

1)Difference between JSON and XML

JSON (JavaScript Object Notation) and XML (eXtensibleMarkup Language) are both popular data interchange formats, but they have some key differences:

Syntax: JSON is simpler and more lightweight than XML. JSON uses key-value pairs and arrays to represent data, while XML uses tags and attributes within a hierarchical structure.

Readability: JSON is generally easier for humans to read and write due to its concise syntax, while XML can be more verbose and complex, especially for large datasets.

Data Types: JSON supports a limited set of data types, including strings, numbers, booleans, arrays, and objects. XML is more flexible in this regard and can represent a wider range of data types and structures.

Parsing: JSON can be parsed more quickly and efficiently by JavaScript engines, making it a preferred format for data exchange in web applications. XML parsing requires more resources and can be slower.

Usage: JSON is commonly used for web APIs, configuration files, and data exchange between web servers and clients. XML is often used in legacy systems, document storage, and data interchange between different platforms and technologies.

4)Difference between Authorization and Authentication

Authentication and authorization are two related but distinct concepts in the context of security and access control:

Authentication:

- Verifies the identity of a user, device, or system
- Ensures that someone or something is who they claim to be
- Typically involves credentials like passwords, tokens, or biometrics
- Grants access to a system, network, or application

Authorization:

- Determines what actions a authenticated user can perform
- Controls access to specific resources, data, or functionality
- Defines what a user can do, based on their role, privileges, or permissions
- Ensures that a user only has access to what they are allowed to access

To illustrate the difference:

- Authentication is like showing your ID to get into a building (verifying who you are).
- Authorization is like having a keycard that grants access to specific rooms or areas within the building (controlling what you can do).

In summary, authentication verifies identity, while authorization determines the level of access and control once authenticated.

2) Create 3 xml files for dept, year, student ### XML Files:

1. CSE(c).xml:

xml

```
<Departments>
  <Department>
    <ID>1A</ID>
    <name>Computer Science</Name>
    <Head>Dr.K rameshwariah</Head>
  </Department>
  <Department>
    <ID>2A</ID>
    <Name>Electrical Engineering</Name>
    <Head>Dr.A.Nikhitha</Head>
  </Department>
</Departments>
```

2. Year.xml:

xml

```
<Years>
  <Year>
    <ID>1st</ID>
    <Name>humanities and sciences</Name>
  </Year>
  <Year>
    <ID>2nd</ID>
    <Name>2nd years</Name>
  </Year>
  <Year>
    <ID>3</ID>
    <Name>3rd years</Name>
  </Year>
  <Year>
    <ID>4</ID>
    <Name>final years</Name>
  </Year>
</Years>
```

3. Student.xml:

xml

```
<Students>
  <Student>
    <ID>1256</ID>
    <Name>pranitha</>
    <DepartmentID>1A</DepartmentID>
    <YearID>3</YearID>
    <GPA>9.8</GPA>
  </Student>
```

```
<Student>
  <ID>6629</ID>
  <Name>nikhitha</Name>
  <DepartmentID>2A</DepartmentID>
  <YearID>3</YearID>
  <GPA>.8</GPA>
</Student>
</Students>
```

JSON Files:

1. Department.json:

json

```
{
  "Departments": [
    {
      "ID": "1A",
      "Name": "Computer Science",
      "Head": "Dr.k.rameshwariah"
    },
    {
      "ID": "2A",
      "Name": "Electrical Engineering",
      "Head": "Dr.k.Nikhitha"
    }
  ]
}
```

2. Year.json:

json

```
{
  "Years": [
    {
      "ID": 1,
      "Name": "humanities and sciences"
    },
    {
      "ID": 2,
      "Name": "2nd years"
    },
    {
      "ID": 3,
      "Name": "3rd years"
    },
    {
      "ID": 4,
      "Name": "final years"
    }
  ]
}
```

3. Student.json:

json

```
{
  "Students": [
    {
      "ID": 1256,
```

```

    "Name": "pranitha",
    "DepartmentID": "1A"
    "YearID": 3,
    "GPA": 8.34
  },
  {
    "ID": 6621,
    "Name": "Sujatha",
    "DepartmentID": "2A",
    "YearID": 2,
    "GPA": 7.8
  }
]
}

```

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

```

<Departments>
  <Department>
    <ID>1A</ID>
    <Name>Computer Science</Name>
    <Head>Dr. K rameshwariah</Head>
    <Year>
      <ID>3</ID>
      <Name>3rd years</Name>
      <Students>
        <Student>
          <ID>1256</ID>
          <Name>pranitha</Name>

```

```
        <GPA>8.34</GPA>
    </Student>
    <Student>
        <ID>6629</ID>
        <Name>keerthana</Name>
        <GPA>6.8</GPA>
    </Student>
</Students>
</Year>
</Department>
<Department>
    <ID>2A</ID>
    <Name>Electrical Engineering</Name>
    <Head>Dr. A.Nikhitha</Head>
    <Year>
        <ID>3</ID>
        <Name>3rd years</Name>
        <Students>
            <Student>
                <ID>6629</ID>
                <Name>lasya</Name>
                <GPA>.8</GPA>
            </Student>
        </Students>
    </Year>
</Department>
</Departments>
```


