| **JSON** | **XML** |
| --- | --- |
| It is java script object notation | It is extensible markup language. |
| It is based on JavaScript language. | It is derived from SGML. |
| It is a way of representing objects. | It is a markup language and uses tag structure to represent data items. |
| It does not provides any support for namespaces. | It supports namespaces. |
| It supports array. | It does not supports arrays. |
| Its files are very easy to read as compared to XML. | Its documents are comparatively difficult to read and interpret. |
| It doesn’t use end tag. | It has start and end tags. |
| It is less secured. | It is more secured than JSON. |
| It doesn’t supports comments. | It supports comments. |
| Its supports only UTF-8 encoding. | It supports various encoding. |

**1 .Difference between JSON and XML:**

**2 .Create 3 XML and JSON files for department, year, student:**

Xml files:

<departments>

    <department>

        <name>Computer Science</name>

    </department>

    <department>

        <name>Electrical Engineering</name>

    </department>

</departments>

<years>

<year>

<sec>A</sec>

</year>

<year>

<sec>B</sec>

</year>

</years>

<students>

<student>

<id>1</id>

</student>

<student>

<id>2</id>

</student>

</students>

Json files:

{

  "departments": [

    {

      "id": 1,

      "name": "Computer Science",

      "location": "Building A"

    },

    {

      "id": 2,

      "name": "Electrical Engineering",

      "location": "Building B"

    }

  ]

}

{

  "years": [

    {

      “sec”: A,

    },

    {

      "sec”: B,

    },

  ]

}

{

  "students": [

    {

      “id”: 1,

    },

    {

      "id": 2,

    }

  ]

}

**3.create a file with department as root, year as subroot and student as an element:**

< Department1>

  <Year1>

    <Student>1</student>

    <Student>2</student>

</year1>

  <Year2>

    <Student>3</student>

    <Student4></student>

</year>

<Department1>

**4.Difference between Authorization and Authentication:**

| **Authentication** | **Authorization** |
| --- | --- |
| In the [authentication](https://www.geeksforgeeks.org/authentication-in-computer-network/) process, the identity of users are checked for providing the access to the system. | While in [authorization](https://www.geeksforgeeks.org/what-is-aaa-authentication-authorization-and-accounting/) process, a the person’s or user’s authorities are checked for accessing the resources. |
| In the authentication process, users or persons are verified. | While in this process, users or persons are validated. |
| It is done before the authorization process. | While this process is done after the authentication process. |
| It needs usually the user’s login details. | While it needs the user’s privilege or security levels. |
| Authentication determines whether the person is user or not. | While it determines **What permission does the user have?** |
| Generally, transmit information through an ID Token. | Generally, transmit information through an Access Token. |
| The OpenID Connect (OIDC) protocol is an authentication protocol that is generally in charge of user authentication process. | The OAuth 2.0 protocol governs the overall system of user authorization process. |

**5.create a login screen:**

<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Login</title>  
    <style>  
        body {  
            font-family: Arial, sans-serif;  
            background-color: #f0f0f0;  
            display: flex;  
            justify-content: center;  
            align-items: center;  
            height: 100vh;  
        }  
        .login-container {  
            background-color: white;  
            padding: 20px;  
            border-radius: 5px;  
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
        }  
        .login-container h2 {  
            margin-bottom: 20px;  
        }  
        .login-container input[type="text"], .login-container input[type="password"] {  
            width: 100%;  
            padding: 10px;  
            margin: 10px 0;  
            border: 1px solid #ccc;  
            border-radius: 5px;  
        }  
        .login-container input[type="submit"] {  
            width: 100%;  
            padding: 10px;  
            background-color: #5cb85c;  
            color: white;  
            border: none;  
            border-radius: 5px;  
            cursor: pointer;  
        }  
    </style>  
</head>  
<body>  
    <div class="login-container">  
        <h2>Login</h2>  
        <form action="/login" method="POST">  
            <input type="text" name="username" placeholder="Username" required>  
            <input type="password" name="password" placeholder="Password" required>  
            <input type="submit" value="Login">  
        </form>  
    </div>  
</body>  
</html>

**6.create a user creation screen by using all elements in it(like list, radio button, drop down, checkbox)**

**HTML and CSS for a User Creation Screen:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Create User</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #f0f0f0;

            display: flex;

            justify-content: center;

            align-items: center;

            height: 100vh;

        }

        .create-user-container {

            background-color: white;

            padding: 20px;

            border-radius: 5px;

            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

        }

        .create-user-container h2 {

            margin-bottom: 20px;

        }

        .create-user-container label {

            display: block;

            margin: 10px 0 5px;

        }

        .create-user-container input[type="text"], .create-user-container input[type="password"], .create-user-container select {

            width: 100%;

            padding: 10px;

            margin: 10px 0;

            border: 1px solid #ccc;

            border-radius: 5px;

        }

        .create-user-container input[type="radio"], .create-user-container input[type="checkbox"] {

            margin-right: 10px;

        }

        .create-user-container input[type="submit"] {

            width: 100%;

            padding: 10px;

            background-color: #5cb85c;

            color: white;

            border: none;

            border-radius: 5px;

            cursor: pointer;

