MODULE END ASSIGNMENT: 02

SIGN UP FORM VALIDATION USING HTML, CSS & BOOTSTRAP AND JAVASCRIPT

This documentation outlines the implementation and best practices for building a sign up form with client-side validation. The goal is to ensure data integrity, enhance usability, and provide immediate feedback to users.

1. Introduction

This project focuses on creating a sleek, responsive Sign Up form for a web application. Using HTML, CSS, Bootstrap, and JavaScript, the form will emphasize strong client-side validation to enhance user experience and safeguard data integrity. It ensures real-time feedback for user inputs and restricts submission until all validations are successfully met.

2. Project Structure

```
signup-form-validation/
├── index.html              # Main HTML file containing the form
├── css/
│   └── styles.css          # Custom CSS styles (optional)
├── js/
│   └── validation.js       # JavaScript for form validation
└── README.md               # Project documentation
```

3. Setting Up the Environment

- **Prerequisites:**
  - Web browser (Chrome, Firefox, Edge, etc.)

- Code editor (VS Code, Sublime, etc.)

- **Bootstrap Integration:** Use Bootstrap CDN in your HTML `<head>`:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
```

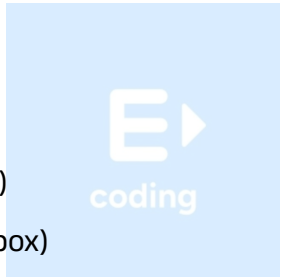
- **Folder Setup:**

  - Create the above directory structure.
  - Place your HTML, CSS, and JS files accordingly.

4. Layout

The sign up form will include the following fields:

- Full Name (text)
- Email Address (email)
- Password (password)
- Confirm Password (password)
- Terms and Conditions (checkbox)
- Submit Button



Bootstrap will be used for responsive layout and styling.

Example Layout:

```
<form class="row g-3 needs-validation" novalidate>
  <div class="col-12">
     <label for="fullName" class="form-label">Full Name</label>
     <input type="text" class="form-control" id="fullName" required>
     <div class="invalid-feedback">Please enter your full name.</div>
  </div>
  <div class="col-12">
     <label for="email" class="form-label">Email address</label>
     <input type="email" class="form-control" id="email" required>
```

```
      <div class="invalid-feedback">Please provide a valid
email.</div>
  </div>
  <div class="col-md-6">
      <label for="password" class="form-label">Password</label>
      <input type="password" class="form-control" id="password"
required minlength="8">
      <div class="invalid-feedback">Password must be at least 8
characters.</div>
  </div>
  <div class="col-md-6">
      <label for="confirmPassword" class="form-label">Confirm
Password</label>
      <input type="password" class="form-control" id="confirmPassword"
required>
      <div class="invalid-feedback">Passwords do not match.</div>
  </div>
  <div class="col-12">
      <div class="form-check">
      <input class="form-check-input" type="checkbox" id="terms"
required>
      <label class="form-check-label" for="terms">Agree to terms and
conditions</label>
      <div class="invalid-feedback">You must agree before
submitting.</div>
      </div>
  </div>
  <div class="col-12">
      <button class="btn btn-primary" type="submit">Sign Up</button>
  </div>
</form>
```

5. Detailed Implementation Steps

5.1 HTML Structure

- Use semantic tags and Bootstrap classes for layout.
- Add `required` and other HTML5 validation attributes.

5.2 CSS Styling

- Optionally, add a custom CSS file (`styles.css`) for additional styling.
- Use Bootstrap classes for most styling needs.

5.3 Bootstrap Integration

- Include Bootstrap CSS via CDN.
- Use Bootstrap's grid system and form classes for responsive design.

5.4 JavaScript Validation

- Create a `validation.js` file.
- Use JavaScript to handle:
  - Password and Confirm Password match
  - Real-time validation feedback
  - Preventing form submission if validation fails

Sample JavaScript:

```
document.addEventListener('DOMContentLoaded', function () {
   const form = document.querySelector('.needs-validation');
   form.addEventListener('submit', function (event) {
      const password = document.getElementById('password');
      const confirmPassword =
document.getElementById('confirmPassword');
      if (password.value !== confirmPassword.value) {
        confirmPassword.setCustomValidity('Passwords do not match');
      } else {
        confirmPassword.setCustomValidity('');
      }
      if (!form.checkValidity()) {
      event.preventDefault();
      event.stopPropagation();
```

```
    }
    form.classList.add('was-validated');
  }, false);
});
```

6. Submission Format: Hosting on GitHub

- Create a GitHub repository for your sign-up form validation..
- Upload your HTML, CSS, Bootstrap and JavaScript files to the repository.
- Ensure your assignment is publicly accessible.
- Share the GitHub repository link as your submission.

7. Evaluation Criteria:

S.no	Criteria	Marks
1	Code Quality	5
2	Implementation	10
3	User Interface	5
4	Error Handling	3
5	Documentation	2

8. Conclusion

This project demonstrates the creation of a robust, user-friendly sign up form using HTML, CSS, Bootstrap, and JavaScript. By following best practices in form validation and responsive design, the form ensures a positive user experience and reliable data collection. This

assignment will help you understand the importance of both usability and security in modern web applications.