element with the given id attribute

```
<script>$("#mytxt").css("border","3px solid red");</script>
```

Exercise 2: (Self)

Create an ASP.NET Page Put below controls:

Control	Settings
TextBox	Class name as "myTxt"
TextBox	Class name as "myTxt".
TextBox	Id="txtName"
Button	
Div	Class Name as "myDiv"
Button	Class name as "myTxt"

Write the jQuery with followingeffects:

- All Elements with class name as “myTxt” must be set border in Red color with 4 px. The Font-Family as “Times New Roman”.
- TextBox with txtName as id must have background color as Magenta.
- The Border of Div must be Blue with 10px.

Exercise 3: Working with Hierarchy Selector.

These selectors allows us to easy traverse through the DOM hieratchy of elements and apply effects on it.

Task 1: In the Web Site created in Ex-1 add new Web form name it as ‘WF_Child_Selector.aspx’. Design it as below:

```
<div>
    <ul>
      <li>Microsoft</li>
      <li>
       <ul>
          <li>ASP.NET</li>
          <li>WinForm</li>
          <li>WCF</li>
          <li>WPF</li>
       </ul>
      </li>
      <li>jQuery</li>
    </ul>
     <table style="width: 100%;">
       <tr>
          <td>
            Select Color Theme</td>
          <td>
            <asp:DropDownList ID="lstColors" runat="server">
              <asp:ListItem>red</asp:ListItem>
              <asp:ListItem>blue</asp:ListItem>
              <asp:ListItem>yellow</asp:ListItem>
              <asp:ListItem>black</asp:ListItem>
            </asp:DropDownList>
          </td>
          <td>
             </td>
       </tr>
       <tr>
          <td>
             
            1</td>
```

```
            <td>
                 
                4</td>
            <td>
                 
                7</td>
        </tr>
        <tr>
            <td>
                 
                2</td>
            <td>
                 
                5</td>
            <td>
                 
                8</td>
        </tr>
        <tr>
            <td>
                 
                3</td>
            <td>
                 
                6</td>
            <td>
                 
                9</td>
        </tr>
    </table>
    <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
        DataKeyNames="EmpNo" DataSourceID="SqlDataSource1" AllowPaging="True"
        PageSize="3">
        <Columns>
            <asp:BoundField DataField="EmpNo" HeaderText="EmpNo" InsertVisible="False"
                ReadOnly="True" SortExpression="EmpNo" />
            <asp:BoundField DataField="EmpName" HeaderText="EmpName"
                SortExpression="EmpName" />
            <asp:BoundField DataField="Salary" HeaderText="Salary"
                SortExpression="Salary" />
            <asp:BoundField DataField="DeptName" HeaderText="DeptName"
                SortExpression="DeptName" />
            <asp:BoundField DataField="Designation" HeaderText="Designation"
                SortExpression="Designation" />
        </Columns>
    </asp:GridView>

    <asp:SqlDataSource ID="SqlDataSource1" runat="server"
```

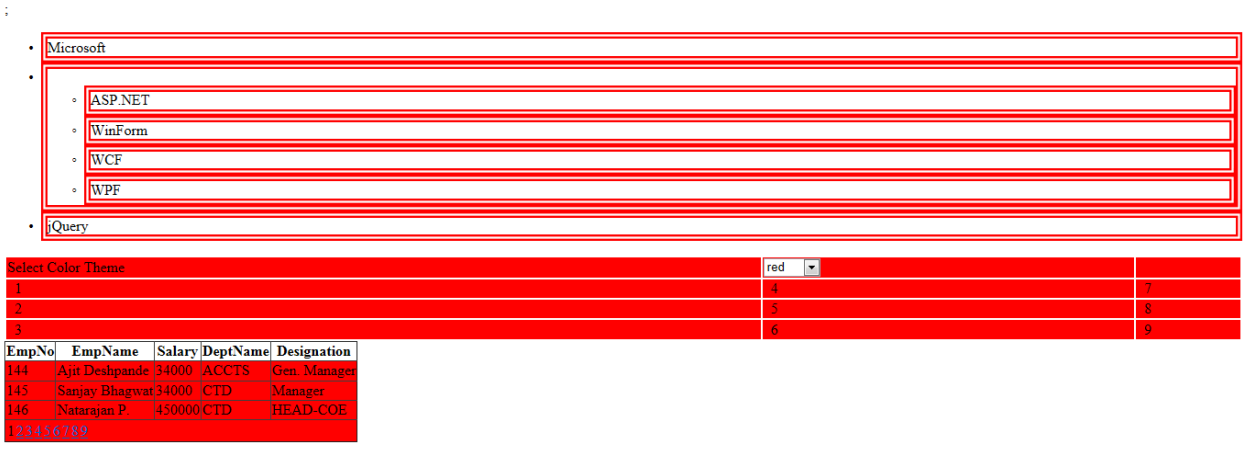
```
        ConnectionString="<%$ ConnectionStrings:CompanyConnectionString %>"
        SelectCommand="SELECT * FROM [EmployeeInfo]"></asp:SqlDataSource>

    </div>
```

Note: You can choose the Data Source to GridView as per you requirements.

