

**VALIDATING FORM DATA**

<b>Name:</b>	Jericho Tabingo Lagang	<b>Date:</b>	09/26/2023
<b>Program:</b>	BSCpE	<b>Section:</b>	A361

**Intended Learning Outcomes:**

1. Write functions to react to events.
2. Validate HTML forms.

Design a simple form that contains the following details:

- First Name
- Middle Initial
- Last Name
- E-Mail Address

Here is an example form:

### Sign Up

The form must be able to validate if the First Name, Middle Initial, and Last Name fields have corresponding inputs (regardless of content). The form must also validate if there is an invalid e-mail address format presented. There must be an alert message to signify that there are invalid inputs.

Here is a sample output:

### Sign Up

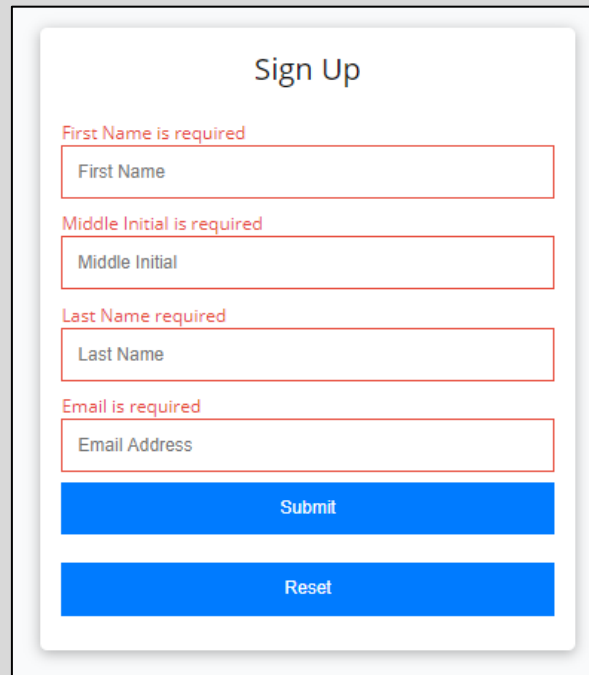
  
  
  
Email is required

### Sign Up

  
  
  
Invalid email format

**VALIDATING FORM DATA**

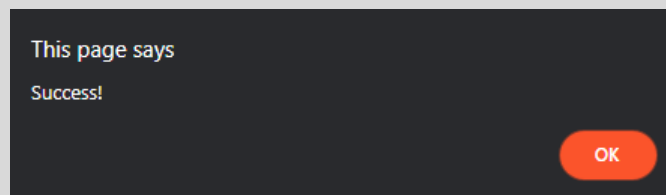
The form must be able to see if there are no inputs placed. Here is a sample output:



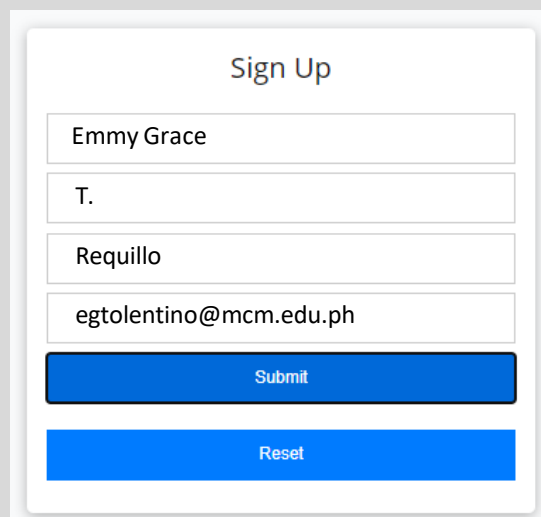
A sample output of a Sign Up form. The form is titled "Sign Up" and contains four input fields, each with a red error message above it: "First Name is required", "Middle Initial is required", "Last Name required", and "Email is required". The input fields are labeled "First Name", "Middle Initial", "Last Name", and "Email Address". Below the input fields are two blue buttons: "Submit" and "Reset".

Once the form validates all inputs, an alert box must be seen that says **Success!**

Here is a sample output:



A sample output of a Success alert box. The box is dark gray with the text "This page says" and "Success!". There is an orange "OK" button in the bottom right corner.



A sample output of a Sign Up form with valid inputs. The form is titled "Sign Up" and contains four input fields, each with a valid input: "Emmy Grace", "T.", "Requillo", and "egtolentino@mcm.edu.ph". Below the input fields are two blue buttons: "Submit" and "Reset".

**HTML Source Code:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" type="text/css" href="form.css">
  <script src="form.js"></script>
  <title>Simple Form</title>
</head>
<body>
  <div class="container">
    <h1>Sign up</h1>
    <form action="#" method="POST" onsubmit="return validateForm()">
      <label for="firstName">First Name:</label>
      <input type="text" id="firstName" name="firstName" required><br><br>

      <label for="middleInitial">Middle Initial:</label>
      <input type="text" id="middleInitial" name="middleInitial"
maxlength="1"><br><br>

      <label for="lastName">Last Name:</label>
      <input type="text" id="lastName" name="lastName" required><br><br>

      <label for="email">E-Mail Address:</label>
      <input type="email" id="email" name="email" required><br><br>

      <input type="submit" value="Submit">
      <input type="reset" value="Reset">
    </form>
  </div>
</body>
</html>
```

**VALIDATING FORM DATA****CSS Source Code:**

```
body {
  margin: 0;
  padding: 0;
  font-family: Arial, sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.container {
  max-width: 400px;
  padding: 20px 100px 20px 100px;
  border: 1px solid #ddd;
  background-color: #f9f9f9;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
  border-radius: 5px;
}

h1 {
  text-align: center;
  margin-bottom: 20px;
  color: #333;
}

form {
  margin: 0 auto;
  padding: 0;
  width: 100%;
  box-sizing: border-box;
}

label {
  display: block;
  margin-bottom: 10px;
  font-weight: bold;
  color: #333;
}

input[type="text"],
input[type="email"] {
  width: 100%;
  padding: 10px;
  margin-bottom: 15px;
  border: 1px solid #ccc;
  border-radius: 4px;
  font-size: 16px;
}

.row {
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

```
input[type="submit"],
input[type="reset"] {
    background-color: #007bff;
    color: #fff;
    border: none;
    padding: 10px 20px;
    font-size: 16px;
    cursor: pointer;
    flex: 1;
}

input[type="submit"]:hover,
input[type="reset"]:hover {
    background-color: #0056b3;
}

.alert {
    color: red;
    margin-top: 5px;
}
```

**JavaScript Source Code:**

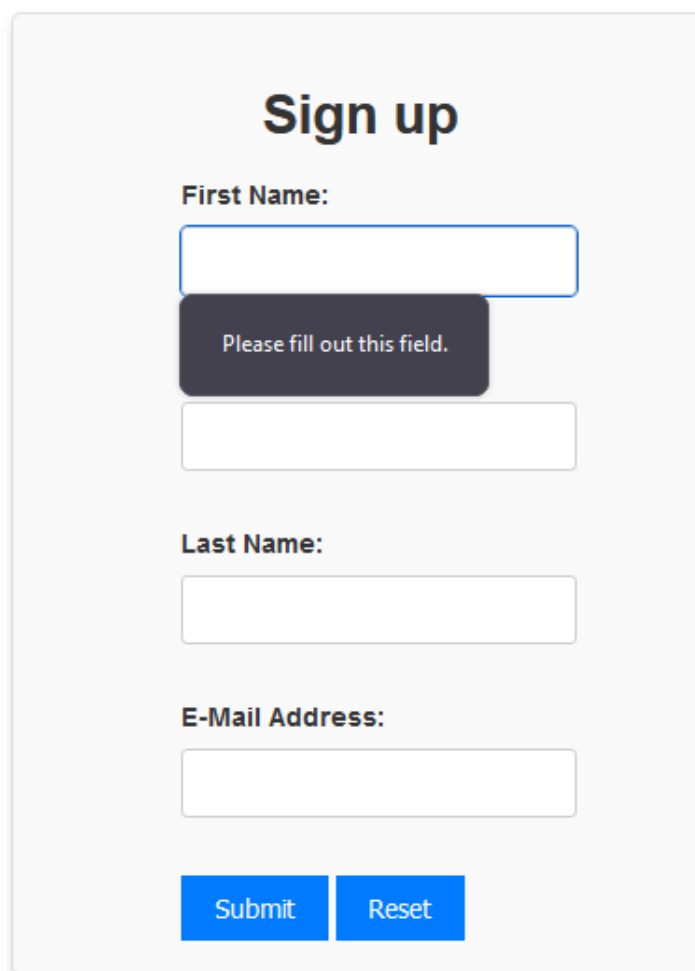
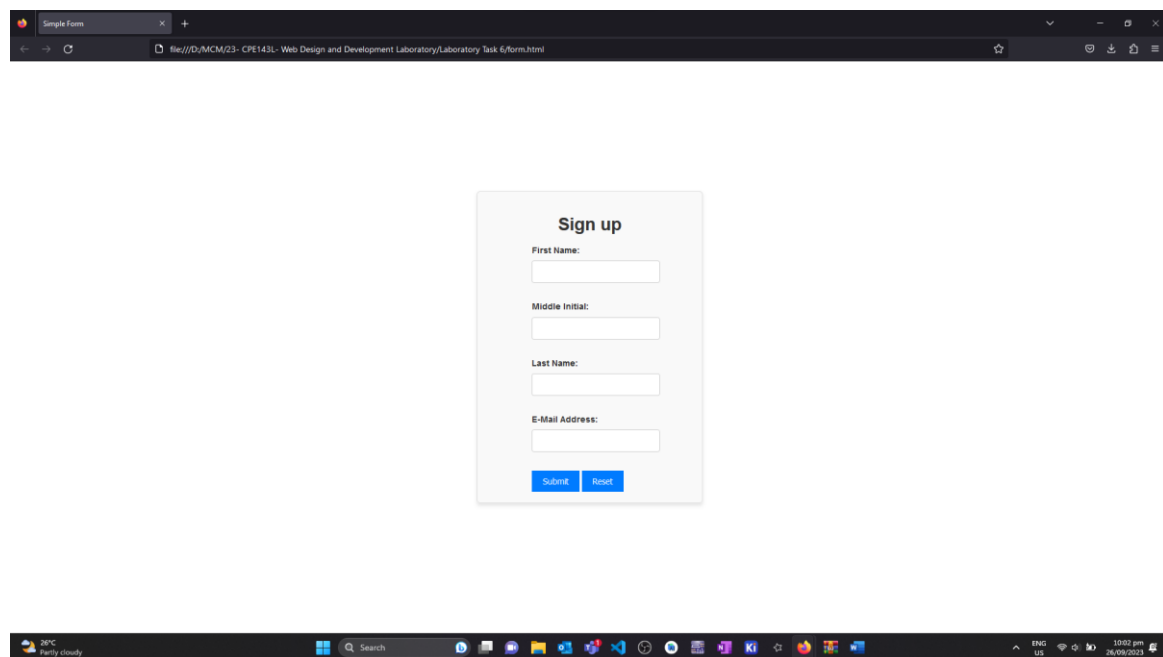
```
function validateForm() {
    const firstName = document.getElementById("firstName").value;
    const middleInitial = document.getElementById("middleInitial").value;
    const lastName = document.getElementById("lastName").value;
    const email = document.getElementById("email").value;

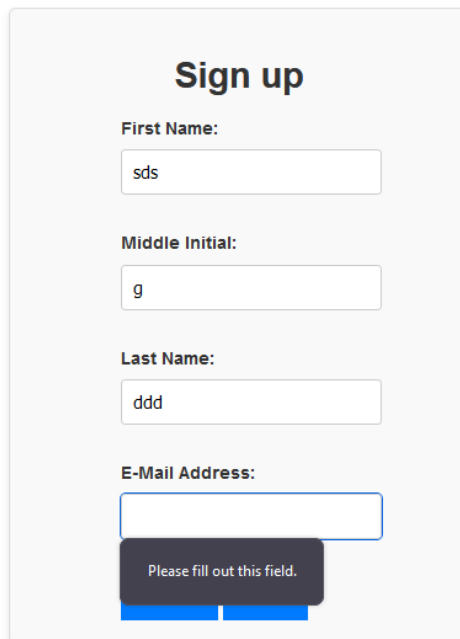
    if (!firstName || !middleInitial || !lastName || !isValidEmail(email)) {
        alert("Invalid inputs. Please check your information.");
        return false;
    }

    return true;
}

function isValidEmail(email) {
    // Basic email format validation using a regular expression
    const emailPattern = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;
    return emailPattern.test(email);
}

alert("Success!");
return true;
```

**Output:**



**Sign up**

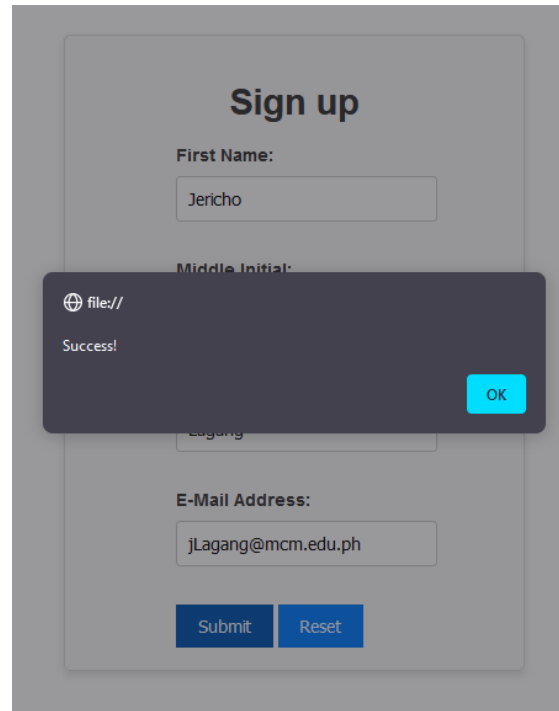
First Name:

Middle Initial:

Last Name:

E-Mail Address:

Please fill out this field.



**Sign up**

First Name:

Middle Initial:

Last Name:

E-Mail Address:

Submit Reset

file://  
Success!  
OK

**Findings and Observations:**

Creating an HTML form with validation rules using JavaScript offers a straightforward way to enhance user input quality. By implementing validation functions, it becomes possible to ensure that required fields are filled, and data formats (like email addresses) adhere to specific patterns. Additionally, feedback mechanisms, such as alert messages, can be used to communicate validation outcomes to users. This approach significantly improves user experience and helps prevent erroneous or incomplete data submissions, ultimately enhancing the reliability of collected information.