**1)Difference b/w JSON and XML**

|  |  |  |
| --- | --- | --- |
| Feature | JSON | XML |
| ****Syntax Simplicity**** | Uses a straightforward, easy-to-read syntax that resembles JavaScript object literals. It uses curly braces **{}** for objects and square brackets **[]** for arrays. | Uses a more complex and verbose syntax with tags to enclose data. Each element has a start tag **<tag>** and an end tag **</tag>**, with attributes contained within tags. |
| ****Readability**** | Generally easier to read and write due to its concise syntax. | Can be harder to read because of the verbose tag structure. |
| ****Data Representation**** | Represents data as key-value pairs, arrays, and nested objects. | Represents data with nested elements, attributes, and text nodes. |
| ****Data Types**** | Supports a limited set of data types: strings, numbers, booleans, arrays, and objects. | Does not inherently support data types. All data is treated as text, and data typing must be handled by schemas (e.g., XSD). |
| Usage | Commonly used in web applications, particularly with JavaScript and APIs | Widely used in a variety of applications, including web services (SOAP), document storage, and configuration files |
| Comments | Does not support comments within the data. | Supports comments using the **<!-- comment -->** syntax |
| Example | {  "students": {  "id": 6648,  "name": "Akshitha",  "GPA": 8.7  }  } | <student1>  <id>6648</id>  <name>Akshitha</name>  <GPA>8.7</GPA>  </student1> |

**2)Difference b/w authentication and authorization?**

| **Aspect** | **Authentication** | **Authorization** |
| --- | --- | --- |
| **Definition** | The process of verifying the identity of a user or system. | The process of determining the permissions or access levels of an authenticated user or system. |
| **Purpose** | To ensure that the user or system is who they claim to be. | To determine what resources and actions the authenticated user or system is allowed to access. |
| **Primary Question** | "Who are you?" | "What are you allowed to do?" |
| **Process** | Typically involves verifying credentials like username/password, biometric data, or tokens. | Involves checking the permissions associated with the authenticated identity against the requested resources or actions. |
| **When It Occurs** | Occurs before authorization. | Occurs after authentication. |
| **Data Used** | Usernames, passwords, OTPs, biometric data, security tokens. | Access control lists (ACLs), roles, policies, permissions. |
| **Outcome** | Confirms the identity of the user or system. | Grants or denies access to specific resources or actions. |
| **Scope** | Identity verification. | Access control and permissions management. |
| **Examples** | Logging in with a username and password, fingerprint scan. | Granting access to a file, allowing the user to perform administrative tasks. |

**3)XML and JSON files?**

XML Files

**Department:**

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

<departments>

<department1>

<id>7Z</id>

<name>CSE</name>

<HOD>Rameshwaraiah</HOD>

</department1>

<department2>

<id>0A</id>

<name>IT</name>

<HOD>Janardhan</HOD>

</department2>

<department3>

<id>2R</id>

<name>ECE</name>

<HOD>Basker</HOD>

</department3>

</departments>

**Year:**

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

<years>

<year1>

<id>1</id>

<students>500</students>

<branches>CSE,ECE,IT,CIVIL</branches>

</year1>

<year2>

<id>2</id>

<students>1500</students>

<branches>CSE</branches>

</year2>

<year3>

<id>3</id>

<students>1000</students>

<branches>ECE</branches>

</year3>

</years>

**Student:**

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

<students>

<student1>

<id>6648</id>

<name>Akshitha</name>

<GPA>8.7</GPA>

</student1>

<student2>

<id>217Z1A6648</id>

<name>Akshitha Reddy</name>

<GPA>8.8</GPA>

</student2>

<student3>

<id>207Z1A6648</id>

<name>Prabha</name>

<GPA>9.4</GPA>

</student3>

</students>

**JSON Files**

**Department:**

{

"departments": [

{

"id": 7Z,

"name": "CSE",

"HOD": " Rameshwaraiah "

},

{

"id": 0A,

"name": "IT",

"HOD": "Janardhan"

},

{

"id": 2R,

"name": "ECE",

"HOD": "Basker"

}

]

}

**Year:**

{

"years": [

{

"id": 1,

“students”:1000,

"name": "Juniors”

},

{

"id": 2,

“students”:1500,

"name": "Super Seniors"

},

{

"id": 3,

“students”:500,

“name”: "Juniors"

}

]

}

**Student:**

{

"students": [

{

"id": 6648,

"name": "Akshitha",

"GPA": 8.7

},

{

"id": 217Z1A6648,

"name": "Akshitha Reddy",

"GPA":8.8

},

{

"id": 207Z1A6648,

"name": "Prabha",

"GPA": 9.4

}

]

}

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

**JSON File**

{

"department": {

"id": 1,

"name": "Computer Science",

"head": "Dr. John Doe",

"years": [

{

"id": 1,

"name": "Freshman",

"students": [

{

"id": 1,

"name": "Emily Johnson"

},

{

"id": 2,

"name": "Michael Brown"

}

]

},

{

"id": 2,

"name": "Sophomore",

"students": [

{

"id": 3,

"name": "Jessica Wilson"

},

{

"id": 4,

"name": "Daniel White"

}

]

},

{

"id": 3,

"name": "Junior",

"students": [

{

"id": 5,

"name": "Sarah Davis"

},

{

"id": 6,

"name": "James Miller"

}

]

},

{

"id": 4,

"name": "Senior",

"students": [

{

"id": 7,

"name": "Linda Harris"

},

{

"id": 8,

"name": "Robert Clark"

}

]

}

]

}

}

**XML File**

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

<department>

<id>1</id>

<name>Computer Science</name>

<head>Dr. John Doe</head>

<year>

<id>1</id>

<name>Freshman</name>

<student>

<id>1</id>

<name>Emily Johnson</name>

</student>

<student>

<id>2</id>

<name>Michael Brown</name>

</student>

</year>

<year>

<id>2</id>

<name>Sophomore</name>

<student>

<id>3</id>

<name>Jessica Wilson</name>

</student>

<student>

<id>4</id>

<name>Daniel White</name>

</student>

</year>

<year>

<id>3</id>

<name>Junior</name>

<student>

<id>5</id>

<name>Sarah Davis</name>

</student>

<student>

<id>6</id>

<name>James Miller</name>

</student>

</year>

<year>

<id>4</id>

<name>Senior</name>

<student>

<id>7</id>

<name>Linda Harris</name>

</student>

<student>

<id>8</id>

<name>Robert Clark</name>

</student>

</year>

</department>

**5)Simple Login Page?**

<!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>

</head>

<body>

<div class="login-container">

<h2>Login</h2>

<form action="/login" method="post">

<div class="input-group">

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

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

</div>

<div class="input-group">

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

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

</div>

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

</form>

</div>

</body>

</html>



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

<!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;

margin: 0;

padding: 20px;

background-color: #f7f7f7;

}

.container {

max-width: 600px;

margin: auto;

background: white;

padding: 20px;

border-radius: 8px;

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

}

.form-group {

margin-bottom: 15px;

}

.form-group label {

display: block;

margin-bottom: 5px;

}

.form-group input,

.form-group select,

.form-group textarea {

width: 100%;

padding: 8px;

box-sizing: border-box;

}

.form-group input[type="checkbox"],

.form-group input[type="radio"] {

width: auto;

}

.form-group .options {

display: flex;

flex-wrap: wrap;

}

.form-group .options label {

margin-right: 10px;

}

.form-group button {

padding: 10px 15px;

background: #007bff;

border: none;

color: white;

cursor: pointer;

border-radius: 4px;

}

.form-group button:hover {

background: #0056b3;

}

</style>

</head>

<body>

<div class="container">

<h2>Create New User</h2>

<form>

<!-- Text Fields -->

<div class="form-group">

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

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

</div>

<div class="form-group">

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

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

</div>

<div class="form-group">

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

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

</div>

<!-- Dropdown -->

<div class="form-group">

<label for="role">Role:</label>

<select id="role" name="role" required>

<option value="">Select a role</option>

<option value="admin">Admin</option>

<option value="user">User</option>

<option value="guest">Guest</option>

</select>

</div>

<!-- Radio Buttons -->

<div class="form-group">

<label>Gender:</label>

<div class="options">

<label><input type="radio" name="gender" value="male" required> Male</label>

<label><input type="radio" name="gender" value="female" required> Female</label>

<label><input type="radio" name="gender" value="other" required> Other</label>

</div>

</div>

<!-- Checkboxes -->

<div class="form-group">

<label>Interests:</label>

<div class="options">

<label><input type="checkbox" name="interests" value="sports"> Sports</label>

<label><input type="checkbox" name="interests" value="music"> Music</label>

<label><input type="checkbox" name="interests" value="travel"> Travel</label>

<label><input type="checkbox" name="interests" value="technology"> Technology</label>

</div>

</div>

<!-- List (Multi-select) -->

<div class="form-group">

<label for="languages">Languages Spoken:</label>

<select id="languages" name="languages" multiple>

<option value="english">English</option>

<option value="spanish">Spanish</option>

<option value="french">French</option>

<option value="german">German</option>

<option value="chinese">Chinese</option>

</select>

</div>

<!-- Submit Button -->

<div class="form-group">

<button type="submit">Create User</button>

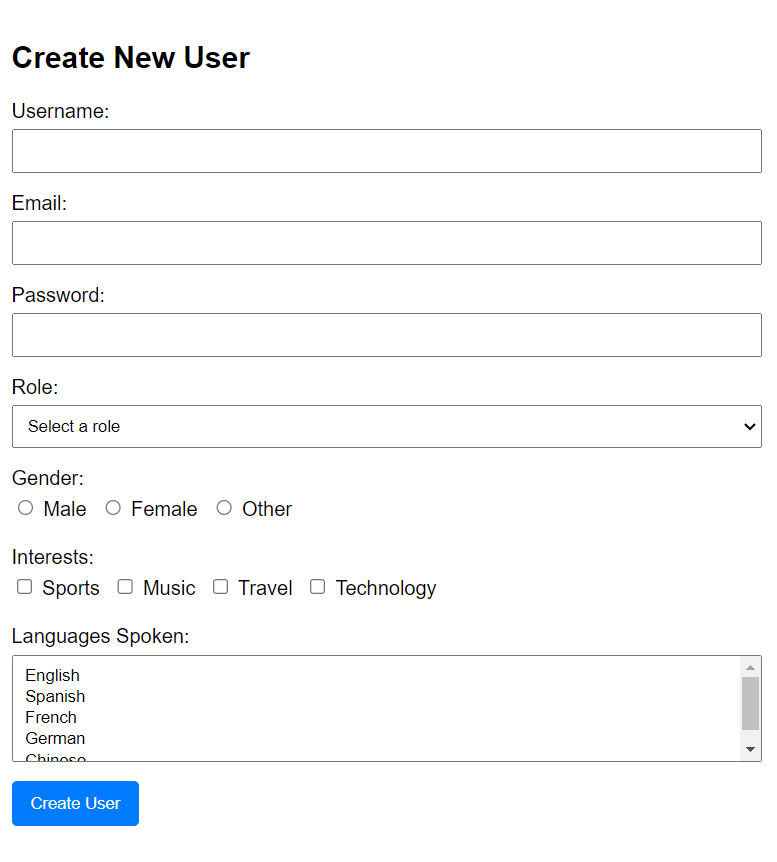
</div>

</form>

</div>

</body>

</html>



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

<!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>

table {

width: 100%;

border-collapse: collapse;

}

table, th, td {

border: 1px solid black;

}

th, td {

padding: 10px;

text-align: left;