6.USER CREATION SCREEN USING(LIST,RADIO BUTTON,DROPDOWN,CHECK BOX)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>User Creation Screen</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      margin: 0;
      padding: 20px;
    }

    form {
      background-color: #ffffff;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }

    input[type="text"],
    input[type="email"],
    select,
    textarea {
      width: 100%;
```

```
padding: 10px;
margin: 5px 0;
box-sizing: border-box;
border: 1px solid #ccc;
border-radius: 5px;
}
```

```
input[type="radio"],
input[type="checkbox"] {
    margin-right: 5px;
}
```

```
input[type="submit"] {
    background-color: #4CAF50;
    color: white;
    cursor: pointer;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
}
```

```
input[type="submit"]:hover {
    background-color: #45a049;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<form action="create_user.php" method="post">
```

```
<h2>User Creation</h2>
```

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

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

<label for="email">Email:</label>

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

<label for="gender">Gender:</label>

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

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

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

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

<label for="country">Country:</label>

<select id="country" name="country">

<option value="usa">USA</option>

<option value="canada">Canada</option>

<option value="uk">UK</option>

<option value="australia">Australia</option>

</select>

<label for="interests">Interests:</label>

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

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

<input type="checkbox" id="music" name="interests" value="music">

<label for="music">Music</label>

<input type="checkbox" id="movies" name="interests" value="movies">

<label for="movies">Movies</label>

<label for="bio">Bio:</label>

```
<textarea id="bio" name="bio" rows="4"></textarea>

<input type="submit" value="Create User">
</form>
</body>
</html>
```

8) Create a HTML page with Google Map

////we have to add our own API key from googleapi keys

```
<!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=i
nitMap">
</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/embed/-oOoTluloL8M" frameborder="0"
allowfullscreen></iframe>
</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>
```

10)Create a HTML page with Audio file

NOTE: replace your audio file with any audio file in your pc

```
<!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>

```

7)

```

<!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: #f4f4f4;
      margin: 0;
      padding: 20px;
    }

    table {
      width: 100%;
      border-collapse: collapse;

```



```
margin-bottom: 20px;  
}
```

```
table, th, td {  
  border: 1px solid #ddd;  
  padding: 8px;  
  text-align: left;  
}
```

```
th {  
  background-color: #f2f2f2;  
}
```

```
button {  
  padding: 8px 12px;  
  border: none;  
  border-radius: 4px;  
  cursor: pointer;  
}
```

```
.delete-btn {  
  background-color: #ff6347;  
  color: white;  
}
```

```
.delete-btn:hover {  
  background-color: #d43f1d;  
}
```

```
.confirm-delete {
```

```
display: none;
position: absolute;
top: 50%;
left: 50%;
transform: translate(-50%, -50%);
background-color: white;
padding: 20px;
border: 1px solid #ccc;
border-radius: 5px;
box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
z-index: 999;
}
```

```
.confirm-delete p {
  margin: 0 0 10px;
}
```

```
.btn-container {
  text-align: center;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>List of Users</h2>
```

```
<table>
```

```
<thead>
```

```
<tr>
```

```
<th>ID</th>
```

```
<th>Name</th>
```

```
<th>Email</th>
```

```

        <th>Action</th>
    </tr>
</thead>
<tbody>
    <tr>
        <td>1</td>
        <td>John Doe</td>
        <td>john@example.com</td>
        <td>
            <button onclick="updateUser(1)">Update</button>
            <button onclick="confirmDelete(1)"
class="delete-btn">Delete</button>
        </td>
    </tr>
    <!-- More rows for other users -->
</tbody>
</table>

<div id="confirm-delete" class="confirm-delete">
    <p>Are you sure you want to delete this user?</p>
    <button onclick="deleteUser()" id="delete-btn"
class="delete-btn">Delete</button>
    <button onclick="cancelDelete()" id="cancel-btn">Cancel</button>
</div>

<script>
    let userIdToDelete;

    function confirmDelete(userId) {
        userIdToDelete = userId;

```

```

        document.getElementById('confirm-delete').style.display = 'block';
    }

    function cancelDelete() {
        document.getElementById('confirm-delete').style.display = 'none';
    }

    function deleteUser() {
        // Send request to server to delete user with userIdToDelete
        console.log('Deleting user with ID:', userIdToDelete);
        // After deletion, hide confirmation dialog
        document.getElementById('confirm-delete').style.display = 'none';
    }

    function updateUser(userId) {
        // Redirect or show update form for the selected user
        console.log('Updating user with ID:', userId);
    }
</script>
</body>
</HTML>

```

5.Login page using html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Page</title>
    <style>

```

```
body {  
  font-family: Arial, sans-serif;  
  background-color: #f4f4f4;  
  margin: 0;  
  padding: 0;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 100vh;  
}
```

```
form {  
  background-color: #ffffff;  
  padding: 20px;  
  border-radius: 8px;  
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
}
```

```
input[type="text"],  
input[type="password"],  
input[type="submit"] {  
  width: 100%;  
  padding: 10px;  
  margin: 5px 0;  
  box-sizing: border-box;  
  border: 1px solid #ccc;  
  border-radius: 5px;  
}
```

```
input[type="submit"] {
```

```
background-color: #4CAF50;
color: white;
cursor: pointer;
}
```

```
input[type="submit"]:hover {
background-color: #45a049;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<form action="login.php" method="post">
```

```
<h2>Login</h2>
```

```
<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>
```

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

```
</form>
```

```
</body>
```

```
</html>
```