Department of CSE
SSN College of Engineering

Vishakan Subramanian - 18 5001 196 - Semester VI

13 April 2021

UCS 1611 - Internet Programming Lab

Exercise 6: Autocomplete Feature Using AJAX

Learning Objective:

1. Develop an AJAX program that implements the Autocomplete feature for filling the country names.
2. 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.
3. When “Na” is typed only “Namibia” and “Nauru” should be listed for selection.
4. Use Servlet for server side scripting.
5. Use MySQL database to store the countries names.

Code - Search Page HTML:

```html
<!DOCTYPE html>
<html>

<head>
    <title>Autocomplete Using AJAX</title>
    <link rel="stylesheet" href="styles.css">
</head>

<body>

    <h1>Select Your Country</h1><hr><br>

    <div align="center">
        <p id="info"></p>
    </div>

    <div id="dropdown" align="center" class="dropdown-list">
        <input list="country-data" id="country-input" name="country-input"
    onkeyup="filterCountries()" />
        <datalist id="country-data">
        </datalist>
    </div>



    <footer>
        &copy; Vishakan Subramanian 2021
    </footer>

    <script src="scripts/index.js"></script>
</body>

</html>
```

Code - CSS Code:

```css
@import url('https://fonts.googleapis.com/css2?family=Montserrat&display=
    swap');

body {
    background: rgb(103,212,159);
    background: radial-gradient(circle, rgba(103,212,159,1) 0%, rgba
    (113,217,236,1) 100%);
    font-family: 'Montserrat', sans-serif;
    font-size: 150%;

}

h1 {
    text-align: center;
}

input{
    font-size: 25px;
    font-family: 'Montserrat', sans-serif;
    font-weight: bold;

}

p{
    text-align: center;
    font-weight: bold;
    color: red;
    font-size: 30px;
}

footer {
    position: fixed;
    bottom: 0;
    color: black;
    font-weight: bold;
    border-top: 3px solid black;
    padding: 5px 5px 5px 5px;
    text-align: center;
    width: 100%;
}
```

Code - Search Page JavaScript:

```javascript
filterCountries = () => {
    //filter the countries based on name

    let input = document.getElementById("country-input");
    let value = input.value.trim().toUpperCase();

    let xhttp = new XMLHttpRequest();
    let url = `FetchServlet?queryString=${value}`;

    xhttp.onreadystatechange = function () {
        if(this.readyState == 4 && this.status == 200){
            autoComplete(value, this);
        }
    };

    xhttp.open('GET', url, true);
    xhttp.send();
}

autoComplete = (value, xml) => {
    //populate the datalist's options by parsing the XML document response

    let i = 0;
    let xmlDoc = xml.responseXML;
    console.log(xmlDoc);

    let root = document.getElementsByTagName("datalist");
    root = root[0];
    removeAllChildNodes(root);  //remove all option nodes from the
   datalist

    let x = xmlDoc.getElementsByTagName("country");

    //displaying the number of records found
    let info = document.getElementById("info");
    info.innerHTML = "";    //clear the existing content

    if(value != ""){
        //display if queryString isn't an empty string
        info.innerHTML = `Found ${x.length} record(s) that match ${value
   }`;
    }

    for(i = 0; i < x.length; i++){
        //set the option nodes with its values for the datalist
        let item = x[i].childNodes[0].nodeValue;
        let optionNode = document.createElement("option");
        optionNode.setAttribute("value", item);
        root.appendChild(optionNode);
    }
}

removeAllChildNodes = (parent) => {
    //to remove all existing child nodes of an HTML element

    while (parent.firstChild) {
        parent.removeChild(parent.firstChild);
    }
}
```

Code - Web XML:

```
<web-app>
    <servlet>
        <servlet-name>FetchServlet</servlet-name>
        <servlet-class>FetchServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>FetchServlet</servlet-name>
        <url-pattern>/FetchServlet</url-pattern>
    </servlet-mapping>

    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
    </welcome-file-list>
</web-app>
```

Code - FetchServlet:

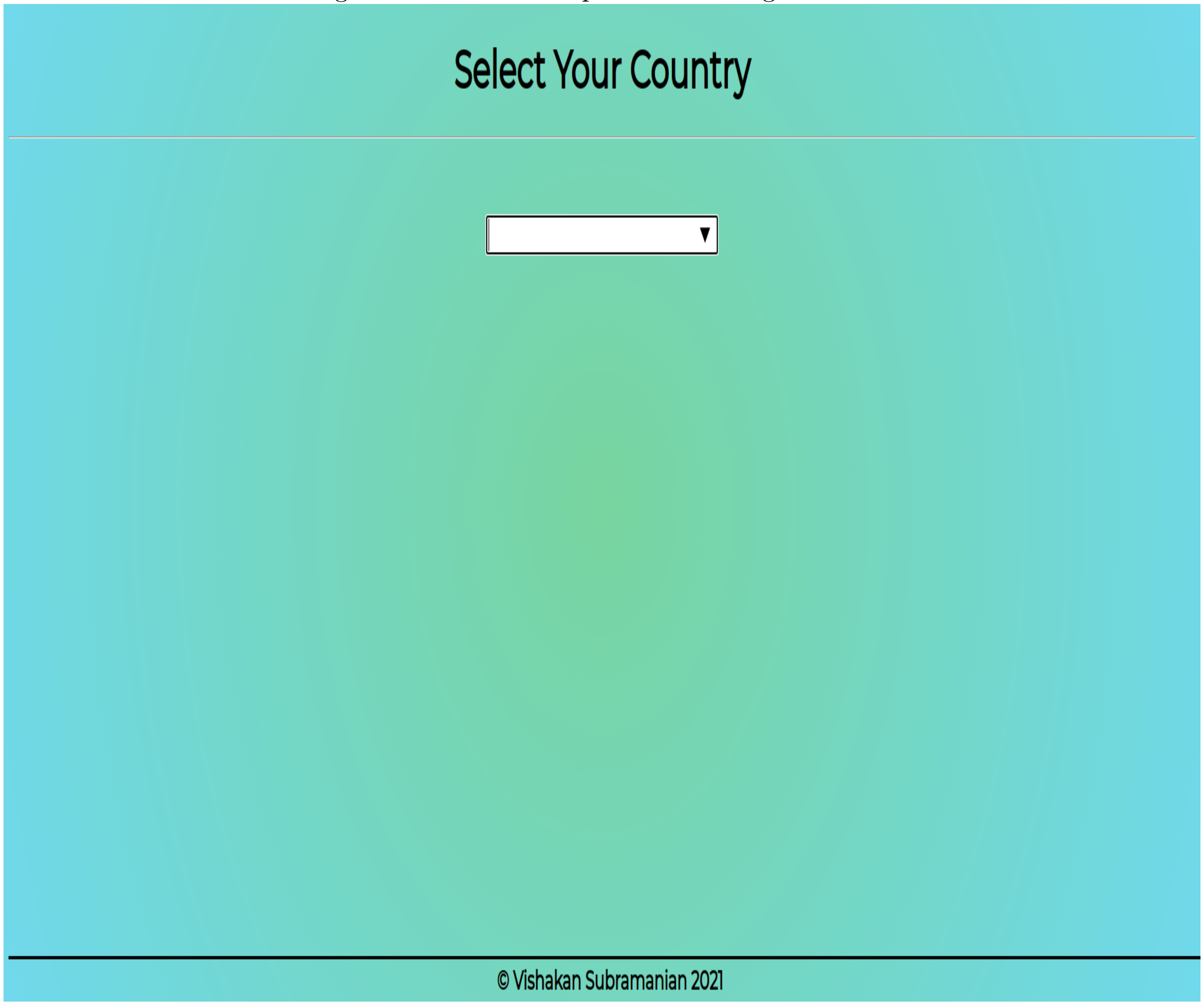
```java
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class FetchServlet extends HttpServlet{
    public void doGet(HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException{

        String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
        String DB_URL = "jdbc:mysql://localhost:3306/mysql?autoReconnect=true&useSSL=false";
        String USER = "admin";
        String PASS = "password";

        PrintWriter out = res.getWriter();

        String queryString = req.getParameter("queryString");

        //setting content type to XML
        res.setContentType("text/xml;charset=UTF-8");

        //Writing a raw XML document
        out.append("<?xml version=\"1.0\" encoding=\"UTF-8\"?>");
        out.append("<country_list>");

        try{
            Class.forName(JDBC_DRIVER);
            Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
            String query = "SELECT name FROM country WHERE name REGEXP \"^" + queryString + "\"";
            Statement stmt = conn.createStatement();

            ResultSet rs = stmt.executeQuery(query);

            while(rs.next()){   //appending queried countries to the XML
                String country = rs.getString("name");
                out.append("<country>" + country + "</country>");
            }

            out.append("</country_list>");

            conn.close();   //must close connection here as we may make too many requests
        }


        catch(SQLException se){
            se.printStackTrace();
            out.println(se);
        }
        catch(Exception e){
            e.printStackTrace();
            out.println(e);
        }
    }
}
```

