

## Lab Cycle 2

**CO2: Implement client-side validation techniques using scripting languages.**

### PROGRAM 6:

Create an HTML page to demonstrate the use of various predefined functions in a string and math object in java script.

### PROGRAM CODE:

#### Functions.html

```
<html>
<head>
  <title>JavaScript Functions</title>
</style>
  body {
    font-family: sans-serif;
    padding: 20px;
  }
  .container {
    max-width: 600px;
    margin: auto;
    border: 1px solid #ccc;
    padding: 20px;
    background-color: #e3e3e3;
    border-radius: 8px;
  }
  input[type="text"] {
    width: 95%;
    padding: 8px;
    margin-bottom: 10px;
  }
  button {
    margin: 5px;
    padding: 8px 12px;
    cursor: pointer;
    background-color: #dbffc8;
    border: 1px solid #4CAF50;
  }
  #result, #result2 {
    font-weight: bold;
    color: #0056b3;
```

```
}
h2 {
  border-bottom: 1px solid #eee;
  padding-bottom: 5px;
}
h1 {
  text-align: center;
}
section {
  display: flex;
  justify-content: space-around;
  flex-wrap: wrap;
  gap: 20px;
}
</style>
</head>
<body>

<h1>String and Math Functions</h1>

<section>
  <div class="container">

    <label for="inputBox">Enter text:</label>
    <input type="text" id="inputBox" placeholder="Your input here...">

    <p>Result: <span id="result"></span></p>

    <h2>String Functions</h2>
    <button onclick="getStringLength()">Length</button>
    <button onclick="convertToUpperCase()">To Upper Case</button>
    <button onclick="convertToLowerCase()">To Lower Case</button>
    <button onclick="getCharacterAt()">Character At Index 2</button>
    <button onclick="sliceString()">Slice (0 to 4)</button>
    <button onclick="replaceText()">Replace 'haha' with 'hehee'</button>
  </div>

  <div class="container">

    <label for="inputBox">Enter a number:</label>
    <input type="text" id="inputBox2" placeholder="Your input here...">
```

```
<p>Result: <span id="result2"></span></p>

<h2>Math Functions (Input should be a number)</h2>
<button onclick="getSquareRoot()">Square Root</button>
<button onclick="roundNumber()">Round</button>
<button onclick="getCeil()">Ceiling</button>
<button onclick="getFloor()">Floor</button>
<button onclick="getAbsolute()">Absolute Value</button>
<button onclick="showRandom()">Random Number</button>
</div>
</section>

<script>
function getInputElement(box) {
    return document.getElementById(box);
}

function getResultElement(resultBox) {
    return document.getElementById(resultBox);
}

function getStringLength() {
    var input = getInputElement('inputBox').value;
    getResultElement('result').innerText = "Length is: " + input.length;
}

function convertToUpperCase() {
    var input = getInputElement('inputBox').value;
    getResultElement('result').innerText = input.toUpperCase();
}

function convertToLowerCase() {
    var input = getInputElement('inputBox').value;
    getResultElement('result').innerText = input.toLowerCase();
}

function getCharacterAt() {
    var input = getInputElement('inputBox').value;
    if (input.length > 2) {
        getResultElement('result').innerText = "Character at index 2 is: " + input.charAt(2);
    }
}
```

```
    } else {
        getResultElement('result').innerText = "Input is too short.";
    }
}

function sliceString() {
    var input = getInputElement('inputBox').value;
    getResultElement('result').innerText = "Sliced string is: " + input.slice(0, 4);
}

function replaceText() {
    var input = getInputElement('inputBox').value;
    getResultElement('result').innerText = input.replace("haha", "hehee");
}

function getSquareRoot() {
    var input = getInputElement('inputBox2').value;
    var number = parseFloat(input);
    if (isNaN(number)) {
        getResultElement('result2').innerText = "Please enter a valid number.";
    } else {
        getResultElement('result2').innerText = "Square root is: " + Math.sqrt(number);
    }
}

function roundNumber() {
    var input = getInputElement('inputBox2').value;
    var number = parseFloat(input);
    if (isNaN(number)) {
        getResultElement('result2').innerText = "Please enter a valid number.";
    } else {
        getResultElement('result2').innerText = "Rounded value is: " + Math.round(number);
    }
}

function getCeil() {
    var input = getInputElement('inputBox2').value;
    var number = parseFloat(input);
    if (isNaN(number)) {
        getResultElement('result2').innerText = "Please enter a valid number.";
    } else {
```

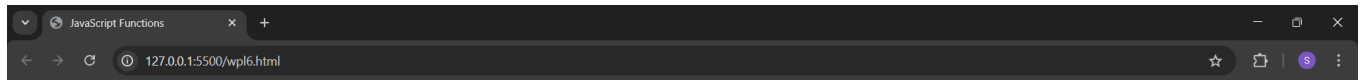
```
        getResultElement('result2').innerText = "Ceiling value is: " + Math.ceil(number);
    }
}

function getFloor() {
    var input = getInputElement('inputBox2').value;
    var number = parseFloat(input);
    if (isNaN(number)) {
        getResultElement('result2').innerText = "Please enter a valid number.";
    } else {
        getResultElement('result2').innerText = "Floor value is: " + Math.floor(number);
    }
}

function getAbsolute() {
    var input = getInputElement('inputBox2').value;
    var number = parseFloat(input);
    if (isNaN(number)) {
        getResultElement('result2').innerText = "Please enter a valid number.";
    } else {
        getResultElement('result2').innerText = "Absolute value is: " + Math.abs(number);
    }
}

function showRandom() {
    getResultElement('result2').innerText = "Random number (0-1): " + Math.random();
}
</script>

</body>
</html>
```

**OUTPUT:****String and Math Functions**

Enter text:

Result: **hehee world**

**String Functions**

Enter a number:

Result: **Square root is: 5**

**Math Functions (Input should be a number)**

**PROGRAM 7:**

Generate a calendar using JavaScript code by getting the year and month from the user.

**PROGRAM CODE:****calender.html**

```
<html>
<head>

<title>Calendar</title>
<style>
  body {
    font-family: Arial, Helvetica, sans-serif;
    background: #ffffff;
    color: #000000;
    padding: 20px;
  }

  .controls {
    max-width: 420px;
    border: 2px solid #000;
    padding: 10px;
    background: #f9f9f9;
  }

  .controls label {
    display: block;
    margin-bottom: 6px;
    font-weight: bold;
  }

  .controls input[type="number"] {
    width: 100%;
    padding: 6px;
    font-size: 16px;
    border: 1px solid #000;
    margin-bottom: 8px;
    box-sizing: border-box;
    background: #ffffff;
  }
```

```
.controls .buttons {
  margin-top: 6px;
}

button {
  padding: 6px 10px;
  font-size: 16px;
  border: 1px solid #000;
  background: #e0e0e0;
  cursor: pointer;
  margin-right: 6px;
}

.calendar {
  max-width: 420px;
  margin-top: 14px;
  border: 2px solid #000;
  padding: 6px;
  background: #fff;
}

table {
  width: 100%;
  border-collapse: collapse;
  font-size: 14px;
}

th, td {
  border: 1px solid #000;
  padding: 8px;
  text-align: center;
}

th {
  background: #dddddd;
  font-weight: bold;
}

td.inactive {
  color: #888888;
```

```
    background: #fafafa;
  }

  td.today {
    background: #ffff99;
  }

  .cal-header {
    margin-bottom: 8px;
    font-weight: bold;
    text-align: left;
  }

  .footer {
    margin-top: 8px;
    font-size: 13px;
  }
</style>
</head>
<body>
  <div class="controls">
    <label>
      Year
      <input id="yearInput" type="number" min="1" max="9999" value="" />
    </label>
    <label>
      Month (1-12)
      <input id="monthInput" type="number" min="1" max="12" value="" />
    </label>
    <div class="buttons">
      <button id="showBtn">Show</button>
    </div>
  </div>

  <div id="calendarContainer" class="calendar"></div>

  <script>
    var yearInput = document.getElementById('yearInput');
    var monthInput = document.getElementById('monthInput');
    var showBtn = document.getElementById('showBtn');
    var container = document.getElementById('calendarContainer');
```

```
var monthNames =
["January","February","March","April","May","June","July","August","September","October","November","December"];
var dayNames = ["Sun","Mon","Tue","Wed","Thu","Fri","Sat"];

function renderCalendar(year, month) {
    year = parseInt(year, 10);
    month = parseInt(month, 10);

    if (isNaN(year) || isNaN(month) || month < 1 || month > 12) {
        container.innerHTML = '<div class="cal-header">Invalid year or month</div>';
        return;
    }

    var firstDayIndex = new Date(year, month - 1, 1).getDay();
    var daysInMonth = new Date(year, month, 0).getDate();
    var prevMonthDays = new Date(year, month - 1, 0).getDate();

    var html = "";
    html += '<div class="cal-header">' + monthNames[month - 1] + ' ' + year + '</div>';
    html += '<table>';
    html += '<thead><tr>';
    for (var d = 0; d < 7; d++) {
        html += '<th>' + dayNames[d] + '</th>';
    }
    html += '</tr></thead><tbody>';

    var day = 1;
    var started = false;

    for (var week = 0; week < 6; week++) {
        if (started && day > daysInMonth) {
            break;
        }

        html += '<tr>';
        for (var dow = 0; dow < 7; dow++) {
            var cell = "";
            if (!started && dow === firstDayIndex) {
                started = true;
            }
        }
    }
}
```

```
    }

    if (!started) {
        var num = prevMonthDays - (firstDayIndex - 1 - dow);
        cell = '<td class="inactive">' + num + '</td>';
    } else if (day > daysInMonth) {
        var nextDay = day - daysInMonth;
        cell = '<td class="inactive">' + nextDay + '</td>';
        day++;
    } else {
        cell = '<td>' + day + '</td>';
        day++;
    }
    html += cell;
}
html += '</tr>';
}

html += '</tbody></table>';
container.innerHTML = html;
}

function showFromInputs() {
    var now = new Date();
    var defaultYear = now.getFullYear();
    var defaultMonth = now.getMonth() + 1;

    var y = yearInput.value || defaultYear;
    var m = monthInput.value || defaultMonth;
    renderCalendar(y, m);
}

var now = new Date();
yearInput.value = now.getFullYear();
monthInput.value = now.getMonth() + 1;
renderCalendar(yearInput.value, monthInput.value);

showBtn.addEventListener('click', showFromInputs);

yearInput.addEventListener('keydown', function (e) {
    if (e.key === 'Enter') {
```

```
        showFromInputs();
    }
});
monthInput.addEventListener('keydown', function (e) {
    if (e.key === 'Enter') {
        showFromInputs();
    }
});

monthInput.addEventListener('input', function () {
    var m = parseInt(monthInput.value, 10);
    if (isNaN(m)) return;
    if (m < 1) monthInput.value = 1;
    if (m > 12) monthInput.value = 12;
});

yearInput.addEventListener('input', function () {
    var y = parseInt(yearInput.value, 10);
    if (isNaN(y)) return;
    if (y < 1) yearInput.value = 1;
    if (y > 9999) yearInput.value = 9999;
});
</script>
</body>
</html>
```

**OUTPUT:**

Year  
2024

Month (1-12)  
12

Show

December 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

**PROGRAM 8:**

Create a program using the JavaScript Event Listener to add multiple events on the same button.

**PROGRAM CODE:****btnEvents.html**

```
<html>
<head>
  <title>Js Multiple Events</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 2rem; background: #fff; color: #111; }
    h1 { font-size: 1.1rem; margin-bottom: 0.5rem; }
    #btn { padding: 0.5rem 0.8rem; margin-bottom: 0.5rem; }
  </style>
</head>
<body>
  <div class="container">
    <h1>Multiple events on button</h1>

    <button id="btn" type="button">Press Me</button>
  </div>

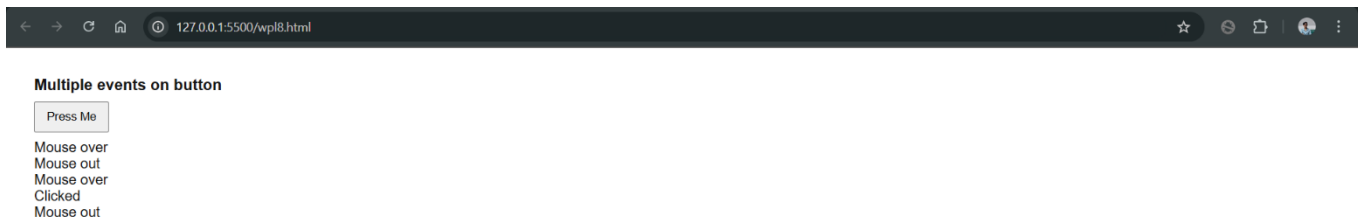
  <script>
    const btn = document.getElementById('btn');
```

```
btn.addEventListener('mouseover', mOn);
btn.addEventListener('click', mClick);
btn.addEventListener('mouseout', mMiss);

function appendEntry(text) {
  const tn = document.createTextNode(text);
  document.body.appendChild(tn);
  document.body.appendChild(document.createElement('br'));
}

function mOn() {
  appendEntry('Mouse over');
}
function mClick() {
  appendEntry('Clicked');
}
function mMiss() {
  appendEntry('Mouse out');
}
</script>
</body>
</html>
```

**OUTPUT:**



### PROGRAM 9:

Develop a JavaScript Event Handling mechanism to change the background colour of an HTML page.

### PROGRAM CODE:

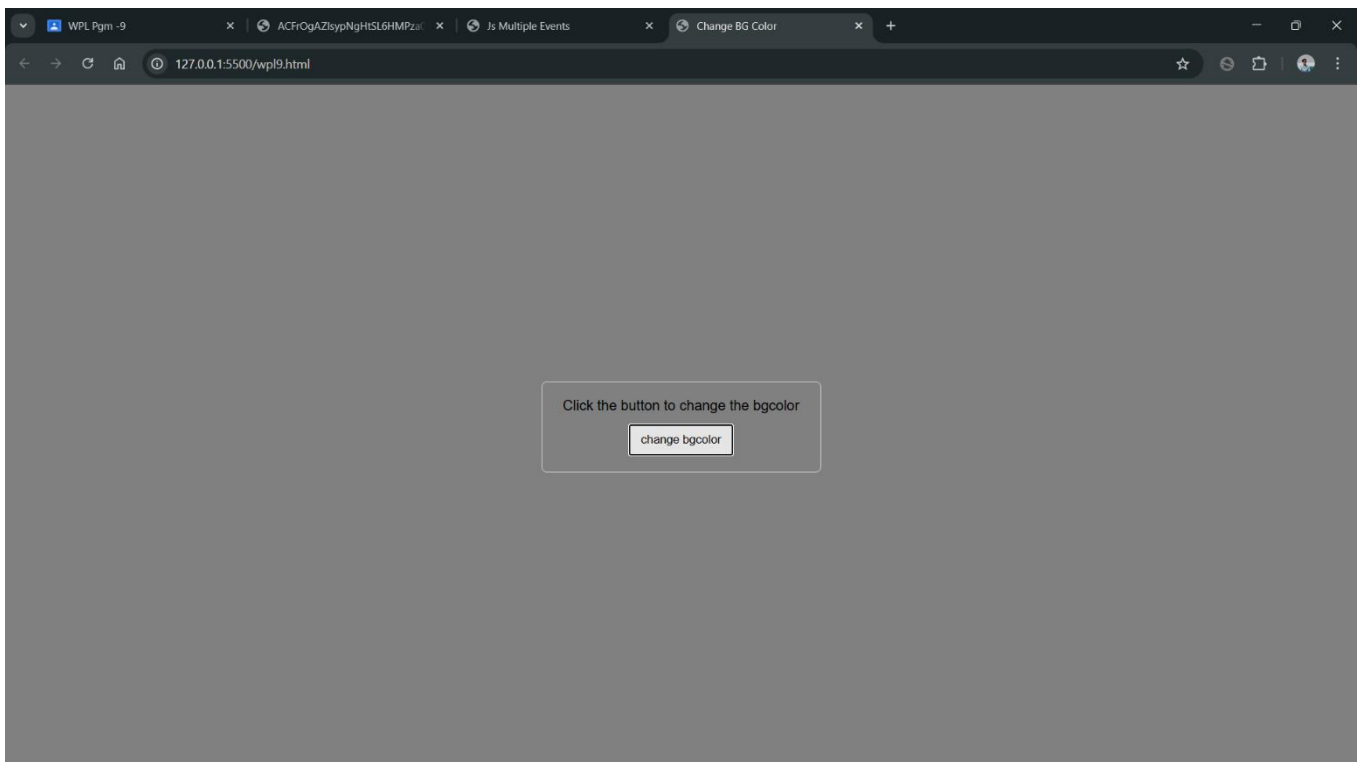
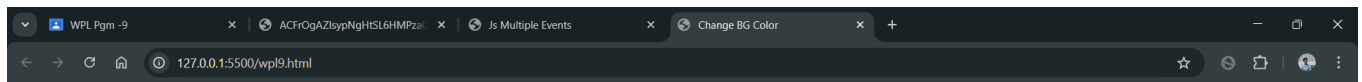
#### bgColor.html

```
<html>
<head>
  <title>Change BG Color</title>
  <style>
    body {
      font-family: sans-serif;
```

```
margin: 0;
background: white;
display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
min-height: 100vh;
}
.box {
width: 280px;
padding: 18px;
border: 1px solid #ccc;
border-radius: 6px;
text-align: center;
}
button {
margin-top: 12px;
padding: 8px 12px;
cursor: pointer;
}
</style>
</head>
<body>
<div class="box">
<div>Click the button to change the bgcolor</div>
<button id="changeBtn">change bgcolor</button>
</div>

<script>
document.getElementById('changeBtn').addEventListener('click', function () {
document.body.style.backgroundColor = 'grey';
});
</script>
</body>
</html>
```

**OUTPUT:**

**PROGRAM 10:**

Create an HTML page to display a new image and text when the mouse comes over the existing content in the page using JavaScript Event Handling mechanism.

**PROGRAM CODE:**

**index.html**

<html>

```
<head>
  <title>Mouse Over Image</title>
  <style>
    body { font-family: Arial, sans-serif; padding: 30px; background: #f7f7f7; }
    #card { width: 320px; padding: 10px; background: #fff; border: 1px solid #ddd; border-
radius: 6px; text-align: center; cursor: pointer; }
    #card img { max-width: 100%; height: auto; display: block; margin: 0 auto 10px; }
    #caption { color: #333; margin: 0; }
  </style>
</head>
<body>
  <h2>Simple Mouseover Events</h2>

  <div id="card" aria-label="Interactive card">
    
    <p id="caption">This is the original image. Move your mouse over the box.</p>
  </div>

  <script>
    var card = document.getElementById('card');
    var pic = document.getElementById('pic');
    var caption = document.getElementById('caption');

    var originalSrc = pic.src;
    var originalText = caption.textContent;

    var hoverSrc = "./new.png";
    var hoverText = "Nice, You hovered over it, here's new text and image.";

    card.addEventListener('mouseover', function () {
      pic.src = hoverSrc;
      caption.textContent = hoverText;
    });

    card.addEventListener('mouseout', function () {
      pic.src = originalSrc;
      caption.textContent = originalText;
    });
  </script>
</body>
</html>
```

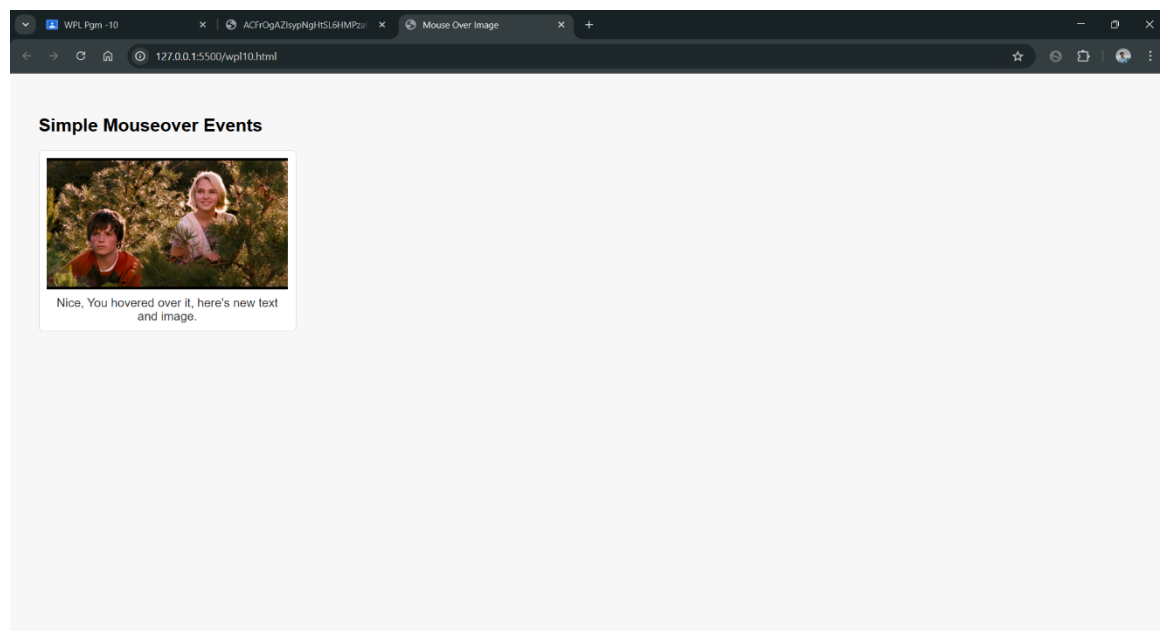
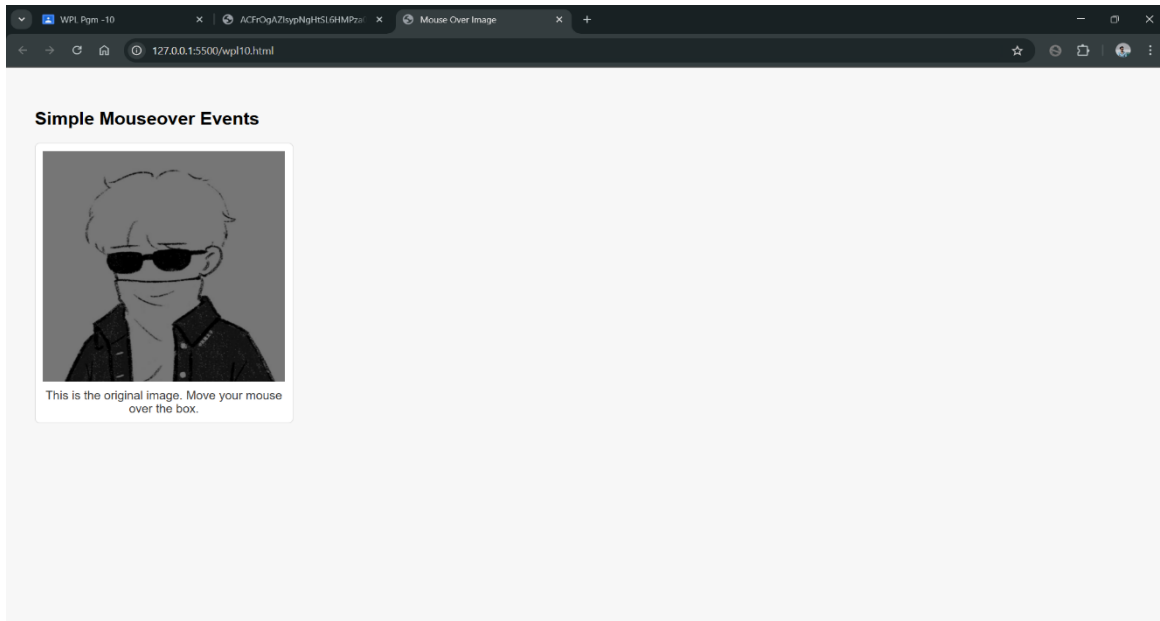
**relative.css:**

```
.box-container { border:
3px solid #0066cc;
padding: 20px; margin:
40px; height: 300px;
background-color: #f0f8ff;
} .box { width: 150px;
height: 100px; background-
color: #4682b4; color:
white; text-align: center;
font-size: 1.2em; margin:
10px;
}
.relative-box { position:
relative; right: 40px;
bottom: 20px; background-
color: #ff6347; z-index: 10;
}
.normal-box { background-color:
#ccc;
color: #333;
}
```

**absolute.css**

```
.box { width: 150px; height:
100px; background-color:
#4682b4; color: white; text-
align: center; line-height:
100px; font-size: 1.2em;
margin: 10px;
}
.absolute-container {
position: relative; border:
3px solid #556b2f;
padding: 20px; margin:
40px; height: 300px;
background-color: #f0fff0;
}
```

```
.absolute-box { position:
absolute; right: 30px;
bottom: 30px; background-
color: #da70d6;
z-index: 10; margin:
0;
}
```

**OUTPUT:****PROGRAM 11:**

Create a HTML registration form and validate the form using JavaScript code. The validation should include the following:

1. Name validation/Mandatory field validation
2. password validation

## 3. Email validation

## 4. Mobile number validation

**PROGRAM CODE:****form.html**

```

<html>
<head>
  <title>Registration Form</title>
  <style>
    body { font-family: Arial, sans-serif; background:#f7f7f7; padding:20px; }
    .container { max-width:420px; margin:40px auto; padding:20px; border: 1px solid
black}
    h2 { margin-top:0; font-weight:600; }
    .field { margin-bottom:14px; }
    label { display:block; font-size:14px; margin-bottom:6px; }
    input[type="text"], input[type="email"], input[type="password"], input[type="tel"] {
      width:100%; padding:8px 10px; border:1px solid #ccc; box-sizing:border-box;
    }
    .error { color:#b00020; font-size:13px; margin-top:6px; display:none; }
    .success { color:green; font-size:14px; margin-top:8px; display:none; }
    button { padding:10px 14px; border:0; background:#0078d4; color:#fff; cursor:pointer;
  }
    button:disabled { opacity:0.6; cursor:not-allowed; }
  </style>
</head>
<body>
  <div class="container">
    <h2>Register</h2>
    <form id="regForm" novalidate>
      <div class="field">
        <label for="name">Full name</label>
        <input id="name" name="name" type="text" autocomplete="name" />
        <div id="nameErr" class="error">Please enter your name (letters and spaces
only).</div>
      </div>

      <div class="field">
        <label for="email">Email</label>
        <input id="email" name="email" type="email" autocomplete="email" />
        <div id="emailErr" class="error">Please enter a valid email address.</div>
      </div>

      <div class="field">

```

```
<label for="mobile">Mobile number</label>
<input id="mobile" name="mobile" type="tel" inputmode="numeric"
placeholder="10 digits" />
<div id="mobileErr" class="error">Please enter a valid 10-digit mobile
number.</div>
</div>

<div class="field">
  <label for="password">Password</label>
  <input id="password" name="password" type="password" autocomplete="new-
password" />
  <div id="passwordErr" class="error">Password must be at least 8 chars, include
upper and lower case letters and a number.</div>
</div>

<div style="margin-top:12px;">
  <button type="submit" id="submitBtn">Create account</button>
</div>

<div id="formSuccess" class="success">Registration successful!</div>
</form>
</div>

<script>
const form = document.getElementById('regForm');
const nameInput = document.getElementById('name');
const emailInput = document.getElementById('email');
const mobileInput = document.getElementById('mobile');
const passwordInput = document.getElementById('password');

const nameErr = document.getElementById('nameErr');
const emailErr = document.getElementById('emailErr');
const mobileErr = document.getElementById('mobileErr');
const passwordErr = document.getElementById('passwordErr');
const formSuccess = document.getElementById('formSuccess');

function showError(el, msgEl) {
  el.setAttribute('aria-invalid', 'true');
  msgEl.style.display = 'block';
}

function clearError(el, msgEl) {
  el.removeAttribute('aria-invalid');
  msgEl.style.display = 'none';
}
```

```
}

function validateName() {
  const val = nameInput.value.trim();
  const ok = /^[A-Za-z ]{2,}$/.test(val);
  if (!ok) showError(nameInput, nameErr); else clearError(nameInput, nameErr);
  return ok;
}

function validateEmail() {
  const val = emailInput.value.trim();
  const ok = /^[^\s@]+@[^\s@]+\.[^\s@]{2,}$/.test(val);
  if (!ok) showError(emailInput, emailErr); else clearError(emailInput, emailErr);
  return ok;
}

function validateMobile() {
  const val = mobileInput.value.replace(/\D/g, "");
  const ok = /^\d{10}$/.test(val);
  if (!ok) showError(mobileInput, mobileErr); else clearError(mobileInput, mobileErr);
  return ok;
}

function validatePassword() {
  const val = passwordInput.value;
  const ok = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d){8,}$/.test(val);
  if (!ok) showError(passwordInput, passwordErr); else clearError(passwordInput,
passwordErr);
  return ok;
}

nameInput.addEventListener('input', validateName);
emailInput.addEventListener('input', validateEmail);
mobileInput.addEventListener('input', validateMobile);
passwordInput.addEventListener('input', validatePassword);

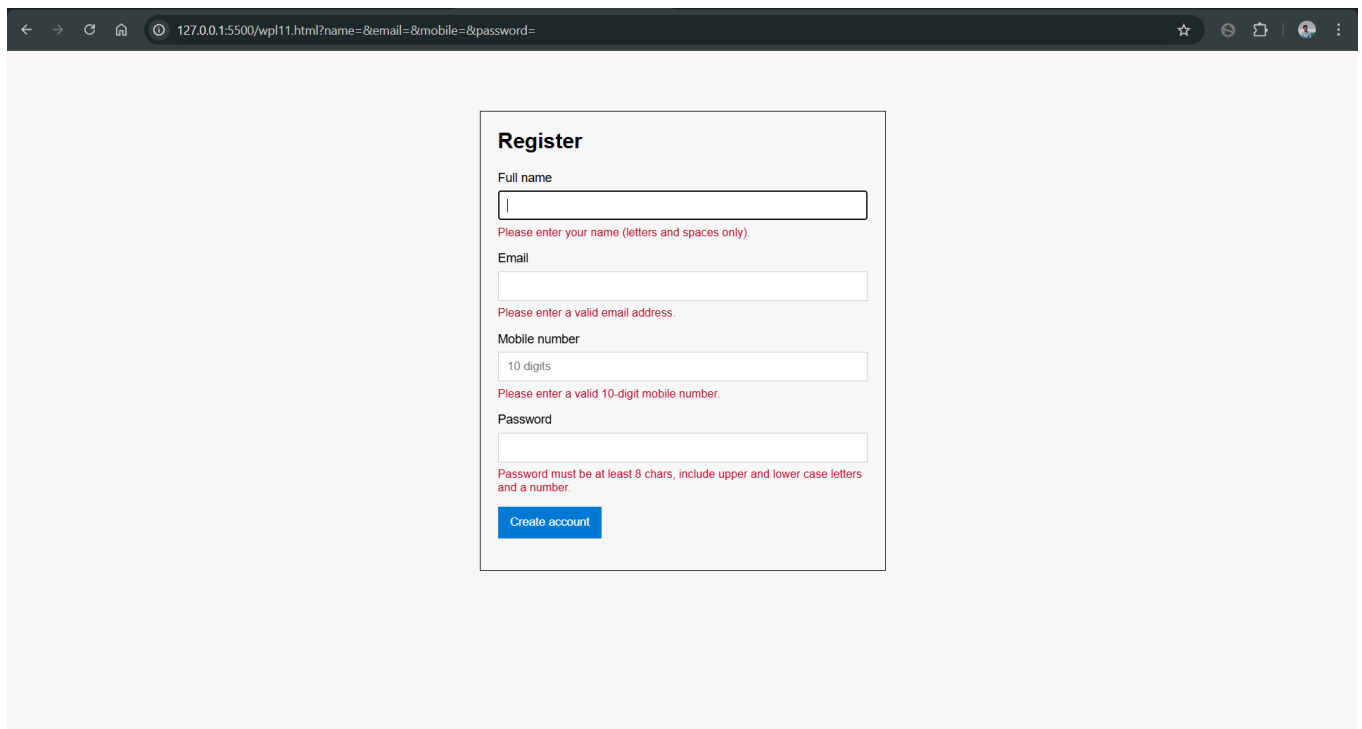
form.addEventListener('submit', function (e) {
  e.preventDefault();
  formSuccess.style.display = 'none';

  const okName = validateName();
  const okEmail = validateEmail();
  const okMobile = validateMobile();
```

```
const okPassword = validatePassword();

if (okName && okEmail && okMobile && okPassword) {
  formSuccess.style.display = 'block';
  form.reset();
  [nameErr, emailErr, mobileErr, passwordErr].forEach(el => el.style.display = 'none');
} else {
  const firstInvalid = form.querySelector('[aria-invalid="true"]');
  if (firstInvalid) firstInvalid.focus();
}
});
</script>
</body>
</html>
```

## OUTPUT:



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5500/wpl11.html?name=&email=&mobile=&password=". The main content area features a registration form titled "Register". The form contains four input fields: "Full name", "Email", "Mobile number", and "Password". Each field has a corresponding error message in red text below it: "Please enter your name (letters and spaces only).", "Please enter a valid email address.", "Please enter a valid 10-digit mobile number.", and "Password must be at least 8 chars, include upper and lower case letters and a number.". At the bottom of the form is a blue button labeled "Create account".

**PROGRAM 12:**

Design and develop a complete online exam using Html, CSS and JavaScript. Minimum 4-5 questions should be in the online exam, and the results should be displayed on completion.

**PROGRAM CODE:****exam.html**

```
<html>
<head>
  <title>Online Exam</title>
  <style>
    body {
      font-family: Arial, Helvetica, sans-serif;
      background: #f0f4f8;
      margin: 0;
      padding: 20px;
      color: #222;
    }

    .wrap {
      max-width: 800px;
      margin: 30px auto;
      padding: 10px;
    }

    .card {
      background: #fff;
      border: 1px solid #ccc;
      border-radius: 6px;
      padding: 20px;
    }

    header {
      margin-bottom: 12px;
    }
    h1 {
      font-size: 18px;
      margin: 0;
```

```
    color: #111;
}

.questions {
    margin-top: 10px;
}

.q {
    padding: 10px;
    border: 1px solid #e0e0e0;
    border-radius: 4px;
    margin-bottom: 10px;
    background: #fafafa;
}

.q h3 {
    font-size: 14px;
    margin: 0 0 8px 0;
}

.options label {
    display: block;
    padding: 6px;
    margin-bottom: 6px;
    border-radius: 4px;
    cursor: pointer;
    background: transparent;
    border: 1px solid transparent;
}

.options label:hover {
    background: #fff;
    border-color: #e6e6e6;
}

.options input {
    margin-right: 8px;
}

.controls {
    margin-top: 12px;
}
```

```
button {
  padding: 8px 12px;
  border: 0;
  background: #007bff;
  color: #fff;
  border-radius: 4px;
  cursor: pointer;
  font-weight: 600;
}
button.secondary,
button[disabled] {
  background: #e9ecef;
  color: #6c757d;
  border: 1px solid #ccc;
  cursor: default;
}
</style>
</head>
<body>
<div class="wrap">
  <div class="card" id="examCard">
    <header>
      <div>
        <h1>Online Exam</h1>
      </div>
    </header>

    <form id="examForm" class="questions" autocomplete="off" aria-live="polite">
      <section class="q" data-q="1">
        <h3>1. Which HTML element is used to define the largest heading?</h3>
        <div class="options">
          <label><input type="radio" name="q1" value="a"> &lt;heading&gt;</label>
          <label><input type="radio" name="q1" value="b"> &lt;h1&gt;</label>
          <label><input type="radio" name="q1" value="c"> &lt;head&gt;</label>
          <label><input type="radio" name="q1" value="d"> &lt;title&gt;</label>
        </div>
      </section>
```

```
<section class="q" data-q="2">
```

```
<h3>2. Which CSS property controls the text size?</h3>
```

```
<div class="options">
```

```
<label><input type="radio" name="q2" value="a"> font-style</label>
```

```
<label><input type="radio" name="q2" value="b"> text-size</label>
```

```
<label><input type="radio" name="q2" value="c"> font-size</label>
```

```
<label><input type="radio" name="q2" value="d"> text-style</label>
```

```
</div>
```

```
</section>
```

```
<section class="q" data-q="3">
```

```
<h3>3. In JavaScript, which keyword is used to declare a block-scoped  
variable?</h3>
```

```
<div class="options">
```

```
<label><input type="radio" name="q3" value="a"> var</label>
```

```
<label><input type="radio" name="q3" value="b"> let</label>
```

```
<label><input type="radio" name="q3" value="c"> const</label>
```

```
<label><input type="radio" name="q3" value="d"> both b and c</label>
```

```
</div>
```

```
</section>
```

```
<section class="q" data-q="4">
```

```
<h3>4. Which HTTP status code means "Not Found"?</h3>
```

```
<div class="options">
```

```
<label><input type="radio" name="q4" value="a"> 200</label>