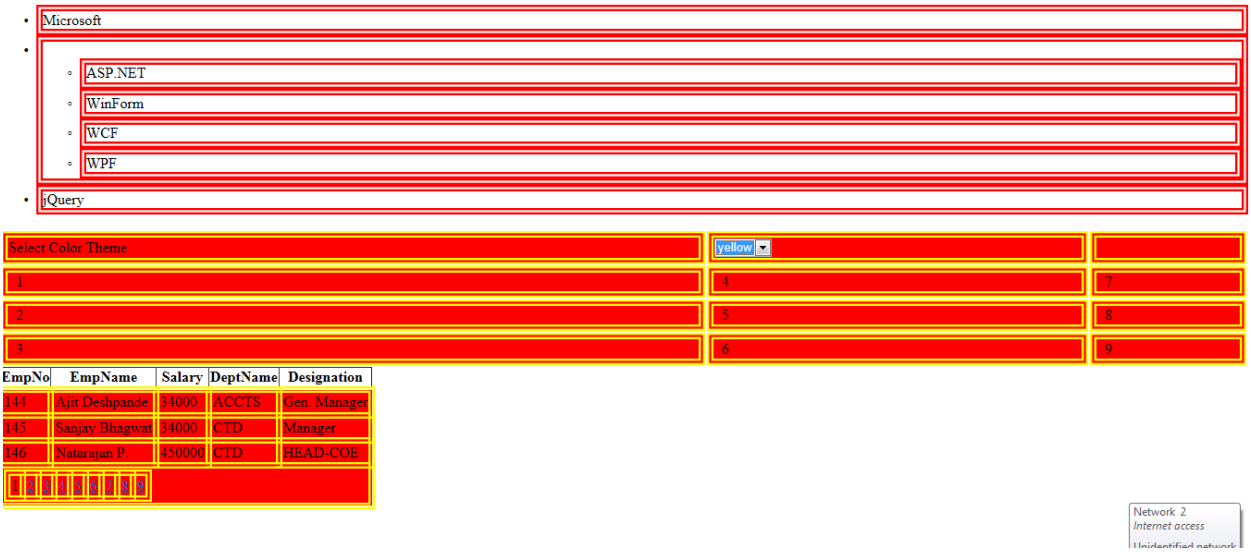
Task 2: In this Task we will be using “Child Selector “(“P>C”)”. This means that ‘Selects all direct child elements specified by "C" of elements specified by "P".’ This is supported by all modern browsers. In our case we have ‘ul’ which is parent of ‘li’. Also the Table contains ‘tr’ and ‘td’ where each ‘tr’ is parent of ‘td’. So now we can apply effects on it as below:

```
<script type="text/javascript">
        $(document).ready(function () {
            $("ul>li").css("border", "7px double red");
            //            $("tr>td").css("border", "7px double yellow");
                    $("tr>td").css("background-color", "red");

                    $("#lstColors").bind("keydown change", function (e) {
                        var data = $(this).val();
                        //alert(data);
                        $("tr>td").css("border", "7px double " + data);
                    });

        });
    </script>
```

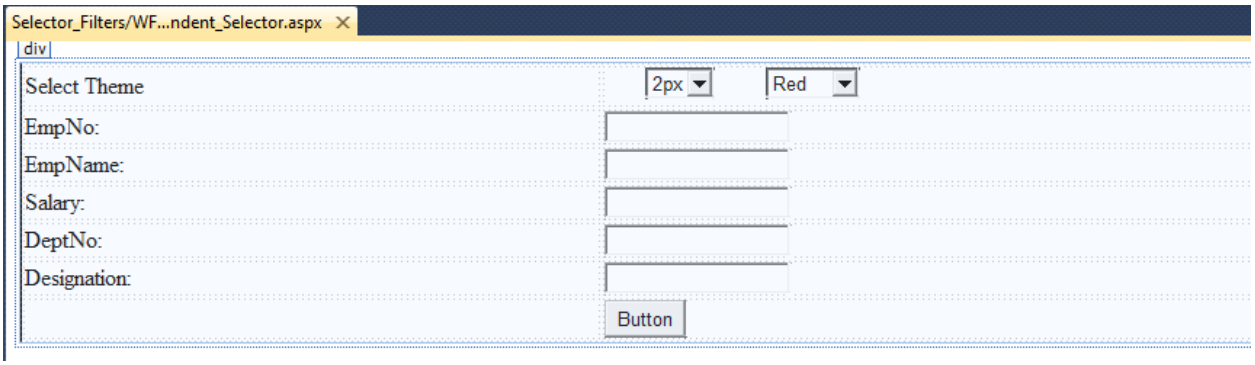
Task 3: Run the page, the result will be as below:



Change the color from the ComboBox the effects will be as below:



Task 4: Add a new Web Form in the Web Site, name it as 'WF_DEscendent_Selector.aspx'. Design it as below:



The Data in the DropDown are:

ComboBox 1: 2px, 3px,4px,5px

ComboBox 2: Red, Blue, Yellow, Green, Yellow.

Task 5: In this task we will see the 'Descendent Selector'. "This selects al elements that are descendants of a given ancestor". In our case all textboxes are descendants of the 'form'. So we will apply effects on all TextBoxes with the below jQuery:

```
<script type="text/javascript">
    $(document).ready(function () {
      var data;
```

```
        var d1;
        var d2;
        $("#lstSize").bind("keypress change", function () {
            d1 = $(this).val();
        });

        $("#lstColor").bind("keypress change", function () {
            d2 = $(this).val();
            data = d1 + " double " + d2;
            $("form input:text").css("border", data);
        });
        $("#Button1").click(function (e) {
            e.preventDefault();
        });
    });
</script>
```

In the above code the expression $(form input:text) represents the descendant selector where the effects are applied based upon values selected form the comboboxes.

Task 6: Run the page and select values from ComboBoxes, the result will be as below:

Exercise 4: Working with Basic Filters.

Task 1: In the Web Site created in Ex-1, add a new Web Form in 'Filters' folder, name it as 'WF_BasicFilter_eq.aspx'. On this page add a GridView with some DataBinding with it using DataSource of your choice.

Task 2: We know that GridView is rendered as <table> so we can apply filters on it. Here we will change the row background of the second row as red using ":eq" filter as below:

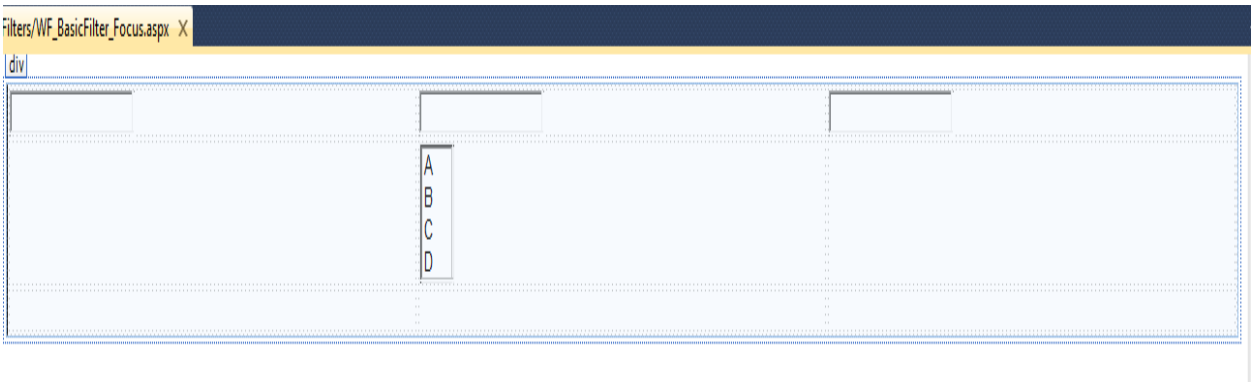
```
$(document).ready(function () {
        $("tr:eq(2)").css("background-color", "Red");
```

```
});
```

Task 3: Test the page the result will be as below:

CompanyId	CompanyName	Q1	Q2	Q3	Q4
1	TS. Soft	400.0000	200.0000	300.0000	800.0000
2	MS. Soft	400.0000	300.0000	300.0000	600.0000
3	LS. Soft	200.0000	300.0000	400.0000	300.0000
4	TMS. Soft	300.0000	400.0000	100.0000	500.0000

Task 4: In this Task we will see the ':focus' filter. This selects the element which is currently focused. In the web site add a new Web Form, name it as 'WF_BasicFilter_Focus.aspx'. Design the Web Form as below:



Now here we want to change the color of every element on the page when it is focused. (Note: For this task we are going to define event for <table> using 'delegate' method. This is not available in the current jQuery version so we need to make use of 1.5.1 do please download it or get it from any Asp.NET MVC prohect).

