

Lab12(B.Bhagyasri)

1. Create a webpage with Registration Form to accept the following details from user.

Firstname

Lastname

Mobile number

Email Id

Qualification (Drop down list) (BE, BTech, BSc, BCA)

Year of Completion Drop down List (2023, 2022, 2021, 2020)

Technical Skills Drop down List (Web designing, Core Java, Mysql)

(Spring, spring Boot with REST API)

(core Java)

(javascript)

Interested in Relocation (yes/No)

submit reset

read and submit the data need to be stored in javascript object and in the same page display using table.

Program:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Reg.details</title>
```

```
<style>
```

```
table {
```

```
border-collapse: collapse;
```

```
width: 70%;
```

```
margin: auto;
```

```
}
```

```
th, td {
```

```
        border: 1px solid black;

        padding: 8px;

        text-align: left;

    }

</style>

</head>

<script>

function storeData() {

    var firstName = document.getElementById("firstName").value;

    var lastName = document.getElementById("lastName").value;

    var mobileNo = document.getElementById("mobileNo").value;

    var email = document.getElementById("email").value;

    var qualification = document.getElementById("qualification").value;

    var yearOfCompletion = document.getElementById("yearOfCompletion").value;

    var technicalSkills = document.getElementById("technicalSkills").value;

    var relocationYes = document.getElementById("relocationYes").checked;

    var relocationNo = document.getElementById("relocationNo").checked;


    var interestedInRelocation = relocationYes ? "Yes" : (relocationNo ? "No" : "");


    var data = {

        firstName: firstName,

        lastName: lastName,

        mobileNo: mobileNo,

        email: email,

        qualification: qualification,

        yearOfCompletion: yearOfCompletion,

        technicalSkills: technicalSkills,

        interestedInRelocation: interestedInRelocation
```

```
};
```

```
var storedData = localStorage.getItem("submittedData");
if (storedData) {
    storedData = JSON.parse(storedData);
    storedData.push(data);
} else {
    storedData = [data];
}localStorage.setItem("submittedData", JSON.stringify(storedData));
displayData(); // Call function to display the stored data
alert("Data submitted and stored!");
} function resetForm() {
    document.getElementById("registrationForm").reset();
}function displayData() {
    var storedData = localStorage.getItem("submittedData");
    if (storedData) {
        storedData = JSON.parse(storedData);
        var tableBody = document.getElementById("dataDisplay");
        // Clear existing table rows
        tableBody.innerHTML = "";
        // Populate the table rows with the stored data
        for (var i = 0; i < storedData.length; i++) {
            var row = tableBody.insertRow();
            var cell1 = row.insertCell(0);
            var cell2 = row.insertCell(1);
            var cell3 = row.insertCell(2);
            var cell4 = row.insertCell(3);
            var cell5 = row.insertCell(4);
            var cell6 = row.insertCell(5);
```

```

        var cell7 = row.insertCell(6);
        var cell8 = row.insertCell(7);

        cell1.innerHTML = storedData[i].firstName;
        cell2.innerHTML = storedData[i].lastName;
        cell3.innerHTML = storedData[i].mobileNo;
        cell4.innerHTML = storedData[i].email;
        cell5.innerHTML = storedData[i].qualification;
        cell6.innerHTML = storedData[i].yearOfCompletion;
        cell7.innerHTML = storedData[i].technicalSkills;
        cell8.innerHTML = storedData[i].interestedInRelocation;
    }
}
}
</script>
<body>
    <h2>Registration Form</h2>
    <main>
        <form id="registrationForm" onsubmit="storeData(); return false;">
            <label for="firstName">Firstname: </label>
            <input type="text" id="firstName" required><br><br>

            <label for="lastName">Lastname: </label>
            <input type="text" id="lastName" required><br><br>

            <label for="mobileNo">MobileNo: </label>
            <input type="text" id="mobileNo" required><br><br>

            <label for="email">Email: </label>
            <input type="email" id="email" required><br><br>
        </form>
    </main>
</body>
</html>

```

<label for="qualification">Qualification:</label>

<select id="qualification" required>

<option value="BE">BE</option>

<option value="B.tech">B.tech</option>

<option value="Bsc">Bsc</option>

<option value="BCA">BCA</option>

</select>

<label for="yearOfCompletion">Year of Completion:</label>

<select id="yearOfCompletion" required>

<option value="2023">2023</option>

<option value="2022">2022</option>

<option value="2021">2021</option>

<option value="2020">2020</option>

</select>

<label for="technicalSkills">Technical Skills:</label>

<select id="technicalSkills" required>

<option value="web designing">web Designing</option>

<option value="Core Java">Core Java</option>

<option value="Mysql">Mysql</option>

<option value="Spring">Spring</option>

<option value="spring Boot with REST API">spring Boot with REST API</option>

<option value="Javascript">Javascript</option>

</select>

<label for="relocation">Interested in Relocation:</label>

<input type="radio" id="relocationYes" name="relocation" required>

