WEBFORMS ASSIGNMENT

1. Create a webapge 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.

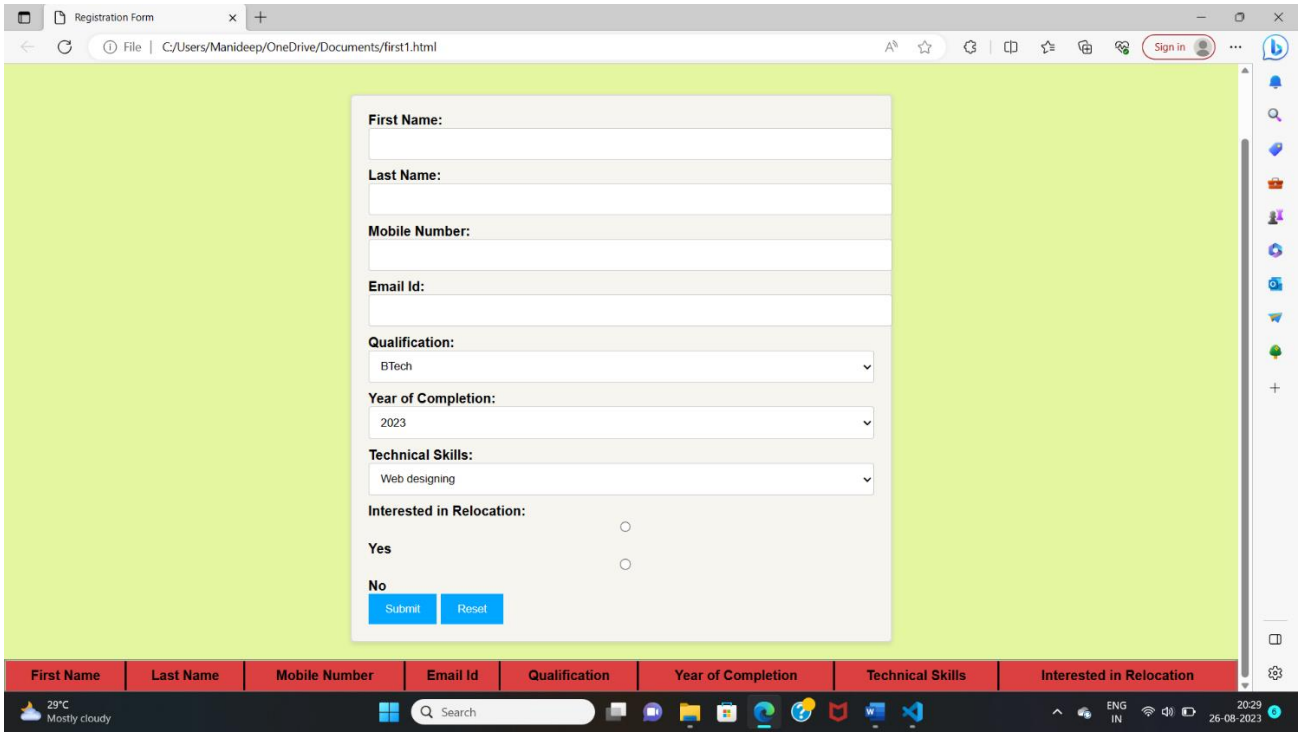
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Registration Form</title>
<style>
body {
font-family: Arial, sans-serif;
background-color:#ffffff;
margin: 0;
padding: 0;
}
h1 {
text-align: center;
padding: 20px 0;
}
form {
background-color:#e38787;
padding: 20px;
border-radius: 5px;
box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
max-width: 600px;
margin: 0 auto;
```

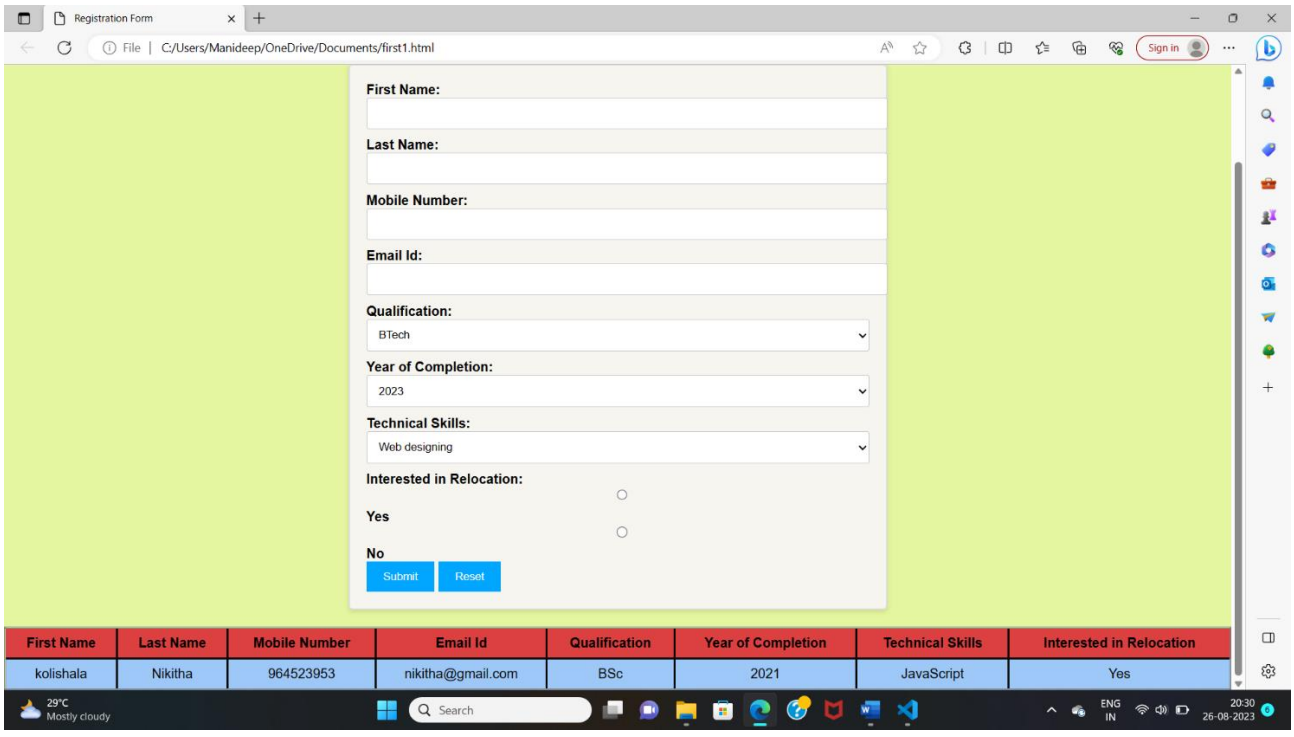
```
border: 2px solid rgba(0, 0, 0, 0.1);
}
label {
font-weight: bold;
}
input[type="text"],
input[type="tel"],
input[type="email"],
select,
input[type="radio"] {
width: 100%;
padding: 10px;
margin-bottom: 10px;
border: 1px solid #ccc;
border-radius: 3px;
}
select[multiple] {
height: auto;
}
input[type="submit"],
input[type="reset"] {
background-color: #007bff;
color: #ffffff;
border: none;
padding: 10px 20px;
cursor: pointer;
}
input[type="submit"]:hover,
input[type="reset"]:hover {
background-color: #b000b3;
}
table {
width: 100%;
border-collapse: collapse;
margin-top: 20px;
border-style: double;
border-width: 3px;
}
th, td {
border: 3px solid #090808;
padding: 8px;
text-align: center;
}
th {
background-color: #5768c8;
}
tr:nth-child(even) {
background-color: #9ec8f7;
}
</style>
</head>
<body>
<h1>Registration Form</h1>
<form id="registrationForm">
<label for="firstname">First Name:</label>
<input type="text" id="firstname" name="firstname" required><br>
<label for="lastname">Last Name:</label>
<input type="text" id="lastname" name="lastname" required><br>
<label for="mobile">Mobile Number:</label>
```

