Q1. Implement <iframe> tag by embedding the following elements from another webpage: <header>, <nav>, <section>, <article>, <footer>. Add a navigation bar inside <nav> with links to different sections of the page using the <a> tag.

```
<!DOCTYPE html>
<head> <title>Document</title></head>
<body>
  <iframe src="home.html" frameborder="10" width="400" height="400"></iframe>
</body></html>
<!DOCTYPE html>
<head> <title>Document</title> </head>
<body> <header> <h2>Hello this is a header</h2>
<nav>  <a href="home.html"> Homepage</a><a href="help.html"> Contact Us</a>
<a href="list.html"> Services provided by us</a><a href="#id1"> Discount</a> <a href="#id1"> Discount</a> </a>
href="#id2">Important News!!</a> </nav>
  </header>
  <section> <h2>About us</h2>
    We are a great corporation, who comes forward as a leading tech company.
  </section>
  <article> <h2 id="id1">Sale Sale Flat 50% off</h2>
    We are offering discount if you purchase our premium today. Please go to our official website to
purchase the premium...
  </article>
  <article> <h2 id="id2"> Server Maintainence</h2>
    The server will be on break comming friday, so please plan accordingly...
  </article>
<footer>Terms and conditionSupportRead about our sourcesIi></footer>
</body></html>
```

Hello this is a header

- Homepage
- Contact Us
- Services provided by us
- <u>Discount</u>
- Important News!!

About us

We are a great corporation, who comes forward as a leading tech company.

Sale Sale Flat 50% off

We are offering discount if you purchase our premium today. Please go to our official website to purchase the premium...

Server Maintainence

The server will be on break comming friday, so please plan accordingly...

- Terms and condition
- Support
- Read about our sources

Q2. Embed a table from another webpage using <iframe>. The structure of the table should be as:-

	Seminar			
Day		Tania		
	Begin	End	Topic	
Monday	8:00 a.m.	5:00 n m	Introduction to XML	
IVIOIIday	0.00 a.III.	5:00 p.m.	Validity: DTD and Relax NG	
	8:00 a.m.	11:00 a.m.	XPath	
Tuesday	11:00 a.m.	2:00 p.m.		
	2:00 p.m.	5:00 p.m.	XSL Transformations	
Wednesday	8:00 a.m.	12:00 p.m.	XSL Formatting Objects	

```
<!DOCTYPE html>
<head><title>Document</title>
</head> <body>
 <iframe src="home.html" frameborder="20" width="600" height="600"></iframe>
</body> </html>
  Day  Schedule 
  Schedule Topic 
  Begin Begin
  Monday 8:00 am 5:00 pm
Introduction to XML
 Validity:DTD and Relax NG
 Tuesday  8:00 am   11:00 am
rowspan="1">XPath
 11:00 am2:00 pmrowspan="2">XSL Tranformations
 2:00 pm5:00 pm
 Wednesday8:00 am12"00 pmXML Formatting
Objects
```

	Schedule			
Day	Schedule		Tr i .	
	Begin	Begin	Topic	
	8:00 am	5:00 pm	Introduction to XML	
Monday			Validity:DTD and Relax NG	
	8:00 am	11:00 am	XPath	
Tuesday	11:00 am	2:00 pm	XSL Tranformations	
	2:00 pm	5:00 pm		
Wednesday	8:00 am	12"00 pm	XML Formatting Objects	

This the the original website. And above is an iframe.

Q3. Embed <map> and <area> from another web page using <iframe>. Create a clickable map of a school or college campus.

Use an tag to display a campus map.

Use a <map> with multiple <area> tags to create clickable zones for different buildings (e.g., Library, Cafeteria, Sports Complex).

Each <area> should link to a new page with information about that building. Use alt attributes to describe each area.

<map name="map1">

<area shape="rect" coords="77,3, 408,80" href="https://en.wikipedia.org/wiki/Assembly_hall"
alt="About hall">

<area shape="circle" coords="251,240,66" href="https://en.wikipedia.org/wiki/Statue"
alt="Statue">

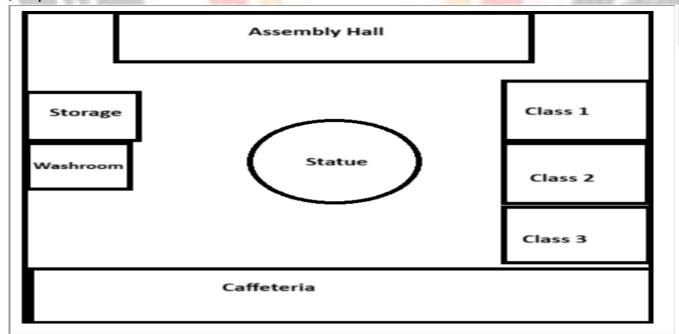
<area shape="rect" coords="4,129,92,204" href="https://en.wikipedia.org/wiki/Storage"
alt="Storage area">

<area shape="rect" coords="6,210,87,283" href="https://en.wikipedia.org/wiki/Bathroom"
alt="Washroom">

<area shape="rect" coords="8,413,504,497" href="https://en.wikipedia.org/wiki/Cafeteria"
alt="Cafeteria">

<area shape="rect" coords="387,112,501,400" href="https://en.wikipedia.org/wiki/Class"
alt="Classes">

</map>



This the the original website. And above is an iframe.

Q.4 Style a webpage by changing the font, color, and size of headings and paragraphs.



Q.5 Create a button with background color, padding, and rounded corners. Add a hover effect that changes the background color.

```
<!DOCTYPE html>
<head> <title> My Document</title> </head>
<body>
    <button> Press Me</button>
    <style>
```

```
button{
    padding: 10px;
    width: 150px;
    height: 100px;
    border-radius: 60px;
    background-color: aqua;
    color: red;
    font-weight: 550;
    font-size: medium;
}
button:hover{
    background-color:violet
}
</style> </html>
```





Q.6 Display an image and add a border-radius for rounded corners and add a shadow effect to the image.

```
<!DOCTYPE html>
<head> <title> My Document</title> </head>
<body> <img src="/Assignments/download.png" alt="Image" height="200" width="400">
<style> img{ border: 5px solid red; border-radius: 20px; box-shadow: 10px 10px 20px black; }
</style>
</body> </HTMl>
```



Q.7 Create an animated gradient border around a div using @keyframes and border-image-source.

Apply an animated gradient border that changes colors.

```
Use border-image-source instead of solid borders.
body{
    display: flex; align-items: center; justify-content: center; }
  button{
    height: 100px; width: 200px;
    padding: 1px 1px; background-color:azure; border: none; color: gray;
    border-radius: 60px; font-family: Auga; font-weight: 700; position: relative;
    display: flex; align-items: center; justify-content: center; font-size: 20px;
  button::after{
    content: "; position: absolute; height: 107%; width: 104%;
    background-image: linear-gradient(to bottom right,blue,purple); border-radius: 60px;
    z-index: -1; }
  button:hover{ animation: animate 1s linear infinite; z-index: 0; }
@keyframes animate{ 0%{ border-image: linear-gradient(to bottom right, blue, purple); }
    25%{ border-image: linear-gradient(to top right,blue,purple); }
    50%{ border-image: linear-gradient(to top center,blue,purple); }
    75%{ border-image: linear-gradient(to top left,blue,purple); }
    100% { border-image: linear-gradient(to bottom left,blue,purple); } }
</style> </body>
```



Q.8 Create a parallax scrolling effect where the background image moves at a different speed than the content.

Use background-attachment: fixed; for the parallax effect. Ensure the content scrolls over the background smoothly.

```
<body>
<div class="section1"> <div class="text">Parallax</div> </div>
<div class="section2"> <div class="text">Normal</div> </div>
<style> html{ height: 100vh; overflow: hidden; }
  body{ margin: 0; height: 100vh; width: 100vw; background: black; perspective: 1px;
    transform-style: preserve-3d; overflow-x: hidden; overflow-y: scroll;
  .section1..section2{
    width: 100%; min-height: 100vh; position: relative; transform-style: preserve-3d; }
  .section2{ background: rgb(68,35,19); }
  .section1::before{ content: ""; width: 100%; height: 100%; position: absolute;
     background: url(back.jpg) top center; background-size: cover;
     transform: translateZ(-1px) scale(2,2); filter: blur(2px); }
  .text{ top: 30%; left: 50%; position: absolute; font-size: 15vw; color: white;
    text-shadow: 2px 2px 5px rgba(0,0,0,.3), 5px 5px 70px rgba(225,225,225,.5);
    transform: scale(1,1.1) translate(-50%, 10%); }
</style>
```



Q.9 Use Flexbox to center a box in the middle of the page:-Create a div box and center it both vertically and horizontally.

Add padding, background color, and a shadow to the box.

```
<div >Hello</div>
<style>
    div{
        background-color: aquamarine; color: white; height: 100px; width: 100px;
        display: flex; align-items: center; justify-content: center; text-align: center;
        padding: 10 px; box-shadow: 10px 10px 20px black; }
</style>
```

Hello

Q.10 Create a Student Registration Form using HTML that includes the following fields: Form Requirements:

Student Name – (Text input, Required)

Email – (Email input, Required, Must be in a valid email format)

Phone Number – (Number input, Required, Must be exactly 10 digits)

Date of Birth – (Date input, Required)

Gender – (Radio buttons: Male, Female, Other, Required)

Course Selection – (Dropdown with at least 3 options, Required)

Address - (Textarea, Required)

Password – (Password input, Required, Minimum 6 characters)

Confirm Password – (Password input, Must match Password)

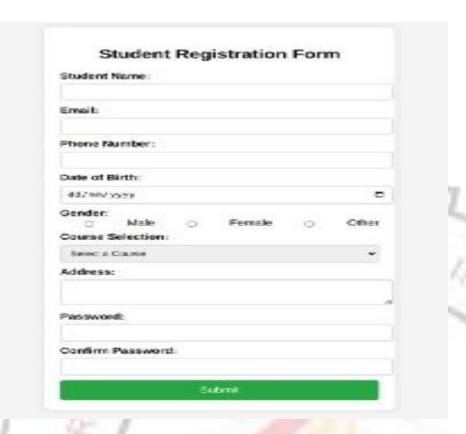
Submit Button – (Button to submit the form)

```
<style> body {
font-family: Arial, sans-serif; background-color: #f4f4f4; display: flex; justify-content: center;
align-items: center; height: 100vh; }
.container { background: white; padding: 20px; border-radius: 10px; width: 400px;
box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1); }
h2 { text-align: center; }
```

```
label { font-weight: bold; }
input, select, textarea { width: 100%; padding: 8px; margin: 5px 0 10px; border: 1px solid #ccc;
border-radius: 5px; } .gender-group { display: flex; gap: 10px; }
button { background-color: #28a745; color: white; border: none; padding: 10px;
border-radius: 5px; font-size: 16px; cursor: pointer; width: 100%; }
button:hover { background-color: #218838; } .error { color: red; font-size: 14px; }
</style> </head> <body> <div class="container"> <h2>Student Registration Form</h2>
<form id="registrationForm"> <label for="name">Student Name:</label>
<input type="text" id="name" required> <span id="nameError" class="error"></span>
<label for="email">Email:</label> <input type="email" id="email" required>
<span id="emailError" class="error"></span> <label for="phone">Phone Number:</label>
<input type="number" id="phone<mark>" required> <spa</mark>n id="phoneError" class="error"></span>
<label for="dob">Date of Birth:</label> <input type="date" id="dob" required>
<span id="dobError" class="error"></span> <label>Gender:</label>
<div class="gender-grou<mark>p"> <input</mark> type="radio" name="gen<mark>der" value=</mark>"Male" required> Male
<input type="radio" name="gender" value="Female" required> Female
<input type="radio" name="gender" value="Other" required> Other
</div> <span id="genderError" class="error"></span>
<label for="course">Course Selection:</label> <select id="course" required>
<option value="">Select a Course</option> <option value="B.Tech">B.Tech</option>
<option value="MBA">MBA</option> <option value="B.Sc">B.Sc</option>
</select> <span id="courseError" class="error"></span>
<label for="address">Address:</label> <textarea id="address" required></textarea>
<span id="addressError" class="error"></span><label for="password">Password:</label>
<input type="password" id="password" required> <span id="passwordError"</pre>
class="error"></span>
```

```
<label for="confirmPassword">Confirm Password:</label>
<input type="password" id="confirmPassword" required>
<span id="confirmPasswordError" class="error"></span>
<button type="submit">Submit</button> </form> </div> <script>
document.getElementById("registrationForm").addEventListener("submit", function(event)
{ let valid = true;
const name = document.getElementById("name").value.trim();
if (name === "") {
document.getElementById("nameError").innerText = "Name is required!
valid = false;
} else { document.getElementById("nameError").innerText = ""; }
const email = document.getEle<mark>mentById("email")</mark>.value.trim();
const emailPattern = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2.}$/;
if (!emailPattern.test(email)) {
document.getElementById("emailError").innerText = "Enter a valid email!";
valid = false;
} else { document.getElementByld("emailError").innerText = ""; }
const phone = document.getElementById("phone").value.trim();
if (phone.length !== 10 || isNaN(phone)) {
document.getElementById("phoneError").innerText = "Phone number must be 10 digits!";
valid = false;
} else { document.getElementById("phoneError").innerText = ""; }
const dob = document.getElementById("dob").value;
if (!dob) {document.getElementById("dobError").innerText = "Date of birth is required!";
valid = false;
} else { document.getElementById("dobError").innerText = ""; }
```

```
const gender = document.querySelector('input[name="gender"]:checked');
if (!gender) {
document.getElementById("genderError").innerText = "Select a gender!";
valid = false;
} else { document.getElementById("genderError").innerText = ""; }
const course = document.getElementById("course").value;
if (course === "") { document.getElementById("courseError").innerText = "Select a course!";
valid = false;
} else { document.getElementById("courseError").innerText = ""; }
const address = document.getElementById("address").value.trim();
if (address === "") { document.getElementById("addressError").innerText = "Address is
required!";
valid = false;
} else {document.getElementById("addressError").innerText = ""; }
const password = document.getElementById("password").value.trim();
if (password.length < 6) {
document.getElementById("passwordError").innerText = "Password must be at least 6
characters!"; valid = false;
} else { document.getElementById("passwordError").innerText = ""; }
const confirmPassword = document.getElementById("confirmPassword").value.trim();
if (confirmPassword !== password) {
document.getElementById("confirmPasswordError").innerText = "Passwords do not
match!";
valid = false; } else { document.getElementById("confirmPasswordError").innerText = ""; }
if (!valid) { event.preventDefault(); } });</script> </body> </html>
```



Q.11 To-Do List – Create a simple to-do list where users can add and remove tasks.

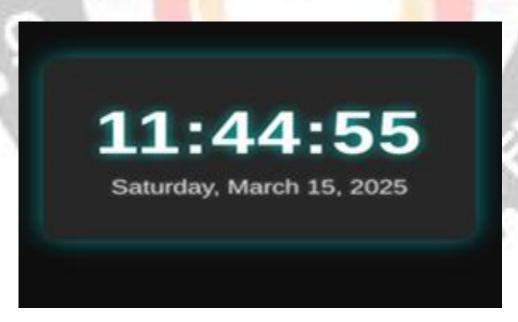
```
<style>
body { font-family: Arial, sans-serif; background: #f4f4f4; text-align: center;
margin: 50px; }
.container { max-width: 400px; background: white; padding: 20px;
border-radius: 10px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); margin: auto;
h2 { margin-bottom: 15px; color: #333; }
input { width: 70%; padding: 10px; border: 1px solid #ccc;
border-radius: 5px; font-size: 16px; }
button {
background: #28a745; color: white; border: none;
padding: 10px 15px; font-size: 16px; border-radius: 5px;
cursor: pointer; margin-left: 5px;
button:hover { background: #218838;
ul { list-style: none; padding: 0; margin-top: 20px; }
li {
background: #fff; padding: 10px; border-radius: 5px; display: flex; justify-content: space-between;
align-items: center; margin-bottom: 10px; box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
transition: all 0.3s ease-in-out; }
li.fade-out { opacity: 0; transform: translateX(20px);
```

```
transition: opacity 0.3s ease, transform 0.3s ease; }
.remove-btn { background: red; color: white; border: none; padding: 5px 10px;
cursor: pointer; border-radius: 5px; font-size: 14px; }
.remove-btn:hover { background: darkred; }
</style> </head> <body>
<div class="container"> <h2>To-Do List</h2>
<input type="text" id="task" placeholder="Enter Task"> <button onclick="addTask()">Add Task</button>
ul id="taskList"></div></script>
document.addEventListener("DOMContentLoaded", loadTasks);
document.getElementById("task").addEventListener("keypress", function(event) {
if (event.key === "Enter") { addTask(); } });
function addTask() { let taskInput = document.getElementById("task");
let task = taskInput.value.trim(); if (task === "") { alert("Please enter a task"); return; }
let list = document.getElementById("taskList"); let li = document.createElement("li");
li.innerHTML = `${task} <button class="remove-btn"</pre>
onclick='removeTask(this)'>Remove</button>';
list.appendChild(li); saveTasks(); taskInput.value = ""; taskInput.focus(); }
function removeTask(element) { let li = element.parentNode;
li.classList.add("fade-out"); setTimeout(() => { li.remove(); saveTasks(); }, 300); }
function saveTasks() { let tasks = [];
document.querySelectorAll("#taskList li").forEach(item => {
tasks.push(item.firstChild.textContent); });
localStorage.setItem("tasks", JSON.stringify(tasks)); }
function loadTasks() { let savedTasks = JSON.parse(localStorage.getItem("tasks")) || [];
let list = document.getElementById("taskList");
savedTasks.forEach(task => { let li = document.createElement("li");
li.innerHTML = `${task} <button class="remove-btn" onclick='removeTask(this)'>Remove</button>`;
list.appendChild(li); }); } </script> </body> </html>
```

Enter Task	
Add Tas	sk
coding	Remove
sleeping	Remove
chocking	Remove
classes	Remove
dinner	Remove

Q12. Digital Clock – Display the current time that updates every second.

```
<style>
body {display: flex; justify-content: center; align-items: center;
height: 100vh; background: #0e0e0e; color: #fff; font-family: 'Arial', sans-serif; margin: 0; }
.clock-container { text-align: center;
background: rgba(255, 255, 255, 0.1); padding: 40px; border-radius: 15px;
box-shadow: 0px 0px 20px rgba(0, 255, 255, 0.5); backdrop-filter: blur(10px); }
.clock { font-size: 60px; font-weight: bold; letter-spacing: 2px;
text-shadow: 0px 0px 10px rgba(0, 255, 255, 0.8); transition: all 0.2s ease-in-out; }
.clock:hover { transform: scale(1.05); }
.date { font-size: 20px; margin-top: 10px; color: #ccc; text-shadow: 0px 0px 5px rgba(255, 255, 255,
0.5); }
</style> </head> <body> <div class="clock-container"> <div class="clock" id="clock">00:00:00</div>
<div class="date" id="date">Loading...</div> </div> <script>
function updateClock() { let now = new Date();
let hours = now.getHours().toString().padStart(2, '0');
let minutes = now.getMinutes().toString().padStart(2, '0');
let seconds = now.getSeconds().toString().padStart(2, '0');
let timeString = `${hours}:${minutes}:${seconds}`;
document.getElementById("clock").innerText = timeString;
let options = { weekday: 'long', month: 'long', day: 'numeric', year: 'numeric' };
document.getElementById("date").innerText = now.toLocaleDateString('en-US', options); }
setInterval(updateClock, 1000); updateClock(); </script> </body> </html>
```



Q 13. Color Changer – Allow users to change the background color by clicking buttons.

Image Slider – Create a basic image carousel that auto-plays or changes images on button clicks.