}

</style>

</head>

<body>

<h1>User Management</h1>

<table>

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>Email</th>

<th>Actions</th>

</tr>

</thead>

<tbody id="userTable">

<!-- Users will be populated here -->

</tbody>

</table>

<!-- Edit User Modal -->

<div id="editModal" style="display: none; position: fixed; top: 50%; left: 50%; transform: translate(-50%, -50%); background: white; padding: 20px; border: 1px solid black;">

<h2>Edit User</h2>

<form id="editForm">

<input type="hidden" id="editUserId">

<div>

<label for="editUserName">Name:</label>

<input type="text" id="editUserName">

</div>

<div>

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

<input type="text" id="editUserEmail">

</div>

<button type="button" onclick="saveUser()">Save</button>

<button type="button" onclick="closeEditModal()">Cancel</button>

</form>

</div>

<script>

const users = [

{ id: 1, name: 'John Doe', email: 'john@example.com' },

{ id: 2, name: 'Jane Smith', email: 'jane@example.com' },

{ id: 3, name: 'Bob Johnson', email: 'bob@example.com' }

];

document.addEventListener('DOMContentLoaded', function () {

populateUsers();

});

function populateUsers() {

const userTable = document.getElementById('userTable');

userTable.innerHTML = '';

users.forEach(user => {

const row = document.createElement('tr');

row.innerHTML = `

<td>${user.id}</td>

<td>${user.name}</td>

<td>${user.email}</td>

<td>

<button onclick="editUser(${user.id})">Edit</button>

<button onclick="confirmDeleteUser(${user.id})">Delete</button>

</td>

`;

userTable.appendChild(row);

});

}

function editUser(userId) {

const user = users.find(u => u.id === userId);

document.getElementById('editUserId').value = user.id;

document.getElementById('editUserName').value = user.name;

document.getElementById('editUserEmail').value = user.email;

document.getElementById('editModal').style.display = 'block';

}

function closeEditModal() {

document.getElementById('editModal').style.display = 'none';

}

function saveUser() {

const userId = document.getElementById('editUserId').value;

const userName = document.getElementById('editUserName').value;

const userEmail = document.getElementById('editUserEmail').value;

const user = users.find(u => u.id == userId);

user.name = userName;

user.email = userEmail;

closeEditModal();

populateUsers();

}

function confirmDeleteUser(userId) {

if (confirm("Are you sure you want to delete this user?")) {

deleteUser(userId);

}

}

function deleteUser(userId) {

const index = users.findIndex(u => u.id === userId);

if (index > -1) {

users.splice(index, 1);

}

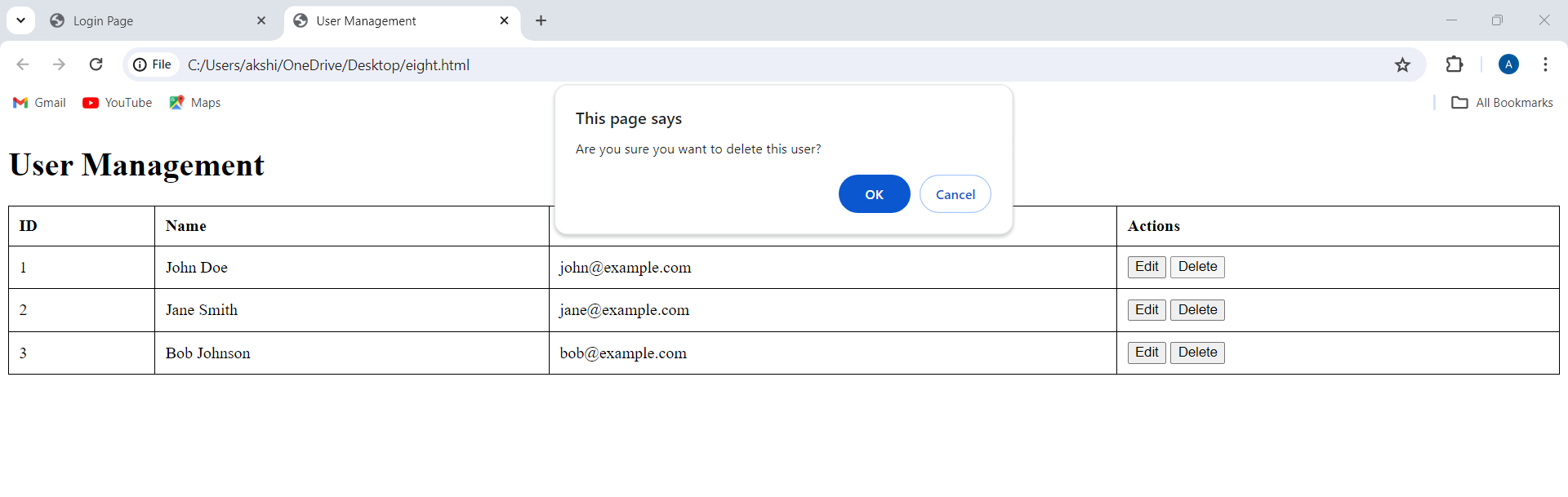
populateUsers();

