1. Difference between JSON and XML

Feature	JSON (JavaScript Object Notation)	XML (eXtensible Markup Language)
Syntax	Lightweight, easy-to-read, key-value pairs	Verbose, uses nested elements and attributes
Data Representation	Objects and arrays	Hierarchical elements
Data Types	Strings, numbers, booleans, arrays, objects, null	Primarily text; data types are application-specific
Human Readability	More human-readable	Less human-readable
Schema Support	JSON Schema (less mature)	XML Schema (XSD), DTD, RelaxNG
Namespaces	Not supported	Supported
Parsing Complexity	Simpler and faster	More complex and slower
Transformation	Limited	Extensive support (XSLT)
Usage	Web development, APIs, configuration files	Legacy systems, document-centric applications
Popularity	More popular in modern web applications	Still used in enterprise and legacy systems
Integration	Easily integrated with web technologies	Often used with document standards
Validation	Less formal, typically application-specific	Extensive validation options

EXAMPLES of XML and JASON:

JASON	XML
{“name”:”Ruchitha”, ”age”:20, “isStuding”:True, “courses”:["Java”,”Science”],	<person> <name>Ruchitha</name> <age>20</age> <isStuding>True</isStuding>

“address”:{
“street”:”24 Main Street”,
“city”:”seoul”
}
}
	<courses>
	<course>Java</course>
	<course>C++</course>
	</courses>
	<address>
	<street>24 Main street</street>
	<city>seoul</city>
	</address>
	</person>

2.create three XML and Jason files for department, student, years.

XML students

```
<?xml version="1.0" encoding="UTF-8"?>

<students>

  <student>

    <id>1</id>

    <name>abc</name>

    <age>20</age>

  </student>

  <student>

    <id>2</id>

    <name>pqr</name>

    <age>22</age>

  </student>

  <student>

    <id>3</id>
    <name>ijk</name>

    <age>21</age>

  </student>

  <student>

    <id>4</id>

    <name>xyz</name>

    <age>23</age>

  </student>

</students>
```

XML department

```
<?xml version="1.0" encoding="UTF-8"?>

<department>

<count>6</count>

<dept>

<name>Computer Science and Engineering</name>

<code>CSE</code>

<head>DR.k.Rameshwaraiah</head>

</dept>

<dept>

<name>Electronics and Communication Engineering</name>

<code>ECE</code>

<head>DR.S.Ravi chand</head>

</dept>

<dept>

<name>Humanities and sciences</name>

<code>H& S</code>
```

```
<head>DR.E.Chandra shekar</head>
</dept>
<dept>
<name>Civil Engineering</name>
<code>CE</code>
<head>DR.Y.Srinivas</head>
</dept>
<dept>
<name>Mechanical Engineering</name>
<code>ME</code>
<head>DR.G.Janardhana Raju</head>
</dept>
<dept>
<name>Electrical and Electonics Engineering</name>
<code>EEE</code>
<head>DR.P.Ramesh</head>
</dept>
</department>
```

XML years

```
<?xml version="1.0" encoding="UTF-8"?>
<collegeYears>
  <year>
    <id>1</id>
    <sections>9</sections>
    <startDate>2023-08-15</startDate>
    <endDate>2024-05-20</endDate>
  </year>
  <year>
    <id>2</id>
     <sections>8</sections>
    <startDate>2024-08-15</startDate>
    <endDate>2025-05-20</endDate>
  </year>
  <year>
    <id>3</id>
     <sections>6</sections>
    <startDate>2025-08-15</startDate>
    <endDate>2026-05-20</endDate>
  </year>
  <year>
    <id>4</id>
     <sections>4</sections>
    <startDate>2026-08-15</startDate>
    <endDate>2027-05-20</endDate>
  </year>
</collegeYears>
```

JSON Department

```
{
  "Departments": [
    {
      "Name": "Computer Science and Engineering",
"Code": "CSE",
      "Head": "Dr.k.Rameshwariah"

    },

    {

      "Name": "Electrical and Electonics Engineering",

"Code": "EEE",

      "Head": "DR.P.Ramesh"

    }

{

"name":"Electronics and Communication Engineering",

"code":"ECE",

"head":"DR.S.Ravi chand"

}

{

"name":"Humanities and sciences",

"code":"H& S",

"head":"DR.E.Chandra shekar"

}

]

}
```

