

UCS1601-INTERNET PROGRAMMING

Exercise 6: Autocomplete Feature using AJAX

NAME: YASIR MANZOOR ITOO

REGNO: 185001203

Develop an AJAX program that implements the Autocomplete feature for filling the country names as given below:

For Example, when N is entered all the names of the country starting with “N” should be listed in drop-down list for user’s easy selection. When “Na” is typed only “Namibia” and “Nauru” should be listed for selection. Use Servlet for server side scripting. Use MySQL database to store the countries names.

PROGRAM:

Html file

```
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">
    <title>Ajax Assignment</title>
    <link href="style.css" rel="stylesheet" type="text/css"/>
    <script src="index.js"></script>
    <script src="style.js"></script>
    <style>
    </style>
  </head>
  <body>
    <h1 style="color: black;">Country Names</h1>
    <input type="text" list="select" onkeyup="disp();" id="input"
name="countries">
    <datalist id="select">
    </datalist>
    <br><br><br>
  </body>
</html>
```

Style.js

```
(function() {

    var CuteSelect = CuteSelect || {};

    FIRSTLOAD = true;
    SOMETHINGOPEN = false;

    CuteSelect.tools = { canRun: function() { var myNav =
        navigator.userAgent.toLowerCase(); var browser =
        (myNav.indexOf('msie') != -1) ?
parseInt(myNav.split('msie')[1]) : false;
        if(browser) { return (browser > 8) ?
            true : false;
        } else { return true; }
    }, unqid: function() { var n=
        Math.floor(Math.random()*11); var k =
        Math.floor(Math.random()* 1000000); var m =
        String.fromCharCode(n)+k; return m; }, hideEverything:
        function() { if(SOMETHINGOPEN) { SOMETHINGOPEN = false;
        targetThis = false; var cells =
        document.getElementsByTagName('div'); for (var i = 0; i
        < cells.length; i++) {

if(cells[i].hasAttribute('data-cuteselect-options')) {
            var parent = cells[i].parentNode;
            cells[i].style.opacity = '0';
            cells[i].style.display = 'none';
        }
        } }, getStyle: function() { var css =
        ''; var stylesheets = document.styleSheets;
        var css = '';

        for(s = 0; s < stylesheets.length; s++) {
            var classes = stylesheets[s].rules ||
stylesheets[s].cssRules;
```

```

        for (var x = 0; x < classes.length; x++) {
            if(classes[x].selectorText != undefined) {
                var selectPosition =
classes[x].selectorText.indexOf('select');
                var optionPosition =
classes[x].selectorText.indexOf('option');
                var selectChar =
classes[x].selectorText.charAt(selectPosition - 1);
                var optionChar =
classes[x].selectorText.charAt(optionPosition - 1);
                if(selectPosition >= 0 && optionPosition >= 0
&& (selectChar == '"' || selectChar == '}' || selectChar == ' ') &&
(optionChar == '"' || optionChar == '}' || optionChar == ' ')) {
                    text = (classes[x].cssText) ?
classes[x].cssText : classes[x].style.cssText;
                    css += text.replace(/\boption\b/g,
'[data-cuteselect-value]').replace(/\bselect\b/g,
'[data-cuteselect-item]');
                    continue;
                } if(selectPosition >= 0)
                { var character =
classes[x].selectorText.charAt(selectPosition - 1);
                    if(character == '"' || character == '}' ||
character == ' ') {
                        text = (classes[x].cssText) ?
classes[x].cssText : classes[x].style.cssText;
                        css += text.replace(/\bselect\b/g,
'[data-cuteselect-item]');
                    } } if(optionPosition
                    >= 0) { var character =
classes[x].selectorText.charAt(optionPosition - 1);
                        if(character == '"' || character == '}' ||
character == ' ') {
                            text = (classes[x].cssText) ?
classes[x].cssText : classes[x].style.cssText;
                            css += text.replace(/\boption\b/g,
'[data-cuteselect-value]');
                        }
                    }
                }
            }
        }

```

```

        }
    }

    return css;
}, createSelect:
function(item) {

    // Create custom select var node =
    document.createElement("div");
    if(item.hasAttribute('id')) { // Catch ID
        node.setAttribute('id', item.getAttribute('id'));
        item.removeAttribute('id');
    } if(item.hasAttribute('class')) { // Catch Class
        node.setAttribute('class', item.getAttribute('class'));
        item.removeAttribute('class');
    }

    // Hide select
    item.style.display = 'none';

    // Get Default value (caption) var caption =
    null; var cells =
    item.getElementsByTagName('option'); for (var i
    = 0; i < cells.length; i++) { caption =
    cells[0].innerHTML;
    if(cells[i].hasAttribute('selected')) { caption
    = cells[i].innerHTML; break;
        }
    }

    // Get select options var options = '<div data-
    cuteselect-title>' + caption +
    '</div><div data-cuteselect-options><div data-cuteselect-options-
    container>';

    var cells = item.getElementsByTagName('option'); for
    (var i = 0; i < cells.length; i++) {
        if(cells[i].hasAttribute('disabled')) { continue; }
        if(cells[i].hasAttribute('class')) { var optionStyle =
        ' class="' + cells[i].getAttribute('class') + '"'; }
        else { var optionStyle = ''; }
    }