```
<input type="tel" id="mobile" name="mobile" required><br>
<label for="email">Email Id:</label>
<input type="email" id="email" name="email" required><br>
<label for="qualification">Qualification:</label>
<select id="qualification" name="qualification">
<option value="BTech">BTech</option>
<option value="BE">BE</option>
<option value="BSc">BSc</option>
<option value="BCA">BCA</option>
</select><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>
</select><br>
<label for="skills">Technical Skills:</label>
<select id="skills" name="skills">
<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>
<label for="relocation">Interested in Relocation:</label>
<input type="radio" id="relocationYes" name="relocation" value="Yes"
required>
<label for="relocationYes">Yes</label>
<input type="radio" id="relocationNo" name="relocation" value="No"
required>
<label for="relocationNo">No</label><br>
<input type="submit" value="Submit">
<input type="reset" value="Reset">
</form>
<table id="dataTable">
<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 Skills</th>
<th>Interested in Relocation</th>
</tr>
</table>
<script>
const registrationForm = document.getElementById("registrationForm");
const dataTable = document.getElementById("dataTable");
registrationForm.addEventListener("submit", function(event) {
event.preventDefault();
const formData = new FormData(registrationForm);
const dataObject = {};
formData.forEach((value, key) => {
if (key === "skills") {
dataObject[key] = Array.from(formData.getAll(key));
```

```
} else {
dataObject[key] = value;
}
});
const newRow = dataTable.insertRow();
for (const key in dataObject) {
const cell = newRow.insertCell();
cell.textContent = dataObject[key];
}
registrationForm.reset();
});
</script>
</body>
</html>
```

OUTPUT:





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

```
<!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:beige;
margin: 0;
padding: 0;
}
h1 {
text-align: center;
padding: 20px 0;
}
table {
width: 80%;
margin: 20px auto;
border-collapse: collapse;
border: double;
border: 3px solid #100f0f;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
th, td {
padding: 10px;
text-align: left;
}
th {
background-color: #63c2f5;
}
tr:nth-child(even) {
background-color: #ee7878;
}
</style>
</head>
<body>
<h1>Advantages of Spring Boot</h1>
<table>
<tr>
<th>Advantage</th>
<th>Explanation</th>
</tr>
<tr>
<td>1. Rapid Development</td>
<td>Spring Boot simplifies and accelerates application development by
providing default configurations and eliminating boilerplate code.</td>
</tr>
<tr>
<td>2. Microservices</td>
<td>Spring Boot is well-suited for building microservices-based
architectures, making it easier to develop, deploy, and manage individual
components.</td>
```

```
</tr>
<tr>
<td>3. Embedded Servers</td>
<td>Spring Boot includes embedded servers like Tomcat and Jetty,
making it convenient to package and run applications as standalone JARs.</td>
</tr>
<tr>
<td>4. Opinionated Defaults</td>
<td>Spring Boot provides sensible defaults for various configurations,
reducing the need for manual setup and allowing developers to focus on
coding.</td>
</tr>
<tr>
<td>5. Auto Configuration</td>
<td>Spring Boot automatically configures components based on the
project's classpath, minimizing the need for explicit configuration.</td>
</tr>
</table>
</body>
</html>
```

Output: