*PRN: 21070126039 Name: Jainil Patel Batch: AI/ML A2*

*Full Stack Developments - LAB 5*

*Interactive Registration Form with JavaScript Validation*

**Problem**:

Design a registration form for a website and implement JavaScript event handling to ensure proper validation of user input. Your form should include the following fields: First Name, Last Name, Email, and Password.

**Requirements:**

* Implement event handling to validate the following conditions:
* All fields are required.
* Email field must contain a valid email address format.
* Password field must be at least 8 characters long.
* Display appropriate error messages next to each field if validation fails.
* Prevent the form from being submitted if there are validation errors.
* Provide visual feedback to the user indicating which fields have errors.
* Ensure that the form submission is successful only when all fields are properly filled out.

Write a JavaScript function that handles the form submission event and integrates it with the HTML form.

Test your form thoroughly by entering various inputs and verifying that the validation works as expected.

Note: Your implementation should be well-structured, readable, and demonstrate good programming practices. Additionally, consider enhancing the user experience by using CSS to style the form and error messages appropriately.

**HTML Tags used:**

1. **<html>: Defines the root element of an HTML document.**

<html>

*<!-- HTML content goes here -->*

</html>

1. **<head>: Contains meta-information about the HTML document, such as title, links to stylesheets, and scripts.**

<head>

    <title>My Page Title</title>

*<!-- Other meta-information goes here -->*

</head>

1. **<title>: Sets the title of the HTML document displayed in the browser's title bar or tab.**

<title>My Page Title</title>

1. **<style>: Defines CSS styles that apply to elements within the document.**

<style>

    body {

        font-family: Arial, sans-serif;

    }

</style>

1. **<body>: Contains the content of the HTML document that is displayed in the browser window.**

<body>

    <h1>Hello, World!</h1>

    <p>This is a paragraph.</p>

</body>

1. **<script>: Defines client-side JavaScript code within the HTML document, allowing for dynamic behavior and interaction.**

<script>

    .

    .

    .

</script>

1. **<form>: Defines an HTML form for user input. It can contain input elements like text fields, checkboxes, radio buttons, submit buttons, and more.**

<form *id*="validationForm">

    .

    .

    .

</form>

1. **<input type="button”> :** **Defines a clickable button that can trigger JavaScript functions**

<input *type*="button" *value*="Activate Textboxes" *onclick*="activateTextboxes()" >

1. **<input type="submit”>: Defines a submit button for submitting form data to a server.**

<input *type*="submit" *value*="Submit" *onclick*="submitForm(event)">

**Code:**

<!DOCTYPE *html*>

<html *lang*="en">

<head>

    <meta *charset*="UTF-8">

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

    <title>Lab 5</title>

    <link *rel*="stylesheet" *type*="text/css" *href*="style.css" >

</head>

<body>

    <form *id*="registrationForm" *action*="#" *method*="post">

        <h2>Registration Form</h2>

        <label *for*="firstName">First Name:</label><br>

        <input *type*="text" *id*="firstName" *name*="firstName">

        <span *id*="firstNameError"  *class*="error">

        </span><br>

        <label *for*="lastName">Last Name:</label><br>

        <input *type*="text" *id*="lastName" *name*="lastName">

        <span *id*="lastNameError"  *class*="error">

        </span><br>

        <label *for*="email">Email:</label><br>

        <input *type*="email" *id*="email" *name*="email">

        <span *id*="emailError"  *class*="error">

        </span><br>

        <label *for*="password">Password:</label><br>

        <input *type*="password" *id*="password" *name*="password">

        <span *id*="passwordError"  *class*="error">

        </span><br>

        <input *type*="submit" *value*="Register">

    </form>

    <script *src*="registration.js"></script>

</body>

</html>

**CSS Code:**

**Simple CSS:**

label{

    font-weight: bold;

}

span{

    color: red;

}

body {

    font-family: Arial, sans-serif;

    display: flex;

}

input[*type*="submit"] {

    margin-top: 10px;

    font-weight: bold;

}

**Advanced CSS:**

body{

    font-family: Arial, Helvetica, sans-serif;

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh ;

    background-color: #f9f9f9;

}

form{

    max-width: 600px;

    width: 100%;

    background-color: #fff;

    padding: 10px;

    box-shadow: 0 0 10px 0 rgba(0,0,0,.5);

    border-radius: 8px;

}

label{

    display: block;

    margin-bottom: 2px;

    margin-top: 10px;

    font-weight:bold ;