Code - SQL Script:

```sql
/*Creating the countries table*/

USE mysql;

DROP TABLE country;

CREATE TABLE country(
  name varchar(80) NOT NULL
);


INSERT INTO country VALUES
('AFGHANISTAN'),
('ALBANIA'),
('ALGERIA'),
('ANDORRA'),
('ANGOLA'),
('ANTIGUA AND BARBUDA'),
('ARGENTINA'),
('ARMENIA'),
('AUSTRALIA'),
('AUSTRIA'),
('AZERBAIJAN'),
('THE BAHAMAS'),
('BAHRAIN'),
('BANGLADESH'),
('BARBADOS'),
('BELARUS'),
('BELGIUM'),
('BELIZE'),
('BENIN'),
('BHUTAN'),
('BOLIVIA'),
('BOSNIA AND HERZEGOVINA'),
('BOTSWANA'),
('BRAZIL'),
('BRUNEI'),
('BULGARIA'),
('BURKINA FASO'),
('BURUNDI'),
('CABO VERDE'),
('CAMBODIA'),
('CAMEROON'),
('CANADA'),
('CENTRAL AFRICAN REPUBLIC'),
('CHAD'),
('CHILE'),
('CHINA'),
('COLOMBIA'),
('COMOROS'),
('DEMOCRATIC REPUBLIC OF THE CONGO'),
('REPUBLIC OF THE CONGO'),
('COSTA RICA'),
('COTE D IVOIRE'),
('CROATIA'),
('CUBA'),
('CYPRUS'),
('CZECH REPUBLIC'),
('DENMARK'),
('DJIBOUTI'),
('DOMINICA'),
('DOMINICAN REPUBLIC'),
('EAST TIMOR (TIMOR-LESTE)'),
('ECUADOR'),
('EGYPT'),
('EL SALVADOR'),
('EQUATORIAL GUINEA'),
('ERITREA'),
('ESTONIA'),
('ESWATINI'),
('ETHIOPIA'),
('FIJI'),
('FINLAND'),
('FRANCE'),
('GABON'),
('THE GAMBIA'),
('GEORGIA'),
('GERMANY'),
('GHANA'),
('GREECE'),
('GRENADA'),
('GUATEMALA'),
('GUINEA'),
('GUINEA-BISSAU'),
('GUYANA'),
('HAITI'),
('HONDURAS'),
('HUNGARY'),
('ICELAND'),
('INDIA'),
('INDONESIA'),
('IRAN'),
('IRAQ'),
('IRELAND'),
('ISRAEL'),
('ITALY'),
('JAMAICA'),
('JAPAN'),
('JORDAN'),
('KAZAKHSTAN'),
('KENYA'),
('KIRIBATI'),
('KOREA, NORTH'),
('KOREA, SOUTH'),
('KOSOVO'),
