Assignment:-4

Practical Questions on AJAX

1. Create a button that sends an AJAX GET request and displays the response on the page

```
<html>
<body>
<button onclick="loadData()">Get Data</button>
<div id="output"></div>
<script>
function loadData() {
  var xhr = new XMLHttpRequest();
  xhr.open("GET", "https://jsonplaceholder.typicode.com/posts/1", true);
  xhr.onload = function() {
    if (xhr.status == 200) {
      document.getElementById("output").innerHTML = xhr.responseText;
    }
  };
  xhr.send();
}
</script>
</body>
</html>
```

2. Build a form that collects user data and sends it via AJAX POST to a mock API

```
<html>
<body>
<form onsubmit="sendData(); return false;">
  Name: <input type="text" id="name"><br>
  Email: <input type="email" id="email"><br>
  <button type="submit">Submit</button>
</form>
<script>
function sendData() {
  var xhr = new XMLHttpRequest();
  xhr.open("POST", "https://jsonplaceholder.typicode.com/posts", true);
  xhr.setRequestHeader("Content-Type", "application/json");
  var data = JSON.stringify({
    name: document.getElementById("name").value,
    email: document.getElementById("email").value
  });
  xhr.send(data);
}
</script>
</body>
</html>
```

3. Use AJAX to fetch a list of users and display them in an HTML table

```
<html>
<body>
<table border="1" id="userTable"></table>
<script>
fetch("https://jsonplaceholder.typicode.com/users")
  .then(res => res.json())
  .then(data => {
    let rows = "<tr><th>Name</th><th>Email</th></tr>";
    data.forEach(user => {
      rows += `<tr><td>${user.name}</td><td>${user.email}</td></tr>`;
    });
    document.getElementById("userTable").innerHTML = rows;
  });
</script>
</body>
```

```
</html>
```

4. Log each readyState value during an AJAX request and explain what happens at each stage

```
<script>
var xhr = new XMLHttpRequest();
xhr.onreadystatechange = function() {
  console.log("readyState: " + xhr.readyState);
  if (xhr.readyState == 4 && xhr.status == 200) {
    console.log("Response: ", xhr.responseText);
  }
};
xhr.open("GET", "https://jsonplaceholder.typicode.com/posts/1", true);
xhr.send();
// readyState values: 0=UNSENT, 1=OPENED, 2=HEADERS_RECEIVED, 3=LOADING, 4=DONE
</script>
```

5. Load XML data using AJAX and display specific XML elements in HTML

```
<html>
<body>
<div id="xmlData"></div>
<script>
var xhr = new XMLHttpRequest();
xhr.open("GET", "data.xml", true);
xhr.onreadystatechange = function() {
  if (xhr.readyState == 4 && xhr.status == 200) {
    var xml = xhr.responseXML;
    var items = xml.getElementsByTagName("item");
    let output = "";
    for (let i = 0; i < items.length; i++) {
      output += items[i].getElementsByTagName("name")[0].textContent + "<br>";
    }
    document.getElementById("xmlData").innerHTML = output;
  }
};
xhr.send();
</script>
</body>
</html>
```

6. Write one AJAX call in synchronous mode and one in asynchronous mode. Compare their behavior

```
<script>
// Asynchronous (non-blocking)
var xhrAsync = new XMLHttpRequest();
xhrAsync.open("GET", "https://jsonplaceholder.typicode.com/posts/1", true);
xhrAsync.send();

// Synchronous (blocking)
var xhrSync = new XMLHttpRequest();
xhrSync.open("GET", "https://jsonplaceholder.typicode.com/posts/1", false);
xhrSync.send();
console.log("Synchronous response:", xhrSync.responseText);
</script>
```

7. Use setRequestHeader() to set a content type header in a POST request

```
<script>
var xhr = new XMLHttpRequest();
xhr.open("POST", "https://jsonplaceholder.typicode.com/posts", true);
xhr.setRequestHeader("Content-Type", "application/json");
xhr.send(JSON.stringify({ title: "foo", body: "bar" }));
</script>
```

8. Make a request to a broken URL and handle the error gracefully

```
<script>
var xhr = new XMLHttpRequest();
xhr.open("GET", "https://example.com/invalid", true);
xhr.onload = function() {
  if (xhr.status != 200) {
    alert("Error occurred: " + xhr.status);
  }
};
xhr.onerror = function() {
  alert("Request failed");
};
xhr.send();
</script>
```

9. Populate a second dropdown based on the selection of the first using AJAX

```
<html>
<body>
<select id="category" onchange="loadItems(this.value)">
  <option value="fruits">Fruits</option>
  <option value="vegetables">Vegetables</option>
</select>
<select id="items"></select>
<script>
function loadItems(cat) {
  var data = {
    fruits: ["Apple", "Banana"],
    vegetables: ["Carrot", "Broccoli"]
  };
  var list = data[cat];
  var options = list.map(item => `<option>${item}</option>`).join("");
  document.getElementById("items").innerHTML = options;
}
</script>
</body>
</html>
```

10. Auto-update a part of your webpage every 5 seconds using AJAX

```
<html>
<body>
<div id="autoUpdate"></div>
<script>
function updateContent() {
  var xhr = new XMLHttpRequest();
  xhr.open("GET", "https://jsonplaceholder.typicode.com/posts/1", true);
  xhr.onload = function() {
    document.getElementById("autoUpdate").innerHTML = xhr.responseText;
  };
  xhr.send();
}
setInterval(updateContent, 5000);
</script>
</body>
</html>
```