```
<label for="relocationYes">Yes</label>
<input type="radio" id="relocationNo" name="relocation" required>
<label for="relocationNo">No</label>
<br><br>

<button type="submit">Submit</button>
<button type="button" onclick="resetForm()">Reset</button>
</form>
</main>
```

```
<h2>Registered Users:</h2>
<table>
  <thead>
    <tr>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Mobile Number</th>
      <th>Email Id</th>
      <th>Qualification</th>
      <th>Year of Completion</th>
      <th>Technical</th>
      <th>Interested in location</th>
    </tr>
  </thead>
  <tbody id="dataDisplay"></tbody>
</table>
</body>
</html>
```

Reg.details

File

C:/Users/bodur/OneDrive/Documents/html/StudentRegistration.html

☆

Registration Form

Firstname:

Boduri

Lastname:

Bhagyasri

MobileNo:

8985852830

Email:

boduribhagyasri123@gmail

Qualification:

B.tech

Year of Completion:

2023

Technical Skills:

Core Java

Interested in Relocation:

☒ Yes
 ☐ No

Submit

Reset

Registered Users:

First Name	Last Name	Mobile Number	Email Id	Qualification	Year of Completion	Technical	Interested in location
hbrdf	rfin	rtfg	Bhghi@gmail.com	BE	2023	web designing	Yes
Velaga	Tulasi	8985852830	Bhghi@gmail.com	BE	2023	web designing	No
Boduri	Bhagyasri	8985852830	boduribhagyasri123@gmail.com	B.tech	2023	Core Java	Yes

32°C

Partly sunny

Search

ENG

IN

12:56

25-08-2023

2. Create a webpage using table with Css to write the advantages of Spring boot with explanations.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title> Advantages of springBoot </title>
```

```
</head>
```

```
<body bgcolor="pink" Tablecolor="blue">
```

```
<Table border="5" width ="50%" height="100%" cellpadding="5" cellspacing="0" table
style="background-color:palegoldenrod;margin: 0 auto;" >
```

```
<tr bgcolor="honeydew">
```

```
<h1 style="text-align: center;" >Advantages of Spring Boot </h1>
```

```
<th>Advantage</th>
```

```
<th>Explanation</th>
```

</tr>

<tr>

<td>1. Simplified Configuration</td>

<td>Spring Boot provides sensible defaults and auto-configuration, reducing the need for manual configuration.</td>

</tr>

<tr>

<td>2. Embedded Servers</td>

<td>Spring Boot includes embedded servers like Tomcat, Jetty, or Undertow, simplifying deployment.</td>

</tr><tr>

<td>3. Dependency Management</td>

<td>Spring Boot manages project dependencies through its built-in dependency management system. It ensures that the required versions of libraries are compatible, reducing the likelihood of conflicts. This prevents developers from spending time on manually managing dependencies.</td>

<tr>

<td>4. Microservices Support</td>

<td>Spring Boot's architecture is well-suited for building microservices. It provides features like Spring Cloud for creating and managing microservices, including service discovery, load balancing, and distributed configuration. This allows developers to easily develop and deploy microservices-based applications.</td>

<tr>

<td>5. Production-Ready Actuators</td>

<td>Spring Boot includes production-ready endpoints called "actuators" that provide essential insights into the application's health, metrics, and other operational aspects. These actuator endpoints enable monitoring and management of the application in a production environment.</td>

</tr>

<tr>

<td>6. Auto-Configuration</td>

<td>Spring Boot's auto-configuration feature analyzes the application's classpath and automatically configures beans based on the libraries present. This means developers can focus on writing business logic rather than manually configuring components.</td></tr>

</tr>

<tr>

<td> 7.Reduced Development Time</td>

<td>Spring Boot's opinionated defaults and ready-to-use components significantly reduce the time needed to set up a project. Developers can start coding business logic almost immediately, speeding up the overall development process.</td>

</tr>

</table>

</body>

</html>

