

LOKESHVISWA M

17/07/2025

1.Simple Calculator

Label Name	Element Name	Constraints
Input1	input1	This element is to get the first input. Type should be 'number'
Input2	input2	This element is to get the second input. Type should be 'number'
Select Operation	Operation	A drop down list contains the following values: Select... , ADD, SUBTRACT, MULTIPLY and DIVIDE. Set these values as its option tag text and option tag's 'value' attributes
	Submit	An image for submit must be included using <u>img</u> tag
	Reset	An image for reset must be included using <u>img</u> tag

CODING:

```
<!DOCTYPE html>
<html>
<head>
  <title>Strict Simple Calculator</title>
  <style>
    body {
      font-family: Arial;
      background-color: #eef;
      padding: 20px;
      max-width: 400px;
      margin: auto;
      border: 1px solid #aaa;
      border-radius: 8px;
    }
    h2 {
      text-align: center;
      color: #333;
    }
    label {
      display: block;
```

```

        margin-top: 10px;
        font-weight: bold;
    }
    select, input[type="number"] {
        width: 100%;
        padding: 8px;
        margin-top: 5px;
        border-radius: 5px;
        border: 1px solid #aaa;
    }
    .buttons {
        display: flex;
        justify-content: space-between;
        margin-top: 20px;
    }
    .btn-img {
        width: 48%;
        border: none;
        cursor: pointer;
    }
    .result {
        margin-top: 20px;
        font-size: 18px;
        color: green;
        text-align: center;
    }
}
</style>
</head>
<body>

<h2>Simple Calculator</h2>

<form id="calcForm">
    <label for="input1">Input1:</label>
    <input type="number" id="input1" name="input1" required>

    <label for="input2">Input2:</label>
    <input type="number" id="input2" name="input2" required>

    <label for="operation">Select Operation:</label>
    <select id="operation" name="operation" required>
        <option value="">Select..</option>
        <option value="ADD">ADD</option>
        <option value="SUBTRACT">SUBTRACT</option>
        <option value="MULTIPLY">MULTIPLY</option>
        <option value="DIVIDE">DIVIDE</option>
    </select>

```

```

<div class="buttons">

    
</div>

<div class="result" id="resultDisplay"></div>
</form>

<script>

document.getElementById("submitImg").addEventListener("click", function(e) {
    e.preventDefault();

    const input1 = parseFloat(document.getElementById("input1").value);
    const input2 = parseFloat(document.getElementById("input2").value);
    const operation = document.getElementById("operation").value;
    const resultDisplay = document.getElementById("resultDisplay");

    if (isNaN(input1) || isNaN(input2)) {
        resultDisplay.textContent = "Both inputs must be valid numbers.";
        return;
    }

    let result;

    switch (operation) {
        case "ADD":
            result = input1 + input2;
            break;
        case "SUBTRACT":
            result = input1 - input2;
            break;
        case "MULTIPLY":
            result = input1 * input2;
            break;
        case "DIVIDE":
            if (input2 === 0) {
                resultDisplay.textContent = "Cannot divide by zero.";
                return;
            }
            result = input1 / input2;
            break;
        default:
    }

```

```

        resultDisplay.textContent = "Please select an operation.";
        return;
    }

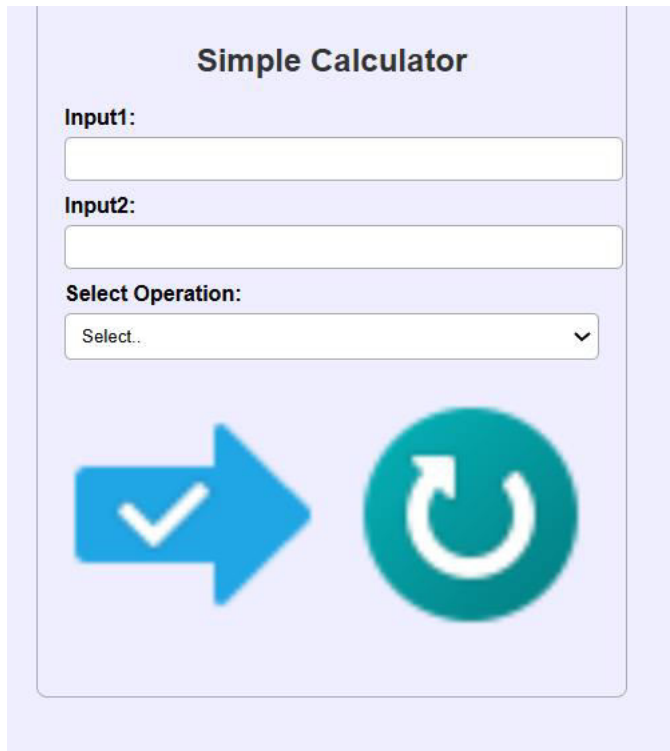
    resultDisplay.textContent = "Result: " + result;
});

document.getElementById("resetImg").addEventListener("click", function(e) {
    e.preventDefault();
    document.getElementById("calcForm").reset();
    document.getElementById("resultDisplay").textContent = "";
});
</script>

</body>
</html>

```

OUTPUT:



The screenshot shows a web application titled "Simple Calculator" with a light purple background. It features two input fields labeled "Input1:" and "Input2:". Below these is a dropdown menu labeled "Select Operation:" with a placeholder text "Select..". At the bottom, there are two large circular buttons: a blue button with a white checkmark and arrow pointing right, and a teal button with a white circular arrow icon.

2. Player Application Form Validation

In the player application form created, do the following validation using JavaScript.

- 1) All fields are mandatory
- 2) Matches played, runs scored, wickets and age should be numbers
- 3) Player name should contain alphabets only
- 4) Email should be in the correct format

If all validation rules are satisfied, success page should be displayed.

CODING;

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8" />
```

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

```
<title>Player Application Form</title>
```

```
<style>
```

```
body {
```

```
    font-family: Verdana, Geneva, Tahoma, sans-serif;
```

```
    background: white;
```

```
    margin: 0;
```

```
    padding: 50px;
```

```
    font-size: large;
```

```
}
```

```
.container {
```

```
    width: 500px;
```

```
    margin: auto;
```

```
    background: #add8e6;
```

```
    border: 2px solid #4682b4;
```

```
padding: 20px;
}
h2 {
background: #4682b4;
color: white;
margin: -20px -20px 20px;
padding: 10px;
text-align: center;
}
form div {
margin-bottom: 12px;
}
label {
display: inline-block;
width: 120px;
font-style: italic;
}
input[type="text"], input[type="email"], select, textarea {
width: 280px;
padding: 4px;
border: 1px solid #ccc;
border-radius: 2px;
}
.checkbox-group, .radio-group {
display: inline-block;
width: 280px;
```

```
}

.checkbox-group label, .radio-group label {

    font-style: normal;

    margin-right: 10px;

}

.buttons {

    text-align: center;

    margin-top: 20px;

}

input[type="submit"], input[type="reset"] {

    padding: 6px 12px;

    margin: 0 10px;

}

.error {

    color: red;

    font-size: 0.9em;

    margin-left: 124px;

}

.success {

    text-align: center;

    color: green;

    font-weight: bold;

    margin-top: 20px;

}

</style>

</head>
```

```
<body>

  <div class="container">

    <h2>Player Details</h2>

    <form id="playerForm" novalidate>

      <div>

        <label for="name">Name*</label>

        <input type="text" id="name" name="name" placeholder="Enter your name" />

        <div class="error" id="errName"></div>

      </div>

      <div>

        <label for="dob">DOB*</label>

        <input type="text" id="dob" name="dob" placeholder="DD/MM/YYYY" />

        <div class="error" id="errDob"></div>

      </div>

      <div>

        <label for="age">Age*</label>

        <input type="text" id="age" name="age" placeholder="Enter your age" />

        <div class="error" id="errAge"></div>

      </div>

      <div>

        <label for="email">Email Id*</label>

        <input type="email" id="email" name="email" placeholder="abc12@xyz.com" />

      </div>

    </form>

  </div>

</body>
```



```
<div class="error" id="errEmail"></div>
```

```
</div>
```

```
<div>
```

```
<label for="country">Country*</label>
```

```
<select id="country" name="country">
```

```
<option value="">Select country</option>
```

```
<option>India</option><option>USA</option><option>UK</option><option>Australia</option>  
>
```

