Experiment-10

Develop a JavaScript program with Ajax (with HTML/CSS) for:

a. Use ajax() method (without Jquery) to add the text content from the text file by sending ajax request.

b. Use ajax() method (with Jquery) to add the text content from the text file by sending ajax request.

c. Illustrate the use of getJSON() method in jQuery

d. Illustrate the use of parseJSON() method to display JSON values.

Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Ajax Demo Program</title>
   <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></scri
pt>
   <style>
```

```
body {
    font-family: Arial, sans-serif;
    line-height: 1.6;
    margin: 0;
    padding: 20px;
    background-color: #f4f4f4;
}
.container {
    max-width: 800px;
    margin: auto;
    background: white;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
}
h1 {
    color: #333;
}
h2 {
```

```
    color: #666;
}
button {
    background-color: #4CAF50;
    border: none;
    color: white;
    padding: 10px 20px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 4px 2px;
    cursor: pointer;
    border-radius: 4px;
}
pre {
    background-color: #f8f8f8;
    border: 1px solid #ddd;
    border-radius: 4px;
```

```
            padding: 10px;
            white-space: pre-wrap;
            word-wrap: break-word;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Ajax Demo Program</h1>

        <h2>a. Ajax-like operation without jQuery</h2>
        <button onclick="operationWithoutJQuery()">Perform Operation (without jQuery)</button>
        <pre id="result-a"></pre>

        <h2>b. Ajax-like operation with jQuery</h2>
        <button onclick="operationWithJQuery()">Perform Operation (with jQuery)</button>
        <pre id="result-b"></pre>
```

```
    <h2>c. jQuery-like getJSON() method</h2>
    <button onclick="getJSONOperation()">Get JSON</button>
    <pre id="result-c"></pre>

    <h2>d. jQuery parseJSON() method</h2>
    <button onclick="parseJSONExample()">Parse JSON</button>
    <pre id="result-d"></pre>
</div>

<script>
    // Simulated data
    const simulatedData = {
        text: "This is a sample text from a simulated server response.",
        json: {
            name: "John Doe",
            age: 30,
            city: "New York"
        }
    };
```

```
    // a. Ajax-like operation without jQuery
    function operationWithoutJQuery() {
        setTimeout(function() {
            document.getElementById("result-a").textContent =
simulatedData.text;
        }, 500);
    }

    // b. Ajax-like operation with jQuery
    function operationWithJQuery() {
        $.Deferred(function(deferred) {
            setTimeout(function() {
                deferred.resolve(simulatedData.text);
            }, 500);
        }).done(function(result) {
            $("#result-b").text(result);
        });
    }
```

```
    // c. jQuery-like getJSON() method
    function getJSONOperation() {
        $.Deferred(function(deferred) {
            setTimeout(function() {
                deferred.resolve(simulatedData.json);
            }, 500);
        }).done(function(result) {
            $("#result-c").text(JSON.stringify(result, null, 2));
        });
    }

    // d. jQuery parseJSON() method
    function parseJSONExample() {
        var jsonString = JSON.stringify(simulatedData.json);
        var jsonObject = $.parseJSON(jsonString);
        $("#result-d").text(JSON.stringify(jsonObject, null, 2));
    }
</script>
```

```
</body>

</html>
```

Explanation

- Purpose of the webpage:
- This is a demonstration page showing different methods of handling asynchronous operations and JSON data, both with and without jQuery. It's designed to help students understand the differences between these approaches.
- The page is divided into four main sections, each demonstrating a different technique:
- Ajax-like operation without jQuery
- Ajax-like operation with jQuery
- jQuery-like getJSON() method
- jQuery parseJSON() method
- Technologies used:
- HTML: For structuring the webpage
- CSS: For styling the page elements
- JavaScript: For performing operations without jQuery
- jQuery: A JavaScript library for easier Ajax and JSON handling

- Ajax-like operation without jQuery:
- This simulates an asynchronous operation using plain JavaScript.
- It uses setTimeout() to mimic a delay in receiving data from a server.
- The result is displayed using vanilla JavaScript DOM manipulation.

- Ajax-like operation with jQuery:
- This demonstrates how to use jQuery's Deferred object to handle asynchronous operations.
- It also uses setTimeout() to simulate a delay.
- The result is displayed using jQuery's DOM manipulation methods.
- jQuery-like getJSON() method:
- This simulates fetching JSON data from a server using jQuery's Deferred object.
- It demonstrates how to handle and display JSON data.
- jQuery parseJSON() method:
- This shows how to parse a JSON string into a JavaScript object using jQuery.
- It then displays the parsed object.
- The code uses a simulatedData object to mimic data that would typically come from a server.
- This allows the demonstration to work without an actual server connection.
- Asynchronous Operations:
- All the operations (except the parseJSON example) use setTimeout() to simulate the delay that would occur when fetching data from a real server.
- This helps students understand the concept of asynchronous programming.
- jQuery Usage:
- The code demonstrates how jQuery can simplify Ajax-like operations and JSON handling.
- It shows the difference between using vanilla JavaScript and jQuery for similar tasks.