

ASSIGNMENT-02:-

Create an image animation using HTML, CSS and JavaScript. It should take the user input to set the interval for changing the image as well as put start and stop buttons.

Source Code:-

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>Image Animation — Interval, Start & Stop</title>
  <style>
    /* Basic layout */
    :root{ --bg:#0f1724; --card:#0b1220; --accent:#7dd3fc; }
    *{box-sizing:border-box}
    body{font-family:Inter, system-ui, -apple-system, Segoe UI, Roboto, "Helvetica
    Neue", Arial; margin:0; min-height:100vh; display:flex; align-items:center;
    justify-content:center; background:linear-gradient(180deg,#071026,#071124 60%);
    color:#e6eef8}

    .app{width:min(920px,95vw); background:linear-gradient(180deg,
    rgba(255,255,255,0.03), rgba(255,255,255,0.01)); border-radius:14px; padding:20px;
    box-shadow:0 8px 30px rgba(2,6,23,0.6)}

    header{display:flex; align-items:center; justify-content:space-between; gap:12px}
    header h1{font-size:18px; margin:0}
    .controls{display:flex; gap:10px; align-items:center}

    .input-row{display:flex; gap:8px; align-items:center}
    input[type="number"]{width:110px; padding:8px 10px; border-radius:8px;
    border:1px solid rgba(255,255,255,0.06); background:transparent; color:inherit}
    label{font-size:13px; color:#cfeffd}

    button{padding:9px 12px; border-radius:9px; border:0; cursor:pointer;
    font-weight:600}
    button:disabled{opacity:0.5; cursor:not-allowed}
    .btn-start{background:linear-gradient(90deg,#06b6d4,#3b82f6); color:#01203a}
    .btn-stop{background:linear-gradient(90deg,#ef4444,#fb7185); color:white}

    .gallery{margin-top:18px; display:grid; grid-template-columns: 1fr 320px; gap:16px}
    .viewer{position:relative; height:420px; border-radius:12px; overflow:hidden;
    background:#061028; display:flex; align-items:center; justify-content:center}

    /* Image stack: animate opacity & transform for a gentle fade/scale */
```

```
.viewer img{position:absolute; inset:0; width:100%; height:100%; object-fit:cover;
opacity:0; transform:scale(1.04); transition:opacity 600ms ease, transform 900ms
cubic-bezier(.2,.9,.3,1)}
```

```
.viewer img.active{opacity:1; transform:scale(1)}
```

```
.sidebar{padding:12px; background:rgba(255,255,255,0.02); border-radius:10px;
height:420px; display:flex; flex-direction:column; gap:12px}
```

```
.meta{font-size:13px}
```

```
.thumbs{display:flex; gap:8px; margin-top:auto}
```

```
.thumbs img{width:64px; height:48px; object-fit:cover; border-radius:6px;
cursor:pointer; opacity:0.8; border:2px solid transparent}
```

```
.thumbs img.sel{border-color:var(--accent); opacity:1}
```

```
footer{margin-top:12px; font-size:13px; color:#9fcfe7}
```

```
/* Small screens */
```

```
@media (max-width:860px){
```

```
.gallery{grid-template-columns:1fr; grid-auto-rows:auto}
```

```
.sidebar{height:auto}
```

```
.viewer{height:300px}
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="app" role="application">
```

```
<header>
```

```
<h1>Image Animation — set interval, start & stop</h1>
```

```
<div class="controls" aria-hidden="false">
```

```
<div class="input-row">
```

```
<label for="interval">Interval (seconds)</label>
```

```
<input id="interval" type="number" min="0.2" step="0.1" value="2">
```

```
</div>
```

```
<button id="startBtn" class="btn-start">Start</button>
```

```
<button id="stopBtn" class="btn-stop" disabled>Stop</button>
```

```
</div>
```

```
</header>
```

```
<div class="gallery">
```

```
<div class="viewer" id="viewer" aria-live="polite" aria-atomic="true">
```

```
<!-- Images will be injected by JS -->
```

```
</div>
```

```
<aside class="sidebar">
```

```
<div class="meta">
```

```
<div><strong>Current:</strong> <span id="currentIndex">—</span> / <span
id="totalCount">—</span></div>
```

```
<div><strong>Interval:</strong> <span id="currentInterval">2.0s</span></div>
```

```
<div style="margin-top:8px; font-size:13px; color:#bfdff4">Tip: Enter a decimal  
value like 0.5 for half a second.</div>  
</div>
```

```
<div style="margin-top:8px;">  
  <label style="font-size:13px">Image list</label>  
  <div class="thumbs" id="thumbs"></div>  
</div>
```

```
<div style="margin-top:12px; font-size:13px; color:#bfe8ff">  
  You can edit the image URLs directly in the script section if you want to use  
  your own images.  
</div>  
</aside>  
</div>
```

```
<footer>  
  Simple image slider with user-controlled interval and start/stop buttons.  
</footer>  
</div>
```

```
<script>  
  const IMAGES = [  
    'https://picsum.photos/seed/pic1/1200/800',  
    'https://picsum.photos/seed/pic2/1200/800',  
    'https://picsum.photos/seed/pic3/1200/800',  
    'https://picsum.photos/seed/pic4/1200/800'  
  ];  
  const viewer = document.getElementById('viewer');  
  const thumbs = document.getElementById('thumbs');  
  const startBtn = document.getElementById('startBtn');  
  const stopBtn = document.getElementById('stopBtn');  
  const intervalInput = document.getElementById('interval');  
  const currentIndexEl = document.getElementById('currentIndex');  
  const totalCountEl = document.getElementById('totalCount');  
  const currentIntervalEl = document.getElementById('currentInterval');  
  
  let imgs = []; // image elements  
  let current = 0;  
  let timer = null;  
  
  function init(){  
    // Build image elements and thumbnails  
    IMAGES.forEach((src, i) => {  
      const img = document.createElement('img');  
      img.src = src;  
      img.alt = `Slide ${i+1}`;  
      if(i===0) img.classList.add('active');
```

```
viewer.appendChild(img);
imgs.push(img);

// thumbs
const t = document.createElement('img');
t.src = src;
t.alt = `Thumb ${i+1}`;
if(i===0) t.classList.add('sel');
t.addEventListener('click', ()=>goTo(i));
thumbs.appendChild(t);
});

totalCountEl.textContent = imgs.length;
currentIndexEl.textContent = current+1;
updateIntervalLabel();

// Event listeners
startBtn.addEventListener('click', start);
stopBtn.addEventListener('click', stop);
intervalInput.addEventListener('change', onIntervalChange);
intervalInput.addEventListener('input', onIntervalChange);

// allow Enter key on interval to start
intervalInput.addEventListener('keydown', (e)=>{
  if(e.key === 'Enter') start();
});

// keyboard controls: space to start/stop, left/right to navigate
window.addEventListener('keydown', (e)=>{
  if(e.code === 'Space'){
    e.preventDefault();
    if(timer) stop(); else start();
  } else if(e.key === 'ArrowRight'){
    stop(); next();
  } else if(e.key === 'ArrowLeft'){
    stop(); prev();
  }
});
}

function updateIntervalLabel(){
  const s = parseFloat(intervalInput.value) || 0;
  currentIntervalEl.textContent = s.toFixed(2) + 's';
}

function onIntervalChange(){
  // sanity: minimum 0.2s
  let v = parseFloat(intervalInput.value);
```

```
if(Number.isNaN(v) || v < 0.2) v = 0.2;
intervallInput.value = v;
updateIntervalLabel();
if(timer){
  // restart timer with new interval
  stop(); start();
}
}

function show(index){
  if(index < 0) index = imgs.length - 1;
  if(index >= imgs.length) index = 0;
  imgs.forEach((im, i)=> im.classList.toggle('active', i===index));
  // update thumbs highlight
  Array.from(thumbs.children).forEach((t, i)=> t.classList.toggle('sel', i===index));
  current = index;
  currentIndexEl.textContent = current+1;
}

function next(){ show(current+1); }
function prev(){ show(current-1); }
function goTo(i){ stop(); show(i); }

function start(){
  // parse interval in seconds
  let seconds = parseFloat(intervallInput.value);
  if(Number.isNaN(seconds) || seconds <= 0){ alert('Please enter a valid interval
(seconds).'); return; }
  // convert to ms
  const ms = Math.max(200, Math.round(seconds*1000));

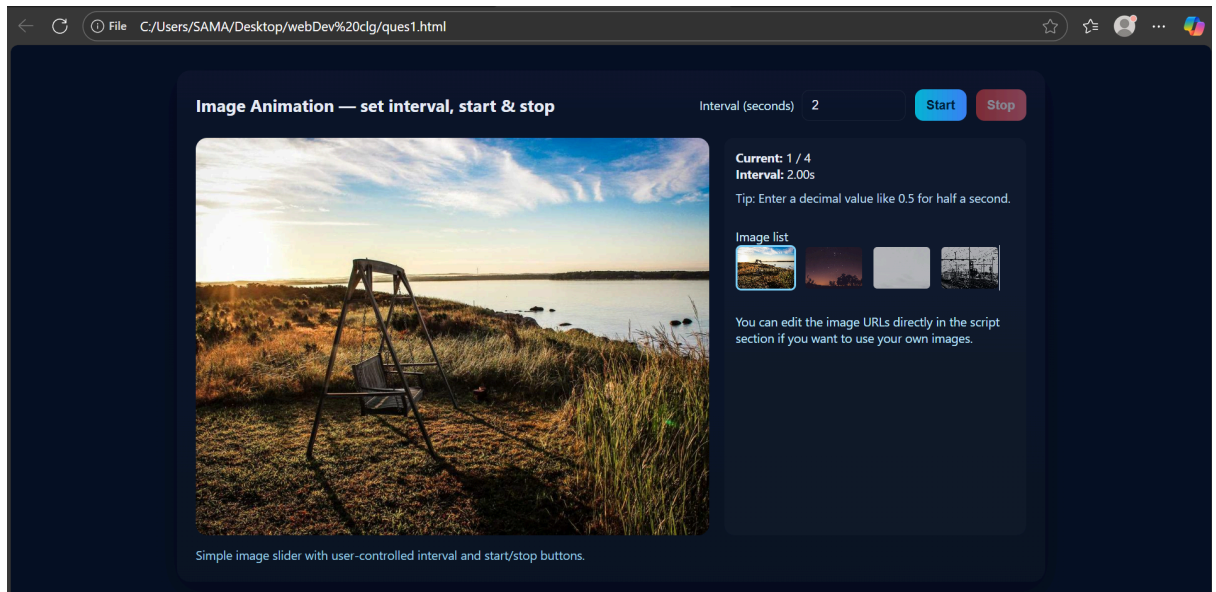
  // disable start button, enable stop
  startBtn.disabled = true;
  stopBtn.disabled = false;
  intervallInput.disabled = true;

  // ensure nothing leftover
  if(timer) clearInterval(timer);
  timer = setInterval(()=>{
    next();
  }, ms);
}

function stop(){
  if(timer) clearInterval(timer);
  timer = null;
  startBtn.disabled = false;
  stopBtn.disabled = true;
}
```

```
    intervalInput.disabled = false;
  }
  // initialize on load
  init();
</script>
</body>
</html>
```

Output:-



Write HTML and JavaScript Program to create simple calculators having following structure and properties, When Focus: Background will be as image, Button hover, first input text is autofocused, result will be on alert and after alert input type set as blank. For currency converters, the first number will be from the first input text and result will be set on other.

Source code:-

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>GCET Simple Calculator</title>
<style>
  body {
    font-family: Arial, sans-serif;
    text-align: center;
    margin-top: 20px;
    transition: background 0.5s;
    background-color: #f4f9f8;
  }

  .header img {
    width: 90%;
    max-width: 900px;
    height: auto;
    display: block;
    margin: 0 auto;
  }

  h1 {
    color: red;
    font-weight: bold;
    margin: 10px 0;
  }

  hr {
    border: 1px solid #ccc;
    width: 90%;
  }

  input {
    width: 250px;
    padding: 8px;
    margin: 6px;
    border-radius: 5px;
    border: 1px solid #ccc;
```

```
}

input:focus {
  background-image:
url('https://images.unsplash.com/photo-1517816428104-797678c7cf0d');
  background-size: cover;
  color: white;
}

.btn {
  background-color: #4CAF50;
  border: none;
  color: white;
  padding: 15px;
  margin: 5px;
  font-size: 16px;
  border-radius: 10px;
  width: 60px;
  cursor: pointer;
  transition: transform 0.2s, background-color 0.2s;
}

.btn:hover {
  background-color: #45a049;
  transform: scale(1.1);
}

.calculator {
  display: inline-block;
  background-color: #e6f7f5;
  padding: 20px;
  border-radius: 15px;
  box-shadow: 0 0 10px rgba(0,0,0,0.2);
}

.row {
  display: flex;
  justify-content: center;
}
</style>
</head>
<body>

<!-- Header Logo -->
<div class="header">
  
</div>
```



```
<h1>!!! GCET !!!</h1>
<hr>

<!-- Calculator Section -->
<div class="calculator">
  <input type="number" id="num1" placeholder="Enter First Number"
autofocus><br>
  <input type="number" id="num2" placeholder="Enter Second Number"><br>

  <div class="row">
    <button class="btn" onclick="calc('+')">+</button>
    <button class="btn" onclick="calc('-')">-</button>
    <button class="btn" onclick="calc('*')">*</button>
    <button class="btn" onclick="calc('/')">/</button>
  </div>
  <div class="row">
    <button class="btn" onclick="calc('%')">%</button>
    <button class="btn" onclick="square()">x2</button>
    <button class="btn" onclick="power()">xy</button>
    <button class="btn" onclick="sqrt()">√</button>
  </div>
  <div class="row">
    <button class="btn" onclick="logVal()">log</button>
    <button class="btn" onclick="sinVal()">sin</button>
    <button class="btn" onclick="currency('INR')">₹</button>
    <button class="btn" onclick="currency('USD')">$</button>
  </div>
</div>

<script>
const num1 = document.getElementById("num1");
const num2 = document.getElementById("num2");

function getValues() {
  const a = parseFloat(num1.value);
  const b = parseFloat(num2.value);
  return {a, b};
}

function clearInputs() {
  num1.value = "";
  num2.value = "";
  num1.focus();
}

function calc(op) {
  let {a, b} = getValues();
  let result;
```

```
if (isNaN(a) || isNaN(b)) {
    alert("Please enter both numbers!");
    return;
}

switch (op) {
    case '+': result = a + b; break;
    case '-': result = a - b; break;
    case '*': result = a * b; break;
    case '/': result = b !== 0 ? a / b : "Cannot divide by zero"; break;
    case '%': result = a % b; break;
}
alert("Result = " + result);
clearInputs();
}

function square() {
    let {a} = getValues();
    if (isNaN(a)) { alert("Enter a number in first input!"); return; }
    alert("Result = " + (a * a));
    clearInputs();
}

function power() {
    let {a, b} = getValues();
    if (isNaN(a) || isNaN(b)) { alert("Enter both numbers!"); return; }
    alert("Result = " + Math.pow(a, b));
    clearInputs();
}

function sqrt() {
    let {a} = getValues();
    if (isNaN(a)) { alert("Enter first number!"); return; }
    alert("√" + a + " = " + Math.sqrt(a));
    clearInputs();
}

function logVal() {
    let {a} = getValues();
    if (isNaN(a)) { alert("Enter first number!"); return; }
    alert("log(" + a + ") = " + Math.log10(a));
    clearInputs();
}

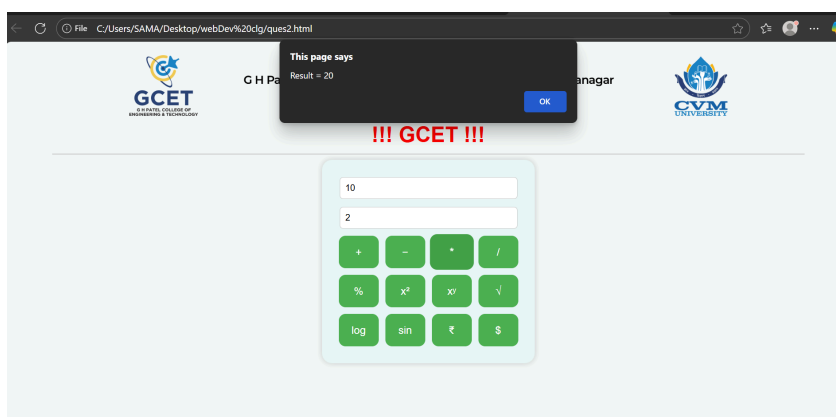
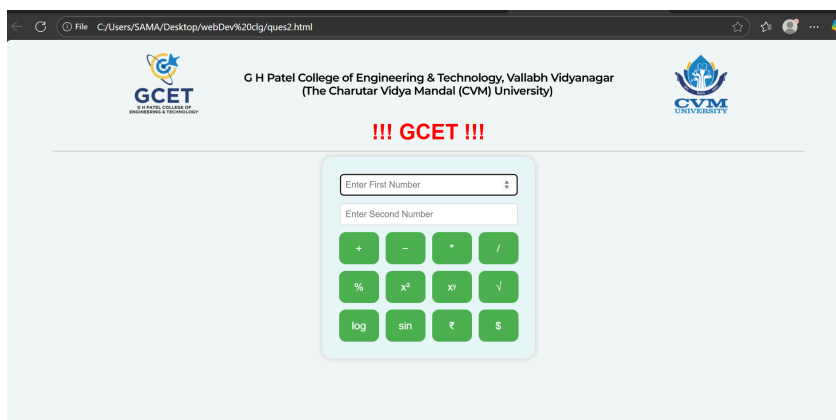
function sinVal() {
    let {a} = getValues();
    if (isNaN(a)) { alert("Enter first number!"); return; }
    alert("sin(" + a + ") = " + Math.sin(a * Math.PI / 180).toFixed(3));
}
```

```
clearInputs();
}

function currency(type) {
  let {a} = getValues();
  if (isNaN(a)) { alert("Enter first number (amount)!"); return; }

  // Example conversion rates
  let converted;
  if (type === 'INR') {
    converted = a * 83; // USD to INR
    num2.value = converted.toFixed(2);
  } else if (type === 'USD') {
    converted = a / 83; // INR to USD
    num2.value = converted.toFixed(2);
  }
}
</script>
</body>
</html>
```

Output:-



Design an HTML, CSS and JavaScript form to prepare the result of 12th Standard. Make it responsive and put all the required validations. Attach the snapshots for the same.

Source code:-

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>12th Standard Result Form</title>

<style>
  body {
    font-family: "Poppins", sans-serif;
    background: linear-gradient(120deg, #a1c4fd, #c2e9fb);
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    margin: 0;
  }

  .container {
    background: white;
    padding: 25px;
    border-radius: 15px;
    box-shadow: 0 4px 15px rgba(0,0,0,0.2);
    width: 90%;
    max-width: 500px;
  }

  h2 {
    text-align: center;
    color: #004aad;
    margin-bottom: 20px;
  }

  label {
    font-weight: bold;
    display: block;
    margin-bottom: 5px;
  }

  input, select {
    width: 100%;
    padding: 8px;
```

```
    margin-bottom: 15px;
    border: 1px solid #ccc;
    border-radius: 8px;
    font-size: 14px;
}

input:focus {
    border-color: #004aad;
    background-color: #f0f8ff;
    outline: none;
}

button {
    width: 48%;
    background: #004aad;
    color: white;
    border: none;
    padding: 10px;
    border-radius: 8px;
    font-size: 16px;
    cursor: pointer;
    transition: 0.3s;
}

button:hover {
    background: #007bff;
}

.btn-group {
    display: flex;
    justify-content: space-between;
}

.result {
    margin-top: 20px;
    text-align: center;
    background: #f8f9fa;
    padding: 15px;
    border-radius: 10px;
    border: 1px solid #ccc;
}

@media (max-width: 500px) {
    button {
        width: 100%;
        margin-bottom: 10px;
    }
}
```

```
.btn-group {
    flex-direction: column;
}
}
</style>
</head>

<body>
<div class="container">
    <h2>12th Standard Result Form</h2>
    <form id="resultForm">
        <label>Student Name:</label>
        <input type="text" id="name" placeholder="Enter full name" required>

        <label>Roll Number:</label>
        <input type="text" id="roll" placeholder="Enter roll number" required
pattern="[0-9]{3,}">

        <label>Stream:</label>
        <select id="stream" required>
            <option value="">Select Stream</option>
            <option value="Science">Science</option>
            <option value="Commerce">Commerce</option>
            <option value="Arts">Arts</option>
        </select>

        <label>Marks (out of 100):</label>
        <input type="number" id="sub1" placeholder="Subject 1" required min="0"
max="100">
        <input type="number" id="sub2" placeholder="Subject 2" required min="0"
max="100">
        <input type="number" id="sub3" placeholder="Subject 3" required min="0"
max="100">
        <input type="number" id="sub4" placeholder="Subject 4" required min="0"
max="100">
        <input type="number" id="sub5" placeholder="Subject 5" required min="0"
max="100">

        <div class="btn-group">
            <button type="button" onclick="calculateResult()">Calculate</button>
            <button type="reset">Reset</button>
        </div>
    </form>

    <div id="resultBox" class="result" style="display:none;">
        <h3>Result Summary</h3>
        <p id="resName"></p>
        <p id="resRoll"></p>
```

```
<p id="resStream"></p>
<p id="resTotal"></p>
<p id="resPercent"></p>
<p id="resGrade"></p>
</div>
</div>

<script>
function calculateResult() {
  let name = document.getElementById("name").value.trim();
  let roll = document.getElementById("roll").value.trim();
  let stream = document.getElementById("stream").value;
  let sub1 = parseFloat(document.getElementById("sub1").value);
  let sub2 = parseFloat(document.getElementById("sub2").value);
  let sub3 = parseFloat(document.getElementById("sub3").value);
  let sub4 = parseFloat(document.getElementById("sub4").value);
  let sub5 = parseFloat(document.getElementById("sub5").value);

  if (!name || !roll || !stream || isNaN(sub1) || isNaN(sub2) || isNaN(sub3) ||
  isNaN(sub4) || isNaN(sub5)) {
    alert(" Please fill all fields correctly!");
    return;
  }

  if ([sub1, sub2, sub3, sub4, sub5].some(m => m < 0 || m > 100)) {
    alert(" Marks should be between 0 and 100!");
    return;
  }

  let total = sub1 + sub2 + sub3 + sub4 + sub5;
  let percent = (total / 500 * 100).toFixed(2);
  let grade = "";

  if (percent >= 90) grade = "A+";
  else if (percent >= 75) grade = "A";
  else if (percent >= 60) grade = "B";
  else if (percent >= 45) grade = "C";
  else if (percent >= 33) grade = "D";
  else grade = "Fail";

  alert(` ${name}'s Result\nTotal: ${total}/500\nPercentage: ${percent}%\nGrade:
  ${grade}`);
  document.getElementById("resultForm").reset();
}
</script>
</body>
</html>
```

Output:-

The first screenshot shows a web browser window with the address bar displaying 'C:/Users/SAMA/Desktop/webDev%20clg/ques3.html'. The page contains a form titled '12th Standard Result Form'. The form fields are: Student Name (Palak Behl), Roll Number (B250396), Stream (Science), and Marks (out of 100) (89, 98, 90, 95, 82). There are 'Calculate' and 'Reset' buttons at the bottom.

The second screenshot shows the same form after the 'Calculate' button is clicked. A dark overlay box appears with the following text: 'This page says', 'Palak Behl's Result', 'Total: 454/500', 'Percentage: 90.80%', 'Grade: A+', and an 'OK' button. The form fields and buttons remain visible in the background.