**1)JSON vs XML: Difference Between JSON and XML**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **JSON** | **XML** |
| **Definition** | It is a file format that stores and transmits the data objects into human-readable text. | It is a markup language designed to store and distribute data across the web  and various APIs. |
| **Full-Form** | JavaScript Object Notation | extensible Markup Language. |
| **Language** | Derived from javascript. | Derived from SGML. |
| **Extension** | .json | .xml |
| **Data Types** | Supports string, number, boolean, and arrays. | Data is in a string format. |
| **Tags** | No tags | Data is represented in tags. |
| **Array** | Use an array to represent the data. | Doesn’t use arrays. |
| **Comments** | Don’t support | Support |
| **NameSpace** | Don’t support | Support |
| **Orientation** | Data Oriented | Document Oriented |
| **Security** | Less Secured | More secured than JSON |

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

XML Files

1. departments.xml

<departments>

<department>

<id>1</id>

<name>Computer Science</name>

<head>Rameshwar sir</head>

</department>

<department>

<id>2</id>

<name>Electrical Engineering</name>

<head>Ramesh sir</head>

</department>

</departments>

2. years.xml

<years>

<year>

<id>1</id>

<name>First Year</name>

<section>A</section>

</year>

<year>

<id>2</id>

<name>Second Year</name>

<section>B</ section>

</year>

<year>

<id>3</id>

<name>Third Year</name>

< section >C</ section >

</year>

</years>

3. students.xml

<students>

<student>

<id>1</id>

<name>Nitheesh</name>

</student>

<student>

<id>2</id>

<name>Divyanka</name>

</student>

</students>

Json files:

1.departments.json

{

"departments": [

{

"id": 1,

"name": "Computer Science",

"head": " Rameshwar sir "

},

{

"id": 2,

"name": "Electrical Engineering",

"head": " Ramesh sir "

},

]

}

2. years.json

{

"years": [

{

"id": 1,

"name": "First Year",

"section": "A",

},

{

"id": 2,

"name": "Second Year",

"section": "B",

},

{

"id": 3,

"name": "Third Year",

"section": "C",

}

]

}

3. students.json

{

"students": [

{

"id": 1,

"name": "Niteesh",

},

{

"id": 2,

"name": "Divyanka",

},

]

}

**3)Create a file with depertment as root,year as subroot and student as an element**

{

"Department1": {

"Year1": [

"Student1",

"Student2",

],

"Year2": [

"Student4",

"Student5",

]

},

"Department2": {

"Year1": [

"Student7",

"Student8",

],

"Year2": [

"Student10",

"Student11",

]

}

}

**4. Difference between Authorization and Authentication**

**Authentication:**

* Definition: The process of verifying the identity of a user or system.
* Purpose: To ensure that the entity is who it claims to be.
* Example: Logging in with a username and password.

**Authorization:**

* Definition: The process of granting or denying access to resources.
* Purpose: To determine what an authenticated entity is allowed to do.
* Example: Access control permissions for files and directories.

### **5. Create a Login Screen**

**HTML and CSS for 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>

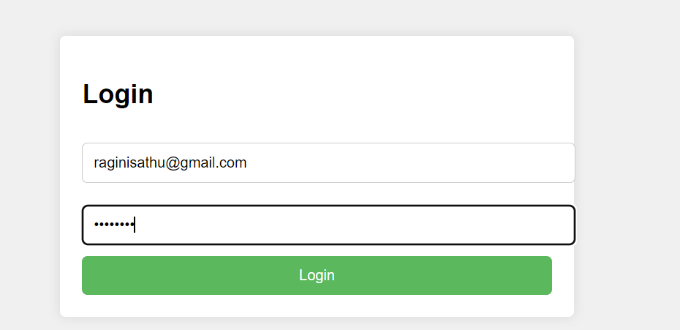
<input type="submit" value="Login">

</form>

</div>

</body>

</html>



### **6. Create a User Creation Screen using all elements**

**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>

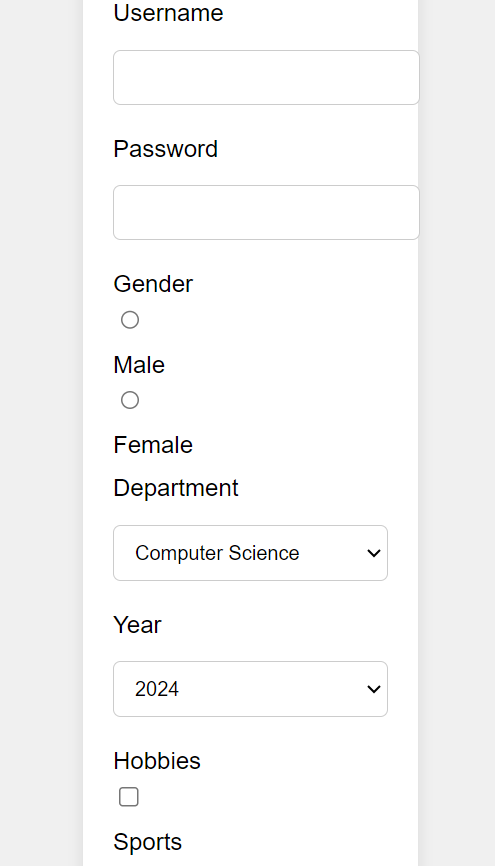
<input type="submit" value="Create User">

</form>

</div>

</body>

</html>



### **7. List all Users, Update user, and Delete user (Popup for confirmation)**

**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;

}

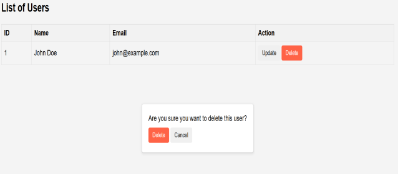
.user-management-container {

background-color: white;

padding: 20px;

border-radius: 5px;

box-shadow: 0 0



**8) 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 -->

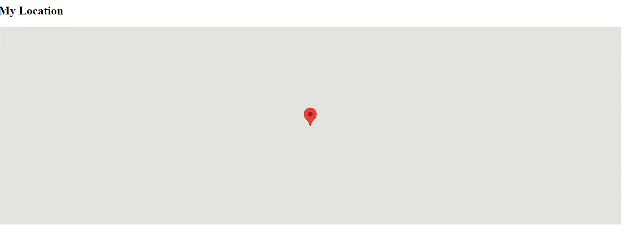
<script async defer

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>

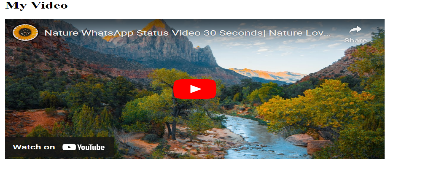
<h2>My Video</h2>

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

<iframe width="600" height="337" src="https://www.youtube.com/watch?v=Y6a1hiDuMOQ " frameborder="0" allowfullscreen></iframe>

</body>

</html>



**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 -->

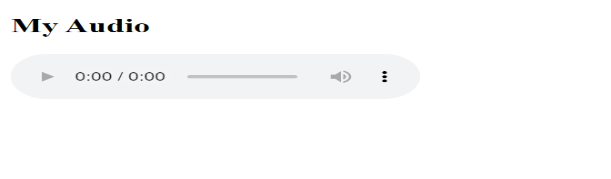
<source src="**your\_audio\_file.mp3**" type="audio/mpeg">

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;

});

</script>

</body>

</html>