}

input{

    width: 100%;

    padding: 8px;

    border-radius: 4px;

    margin-bottom: 15px;

    border: 1px solid #ccc;

    box-sizing: border-box;

}

input[*type*=submit]{

    background-color: #45a049;

    color: white;

}

input[*type*=submit]*:hover*{

    background-color: green;

    color: white;

}

**Js Code (registration.js):**

const form = document.getElementById('registrationForm');

const firstName = document.getElementById('firstName');

const firstNameError = document.getElementById('firstNameError');

const lastName = document.getElementById('lastName');

const lastNameError = document.getElementById('lastNameError');

const email = document.getElementById('email');

const emailError = document.getElementById('emailError');

const password = document.getElementById('password');

const passwordError = document.getElementById('passwordError');

*// Get the form element*

form.addEventListener('submit', function(event) {

    console.log(event);

*// Validate first name*

    let hasError = false;

*if* (firstName.value === '') {

        firstNameError.textContent = 'First name is required';

        hasError = true;

    } *else* {

        firstNameError.textContent = '';

    }

*if* (hasError) {

        event.preventDefault(); *// Prevent form submission if there are errors*

    }

*// Validate last name*

*if* (lastName.value === '') {

        lastNameError.textContent = 'Last name is required';

        hasError = true;

    } *else* {

        lastNameError.textContent = '';

    }

*// Email Validation*

*if* (email.value.trim() === '') {

        emailError.textContent = 'Email is required';

        hasError = true;

    } *else* *if* (!isValidEmail(email.value.trim())) {

        emailError.textContent = 'Invalid email format';

        hasError = true;

    } *else* {

        emailError.textContent = '';

    }

*// Password Validation*

*if* (password.value.trim() === '') {

        passwordError.textContent = 'Password is required';

        hasError = true;

    } *else* *if* (password.value.trim().length < 8) {

        passwordError.textContent = 'Password must be at least 8 characters long';

        hasError = true;

    } *else* *if* (!/[A-Z]/.test(password.value)) {

        passwordError.textContent = 'Password must contain at least one uppercase letter';

        hasError = true;

    } *else* *if* (!/[!@#$%^&\*(),.?":{}|<>]/.test(password.value)) {

        passwordError.textContent = 'Password must contain at least one symbol (!@#$%^&\*(),.?":{}|<>)';

        hasError = true;

    } *else* {

        passwordError.textContent = '';

    }

*if* (hasError) {

        event.preventDefault(); *// Prevent form submission if there are errors*

    }

*// Display confirmation message*

    alert("Thank you for registering!");

*// Optionally, you can reset the form after submission*

    form.reset();

});

*// Email Format Validation Function*

function isValidEmail(email) {

*// Regular expression for validating email format*

    const emailPattern = /*^*[^\s@]+@[^\s@]+\.[^\s@]+*$*/;

*return* emailPattern.test(email);

}

**Explanation of Js Code:**

1. **Element Retrieval:** The code fetches various HTML elements from the DOM using **document.getElementById().** These elements are used to interact with form inputs and error messages.

const form = document.getElementById('registrationForm');

const firstName = document.getElementById('firstName');

const firstNameError = document.getElementById('firstNameError');

const lastName = document.getElementById('lastName');

const lastNameError = document.getElementById('lastNameError');

const email = document.getElementById('email');

const emailError = document.getElementById('emailError');

const password = document.getElementById('password');

const passwordError = document.getElementById('passwordError');

1. **Form Submission Event Listener:** An event listener is added to the form's submit event. When the form is submitted, this function will be executed.