JSON Years

```
{

  "Years": [

    {

      "ID": 1,

      "Sections":9

    },
    {
      "ID": 2,
      "Sections":8
    },
    {
      "ID": 3,
      "Sections":6
    },
    {
      "ID": 4,
      "Sections":4
    }
  ]
}
```

JSON Students

```
{
  "Students": [
    {
      "ID": 6615,
      "Name": "abd",
      "Age":20,
      "YearID": 3
    },
    {
      "ID": 6621,
      "Name": "xyz",
      "Age":20,
      "YearID": 2
    }
  ]
}
```

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

```
<?xml version="1.0" encoding="UTF-8"?>
<department value="IT">
  <year value="2">
    <student>
      <name>abc</name>
      <id>1</id>
      <email>abc@gmail.com</email>
    </student>
    <student>
      <name>ijk</name>
      <id>2</id>
      <email>ijk@gmail.com</email>
    </student>
  </year>
  <year value="3">
    <student>
      <name>abcd</name>
      <id>1</id>
      <email>abcd@gmail.com</email>
    </student>
    <student>
      <name>ijkl</name>
      <id>1</id>
      <email>ijkl@gmail.com</email>
    </student>
  </year>
<year value="4">
    <student>
      <name>abcde</name>
      <id>1</id>
      <email>abcde@gmail.com</email>
    </student>
    <student>
      <name>ijklm</name>
      <id>1</id>
      <email>ijklm@gmail.com</email>
    </student>
  </year>
</department>
```

4. Difference between Authorization and Authentication

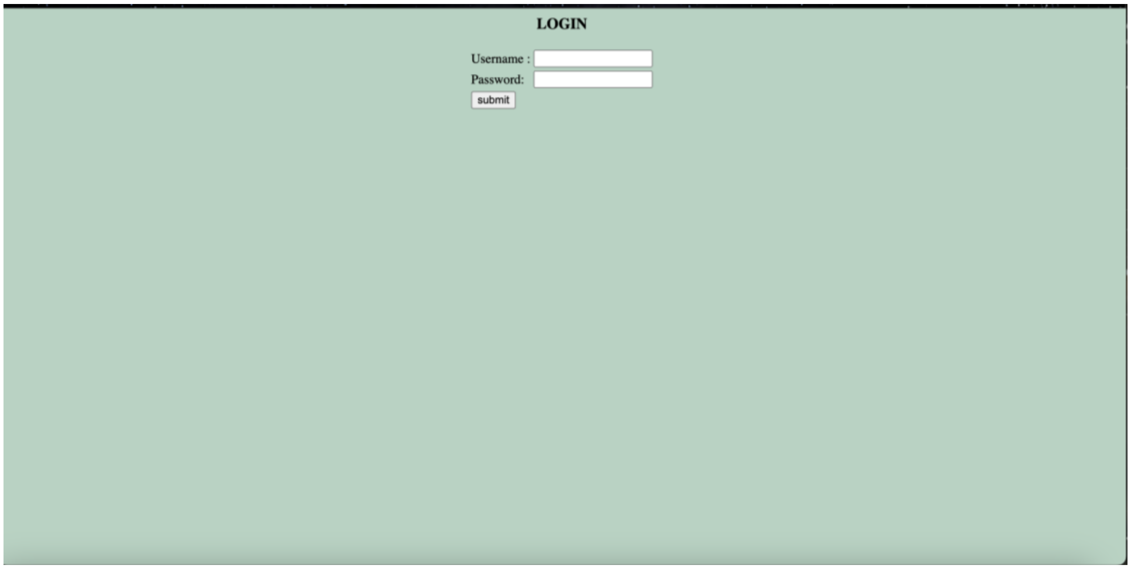
Aspect	Authentication	Authorization
Definition	The process of verifying the identity of a user.	The process of determining what resources a user can access.
Purpose	To confirm the user is who they claim to be.	To grant or deny permissions to resources.