        }

    </style>

</head>

<body>

    <div class="create-user-container">

        <h2>Create User</h2>

        <form action="/create-user" method="POST">

            <label for="username">Username</label>

            <input type="text" id="username" name="username" required>

            <label for="password">Password</label>

            <input type="password" id="password" name="password" required>

            <label>Gender</label>

            <input type="radio" id="male" name="gender" value="male">

            <label for="male">Male</label>

            <input type="radio" id="female" name="gender" value="female">

            <label for="female">Female</label>

            <label for="department">Department</label>

            <select id="department" name="department">

                <option value="cs">Computer Science</option>

                <option value="math">Mathematics</option>

                <option value="phy">Physics</option>

            </select>

            <label for="year">Year</label>

            <select id="year" name="year">

                <option value="2024">2024</option>

                <option value="2025">2025</option>

                <option value="2026">2026</option>

            </select>

            <label>Hobbies</label>

            <input type="checkbox" id="sports" name="hobbies" value="sports">

            <label for="sports">Sports</label>

            <input type="checkbox" id="reading" name="hobbies" value="reading">

            <label for="reading">Reading</label>

            <input type="submit" value="Create User">

        </form>

    </div>

</body>

</html>

**7.list all users, update user and delete user:**

**HTML and JavaScript for User Management:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>User Management</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #f0f0f0;

            display: flex;

            justify-content: center;

            align-items: center;

            height: 100vh;

        }

        .user-management-container {

            background-color: white;

            padding: 20px;

            border-radius: 5px;

            box-shadow: 0 0

**8.create a HTML page with google map:**

<!DOCTYPE html>

<html lang="en">

<head>

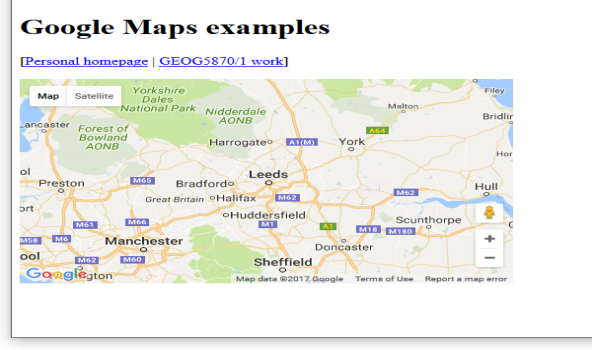
    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Google Map</title>

    <style>

        #map {

****            height: 400px;

            width: 100%;

        }

    </style>

</head>

<body>

    <h2>My Location</h2>

    <div id="map"></div>

    <script>

        function initMap() {

            // Location coordinates

            var myLatLng = {lat: 37.7749, lng: -122.4194};

            // Create a map object and specify the DOM element for display.

            var map = new google.maps.Map(document.getElementById('map'), {

                center: myLatLng,

                zoom: 12 // Zoom level (0 = earth view, 20 = very close)

            });

            // Create a marker and set its position.

            var marker = new google.maps.Marker({

                map: map,

                position: myLatLng,

                title: 'My Location'

            });

        }

    </script>

    <!-- Load the Google Maps JavaScript API with the provided placeholder API key -->

    <script async defer

    src="<https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap>">

    </script>

</body>

</html>

**9.Create a HTML page with video file:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Video Player</title>

</head>

<body>

    <h2>My Video</h2>

    <!-- Replace the src attribute with your YouTube video's embed URL -->

    <iframe width="600" height="337" src="<https://www.youtube.com/watch?v=vQXvyV0zIP4>" frameborder="0" allowfullscreen></iframe>

</body>

</html>

https://ssl.gstatic.com/ui/v1/icons/mail/images/cleardot.gif

Attachments area

[Preview YouTube video The first-ever BMW M5 CS.](https://www.youtube.com/watch?v=vQXvyV0zIP4" \t "_blank)



**10.Create a HTML page with Audio file**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Audio Player</title>

</head>

<body>

    <h2>My Audio</h2>

    <audio controls>

        <!-- Replace the src attribute with the URL of your audio file -->

        <source src="**your\_audio\_file.mp3**" type="audio/mpeg">

        Your browser does not support the audio element.

    </audio>

</body>

</html>

**11.Create a HTML page to upload a file**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>File Upload</title>

</head>

<body>

    <h2>Upload a File</h2>

    <form action="#" method="post" enctype="multipart/form-data" id="uploadForm">

        <input type="file" name="fileToUpload" id="fileToUpload">

        <button type="submit">Upload</button>

    </form>

    <div id="uploadResult"></div>

    <script>

        document.getElementById('uploadForm').addEventListener('submit', function(event) {

            event.preventDefault();

            var fileInput = document.getElementById('fileToUpload');

            var file = fileInput.files[0];

            var formData = new FormData();

            formData.append('file', file);

            // You can use AJAX to send the file to the server

            // Here's just a simple example showing the file name

            var uploadResult = document.getElementById('uploadResult');

            uploadResult.innerHTML = 'File uploaded: ' + [file.name](http://file.name/);

        });

    </script>

</body>

</html>