('KUWAIT'),
('KYRGYZSTAN'),
('LAOS'),
('LATVIA'),
('LEBANON'),
('LESOTHO'),
('LIBERIA'),
('LIBYA'),
('LIECHTENSTEIN'),
('LITHUANIA'),
('LUXEMBOURG'),
('MADAGASCAR'),
('MALAWI'),
('MALAYSIA'),
('MALDIVES'),
('MALI'),
('MALTA'),
('MARSHALL ISLANDS'),
('MAURITANIA'),
('MAURITIUS'),
('MEXICO'),
('FEDERATED STATES OF MICRONESIA'),
('MOLDOVA'),
('MONACO'),
('MONGOLIA'),
('MONTENEGRO'),
('MOROCCO'),
('MOZAMBIQUE'),
('MYANMAR (BURMA)'),
('NAMIBIA'),
('NAURU'),
('NEPAL'),
('NETHERLANDS'),
('NEW ZEALAND'),
('NICARAGUA'),
('NIGER'),
('NIGERIA'),
('NORTH MACEDONIA'),
('NORWAY'),
('OMAN'),
('PAKISTAN'),
('PALAU'),
('PANAMA'),
('PAPUA NEW GUINEA'),
('PARAGUAY'),
('PERU'),
('PHILIPPINES'),
('POLAND'),
('PORTUGAL'),
('QATAR'),
('ROMANIA'),
('RUSSIA'),
('RWANDA'),
('SAINT KITTS AND NEVIS'),
('SAINT LUCIA'),
('SAINT VINCENT AND THE GRENADINES'),
('SAMOA'),
('SAN MARINO'),
('SAO TOME AND PRINCIPE'),
('SAUDI ARABIA'),
('SENEGAL'),
('SERBIA'),
('SEYCHELLES'),
('SIERRA LEONE'),
('SINGAPORE'),
('SLOVAKIA'),
('SLOVENIA'),
('SOLOMON ISLANDS'),
('SOMALIA'),
('SOUTH AFRICA'),
('SPAIN'),
('SRI LANKA'),
('SUDAN'),
('SOUTH SUDAN'),
('SURINAME'),
('SWEDEN'),
('SWITZERLAND'),
('SYRIA'),
('TAIWAN'),
('TAJIKISTAN'),
('TANZANIA'),
('THAILAND'),
('TOGO'),
('TONGA'),
('TRINIDAD AND TOBAGO'),
('TUNISIA'),
('TURKEY'),
('TURKMENISTAN'),
('TUVALU'),
('UGANDA'),
('UKRAINE'),
('UNITED ARAB EMIRATES'),
('UNITED KINGDOM'),
('UNITED STATES'),
('URUGUAY'),
('UZBEKISTAN'),
('VANUATU'),
('VATICAN CITY'),
('VENEZUELA'),
('VIETNAM'),
('YEMEN'),
('ZAMBIA'),
('ZIMBABWE');
```

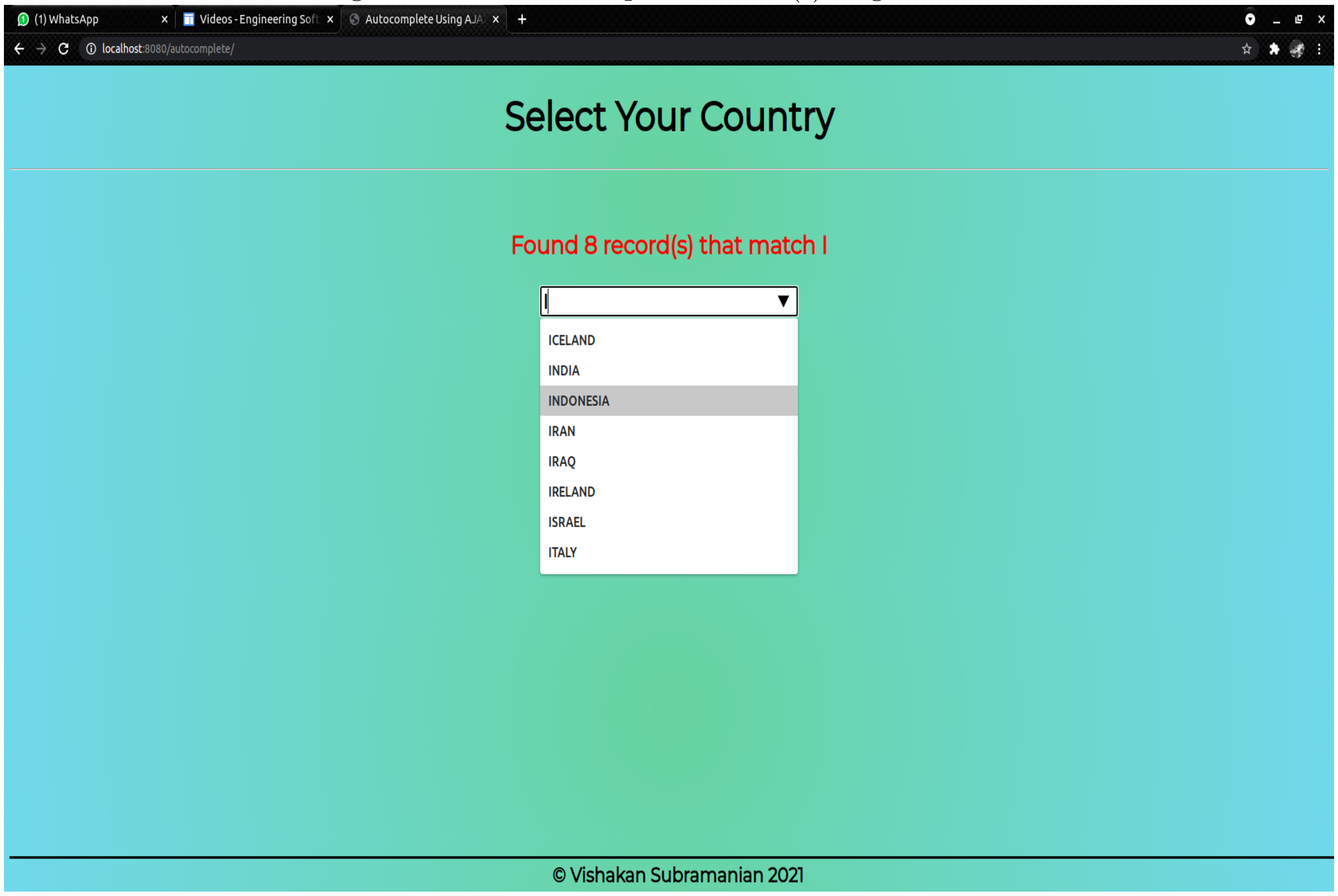
Output - Search Page:

Figure 1: Browser Output: Search Page.



Output - Search (I) Page:

Figure 2: Browser Output: Search (I) Page.



Select Your Country

Found 8 record(s) that match I

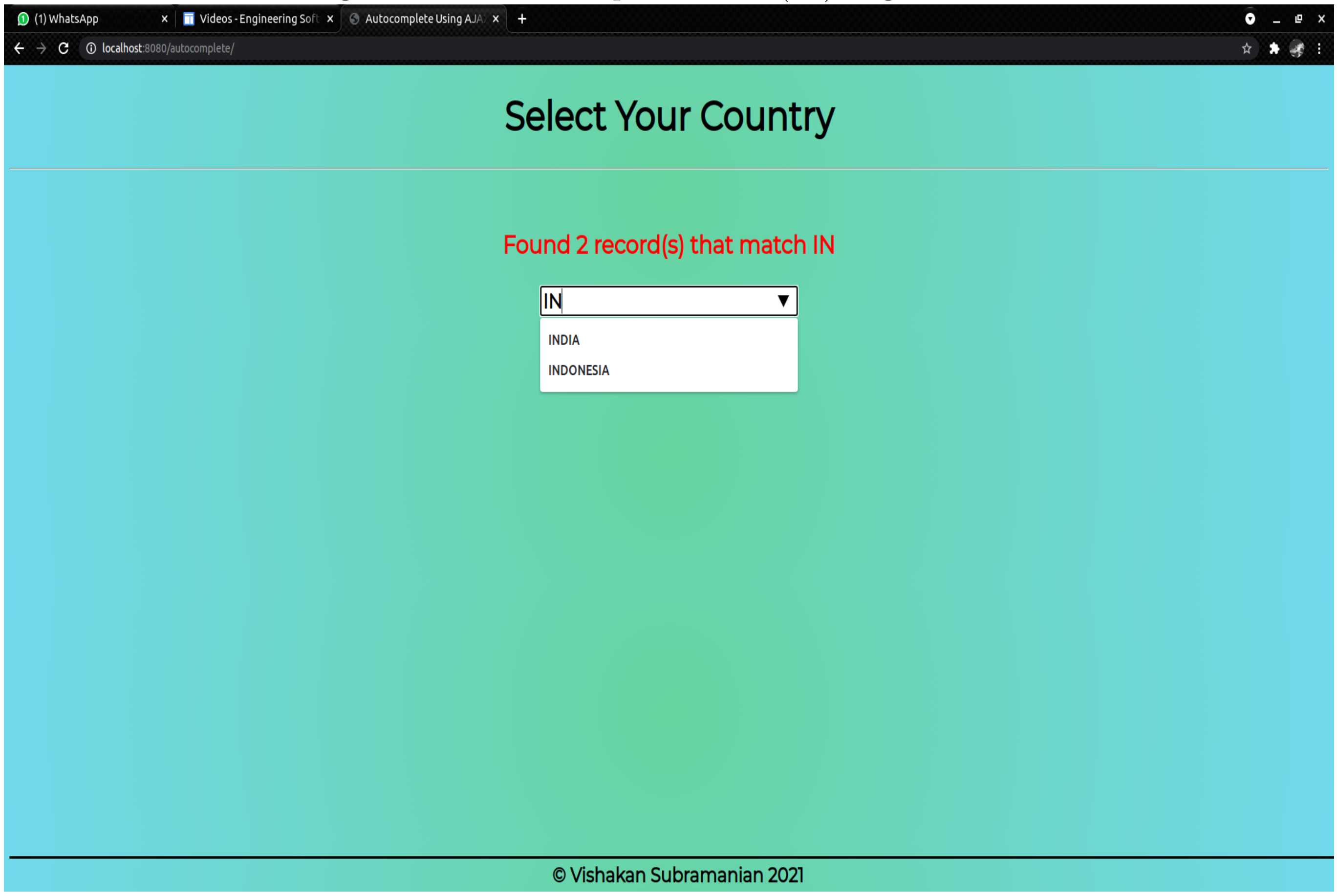
I

ICELAND
INDIA
INDONESIA
IRAN
IRAQ
IRELAND
ISRAEL
ITALY

© Vishakan Subramanian 2021

Output - Search (IN) Page:

Figure 3: Browser Output: Search (IN) Page.



