VIJAYA LAKSHMI SRAVANTi LAB12

Q1. Create a webpage with Registration Form to accept the follwing 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)

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

(Spring, spring Boot with REST API)

(core Java)

(javascript)

Intersted 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.

Student Registration.html:

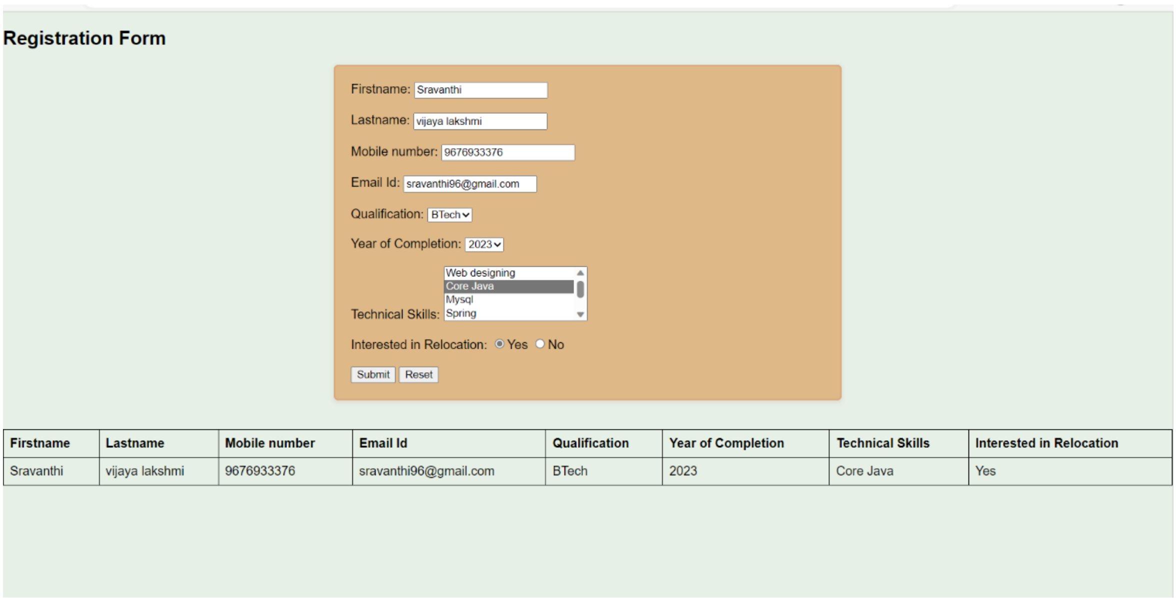
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Registration Form</title>
<style>
body {
font-family: Arial, sans-serif;
background-color:#e0ede1cd;

margin: 0;
padding: 0;
}
h1 {
text-align: center;
padding: 20px 0;
}
form {
background-color: burlywood;
padding: 20px;
border-radius: 5px;
box-shadow: 0 2px 5px rgba(144, 12, 12, 0.1);
max-width: 600px;
margin: 0 auto;
border: 2px solid rgba(167, 25, 25, 0.1); }
table {
border-collapse: collapse;
margin: right;
width: 100%;
}
th, td {
border: 1px solid black;
padding: 8px;
text-align: left;
}
</style>
</head>
<body>
<h2>Registration Form</h2>
<form id="registrationForm">
<label for="firstname">Firstname:</label>
<input type="text" id="firstname" name="firstname" required><br><br>
<label for="lastname">Lastname:</label>
<input type="text" id="lastname" name="lastname" required><br><br>
<label for="mobile">Mobile number:</label>
<input type="text" id="mobile" name="mobile" required><br><br>
<label for="email">Email Id:</label>
<input type="email" id="email" name="email" required><br><br>
<label for="qualification">Qualification:</label>
<select id="qualification" name="qualification">

<option value="BE">BE</option>
<option value="BTech">BTech</option>
<option value="BSc">BSc</option>
<option value="BCA">BCA</option>
<option value="MSc">MCA</option>
</select><br><br>
<label for="year">Year of Completion:</label>
<select id="year" name="year">
<option value="2023">2023</option>
<option value="2022">2022</option>
<option value="2021">2021</option>
<option value="2020">2020</option>
<option value="2019">2019</option>
</select><br><br>
```

