Full Stack Development Lab

Lab Report

Name: Aparna Iyer

PRN: 22070126017

Batch: 2022-2026, A1

Branch: AI-ML

Experiment Number: 2

Title:

Responsive Registration Form with Client-Side Validation.

Aim:

To design and implement a fully responsive registration form using HTML and CSS, focusing on user experience and accessibility across various devices. The form incorporates client-side validation to ensure accurate user input.

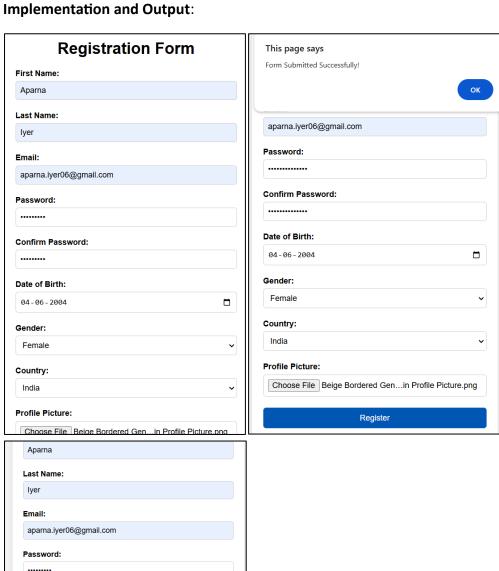
Objectives:

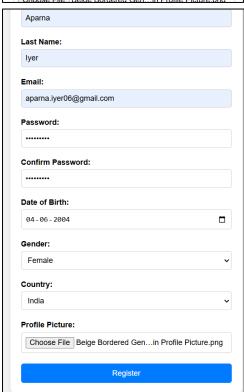
- 1. Design a registration form that includes fields for personal and account information, such as name, email, password, and more.
- 2. Implement responsive design techniques to ensure the form adjusts seamlessly across devices (mobile, tablet, desktop).
- 3. Add client-side validation using HTML5 attributes and JavaScript to enhance data accuracy and prevent submission errors.
- 4. Provide an aesthetically pleasing and user-friendly interface with clear error messages.

Algorithms Used:

- 1. **HTML5 Validation**: Utilize built-in validation attributes like required, email, minlength, and pattern to ensure basic input validation.
- 2. JavaScript Validation: Supplement HTML5 with JavaScript to:
 - o Validate fields like Confirm Password for matching with Password.
 - Display concise error messages dynamically.

3. Responsive Design: Implement media queries to adjust the layout and styles for various screen sizes.





Code Snippets:

```
# style: || || ? | * 1 5 🗆 \
Experiment_2.html X
Experiment_2.html > ...
      <!DOCTYPE html>
      <html lang="en">
          <meta charset="UTF-8">
          <meta name="viewport" content="width=device-width, initial-scale=1.0">
          <title>Registration Form</title>
          <link rel="stylesheet" href="styles.css">
          <div class="container">
              <h2 class="title">Registration Form</h2>
              <form id="registrationForm" onsubmit="return validateForm()">
                  <label for="firstName">First Name:</label>
                  <input type="text" id="firstName" name="firstName" required>
                  <span class="error" id="firstNameError"></span>
                  <label for="lastName">Last Name:</label>
                  <input type="text" id="lastName" name="lastName" required>
                  <span class="error" id="lastNameError"></span>
                  <label for="email">Email:</label>
                  <input type="email" id="email" name="email" required>
                  <span class="error" id="emailError"></span>
                  <label for="password">Password:</label>
                  <input type="password" id="password" name="password" required>
                  <span class="error" id="passwordError"></span>
                  <label for="confirmPassword">Confirm Password:</label>
                  <input type="password" id="confirmPassword" name="confirmPassword" required>
                  <span class="error" id="confirmPasswordError"></span>
                  <label for="dob">Date of Birth:</label>
                  <input type="date" id="dob" name="dob" required>
                  <span class="error" id="dobError"></span>
                  <label for="gender">Gender:</label>
```

```
Experiment_2.html X # style II II To To U
◆ Experiment_2.html > ♦ html > ♦ body > ♦ div.container > ♦ form#registrationForm > ♦ input#email
      <html lang="en">
           <div class="container">
               <form id="registrationForm" onsubmit="return validateForm()">
                    <label for="gender">Gender:</label>
                    <select id="gender" name="gender" required>
                        <option value="">Select</option>
                        <option value="male">Male</option>
                        <option value="female">Female</option>
                        <option value="other">Other</option>
                    <span class="error" id="genderError"></span>
                    <label for="country">Country:</label>
                    <select id="country" name="country" required>
    <option value="">Select</option>
                        <option value="India">India</option>
<option value="USA">USA</option>
                        <option value="UK">UK</option>
                        <option value="Australia">Australia
                    <span class="error" id="countryError"></span>
                    <label for="profilePicture">Profile Picture:</label>
                    <input type="file" id="profilePicture" name="profilePicture" accept="image/*" required>
<span class="error" id="profilePictureError"></span>
                    <button type="submit">Register</button>
           <script>
               function validateForm() {
                    let isValid = true;
                    let errorMessage = "";
                        firstName = document.getElementBvId("firstName").value.trim()
```

```
Experiment_2.html X
◆ Experiment_2.html > ♦ html > ♦ body > ♦ div.container > ♦ form#registrationForm > ♦ input#email
     <html lang="en">
          <script>
              function validateForm() {
                  let firstName = document.getElementById("firstName").value.trim();
                  let lastName = document.getElementById("lastName").value.trim();
                  let email = document.getElementById("email").value.trim();
                  let password = document.getElementById("password").value.trim();
                  let confirmPassword = document.getElementById("confirmPassword").value.trim();
                  let nameRegex = /^[A-Za-z]+$/;
                  let emailRegex = /^[^@\s]+@[^@\s]+\.com$/;
                  if (!nameRegex.test(firstName)) {
                      errorMessage += "First Name should contain only alphabets.\n";
                      isValid = false;
                  if (!nameRegex.test(lastName)) {
                      errorMessage += "Last Name should contain only alphabets.\n";
                      isValid = false;
                  if (!emailRegex.test(email)) {
                      errorMessage += "Email should be in a valid format (e.g., example@domain.com).\n";
                      isValid = false;
                  if (password !== confirmPassword) {
                      errorMessage += "Passwords do not match.\n";
                      isValid = false;
                  if (!isValid) {
                      alert("Please correct the following errors before submitting:\n\n" + errorMessage);
                  alert("Form Submitted Successfully!");
```