Select Your Country

Found 2 record(s) that match IN

IN

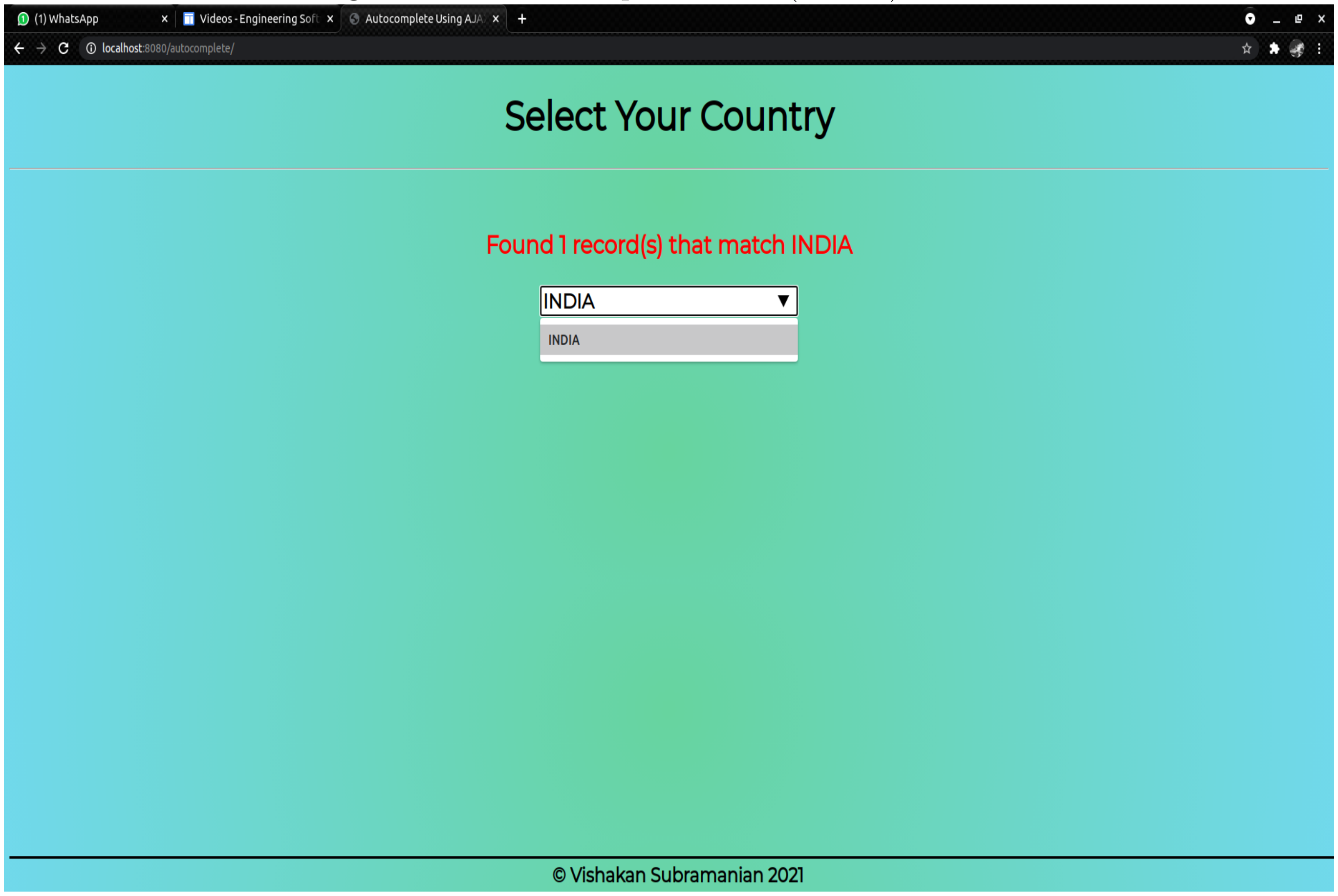
INDIA

INDONESIA

© Vishakan Subramanian 2021

Output - Search (INDIA) Page:

Figure 4: Browser Output: Search (INDIA).