```

```
<label><input type="radio" name="q4" value="b"> 301</label>
```

```
<label><input type="radio" name="q4" value="c"> 404</label>
```

```
<label><input type="radio" name="q4" value="d"> 500</label>
```

```
</div>
```

```
</section>
```

```
<section class="q" data-q="5">
```

```
<h3>5. Which of the following is a semantic HTML element?</h3>
```

```
<div class="options">
```

```
<label><input type="radio" name="q5" value="a"> &lt;div&gt;</label>
```

```
<label><input type="radio" name="q5" value="b"> &lt;span&gt;</label>
```

```
<label><input type="radio" name="q5" value="c"> &lt;article&gt;</label>
```

```
<label><input type="radio" name="q5" value="d"> &lt;br&gt;</label>
```

```
</div>
</section>
```

```
<div class="controls" role="group" aria-label="Exam controls">
  <button type="button" id="submitBtn" class="secondary" disabled>Submit
Answers</button>
</div>
```

```
</form>
</div>
</div>
```

```
<script>
var correct = { q1: 'b', q2: 'c', q3: 'd', q4: 'c', q5: 'c' };
var examForm = document.getElementById('examForm');
var submitBtn = document.getElementById('submitBtn');
```

```
function enableSubmission(enable){
  submitBtn.disabled = !enable;
  submitBtn.classList.toggle('secondary', !enable);
}
```

```
function readAnswers(){
  const data = {};
  const formData = new FormData(examForm);
  for (let i=1;i<=5;i++){
    data['q'+i] = formData.get('q'+i) || null;
  }
  return data;
}
```

```
function gradeAnswers(answers){
  let score=0;
  const details = [];
  for (let i=1;i<=5;i++){
    const key='q'+i;
    const user = answers[key];
    const correctAns = correct[key];
```

```
const isCorrect = user && user === correctAns;
if (isCorrect) score++;
details.push({ q: i, user, correct: correctAns, isCorrect });
}
return { score, details, total: 5 };
}
```

```
function renderResultPopUp(report){
  const percent = Math.round((report.score/report.total)*100);
  let grade = 'F';
  if (percent >= 90) grade='A';
  else if (percent >= 75) grade='B';
  else if (percent >= 60) grade='C';
  else if (percent >= 50) grade='D';
```

```
  let text = `Score: ${report.score} / ${report.total} \nGrade: ${grade}\n\nCorrect
  answers:\n`;
```

```
  report.details.forEach(d => {
    const corrText = d.correct ? d.correct.toUpperCase() : 'N/A';
    const qText = Array.from(document.querySelectorAll('.q'))[d.q-
1].querySelector('h3').textContent;
    text += `Q${d.q}: ${corrText}\n`;
  });
```

```
  alert(text);
}
```

```
examForm.addEventListener('change', function() {
  var answers = readAnswers();
  var any = false;
  for (var k in answers) {
    if (answers.hasOwnProperty(k) && answers[k] !== null) {
      any = true;
      break;
    }
  }
  enableSubmission(any);
});
```

```
submitBtn.addEventListener('click', function() {  
  var answers = readAnswers();  
  var report = gradeAnswers(answers);  
  renderResultPopUp(report);  
  enableSubmission(false);  
});
```

```
enableSubmission(false);
```

```
</script>
```

```
</body>
```

```
</html>
```

## OUTPUT:

Online Exam

1. Which HTML element is used to define the largest heading?

☒ <heading>

☐ <h1>

☐ <head>

☐ <title>

2. Which CSS property controls the text size?

☐ font-style

☐ text-size

☒ font-size

☐ text-style

3. In JavaScript, which keyword is used to declare a block-scoped variable?

☐ var

☐ let

☒ const

☐ both b and c

4. Which HTTP status code means "Not Found"?

☐ 200

☐ 301

☒ 404

☒ font-size

☐ text-style

3. In JavaScript, which keyword is used to declare a block-scoped variable?

☐ var

☐ let

☒ const

☐ both b and c

4. Which HTTP status code means "Not Found"?

☐ 200

☐ 301

☒ 404

☐ 500

5. Which of the following is a semantic HTML element?

☐ <div>

☐ <span>

☐ <article>

☒ <br>

Submit Answers

☒ font-size

☐ text-style

3. In JavaScript, which

☐ var

☐ let

☒ const

☐ both b and c

4. Which HTTP status code means "Not Found"?

☐ 200

☐ 301

☒ 404

☐ 500

5. Which of the following is a semantic HTML element?

☐ <div>

☐ <span>

☐ <article>

☒ <br>

Submit Answers

127.0.0.1:5500 says

Score: 3 / 5

Grade: C

Correct answers:

Q1: B

Q2: C

Q3: D

Q4: C

Q5: C

OK