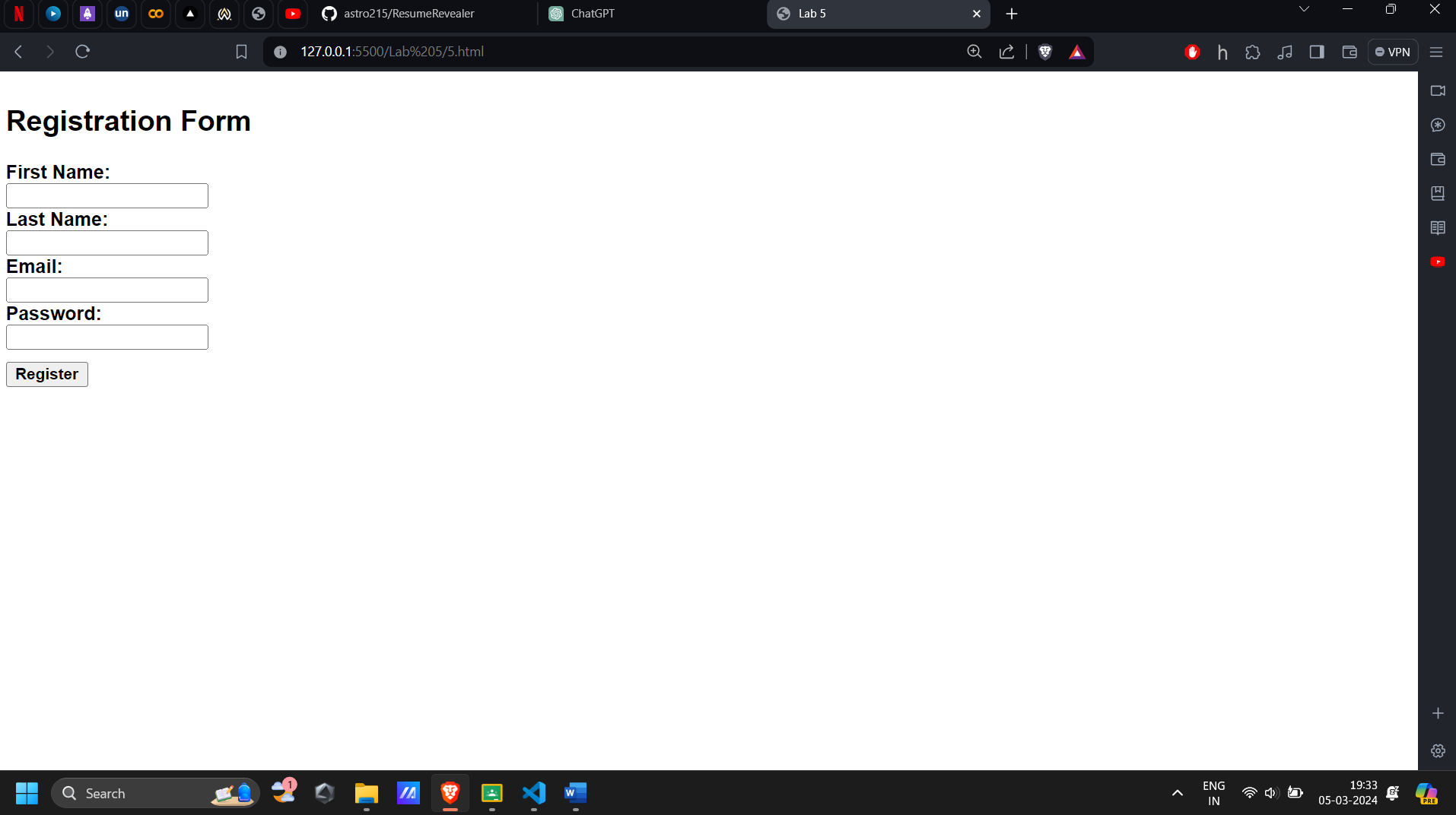
form.addEventListener('submit', function(event) {

*// Code to execute on form submission*

});

1. **Form Validation**: Inside the submit event listener function, the code checks each form field for validation errors.
   1. **First Name Validation**: Checks if the first name is empty. If it's empty, displays an error message.
   2. **Last Name Validation**: Checks if the last name is empty. If it's empty, displays an error message.
   3. **Email Validation**: Checks if the email field is empty or if it's not in a valid email format.
   4. **Password Validation**: Checks if the password field is empty, if it's less than 8 characters, if it contains at least one uppercase letter, and if it contains at least one special character.
2. **Event Prevention**: If any validation error is found (**hasError** is set to **true**), it prevents the form from being submitted by calling **event.preventDefault()**.
3. **Confirmation Message**: After all validations, it displays an alert with a thank you message.
4. **Form Reset**: Optionally, after form submission, the form is reset using **form.reset()**.
5. **Email Format Validation Function**: Defines a function **isValidEmail(email)** which uses a regular expression to validate the email format. It returns **true** if the email format is valid, otherwise **false**.

**Output:**

**Simple CSS:   
**

**Advance CSS:**

**A screenshot of a computer

Description automatically generated**

**Empty Form:**

**A screenshot of a computer

Description automatically generated**

**Partially Filled:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Email Errors:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Password Errors:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Correct From:**

**A screenshot of a computer

Description automatically generated**