}

</script>

</body>

</html>

****

**8)** **Create a HTML page with Google Map**

<!DOCTYPE html>

<html>

<head>

<title>Google Maps Example</title>

<style>

/\* Set the size of the div element that contains the map \*/

#map {

height: 400px; /\* The height is 400 pixels \*/

width: 100%; /\* The width is the width of the web page \*/

}

</style>

</head>

<body>

<h3>My Google Map</h3>

<!-- The div element for the map -->

<div id="map"></div>

<script>

// Initialize and add the map

function initMap() {

// The location to center the map (example: New York City)

const location = { lat: 40.7128, lng: -74.0060 };

// The map, centered at the location

const map = new google.maps.Map(document.getElementById("map"), {

zoom: 10,

center: location,

});

// The marker, positioned at the location

const marker = new google.maps.Marker({

position: location,

map: map,

});

}

</script>

<!-- Load the Google Maps API script with your API key -->

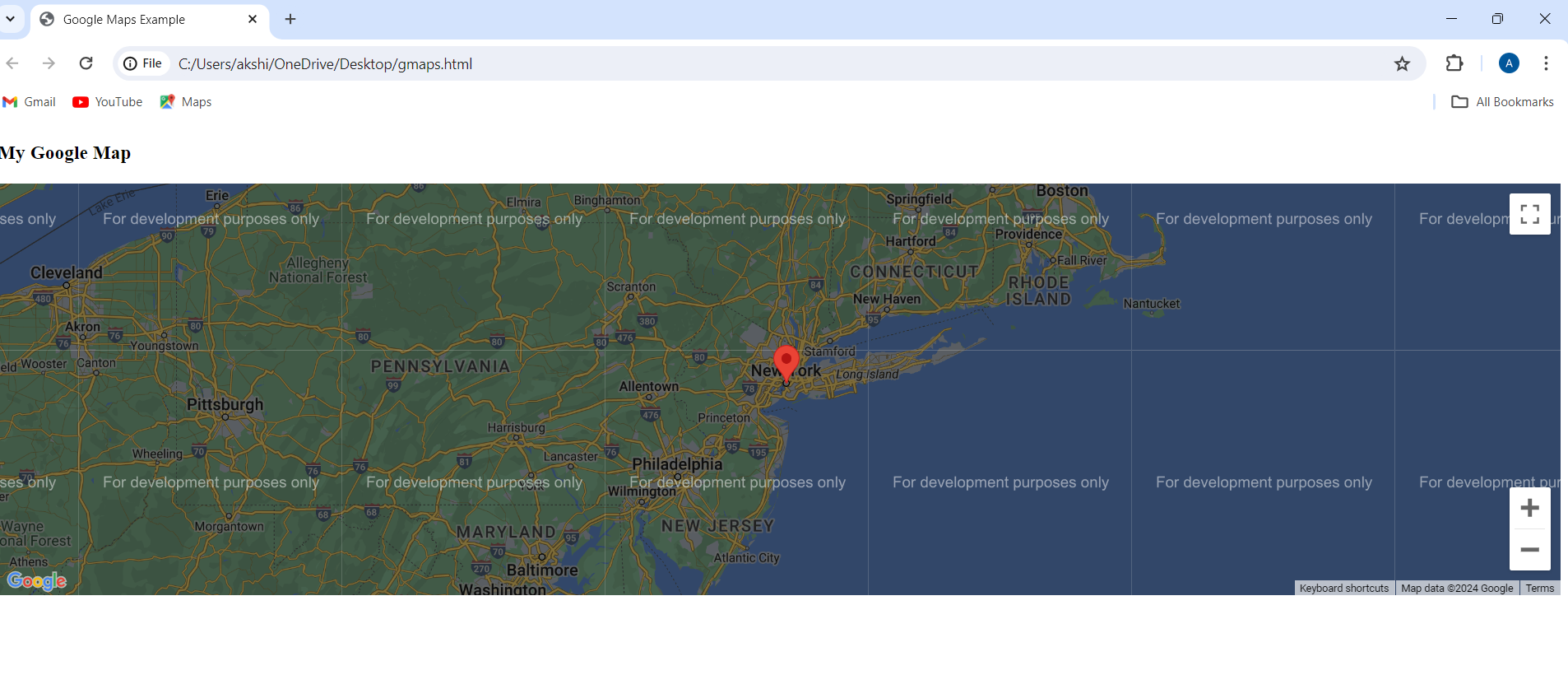
<script async defer

src="https://maps.googleapis.com/maps/api/js?key=YOUR\_API\_KEY&callback=initMap">

</script>

</body>

</html>



**9)Create a HTML File 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="/upload" method="post" enctype="multipart/form-data">

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

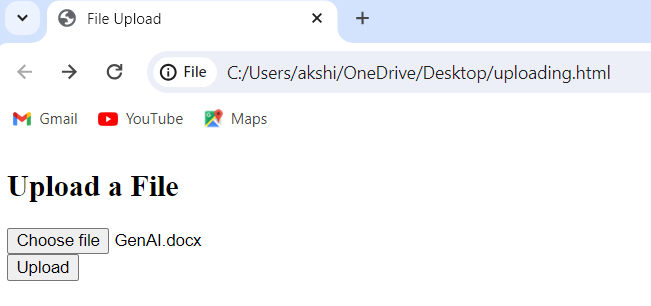
<br>

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

</form>

</body>

</html>



**10)Create a HTML Page with Audio**

<!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>

<audio controls>

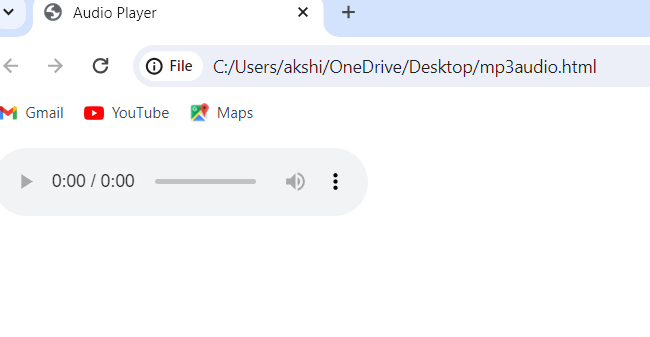
<source src=""C:\Users\akshi\OneDrive\Documents\audio song.mp3"" type="audio/mpeg">

Your browser does not support the audio element.

</audio>

</body>

</html>



**11)Create a HTML Page with Video**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Video Example</title>

<style>

body {

font-family: Arial, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

background-color: #f0f0f0;

}

video {

max-width: 100%;

height: auto;

border: 1px solid #ccc;

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

}

</style>

</head>

<body>

<video controls>

<source src="C:\Users\akshi\OneDrive\Desktop\video.mp4" type="video/mp4">

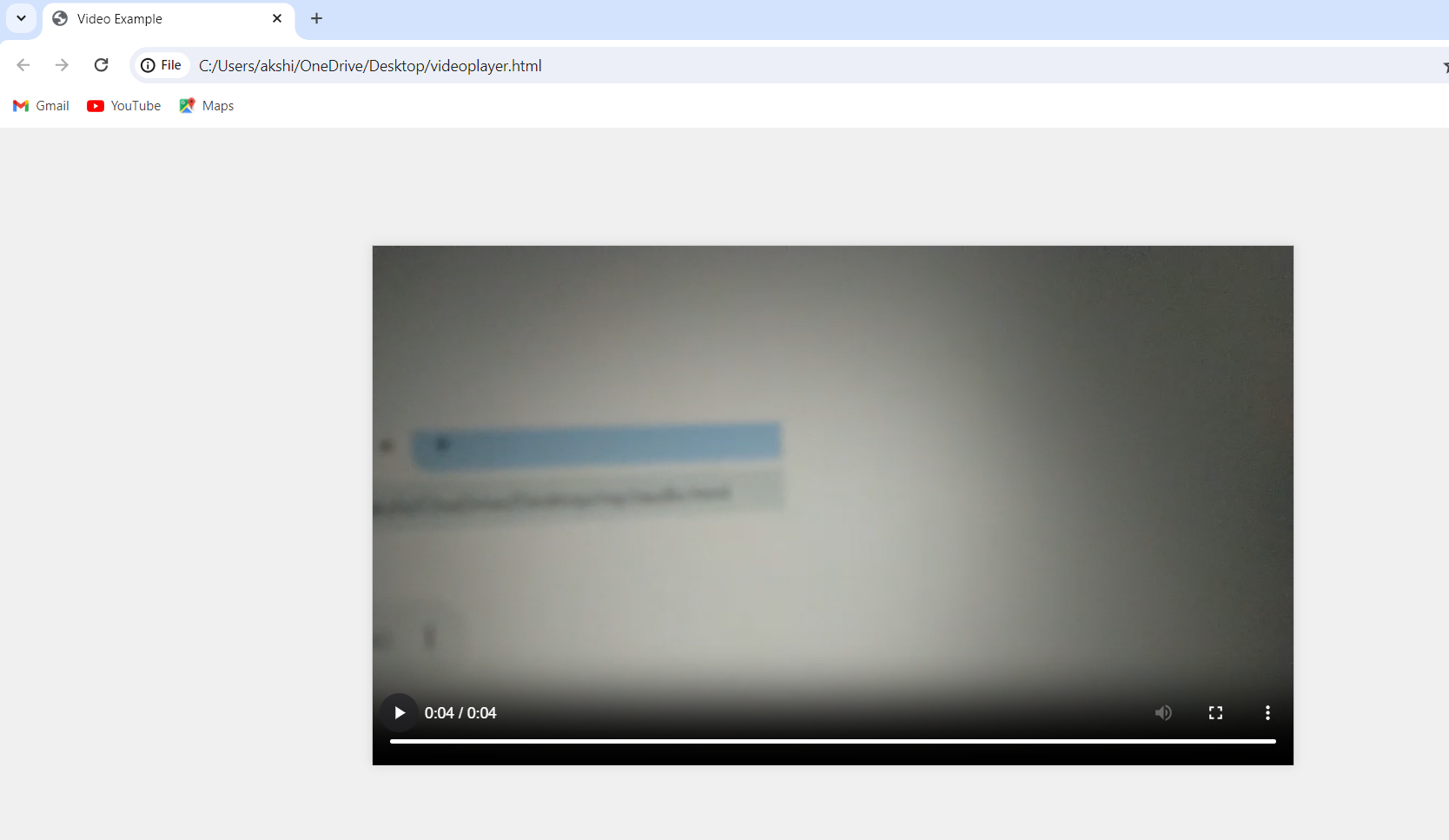
<source src="C:\Users\akshi\OneDrive\Desktop\video.mp4" type="video/ogg">

Your browser does not support the video tag.

</video>

</body>

</html>

****