```

```

        if(cells[i].hasAttribute('id')) { var optionId = '
id=""' + cells[i].getAttribute('id') + ''; } else { var optionId = '';
    }

    if(cells[i].hasAttribute('selected')) { options +=
'<div data-cuteselect-value="" + cells[i].value + "" data-cuteselect-
selected="true"' + optionStyle + optionId + '>' + cells[i].innerHTML +
'</div>'; }

    else { options += '<div data-cuteselect-value="" +
cells[i].value + "" + optionStyle + optionId + '>' +
cells[i].innerHTML + '</div>'; }

    } options +=
    '</div></div>';

    // New select customization
    node.innerHTML = caption;
    node.setAttribute('data-cuteselect-item',
CuteSelect.tools.uniqid());
    node.innerHTML = options; // Display options
    item.setAttribute('data-cuteselect-target',
node.getAttribute('data-cuteselect-item'));
    item.parentNode.insertBefore(node, item.nextSibling);

    // Hide all options
    CuteSelect.tools.hideEverything(); }, show:
    function(item) { if(item.parentNode.hasAttribute('data-
cuteselect-item')) {
var source = item.parentNode.getAttribute('data-cuteselect-item'); }
    else { var source =
item.getAttribute('data-cuteselect-item'); }

    var cells =
    document.getElementsByTagName('select');
    if(item.hasAttribute('data-cuteselect-title')) {
    item = item.parentNode; var cells =
    item.getElementsByTagName('div');
    } else { var cells = item.getElementsByTagName('div'); }
    for (var i = 0; i < cells.length; i++) {
    if(cells[i].hasAttribute('data-cuteselect-options')) {
    targetItem = cells[i]; cells[i].style.display = 'block';

        setTimeout(function() { targetItem.style.opacity =
'1'; }, 10);

```

```

        cells[i].style.position = 'absolute';
        cells[i].style.left = item.offsetLeft + 'px';
        cells[i].style.top = (item.offsetTop +
item.offsetHeight) + 'px';
    }
}

item.focus();

    SOMETHINGOPEN = item.getAttribute('data-cuteselect-
item'); }, selectOption: function(item) { var label =
item.innerHTML; var value = item.getAttribute('data-
cuteselect-value'); var parent =
item.parentNode.parentNode.parentNode; var target =
parent.getAttribute('data-cuteselect-item'); var cells =
parent.getElementsByTagName('div'); for (var i = 0; i <
cells.length; i++) { if(cells[i].hasAttribute('data-
cuteselect-title')) {
cells[i].innerHTML = label; }
}

    // Real select var cells =
document.getElementsByTagName('select'); for (var i
= 0; i < cells.length; i++) { var source =
cells[i].getAttribute('data-cuteselect-target');
    if(source == target) { cells[i].value = value; }
}
    CuteSelect.tools.hideEverything(); },
writeStyles: function() { toWrite =
'<style type="text/css">' +
CuteSelect.tools.getStyle() + ' [data-cuteselect-options] { opacity: 0;
display: none; }</style>';
    document.write(toWrite);
}
};

CuteSelect.event = {
    parse: function() {
        var cells = document.getElementsByTagName('select');
for (var i = 0; i < cells.length; i++) {

```

```

CuteSelect.tools.createSelect(cells[i]); } }, listen: function()
{ document.onkeydown = function(evt) { evt = evt ||
window.event; if (evt.keyCode == 27) {
CuteSelect.tools.hideEverything(); } };

    document.onclick = function(event) { FIRSTLOAD = false;
        if(!event.target.getAttribute('data-cuteselect-
            item'))
&& !event.target.getAttribute('data-cuteselect-value') &&
!event.target.hasAttribute('data-cuteselect-title')) ||
((event.target.hasAttribute('data-cuteselect-item') ||
event.target.hasAttribute('data-cuteselect-title')) && SOMETHINGOPEN))
{ CuteSelect.tools.hideEverything(); return;
    } var action = event.target;
    if(event.target.getAttribute('data-cuteselect-
        value'))
{
        CuteSelect.tools.selectOption(action);
        CuteSelect.tools.hideEverything();
    } else { CuteSelect.tools.show(action);
    } return false;
    } }, manage: function() {
if(CuteSelect.tools.canRun()) { // IE Compatibility
    CuteSelect.event.parse();
    CuteSelect.event.listen();
    CuteSelect.tools.writeStyles();
    }
    }
};

CuteSelect.event.manage();

})();

```

Style.css

```

body { display: flex; flex-
    direction: column;
    justify-content:
    center; align-items:
    center; height: 50vh;
    background: lightblue;

```

```

}
h1 {
    margin: 0 0 0.25em;
}
input[type=text]
{
    background-position: 10px 10px;
}
input[type=text]
{
    background-color: lightblue;
    color: black;
}

```

Java File

```

import java.sql.*; import
java.io.*; import
jakarta.servlet.*; import
jakarta.servlet.http.*;

public class country extends HttpServlet { public void
    doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

        response.setContentType("text");
        PrintWriter pw=response.getWriter();
        Connection con;
        Statement st; ResultSet rs; try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            try {

                con
=DriverManager.getConnection("jdbc:mysql://localhost:3306/world","root"
,"YASIR@123");

                try
                {

                    String filter = request.getParameter("val");
                    String srch = " SELECT * FROM country where name

```



```

    regexp '^"+filter+"' order by name asc;";
    st=con.createStatement();
    rs=st.executeQuery(srch);

    while(rs.next() && filter.length()!=0 ){

pw.println("<option>" +rs.getString(1)+"</option>");

        }
    } catch(SQLException
        e)
    { pw.println(e);
    } } catch
    (SQLException e)
    { pw.println(e);
    } }
    catch(ClassNotFoundException e)
    { pw.println(e);
    }
    pw.close();
}
}

```

OUTPUT:

