

Name: Sudharshan R

Class : CSE-C

Reg : 185001173

Assignment -6 Autocomplete Feature using AJAX

AIM:



To 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

CODE:

Index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Autocomplete Test</title>
</head>
<body>
  <form name="myform">
    <label for="country"> Country : </label>
    <input type="text" id="country" name="country" list="dropdown" onInput
="loadDoc();" >
    <datalist id="dropdown">

  </datalist>
```

```

        <input type="button" value="go" onclick='alert("Your choice :"+document.
t.getElementById("country").value);'>

    </form>
    <script>
    function loadDoc() {
    var xhttp = new XMLHttpRequest();
    var v=document.getElementById("country").value;
    var url="GD?country="+v;

    xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        var val=xhttp.responseText;
        document.getElementById("dropdown").innerHTML=val;
    }
    };
    xhttp.open("GET", url, true);
    xhttp.send();
    }
    </script>
    </body>
</html>

```

Getdata.java

```

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

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

        response.setContentType("text/html");
        int flag=0;
        Connection con;
        Statement st;
        ResultSet rs;
        try
        {
            Class.forName("com.mysql.cj.jdbc.Driver");
            try
            {
                flag=1;
                con =DriverManager.getConnection("jdbc:mysql://localhost:3306/
countries","root","Sun@0097");
                try

```

```

        {
            String srch = " SELECT Name FROM Country WHERE name LIKE '
"+(String)request.getParameter("country")+"%';";
            st=con.createStatement();
            rs=st.executeQuery(srch);
            while(rs.next()){
                response.getWriter().write("<option>" +rs.getString(1)+
"</option>");
            }
        }
        catch(SQLException e)
        {
            response.getWriter().write("error");
        }
    }
    catch (SQLException e)
    {
        flag=2;
    }
}
catch(ClassNotFoundException e)
{
    flag=3;
}
if(flag==2)
{
    response.getWriter().write("error");
}
else if(flag==3)
{
    response.getWriter().write("error");
}
}
}
}

```

Web.xml

```

<web-app>
<servlet>
    <servlet-name>GD</servlet-name>
    <servlet-class>getdata</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>GD</servlet-name>
    <url-pattern>/GD</url-pattern>
</servlet-mapping>
</web-app>

```

SQL Commands:

```
CREATE TABLE Country
(
    CountryID INT
    ,Name VARCHAR(50)
    ,TwoCharCountryCode CHAR(2)
    ,ThreeCharCountryCode CHAR(3)
);
```