Occurs When	At the beginning of a system interaction.	After authentication, once the user's identity is verified.
Based On	User credentials (e.g., username and password).	User's roles and permissions.
Process	Involves validating credentials against a database.	Involves checking access policies and permissions.
Example	Logging in with a username and password.	Accessing specific data or services based on user roles.
Data Used	User-provided credentials (e.g., password, biometrics).	Access control lists, role-based access controls.
Responsibility	Usually managed by authentication systems like login forms.	Managed by authorization systems like access control mechanisms.
Relation to Identity	Confirms the user's identity.	Determines what the authenticated user is allowed to do.
Granularity	Typically binary (authenticated or not).	Can be fine-grained (specific permissions for specific actions).
Visibility to User	Directly visible (user actively participates).	Often transparent to the user (happens after login).
Common Protocols/Standards	OAuth, OpenID Connect, SAML	Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC)

5)Create a Login Screen.

```
<html>
<head>
<title> LOGIN</title>
</head>
<body>
<body style="background-color:B2D3C2">
<form>
<h3><b><center> LOGIN </center> </b></h3>
<table border="0" align="center">
<tr>
<td>Username : </td>
<td><input type ="text"></td>
</tr>
```

```
<tr>
<td>Password: </td>
<td><input type ="password"></td>
</tr>
<tr>
<td><input type="button" value="submit"></td>
</tr>
</form>
</body>
</html>
```



LOGIN

Username :

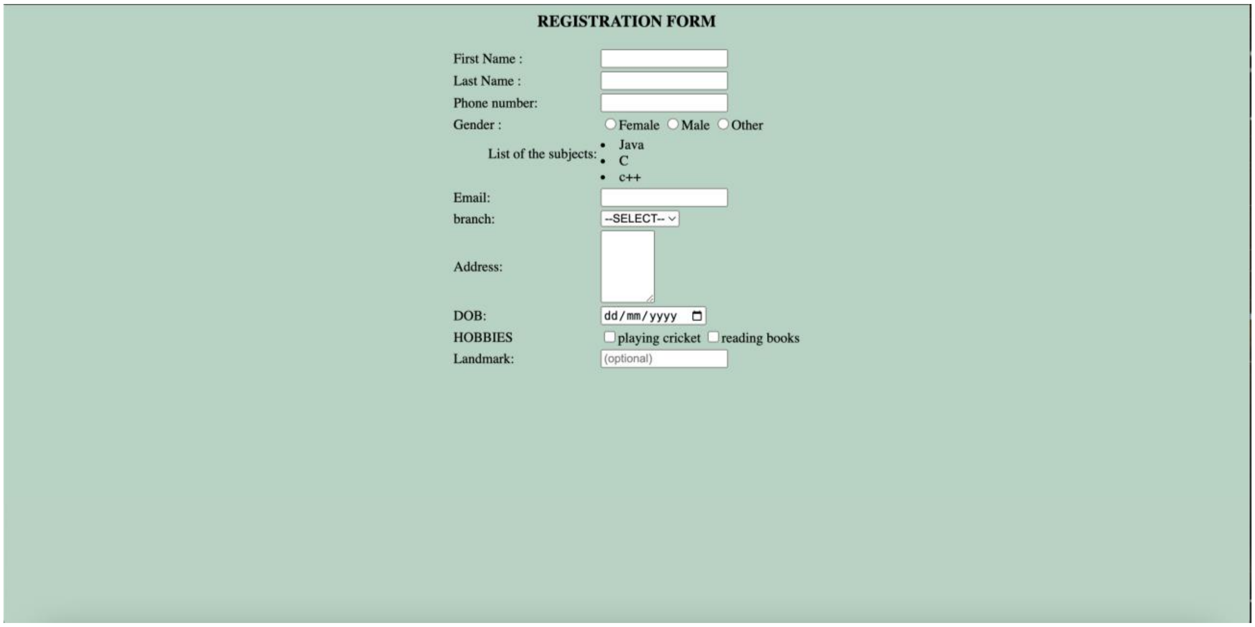
Password:

6. Create a User Creation Screen by using all elements in it(like List, Radio button, Drop down, CheckBox)

```
<html>
<head>
<title> Registration </title>
</head>
<body>
<body style="background-color:B2D3C2">
<form>
<h3><b><center> REGISTRATION FORM </center> </b></h3>
<table border="0" align="center">
<tr>
```

```
<td>First Name : </td>
<td><input type ="text"></td>
</tr>
<tr>
<td>Last Name : </td>
<td><input type ="text"></td>
</tr>
<tr>
<td>Phone number: </td>
<td><input type ="text"></td>
</tr>
<tr>
<td>Gender : </td>
<td><input type ="radio" name="g">Female
   <input type ="radio" name="g">Male
   <input type ="radio" name="g">Other</td>
</tr>
<tr>
<td>
<ul>List of the subjects:</td>
<td><li>Java</li>
<li>C</li>
<li>c++</li></td>
</ul>
</tr>
<tr>
<td>Email: </td>
<td><input type ="text"></td>
</tr>
<tr>
<td>branch: </td>
<td><select name="branch">
<option>--SELECT--</option>
<option>CSE</option>
<option>EEE</option>
<option>ECE</option>
</td>
</tr>
<tr>
<td>Address: </td>
<td> <textarea rows="5" cols="5"> </textarea> </td>
</tr>
<tr>
    <td>DOB: </td>
<td><input type ="date"></td>
```

```
</tr>
<tr>
<td>HOBBIES</td>
<td><input type="checkbox" name="rcb">playing cricket</input>
<input type="checkbox" name="rcb">reading books</input></td>
</tr>
<tr>
<td>Landmark: </td>
<td><input type="text" name="input" placeholder="(optional)"></td>
</tr>
</form>
</body>
</html>
```



REGISTRATION FORM

First Name :

Last Name :

Phone number:

Gender : ○Female ○Male ○Other

List of the subjects:
- Java
- C
- c++

Email:

branch: --SELECT--

Address:

DOB: dd/mm/yyyy

HOBBIES □playing cricket □reading books

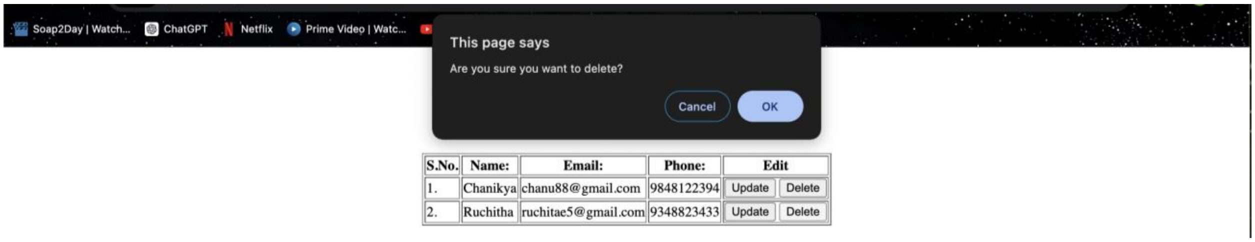
Landmark: (optional)

7.List all Users , Update user and Delete user (Popup for confirmation eg :Are you sure do you want to delete)

```
<html>

<head>
```

```
<script language="Javascript">
function de()
{
confirm("Are you sure you want to delete?");
}
</script>
</head>
<body>
<h1><b><center>List of users</center></b></h1>
<form>
<table border="1" align ="center"><br>
<br>
<br>
<tr>
<th>S.No.</th>
<th>Name:</th>
<th>Email:</th>
<th>Phone:</th>
<th>Edit</th>
</tr>
<tr>
<td> 1.</td>
<td> Chanikya </td>
<td> chanu88@gmail.com </td>
<td>9848122394</td>
<td><input type="Button" value="Update">
```

```
<input type ="Button" value="Delete" onclick="de()"></td>

</tr>

<tr>

<td> 2.</td>

<td> Ruchitha  </td>

<td> ruchitae5@gmail.com </td>

<td>9348823433</td>

<td><input type="Button" value="Update">

<input type ="Button" value="Delete" onclick="de()"></td>

</tr>

</body>

</html>
```



This page says

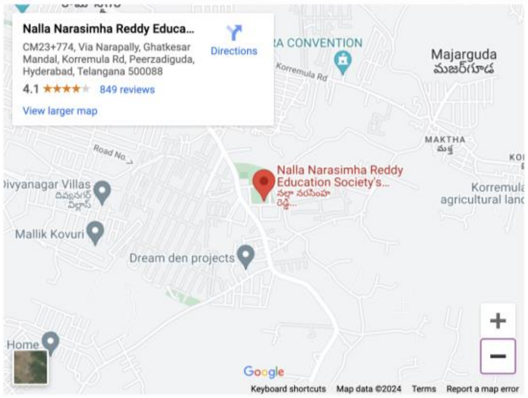
Are you sure you want to delete?