```
</select>
```

```
<div class="error" id="errCountry"></div>
```

```
</div>
```

```
<div>
```

```
<label>Skill*</label>
```

```
<div class="checkbox-group">
```

```
<label><input type="checkbox" name="skill" value="Batting"> Batting</label>
```

```
<label><input type="checkbox" name="skill" value="Bowling"> Bowling</label>
```

```
</div>
```

```
<div class="error" id="errSkill"></div>
```

```
</div>
```

```
<div>
```

```
<label>Bat Style</label>
```

```
<div class="radio-group">
```

<label><input type="radio" name="batStyle" value="Right"> Right Handed</label>

<label><input type="radio" name="batStyle" value="Left"> Left Handed</label>

</div>

</div>

<div>

<label>Bowl Style</label>

<div class="radio-group">

<label><input type="radio" name="bowlStyle" value="Right"> Right Handed</label>

<label><input type="radio" name="bowlStyle" value="Left"> Left Handed</label>

</div>

</div>

<div>

<label for="matches">Matches*</label>

<input type="text" id="matches" name="matches" placeholder="Matches Played" />

<div class="error" id="errMatches"></div>

</div>

<div>

<label for="runs">Runs*</label>

<input type="text" id="runs" name="runs" placeholder="Total runs scored" />

<div class="error" id="errRuns"></div>

</div>

<div>

```
<label for="wickets">Wickets*</label>
```

```
<input type="text" id="wickets" name="wickets" placeholder="Wickets" />
```

```
<div class="error" id="errWickets"></div>
```

```
</div>
```

```
<div>
```

```
<label for="achievements">Achievements</label>
```

```
<textarea id="achievements" name="achievements" rows="3" placeholder="Your  
Achievements in Cricket"></textarea>
```

```
</div>
```

```
<div class="buttons">
```

```
<input type="submit" value="Submit" />
```

```
<input type="reset" value="Reset" />
```

```
</div>
```

```
</form>
```

```
<div id="successMsg" class="success" style="display:none;">
```

```
Form submitted successfully!
```

```
</div>
```

```
</div>
```

```
<script>
```

```
document.getElementById('playerForm').addEventListener('submit', function(e) {
```

```
e.preventDefault();
```

```
clearErrors();
```

```
const data = {  
  name: this.name.value.trim(),  
  dob: this.dob.value.trim(),  
  age: this.age.value.trim(),  
  email: this.email.value.trim(),  
  country: this.country.value,  
  skills: Array.from(this.skill).filter(chk => chk.checked).map(chk => chk.value),  
  matches: this.matches.value.trim(),  
  runs: this.runs.value.trim(),  
  wickets: this.wickets.value.trim()  
};
```

```
let valid = true;
```

```
const isAlphabetic = txt => /^[A-Za-z ]+$/ .test(txt);  
const isEmail = txt => /^[^s@]+@[^s@]+\.[^s@]+$/ .test(txt);  
const isNumber = txt => /^\d+$/ .test(txt);
```