<style> body { font-family: Arial, sans-serif; text-align: center; margin: 0; padding: 0;

```
transition: background-color 0.5s ease-in-out;
}
h1 { color: #333; margin-top: 20px; }
.color-buttons { margin: 20px; }
.color-btn { padding: 10px 20px; margin: 5px; border: none; cursor: pointer; font-size: 16px;
color: white; border-radius: 5px; transition: 0.3s; }
.color-btn:hover { opacity: 0.8; }
.slider-container { position: relative; width: 80%; max-width: 700px; margin: 30px auto;
overflow: hidden; border-radius: 10px; box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2); }
.slider { display: flex; width: 400%; transition: transform 1s ease-in-out; }
.slide { width: 100%;flex: 1 0 100%; }
.slide img { width: 100%; height: 250px; object-fit: cover; border-radius: 10px; }
.prev, .next { position: absolute; top: 50%; transform: translateY(-50%);
background-color: rgba(0, 0, 0, 0.5); color: white; padding: 10px; border: none;
cursor: pointer; border-radius: 50%; font-size: 20px; transition: 0.3s; }
.prev:hover, .next:hover { background-color: black; }
.prev { left: 10px; } .next { right: 10px; } </style> </head> <body>
<h1>Color Changer & Image Slider</h1> <div class="color-buttons">
<button class="color-btn" style="background-color: #ff5733;"</pre>
onclick="changeColor('#ff5733')">Red</button>
<button class="color-btn" style="background-color: #33ff57;"</pre>
onclick="changeColor('#33ff57<mark>')">Green</button></mark>
<button class="color-btn" style="background-color: #3357ff;"</p>
onclick="changeColor('#3357ff')">Blue</button>
<button class="color-btn" style="background-color: #f4c242;"</pre>
onclick="changeColor('#f4c242')">Yellow</button>
<button class="color-btn" style="background-color: #8e44ad;"</p>
onclick="changeColor('#8e44ad')">Purple</button> </div>
<div class="slider-container"> <div class="slider" id="slider">
<div class="slide"><img src="https://gehu.ac.in/assets/images/GEHU-BTL-b80ef66e.webp"</pre>
alt="GEHU Bhimtal"></div>
<div class="slide"><img src="https://lookaside.fbsbx.com/lookaside/crawler/media/?"</pre>
media_id=880431600782829" alt="GEHU Bhimtal"></div>
<div class="slide"><img</pre>
src="https://image-static.collegedunia.com/public/college_data/images/appImage/
57212_cover2.jpg?h=260&w=360&mode=crop" alt="GEHU Bhimtal"></div>
<div class="slide"><img</pre>
src="https://lookaside.instagram.com/seo/google_widget/crawler/?
media_id=3295975471895630683" alt="GEHU Bhimtal"></div>
</div> <button class="prev" onclick="prevSlide()"> </button> <
<button class="next" onclick="nextSlide()"> </button> >
</div> <script> function changeColor(color) { document.body.style.backgroundColor = color;
```

```
let currentSlide = 0; const slides = document.querySelectorAll(".slide");
const totalSlides = slides.length; function updateSlider() {
  document.getElementById("slider").style.transform = `translateX(-${currentSlide * 100}%)`; }
  function nextSlide() {    currentSlide = (currentSlide + 1) % totalSlides;    updateSlider(); }
  function prevSlide() {    currentSlide = (currentSlide - 1 + totalSlides) % totalSlides;
    updateSlider(); }   setInterval(nextSlide, 3000); </script> </body> </html>
```

Color Changer & Image Slider





Q.14 Create an XML file to store information about students, books, or employees.

<database>

<students> <student id="S001"> <name>Ram</name> <age>25</age> <gender>Male</gender>
<course>Computer Science</course> <email>ajay@example.com</email>
<phone>9876543210</phone> </student> <student id="S002"> <name>Sita</name>
<age>22</age> <gender>Female</gender> <course>Data Science</course>
<email>diya@example.com</email> <phone>9876543211</phone> </student> </student> </student>
<!-- Book Information --> <books> <book id="B101"> <title>Introduction to Machine Learning</title>
<author>Tom Mitchell</author> <publisher>McGraw-Hill</publisher> <year>1997</year> <isbn>978-0070428072</isbn> </book> <book id="B102"> <title>Artificial Intelligence: A Modern
Approach</title> <author>Stuart Russell, Peter Norvig</author> <publisher>Pearson</publisher>
<year>2020</year> <isbn>978-0134610993</isbn> </book> <book> <book> <book> <employees>
<employee id="E001"> <name>James</name> <position>Software Engineer</position>
<department>IT</department> <email>rahul@example.com</email> <phone>9876543212</phone>

<salary>90000</salary> </employee> <employee id="E002"> <name>Sia</name>
<position>HR Manager</position> <department>Human Resources</department>
<email>priya@example.com</email> <phone>9876543213</phone> <salary>75000</salary>
</employee> </employees> </database</pre>