```
INSERT INTO Country VALUES
(1,'Afghanistan','AF','AFG')
,(2,'Aland Islands','AX','ALA')
,(3,'Albania','AL','ALB')
,(4,'Algeria','DZ','DZA')
,(5,'American Samoa','AS','ASM')
,(6,'Andorra','AD','AND')
,(7,'Angola','AO','AGO')
,(8,'Anguilla','AI','AIA')
,(9,'Antarctica','AQ','ATA')
,(10,'Antigua and Barbuda','AG','ATG')
,(11,'Argentina','AR','ARG')
,(12,'Armenia','AM','ARM')
,(13,'Aruba','AW','ABW')
,(14,'Australia','AU','AUS')
,(15,'Austria','AT','AUT')
,(16,'Azerbaijan','AZ','AZE')
,(17,'Bahamas','BS','BHS')
,(18,'Bahrain','BH','BHR')
,(19,'Bangladesh','BD','BGD')
```

,(20,'Barbados','BB','BRB')
,(21,'Belarus','BY','BLR')
,(22,'Belgium','BE','BEL')
,(23,'Belize','BZ','BLZ')
,(24,'Benin','BJ','BEN')
,(25,'Bermuda','BM','BMU')
,(26,'Bhutan','BT','BTN')
,(27,'Bolivia','BO','BOL')
,(28,'Bonaire, Sint Eustatius and Saba','BQ','BES')
,(29,'Bosnia and Herzegovina','BA','BIH')
,(30,'Botswana','BW','BWA')
,(31,'Bouvet Island','BV','BVT')
,(32,'Brazil','BR','BRA')
,(33,'British Indian Ocean Territory','IO','IOT')
,(34,'Brunei','BN','BRN')
,(35,'Bulgaria','BG','BGR')
,(36,'Burkina Faso','BF','BFA')
,(37,'Burundi','BI','BDI')
,(38,'Cambodia','KH','KHM')
,(39,'Cameroon','CM','CMR')
,(40,'Canada','CA','CAN')
,(41,'Cape Verde','CV','CPV')
,(42,'Cayman Islands','KY','CYM')
,(43,'Central African Republic','CF','CAF')
,(44,'Chad','TD','TCD')
,(45,'Chile','CL','CHL')
,(46,'China','CN','CHN')
,(47,'Christmas Island','CX','CXR')
,(48,'Cocos (Keeling) Islands','CC','CCK')
,(49,'Colombia','CO','COL')
,(50,'Comoros','KM','COM')

,(51,'Congo','CG','COG')
,(52,'Cook Islands','CK','COK')
,(53,'Costa Rica','CR','CRI')
,(54,'Ivory Coast','CI','CIV')
,(55,'Croatia','HR','HRV')
,(56,'Cuba','CU','CUB')
,(57,'Curacao','CW','CUW')
,(58,'Cyprus','CY','CYP')
,(59,'Czech Republic','CZ','CZE')
,(60,'Democratic Republic of the Congo','CD','COD')
,(61,'Denmark','DK','DNK')
,(62,'Djibouti','DJ','DJI')
,(63,'Dominica','DM','DMA')
,(64,'Dominican Republic','DO','DOM')
,(65,'Ecuador','EC','ECU')
,(66,'Egypt','EG','EGY')
,(67,'El Salvador','SV','SLV')
,(68,'Equatorial Guinea','GQ','GNQ')
,(69,'Eritrea','ER','ERI')
,(70,'Estonia','EE','EST')
,(71,'Ethiopia','ET','ETH')
,(72,'Falkland Islands (Malvinas)','FK','FLK')
,(73,'Faroe Islands','FO','FRO')
,(74,'Fiji','FJ','FJI')
,(75,'Finland','FI','FIN')
,(76,'France','FR','FRA')
,(77,'French Guiana','GF','GUF')
,(78,'French Polynesia','PF','PYF')
,(79,'French Southern Territories','TF','ATF')
,(80,'Gabon','GA','GAB')
,(81,'Gambia','GM','GMB')

,(82,'Georgia','GE','GEO')
,(83,'Germany','DE','DEU')
,(84,'Ghana','GH','GHA')
,(85,'Gibraltar','GI','GIB')
,(86,'Greece','GR','GRC')
,(87,'Greenland','GL','GRL')
,(88,'Grenada','GD','GRD')
,(89,'Guadeloupe','GP','GLP')
,(90,'Guam','GU','GUM')
,(91,'Guatemala','GT','GTM')
,(92,'Guernsey','GG','GGY')
,(93,'Guinea','GN','GIN')
,(94,'Guinea-Bissau','GW','GNB')
,(95,'Guyana','GY','GUY')
,(96,'Haiti','HT','HTI')
,(97,'Heard Island and McDonald Islands','HM','HMD')
,(98,'Honduras','HN','HND')
,(99,'Hong Kong','HK','HKG')
,(100,'Hungary','HU','HUN')
,(101,'Iceland','IS','ISL')
,(102,'India','IN','IND')
,(103,'Indonesia','ID','IDN')
,(104,'Iran','IR','IRN')
,(105,'Iraq','IQ','IRQ')
,(106,'Ireland','IE','IRL')
,(107,'Isle of Man','IM','IMN')
,(108,'Israel','IL','ISR')
,(109,'Italy','IT','ITA')
,(110,'Jamaica','JM','JAM')
,(111,'Japan','JP','JPN')
,(112,'Jersey','JE','JEY')

,(113,'Jordan','JO','JOR')
,(114,'Kazakhstan','KZ','KAZ')
,(115,'Kenya','KE','KEN')
,(116,'Kiribati','KI','KIR')
,(117,'Kosovo','XK','---')
,(118,'Kuwait','KW','KWT')
,(119,'Kyrgyzstan','KG','KGZ')
,(120,'Laos','LA','LAO')
,(121,'Latvia','LV','LVA')
,(122,'Lebanon','LB','LBN')
,(123,'Lesotho','LS','LSO')
,(124,'Liberia','LR','LBR')
,(125,'Libya','LY','LBY')
,(126,'Liechtenstein','LI','LIE')
,(127,'Lithuania','LT','LTU')
,(128,'Luxembourg','LU','LUX')
,(129,'Macao','MO','MAC')
,(130,'Macedonia','MK','MKD')
,(131,'Madagascar','MG','MDG')
,(132,'Malawi','MW','MWI')
,(133,'Malaysia','MY','MYS')
,(134,'Maldives','MV','MDV')
,(135,'Mali','ML','MLI')
,(136,'Malta','MT','MLT')
,(137,'Marshall Islands','MH','MHL')
,(138,'Martinique','MQ','MTQ')
,(139,'Mauritania','MR','MRT')
,(140,'Mauritius','MU','MUS')
,(141,'Mayotte','YT','MYT')
,(142,'Mexico','MX','MEX')
,(143,'Micronesia','FM','FSM')

,(144,'Moldava','MD','MDA')
,(145,'Monaco','MC','MCO')
,(146,'Mongolia','MN','MNG')
,(147,'Montenegro','ME','MNE')
,(148,'Montserrat','MS','MSR')
,(149,'Morocco','MA','MAR')
,(150,'Mozambique','MZ','MOZ')
,(151,'Myanmar (Burma)','MM','MMR')
,(152,'Namibia','NA','NAM')
,(153,'Nauru','NR','NRU')
,(154,'Nepal','NP','NPL')
,(155,'Netherlands','NL','NLD')
,(156,'New Caledonia','NC','NCL')
,(157,'New Zealand','NZ','NZL')
,(158,'Nicaragua','NI','NIC')
,(159,'Niger','NE','NER')
,(160,'Nigeria','NG','NGA')
,(161,'Niue','NU','NIU')
,(162,'Norfolk Island','NF','NFK')
,(163,'North Korea','KP','PRK')
,(164,'Northern Mariana Islands','MP','MNP')
,(165,'Norway','NO','NOR')
,(166,'Oman','OM','OMN')
,(167,'Pakistan','PK','PAK')
,(168,'Palau','PW','PLW')
,(169,'Palestine','PS','PSE')
,(170,'Panama','PA','PAN')
,(171,'Papua New Guinea','PG','PNG')
,(172,'Paraguay','PY','PRY')
,(173,'Peru','PE','PER')
,(174,'Phillipines','PH','PHL')

,(175,'Pitcairn','PN','PCN')
,(176,'Poland','PL','POL')
,(177,'Portugal','PT','PRT')
,(178,'Puerto Rico','PR','PRI')
,(179,'Qatar','QA','QAT')
,(180,'Reunion','RE','REU')
,(181,'Romania','RO','ROU')
,(182,'Russia','RU','RUS')
,(183,'Rwanda','RW','RWA')
,(184,'Saint Barthelemy','BL','BLM')
,(185,'Saint Helena','SH','SHN')
,(186,'Saint Kitts and Nevis','KN','KNA')
,(187,'Saint Lucia','LC','LCA')
,(188,'Saint Martin','MF','MAF')
,(189,'Saint Pierre and Miquelon','PM','SPM')
,(190,'Saint Vincent and the Grenadines','VC','VCT')
,(191,'Samoa','WS','WSM')
,(192,'San Marino','SM','SMR')
,(193,'Sao Tome and Principe','ST','STP')
,(194,'Saudi Arabia','SA','SAU')
,(195,'Senegal','SN','SEN')
,(196,'Serbia','RS','SRB')
,(197,'Seychelles','SC','SYC')
,(198,'Sierra Leone','SL','SLE')
,(199,'Singapore','SG','SGP')
,(200,'Sint Maarten','SX','SXM')
,(201,'Slovakia','SK','SVK')
,(202,'Slovenia','SI','SVN')
,(203,'Solomon Islands','SB','SLB')
,(204,'Somalia','SO','SOM')
,(205,'South Africa','ZA','ZAF')

,(206,'South Georgia and the South Sandwich Islands','GS','SGS')
,(207,'South Korea','KR','KOR')
,(208,'South Sudan','SS','SSD')
,(209,'Spain','ES','ESP')
,(210,'Sri Lanka','LK','LKA')
,(211,'Sudan','SD','SDN')
,(212,'Suriname','SR','SUR')
,(213,'Svalbard and Jan Mayen','SJ','SJM')
,(214,'Swaziland','SZ','SWZ')
,(215,'Sweden','SE','SWE')
,(216,'Switzerland','CH','CHE')
,(217,'Syria','SY','SYR')
,(218,'Taiwan','TW','TWN')
,(219,'Tajikistan','TJ','TJK')
,(220,'Tanzania','TZ','TZA')
,(221,'Thailand','TH','THA')
,(222,'Timor-Leste (East Timor)','TL','TLS')
,(223,'Togo','TG','TGO')
,(224,'Tokelau','TK','TKL')
,(225,'Tonga','TO','TON')
,(226,'Trinidad and Tobago','TT','TTO')
,(227,'Tunisia','TN','TUN')
,(228,'Turkey','TR','TUR')
,(229,'Turkmenistan','TM','TKM')
,(230,'Turks and Caicos Islands','TC','TCA')
,(231,'Tuvalu','TV','TUV')
,(232,'Uganda','UG','UGA')
,(233,'Ukraine','UA','UKR')
,(234,'United Arab Emirates','AE','ARE')
,(235,'United Kingdom','GB','GBR')
,(236,'United States','US','USA')

,(237,'United States Minor Outlying Islands','UM','UMI')
,(238,'Uruguay','UY','URY')
,(239,'Uzbekistan','UZ','UZB')
,(240,'Vanuatu','VU','VUT')
,(241,'Vatican City','VA','VAT')
,(242,'Venezuela','VE','VEN')
,(243,'Vietnam','VN','VNM')
,(244,'Virgin Islands, British','VG','VGB')
,(245,'Virgin Islands, US','VI','VIR')
,(246,'Wallis and Futuna','WF','WLF')
,(247,'Western Sahara','EH','ESH')
,(248,'Yemen','YE','YEM')
,(249,'Zambia','ZM','ZMB')
,(250,'Zimbabwe','ZW','ZWE');

Output Screenshots:

Country :

Namibia

Nauru

Country :

- Namibia
- Nauru
- Nepal
- Netherlands
- New Caledonia
- New Zealand
- Nicaragua
- Niger
- Nigeria
- Niue
- Norfolk Island
- North Korea
- Northern Mariana Islands
- Norway

Learning – Experience:

- Learnt about using AJAX.
- Learnt how to perform sql like operations and wildcards.
- Learnt about datalist and its use in html5 and how to map it to a text input.
- Learnt about xmlhttpready state and status.