```
<label for="skills">Technical Skills:</label>

<select id="skills" name="skills" multiple>

<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><br><br>

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

<input type="radio" id="relocationYes" name="relocation" value="Yes">Yes

<input type="radio" id="relocationNo" name="relocation"value="No">No<br><br>

<input type="button" value="Submit" onclick="submitForm()">

<input type="reset" value="Reset">

</form>

<br><br>

<table id="displayTable">

<thead>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Mobile number</th>

<th>Email Id</th>


<th>Qualification</th>

<th>Year of Completion</th>

<th>Technical Skills</th>

<th>Interested in Relocation</th>

</tr>

</thead>

<tbody id="tableBody">
```

```
</tbody>
</table>
<script>
function submitForm() {
const firstname = document.getElementById("firstname").value;
const lastname = document.getElementById("lastname").value;
const mobile = document.getElementById("mobile").value;
const email = document.getElementById("email").value;
const qualification = document.getElementById("qualification").value;
const year = document.getElementById("year").value;
const skills =Array.from(document.getElementById("skills").selectedOptions).map(option=> option.value);
const relocation =document.querySelector('input[name="relocation"]:checked').value;
const newRow = document.getElementById("tableBody").insertRow();
newRow.insertCell().textContent = firstname;
newRow.insertCell().textContent = lastname;
newRow.insertCell().textContent = mobile;
newRow.insertCell().textContent = email;
newRow.insertCell().textContent = qualification;
newRow.insertCell().textContent = year;
newRow.insertCell().textContent = skills.join(', ');
newRow.insertCell().textContent = relocation;
}
</script>
</body>
</html>
```

Output:



Registration Form

Firstname: Sravanthi

Lastname: vijaya lakshmi

Mobile number: 9676933376

Email Id: sravanthi96@gmail.com

Qualification: BTech

Year of Completion: 2023

Technical Skills: Web designing / Core Java / Mysql / Spring

Interested in Relocation: ● Yes ○ No

Firstname	Lastname	Mobile number	Email Id	Qualification	Year of Completion	Technical Skills	Interested in Relocation
Sravanthi	vijaya lakshmi	9676933376	sravanthi96@gmail.com	BTech	2023	Core Java	Yes

Q2. Create a webpage using table with Css to write the advantages of Spring boot with explanations.

Advantages.html:

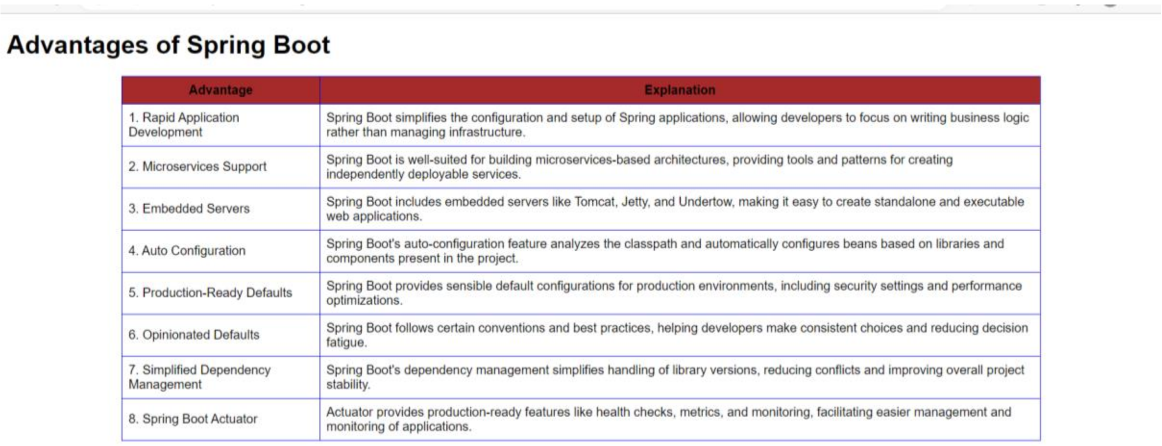
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Advantages of Spring Boot</title>
<style>
body {
font-family: Arial, sans-serif;
background: color #e0ede1cd;
}
```

```
table {
border-collapse: collapse;
width: 80%;
margin: auto;
border: 1px solid #0000dc;
}
th, td {
border: 1px solid #0000dc;
padding: 8px;
}
th {
background-color: #a52a2a;
}
</style>
</head>
<body>
<h1>Advantages of Spring Boot</h1>
<table>
<tr>
<th>Advantage</th>
<th>Explanation</th>
</tr>
<tr>
<td>1. Rapid Application Development</td>
<td>Spring Boot simplifies the configuration and setup of Spring applications,
allowing developers to focus on writing business logic rather than managing
infrastructure.</td>
</tr>
<tr>
```

```html
<td>2. Microservices Support</td>
<td>Spring Boot is well-suited for building microservices-based architectures,
providing tools and patterns for creating independently deployable
services.</td>
</tr>
<tr>
<td>3. Embedded Servers</td>
<td>Spring Boot includes embedded servers like Tomcat, Jetty, and
Undertow, making it easy to create standalone and executable web
applications.</td>
</tr>
<tr>
<td>4. Auto Configuration</td>
<td>Spring Boot's auto-configuration feature analyzes the classpath and
automatically configures beans based on libraries and components present in
the project.</td>
</tr>
<tr>
<td>5. Production-Ready Defaults</td>
<td>Spring Boot provides sensible default configurations for production
environments, including security settings and performance optimizations.</td>
</tr>
<tr>
<td>6. Opinionated Defaults</td>
<td>Spring Boot follows certain conventions and best practices, helping
developers make consistent choices and reducing decision fatigue.</td>
</tr>
<tr>
```

```
<td>7. Simplified Dependency Management</td>
<td>Spring Boot's dependency management simplifies handling of library
versions, reducing conflicts and improving overall project stability.</td>
</tr>
<tr>
<td>8. Spring Boot Actuator</td>
<td>Actuator provides production-ready features like health checks, metrics,
and monitoring, facilitating easier management and monitoring of
applications.</td>
</tr>
</table>
</body>
</html>
```

Output:



Advantages of Spring Boot

Advantage	Explanation
1. Rapid Application Development	Spring Boot simplifies the configuration and setup of Spring applications, allowing developers to focus on writing business logic rather than managing infrastructure.
2. Microservices Support	Spring Boot is well-suited for building microservices-based architectures, providing tools and patterns for creating independently deployable services.
3. Embedded Servers	Spring Boot includes embedded servers like Tomcat, Jetty, and Undertow, making it easy to create standalone and executable web applications.
4. Auto Configuration	Spring Boot's auto-configuration feature analyzes the classpath and automatically configures beans based on libraries and components present in the project.
5. Production-Ready Defaults	Spring Boot provides sensible default configurations for production environments, including security settings and performance optimizations.
6. Opinionated Defaults	Spring Boot follows certain conventions and best practices, helping developers make consistent choices and reducing decision fatigue.
7. Simplified Dependency Management	Spring Boot's dependency management simplifies handling of library versions, reducing conflicts and improving overall project stability.
8. Spring Boot Actuator	Actuator provides production-ready features like health checks, metrics, and monitoring, facilitating easier management and monitoring of applications.