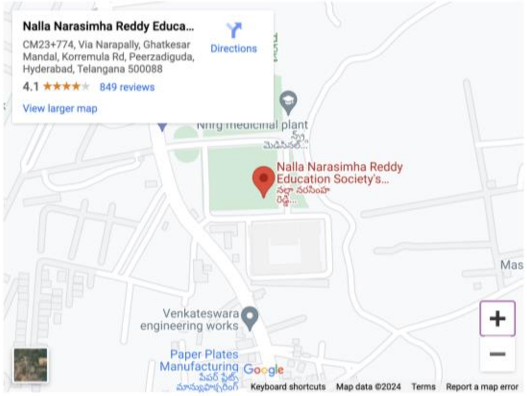
S.No.	Name:	Email:	Phone:	Edit	
1.	Chanikya	chanu88@gmail.com	9848122394	Update	Delete
2.	Ruchitha	ruchitae5@gmail.com	9348823433	Update	Delete

8)Create a HTML page with Google Map.

```
<html>

<body align="center">

<iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3807.1930794820496!2d78.65009197455531!3d17.40251930237024!2m3!1f0!2f0!3
```

```
f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x3bcb90cedb836e63%3A0xae409c
5cd6f1ce0c!2sNalla%20Narasimha%20Reddy%20Education%20Society%E2
%80%99s%20Group%20of%20Institutions!5e0!3m2!1sen!2sin!4v171630988
2498!5m2!1sen!2sin" width="600" height="450" style="border:0;"
allowfullscreen="" loading="lazy" referrerpolicy="no-referrer-when-
downgrade"></iframe>

</body>

</html
```


Nalla Narasimha Reddy Educa...
CM23+774, Via Narapally, Ghatkesar Mandal, Korremula Rd, Peerzadiguda, Hyderabad, Telangana 500088
4.1 ★★★★ 849 reviews
View larger map
Directions
A CONVENTION
Majarguda
మజర్గూడ
Korremula Rd
MAKTHA
Road No. 7
Nalla Narasimha Reddy Education Society's...
Divyanagar Villas
Korremula agricultural land
Mallik Kovuri
Dream den projects
Home
Google
Keyboard shortcuts Map data ©2024 Terms Report a map error


Nalla Narasimha Reddy Educa...
CM23+774, Via Narapally, Ghatkesar Mandal, Korremula Rd, Peerzadiguda, Hyderabad, Telangana 500088
4.1 ★★★★ 849 reviews
View larger map
Directions
Nnrg medicinal plant
Nalla Narasimha Reddy Education Society's...
Mas
Venkateswara engineering works
Paper Plates Manufacturing
Google
Keyboard shortcuts Map data ©2024 Terms Report a map error

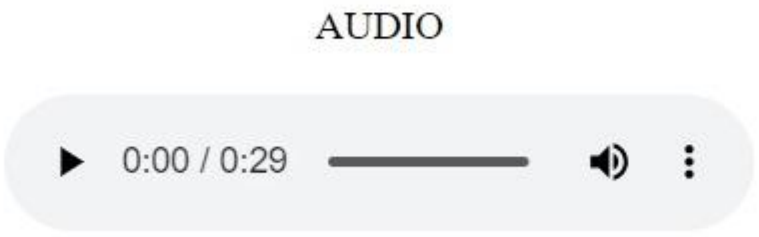
9)Create a HTML page with Video file

```
<html>
```

```
<head>
</head>
<body><center>
VIDEO<br>
<br>
<video width="400" controls>
  <source src="C:/Users/srivy/Downloads/radhe.mp4" type="video/mp4">
</center>
</video>
</body>
</html>
```

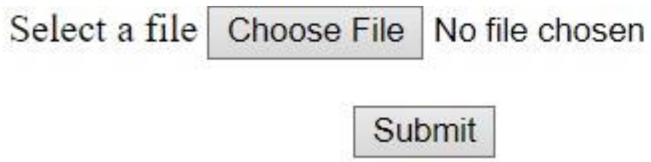
10)Create a HTML page with Audio file

```
<html>

<body><center>

AUDIO<br>

<br>

<audio controls>

  <source src="C:/Users/srivy/Downloads/aud.mp3" type="audio/mpeg"></center>

</audio>

</body>

</html>
```


AUDIO

11)Create a HTML page to upload a file.

```
<html>

<head>

<title> Selection of file </title>

</head>

<body>

<form>

<center>Select  a file

<input type="file" id="myfile" name="myfile"><br>

<br>

<input type="button" value="Submit">

</center>

</form>

</body>

</html>
```


Select a file