Task 5: Write the below script on the page also define a style class of name focused:

```
<style type="text/css">
    .focused
    {
       background-color:Gray;
       }
  </style>
```

```
<script src="../Scripts/jquery-1.5.1.min.js" type="text/javascript"></script>
  <script type="text/javascript">
    $(document).ready(function () {
      $("#tbl").delegate("*", "focus blur", function (e) {
        var txtBox = $(this);
        setTimeout(function () {
          txtBox.toggleClass("focused", txtBox.is(":focus"))
        }, 0);
      });
    });
  </script>
```

The above code the 'delegate', Attach a handler to one or more events for all elements that match the selector, now or in the future, based on a specific set of root elements. Here the root element used is of the type <table> of name 'tbl' and event focus and blur are attached. The 'setTimeOut()' method apply the class of name 'focused' when the element is focused.

Task 6: Run the page the result will be as below:

Exercise 5: Working with Child-Filters

Following are the types of child filters provided:

- Child Filter:
  - :first-child.
    - Selects all elements that are the first child of their parent.
  - :last-child.
    - Selects all elements that are the last child of their parent.
  - :nth-child.
    - Selects all elements that are the nth-child of their parent.
  - :only-child.
    - Selects all elements that are the only child of their parent.

Task 1: In the Web Site created in Ex-1 add a new Web Form, name it as 'WF_Child_Filters'. Add the below Html in it:

```
<div>
  Description: Selects all elements that are the first child of their parent.

    <br />
    <br />
    <br />
    <div>
     <ul>
      <li>Microsoft Developer</li>
        <ul>
          <li>WinForm</li>
          <li>WPF</li>
          <li>WCF</li>
          <li>WF</li>
          <li>MVC</li>
        </ul>
         <li>Microsoft Server</li>
        <ul>
          <li>BizTalk</li>
          <li>SharePoint</li>
          <li>Dynmics</li>
          <li>CRM</li>
        </ul>
     </ul>
    </div>
  </div>
```

Task 2: Add the below style class on the page:

```
<style type="text/css">
    .whenmousehover
    {
      color:Red;
     }
 </style>
```

Task 3: Write the below script:

```
<script src="../Scripts/jquery-1.5.1.min.js" type="text/javascript"></script>
  <script type="text/javascript">
    $(document).ready(function () {
      $("div li:first-child").css("text-decoration", "underline")
      .hover(function () {
```

```
            $(this).addClass("whenmousehover")
            });
           $("div li:last-child").css("text-decoration", "underline")
        .hover(function () {
            $(this).addClass("whenmousehover")
        });
    });
</script>
```

In the above code, it is shown that when the mouse is hover upon on the first and last child of element 'li' under div it will be underlined and the class 'whenmousehover' will be applied on it.

Task 4: Run the page the result will be as below:

Description: Selects all elements that are the first child of their parent.

- Microsoft Developer
  - WinForm
  - WPF
  - WCF
  - WF
  - MVC
- Microsoft Server
  - BizTalk
  - SharePoint
  - Dynmics
  - CRM

Task 4: In the Web Site add a new form, name it as 'WF_Child_Filters_Only_Child.aspx'. In this task we will see the ':only-child' child selector. This means 'Selects all elements that are the only child of their parent and If the parent has other child elements, nothing is matched'.

Task 5: Add the below code in the apsx:

```
<div>
      <asp:Button runat="server" Text="Button 1" ID="Button1"/>
      <asp:Button runat="server" Text="Button 2" ID="Button2"/>
    </div>
     <div>
      <asp:Button runat="server" Text="Button 3" ID="Button3"/>
    </div>
      <div>
      <asp:Button runat="server" Text="Button 4" ID="Button4"/>
    </div>
```

The above design shows that there are some having more than one children.

Task 6: Add the below style class on the page:

```
<style type="text/css">
    .mouseEntered
    {
       background-color:Yellow;
       }
  </style>
```

Task 7: Add the below Script on the page:

```
<script src="../Scripts/jquery-1.5.1.min.js" type="text/javascript"></script>
  <script type="text/javascript">
     $(document).ready(function () {
       $("div").mouseenter(function () {
          $("div input[type=submit]:only-child").val("Only Child").addClass("mouseEntered");
        });
        $("div").mouseleave(function () {
          $("div input[type=submit]:only-child").val("Only Child").removeClass("mouseEntered");
        });
     });
  </script>
```

The above jQuery clearly states that if the <Div> tag has only one submit button type then only the class will be appliued.

Task 8: Run the page and move on the buttons, all <Div> having only one child will only be changed.