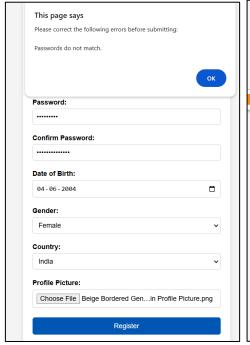
```
◆ Experiment_2.html X
\diamond Experiment_2.html \gt \diamondsuit html \gt \diamondsuit body \gt \diamondsuit div.container \gt \diamondsuit form#registrationForm \gt \diamondsuit input#email
       <html lang="en">
            <script>
                 function validateForm() {
                     alert("Form Submitted Successfully!");
                     document.getElementById("registrationForm").reset();
                     return false;
                 .title {
                     text-align: center;
                     font-size: 1.8rem;
                 .container {
                     width: 90%;
                     max-width: 400px;
                     margin: auto;
                 label, input, select, button {
                     font-size: 0.9rem;
       </body>
```

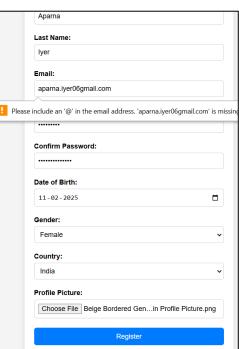
```
Experiment_2.html
                     # styles.css
# styles.css > 😝 button:hover
      /* General Styling */
      body {
           font-family: Arial, sans-serif;
           background-color: ■#f4f4f4;
           display: flex;
           justify-content: center;
          align-items: center;
          height: 100vh;
          margin: 0;
      .container {
           background: ☐ white;
           padding: 20px;
          border-radius: 10px;
          box-shadow: 0px 4px 10px □rgba(0, 0, 0, 0.1);
          width: 100%;
          max-width: 400px;
      /* Title */
      .title {
          text-align: center;
           font-size: 24px;
          margin-bottom: 20px;
      /* Form Styling */
      form {
          display: flex;
           flex-direction: column;
      label {
          margin-bottom: 5px;
          font-weight: bold;
```

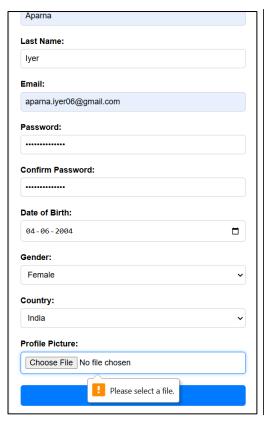
```
Experiment_2.html
                                 ×
                      # styles.css
# styles.css > ધ input:focus
       label {
           margin-bottom: 5px;
           font-weight: bold;
       input, select {
           padding: 8px;
           margin-bottom: 10px;
           border: 1px solid ■#ccc;
           border-radius: 5px;
           font-size: 14px;
       input:focus, select:focus {
 48
           outline: none;
           border-color: ■#007bff;
           box-shadow: 0 0 5px □rgba(0, 123, 255, 0.5);
       .error {
           color: ■red;
           font-size: 12px;
           margin-bottom: 10px;
       button {
           background-color: ■#007bff;
           color: ☐white;
           padding: 10px;
           border: none;
           border-radius: 5px;
           cursor: pointer;
           font-size: 16px;
```

```
Experiment_2.html
                      # styles.css
                                  ×
 # styles.css > 😝 input:focus
       .error {
       /* Button Styling */
       button {
           background-color: ■#007bff;
           color: ■white;
           padding: 10px;
           border: none;
           border-radius: 5px;
           cursor: pointer;
           font-size: 16px;
       button:hover {
           background-color: ■#0056b3;
       @media (max-width: 480px) {
           .container {
               width: 90%;
```

Testing:











Results:

- 1. Successfully created a registration form with a cohesive design that adapts seamlessly across different devices.
- 2. Implemented robust client-side validation to handle various input errors and guide users.
- 3. Provided clear error feedback for a better user experience.
- 4. Achieved a user-friendly interface with modern styling and interactive behaviour.

Challenges:

- 1. Designing a responsive layout compatible with a wide range of devices and screen resolutions.
- 2. Handling edge cases in form validation, such as mismatched passwords or invalid email formats.
- 3. Ensuring an intuitive and visually appealing user interface.

Solutions:

- 1. Used CSS Flexbox and media queries to create a responsive and adaptive layout.
- 2. Combined HTML5 validation with custom JavaScript to handle complex validation requirements.
- 3. Styled form elements using modern CSS techniques, ensuring clarity and ease of use.

Conclusion:

The experiment successfully demonstrates the creation of a fully responsive registration form that prioritizes user experience and accessibility. By combining HTML5, CSS, and JavaScript, the form ensures accurate user input and provides a cohesive design across devices. This approach can be applied to real-world projects to enhance form usability and effectiveness.