Select Your Country

Found 1 record(s) that match INDIA

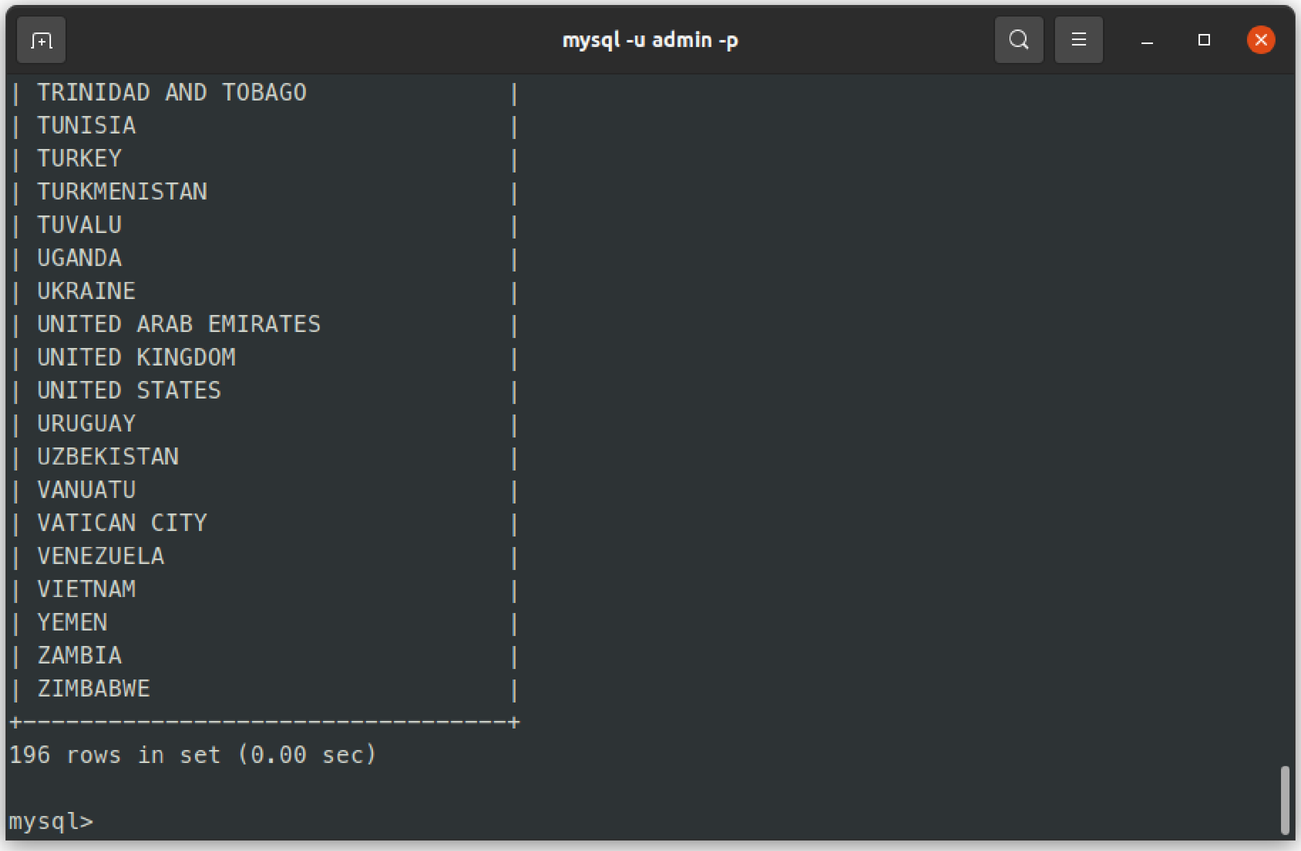
INDIA ▼

INDIA

© Vishakan Subramanian 2021

Output - SQL Table:

Figure 5: Browser Output: SQLTable.



```
| TRINIDAD AND TOBAGO               |
| TUNISIA                           |
| TURKEY                            |
| TURKMENISTAN                      |
| TUVALU                            |
| UGANDA                            |
| UKRAINE                           |
| UNITED ARAB EMIRATES              |
| UNITED KINGDOM                    |
| UNITED STATES                     |
| URUGUAY                           |
| UZBEKISTAN                        |
| VANUATU                           |
| VATICAN CITY                      |
| VENEZUELA                         |
| VIETNAM                           |
| YEMEN                             |
| ZAMBIA                            |
| ZIMBABWE                          |
+-----------------------------------+
196 rows in set (0.00 sec)

mysql>
```

Learning Outcome:

- From the experiment, I learnt to implement **Java Servlets**.
- I learnt how to serve an XML document from the server-side dynamically using Java Servlets.
- I was able to integrate a MySQL database to my Java Servlet program using JDBC connector.
- I was able to pass a GET method call from my client-side using JavaScript with the use of an **XHTTPRequest Object**.
- I understood the basic working and process of an **AJAX request and retrieval**.
- I was able to parse the XML document retrieved using AJAX with the help of JavaScript DOM methods like getElementsByTagName() and getElementById().
- I modified my HTML document dynamically with JavaScript and was able to add the relevant countries as per the user's query string by creating **Node objects** and appending them as a child to a root element in the HTML page, and set it's value attribute using setAttribute() method.
- I was able to remove all child nodes of the root element using removeChild() method.
- I learnt to query SQL using **Regular Expressions**.
- I understood the importance of closing SQL connections in Java once the records had been retrieved. Failure to close the connections leads to too many open connections if the user changes his/her query frequently and the program failing to work in the front-end side due to that.
- I was able to implement a working **Autocomplete Feature** using AJAX.