```
if (!data.name || !isAlphabetic(data.name)) {  
  valid = false;  
  showError('errName', 'Alphabetic name is required');  
}  
  
if (!data.dob) {  
  valid = false;
```

```
        showError('errDob', 'Date of birth is required');
    }
    if (!data.age || !isNumber(data.age)) {
        valid = false;
        showError('errAge', 'Valid numeric age is required');
    }
    if (!data.email || !isEmail(data.email)) {
        valid = false;
        showError('errEmail', 'Valid email is required');
    }
    if (!data.country) {
        valid = false;
        showError('errCountry', 'Country is required');
    }
    if (data.skills.length === 0) {
        valid = false;
        showError('errSkill', 'Select at least one skill');
    }
    if (!data.matches || !isNumber(data.matches)) {
        valid = false;
        showError('errMatches', 'Valid number of matches is required');
    }
    if (!data.runs || !isNumber(data.runs)) {
        valid = false;
        showError('errRuns', 'Valid runs number is required');
    }
}
```

```

    if (!data.wickets || !isNumber(data.wickets)) {
        valid = false;
        showError('errWickets', 'Valid wickets number is required');
    }

    if (valid) {
        document.getElementById('successMsg').style.display = 'block';
        this.style.display = 'none';
    }
});

function showError(id, message) {
    document.getElementById(id).innerText = message;
}

function clearErrors() {
    document.querySelectorAll('.error').forEach(el => el.innerText = "");
    document.getElementById('successMsg').style.display = 'none';
    document.getElementById('playerForm').style.display = 'block';
}
</script>
</body>
</html>

```

OUTPUT:

Player Details

Name*

lokeshviswa

DOB*

12/05/2002

Age*

23

Email Id*

dvsscsc@gmail.com

Country*

India

Skill*

☐ Batting

☒ Bowling

Bat Style

☐ Right Handed

☒ Left Handed

Bowl Style

☐ Right Handed

☒ Left Handed

Matches*

12

Runs*

12234

Wickets*

345

Achievements

Your Achievements in Cricket

Submit

Reset

Player Details

Form submitted successfully!

3. Create a html page with radio buttons to select colors. Selected color should be applied as the background colour of the page.

CODING:

```
<<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Background Color Selector</title>

  <style>

    body {

      transition: background-color 0.4s ease;

      font-size: x-large;

      font-style: oblique;

      padding: 50px;

      text-align: center;

    }

    .color-option {

      margin: 0px 0;
```



```
    }  
  </style>  
</head>  
<body>  
  
  <h1>Select a Background Color</h1>  
  
  <div class="color-option">  
    <label><input type="radio" name="bgColor" value="white" checked> White</label>  
  </div>  
  
  <div class="color-option">  
    <label><input type="radio" name="bgColor" value="lightblue"> Light Blue</label>  
  </div>  
  
  <div class="color-option">  
    <label><input type="radio" name="bgColor" value="lightgreen"> Light Green</label>  
  </div>  
  
  <div class="color-option">  
    <label><input type="radio" name="bgColor" value="lightyellow"> Light Yellow</label>  
  </div>  
  
  <div class="color-option">  
    <label><input type="radio" name="bgColor" value="lightgray"> Light Gray</label>  
  </div>  
  
  <script>  
  
    function changeBackgroundColor(color) {
```

```
document.body.style.backgroundColor = color;

}
```

```
const radioButtons = document.querySelectorAll('input[name="bgColor"]');

radioButtons.forEach(button => {

    button.addEventListener('change', (event) => {

        changeBackgroundColor(event.target.value);

    });

});
```

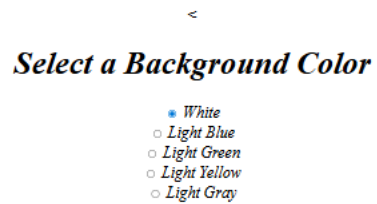
```
changeBackgroundColor(document.querySelector('input[name="bgColor"]:checked').value);
```

```
</script>
```

```
</body>
```

```
</html>
```

OUTPUT:



Select a Background Color

- White
- Light Blue
- Light Green
- Light Yellow
- Light Gray

Select a Background Color

- White
- Light Blue
- Light Green
- Light Yellow
- Light Gray

Select a Background Color

- White
- Light Blue
- Light Green
- Light Yellow
- Light Gray

Select a Background Color

- White
- Light Blue
- Light Green
- Light Yellow
- Light Gray