**Project Name:**

**Registration Form Validator**

**(HTML, CSS,Java Script)**

**Batch 3**

**Group Members List**

**1.Ambili Chandran P R**

**2.Aris P**

**3.Najeeb T A**

**4. Sushanth K**

**5. Vibin P P**

**Project Description:**

The Registration Form Validator project is a fundamental web application built using HTML, CSS, and JavaScript. This project aims to create a user-friendly registration form with robust validation mechanisms to ensure accurate user input.

**Key Features / Requirements:**

1. Interactive UI Design: The project boasts an intuitive user interface designed with HTML and CSS, ensuring a visually appealing and accessible registration form.  
2. Form Validation Logic: JavaScript is employed to implement client-side validation, checking for essential fields such as name, email, password, and date of birth. Real-time feedback is provided to users for immediate correction.  
3. Custom Error Messages: Tailored error messages are displayed for each validation condition,  
offering clear guidance to users on how to rectify their input.  
4. Password Strength Meter: The project includes a dynamic password strength meter that  
assesses the complexity of the entered password, helping users create stronger and more secure  
credentials.  
5. Date Picker: A user-friendly date picker is integrated using JavaScript, enhancing the ease of  
input for the date of birth field.  
6. Submit Functionality: Once all validation conditions are met, the form can be successfully  
submitted, demonstrating the seamless flow of data.

**Outcomes:**  
1. Students will be able to ensuring accurate user input, significance of providing clear and customized error. Messages for different validation conditions.  
2. Students will be able to create their own user-friendly registration strength.

**CSS Code**

The CSS defines the overall style and layout of the form, with the background image, container sizes, input styling, and button hover effects. It includes the following key features:

* Background image is used for setting the background of the page.
* Styling for h4 headings to align them in the center.
* Form input fields are styled with borders, padding, and width adjustments.
* Buttons have a hover effect to change color when the user interacts with them.
* Password strength meter and confirm password validation is integrated for visual feedback on password strength.

   body

   { background-image: url("bg2.jpg");

    background-size: 100% auto ;

    background-position: right;

    background-attachment:fixed;

   }

h4

{

    color:rgb(10, 10, 228);

    text-align: center;

    margin-bottom: 20px;

}

form

{

    position: absolute;

    border: 5px double rgb(24, 181, 216);

    border-radius: 5px;

    padding:40px;

    background-color: whitesmoke;

    margin:0px 0px 0px 0px;

}

.form-size

{

  width:600px;

  margin:0 auto;

}

.container

{

    padding:20px;

    border-width: thick;

    font-family:'Times New Roman', Times, serif;

    font-size:smaller;

    max-width: fit-content;

}

#buton

{

    background-color:rgb(60, 174, 203);

    color:white;

    border-radius: 10px;

    position:absolute;

    width:70px;

    text-align: center;

    height: 30px;

    margin:250px 0px 0px 350px;

}

#buton:hover

{

    background-color:rgb(16, 112, 171);

    color: white;

}

input[type="text"],

input[type="email"],

input[type="number"],

input[type="password"],

input[type="date"],#gender,textarea,#std

 {

    width: 100%;

    padding: 6px;

    margin-bottom: 8px;

    border-radius: 5px;

    border: 1px solid #0f6c74;

    box-sizing: border-box;

}

#save

{

    background-color:rgb(60, 174, 203);

    color:white;

    border-radius: 10px;

    position:absolute;

    width:70px;

    text-align: center;

    height: 30px;

    margin:250px 0px 0px 200px;

}

#save:hover

{

    background-color:rgb(16, 112, 171);

    color: white;

}

textarea

{

   border-radius: 5px;

   margin-bottom: 10px;

}

label

{

    font-weight: bold;

    margin-bottom: 5px;

}

.password-strength

{

   width: 100%;

   margin-top: 10px;

}

.password-meter

{

        height: 4px;

        width: 100%;

        border-radius: 5px;

        background-color: grey;

        margin:10px;

        overflow: hidden;

}

 .meter-bar

{

    height: 100%;

    width: 0;

    background-color: red;

    transition: width 0.3s ease, background-color 0.3s ease;

}

.pwdmeter{

    height: 100%;

    border-radius: 5px;

    transition: width 0.3s;

    color: red;

}

#strengthPwd {

    font-weight: bold;

}

.error {

    color: red;

    font-size: 12px;

}

#regnMessage

{

    color: green;

}

.bgimage{

    background-size: 100% auto;

     background-attachment:fixed;

     background-image: url("bg2.jpg");

     background-position:right;

}

#formErrorMessage

{

    color: red;

    font-size: large;

}

**HTML Code**

This is a basic structure of a registration form with various input fields:

* Student's first and last name.
* Email and mobile number.
* Father and mother name.
* Date of birth (with age limit from 5-18).
* Gender selection via a dropdown.
* Class selection via a dropdown.
* Address field included with no specific validation.
* Password and confirm password fields with validation for complexity and strength.
* Two buttons for "Save" and "Submit."

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Registration Form</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

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

</head>

<body>

    <form novalidate id="studentregistrationForm" onsubmit="return validateForm()">

        <img src="bg.jpg" alt="Form background image">

        <div class="container">

            <div class="form-size">

                <h4>STUDENT REGISTRATION FORM</h4>

                <div class="row mb-3">

                    <div class="col-md-6">

                        <label for="sfname">Student's First Name:</label>

                        <input type="text"  id="sfname"   placeholder="First name" required>

                        <small class="error" id="nameMessage"></small>

                    </div>

                    <div class="col-md-6">

                        <label for="slname">Student's Last Name:</label>

                        <input type="text"  id="slname"   placeholder="Last name" required>

                        <small class="error" id="slnameMessage"></small>

                    </div>

                </div>

                <div class="row mb-3">

                    <div class="col-md-6">

                        <label for="semail">Email:</label>

                        <input type="email" id="semail"  placeholder="Email ID" required>

                        <small class="error" id="emailMessage"></small>

                    </div>

                    <div class="col-md-6">

                        <label for="no">Mobile Number:</label>

                        <input type="text"  id="no" maxlength="10" placeholder="Mobile Number" required>

                        <small class="error" id="noMessage"></small>

                    </div>

                 </div>

                <div class="row mb-3">

                    <div class="col-md-6">

                        <label for="fname">Father Name:</label>

                        <input type="text"  id="fname"  placeholder="Father Name" required>

                        <small class="error" id="fnameMessage"></small>

                    </div>

                    <div class="col-md-6">

                        <label for="mname">Mother Name:</label>

                        <input type="text"  id="mname"  placeholder="Mother name" required>

                        <small class="error" id="mnameMessage"></small>

                    </div>

                </div>

                <div class="row mb-3">

                    <div class="col-md-6">

                        <label for ="date">Birth Date:</label>

                        <input type="date" id="date" placeholder="Date of Birth" required>

                        <small class="error" id="dateMessage"></small>

                    </div>

                    <div class="col-md-6">

                        <label for ="gender">Gender:</label>

                        </br>

                        <select id="gender" name="gender" required>

                            <option value="">Select Gender</option>

                            <option value="male">Male</option>

                            <option value="female">Female</option>

                            <option value="transGender">TransGender</option>

                        </select>

                        <small class="error" id="genderMessage"></small>

                    </div>

                </div>

                <div class="row mb-3">

                    <div class="col-md-6">

                        <label for ="class">Class:</label>

                    </br>

                        <select id="std" name="std" required>

                            <option value="">Select class</option>

                            <option value="First">I</option>

                            <option value="Second">II</option>

                            <option value="Third">III</option>

                            <option value="Forth">IV</option>

                            <option value="Fifth">V</option>

                            <option value="Sixth">VI</option>

                            <option value="Seventh">VII</option>

                            <option value="Eighth">VIII</option>

                            <option value="Ninth">IX</option>

                            <option value="Tenth">X</option>

                        </select>

                        <small class="error" id="classMessage"></small>

                    </div>

                    <div class="col-md-6">

                    <label id="Address">Address:</label>

                    <textarea id="address"  placeholder="House Address" ></textarea>

                </div>

                <div class="row mb-3">

                    <div class="col-md-6">

                        <label for="pwd">Password:</label>

                        <input type="password" id="pwd"  maxlength ="16" placeholder="Password" required>

                        <small class="error" id="pwdMessage"></small>

                        <div class="password-strength">

                            <div class="password-meter" id="pwdmeter">

                                <div class="meter-bar" id="meterbar"></div>

                            </div>

                            <p id="strengthPwd">Strength:</p>

                        </div>

                    </div>

                    <div class="col-md-6">

                        <label for="cpwd">Confirm Password:</label>

                        <input type="password" id="cpwd" placeholder="Confirm Password" required>

                        <small class="error" id="cpwdMessage"></small>

                    </div>

                </div>

                <br>

                <button id="save" type="button" onclick="printform()">Save</button>

                <button id="buton" type="submit">Submit</button>

            </div>

        </div>

        <div id="formErrorMessage" style="display: none; color: red;"></div>

    </form>

    <div class="bgimage"></div>

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

</body>

</html>

**JavaScript Code**

The JavaScript handles input validation for each form field:

* Real-time validation for text fields like first name, last name, and email. Not to enter numbers and special characters in the field.
* Email validation using a regular expression to check for valid domain types (@,.com, .in, No space at first).
* Date validation ensures that the user's age is between 5 and 18 years old. No future date is allowed.
* Mobile number validation checks if exactly 10 digits are provided and no characters allowed.
* Password validation checks for length and complexity (uppercase, lowercase, numbers, special characters).
* The password strength meter dynamically adjusts based on the input strength.
* Confirm password is also checked with the password.
* The form validates all fields before allowing submission, and displays alerts accordingly.

document.getElementById("sfname").addEventListener("input", function (){

    validateSfname();

});

document.getElementById("slname").addEventListener("input", function ()

{

    validateSlname();

});

document.getElementById("semail").addEventListener("input", function () {

    validateEmail();

});

document.getElementById("no").addEventListener("input", function () {

    validateNo();

});

document.getElementById("fname").addEventListener("input", function () {

    validateFname();

});

document.getElementById("mname").addEventListener("input", function () {

    validateMname();

});

document.addEventListener("DOMContentLoaded", function() {

    const today = new Date().toISOString().split('T')[0];

    document.getElementById("date").setAttribute("max", today);

});

document.getElementById('gender').addEventListener('change', function() {

    validateGender();

});

document.getElementById("std").addEventListener("change", function () {

    validateClass();

});

document.getElementById("pwd").addEventListener("input", function () {

    validatePassword();

    validatepwdStrength();

});

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

    validateForm();

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

    if (validateForm()==false) {

        formErrorMessage.textContent=alert("Please fill all the required fields");

        formErrorMessage.style.display="block"

    }

    else

    {

        alert("Registration completed successfully");

    }

});

function validateForm()

{

    const sfnameValid=validateSfname();

    const slnameValid=validateSlname();

    const emailValid=validateEmail();

    const fnameValid=validateFname();

    const mnameValid=validateMname();

    const noValid=validateNo();

    const dateValid=validateDate();

    const genderValid=validateGender();

    const classValid=validateClass();

    const pwdValid=validatePassword();

    const pwdSrtValid=validatepwdStrength();

    return sfnameValid && slnameValid && emailValid && noValid && dateValid &&fnameValid && mnameValid && genderValid && classValid &&pwdValid &&pwdSrtValid ;

}

//  First name validation

function validateSfname()

{

    const sfname = document.getElementById("sfname").value.trim();

    const nameMessage = document.getElementById("nameMessage");

    const namePattern = /^[a-zA-Z\s]+$/;

    if (sfname === "") {

        nameMessage.textContent = "Name required";

        nameMessage.style.display = "block";

        return false;

    } else if (!namePattern.test(sfname)) {

        nameMessage.textContent = "Name can only contain letters and spaces";

        nameMessage.style.display = "block";

        return false;

    } else {

        nameMessage.style.display = "none";

    }

    return true;

}

//  Last name validation

function validateSlname()

{

    const slname = document.getElementById("slname").value.trim();

    const slnameMessage = document.getElementById("slnameMessage");

    const namePattern = /^[a-zA-Z\s]+$/;

    if (slname === "") {

        slnameMessage.textContent = "Name required";

        slnameMessage.style.display = "block";

        return false;

    } else if (!namePattern.test(slname)) {

        slnameMessage.textContent = "Name can only contain letters and spaces";

        slnameMessage.style.display = "block";

        return false;

    } else {

        slnameMessage.style.display = "none";

    }

    return true;

}

// Email validation

function validateEmail() {

    const semail = document.getElementById("semail").value.trim();

    const emailMessage = document.getElementById("emailMessage");

    const emailPattern = /^[^\s@]+@[^\s@]+\.(com|in)$/;

    if (semail==="")

    {

        emailMessage.textContent = "Email required";

        emailMessage.style.display = "block";

        return false;

    } else if (!emailPattern.test(semail)) {

        emailMessage.textContent= "Invalid email";

        emailMessage.style.display = "block";

        console.log(semail);

        return false;

     }

     else  {

        emailMessage.style.display = "none";

     }

    return true;

}

// Mobile number validation

function validateNo() {

    const no = document.getElementById("no").value.trim();

    const noMessage = document.getElementById("noMessage");

    // Regular expression to match exactly 10 digits

    const nopattern = /^\d{10}$/;

    if (no === "") {

        noMessage.textContent = "Mobile number is required.";

        noMessage.style.display = "block";

        return false;

    }

    // Check if the input contains only digits and is exactly 10 digits long

    else if (!nopattern.test(no)) {

        noMessage.textContent = "Please enter a valid 10-digit mobile number.";

        noMessage.style.display = "block";

        return false;

    }

    else {

        noMessage.style.display = "none";

    }

    return true;

}

// Birth date Validation

function validateDate()

{

    const date=document.getElementById("date").value.trim();

    const dateMessage=document.getElementById("dateMessage");

    if(date==="")

    {

        dateMessage.textContent="Date of birth required";

        dateMessage.style.display = "block";

        return false;

    }

    const dob=new Date(date);

    const today = new Date();

    today.setHours(0, 0, 0, 0);

    if (dob >= today) {

        dateMessage.textContent = "Date must be in the past.";

        dateMessage.style.display = "block";

        return false;

    }

    const age = today.getFullYear() - dob.getFullYear();

    const month = today.getMonth() - dob.getMonth();

    const day = today.getDate() - dob.getDate();

    if (month < 0 || (month === 0 && day < 0)) {

        age--;

    }

    if (age < 5){

        dateMessage.textContent = "Age must be greater or equal to 5.";

        dateMessage.style.display = "block"; // Show error message

        return false; // Prevent form submission

    }

        else if (age > 18) {

        dateMessage.textContent = "Age not exceeds 18 years.";

        dateMessage.style.display = "block"; // Show error message

        return false; // Prevent form submission

    }

     else {

        dateMessage.style.display = "none"; // Hide error message if valid

    }

    return true;

}

// Father name validation

function validateFname()

{

    const fname = document.getElementById("fname").value.trim();

    const fnameMessage = document.getElementById("fnameMessage");

    const namePattern = /^[a-zA-Z\s]+$/;

    if (fname === "") {

        fnameMessage.textContent = "Father Name required";

        fnameMessage.style.display = "block";

        return false;

    } else if (!namePattern.test(fname)) {

        fnameMessage.textContent = "Name can only contain letters and spaces";

        fnameMessage.style.display = "block";

        return false;

    } else {

        fnameMessage.style.display = "none";

    }

    return true;

}

//Mother name validation

function validateMname()

{

    const mname = document.getElementById("mname").value.trim();

    const mnameMessage = document.getElementById("mnameMessage");

    const namePattern = /^[a-zA-Z\s]+$/;

    if (mname === "") {

        mnameMessage.textContent = "Mother Name required";

        mnameMessage.style.display = "block";

        return false;

    } else if (!namePattern.test(mname)) {

        mnameMessage.textContent = "Name can only contain letters and spaces";

        mnameMessage.style.display = "block";

        return false;

    } else {

        mnameMessage.style.display = "none";

    }

    return true;

}

// Gender Validation

function validateGender()

{

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

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

    if (gender.value === "") {

        genderMessage.textContent = 'Please select your gender.';

        genderMessage.style.display = "block";

        return false;

    }

    else {

        genderMessage.style.display = "none";

    }

    return true;

}

// Class validation

function validateClass()

{

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

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

    if (std.value === "") {

        classMessage.textContent ="Please select your class";

        classMessage.style.display = "block";

        return false;

    }

    else {

        classMessage.style.display ="none";

    }

    return true;

}

// Password Validation

function validatePassword()

{

    const pwd = document.getElementById("pwd").value.trim();

   const cpwd = document.getElementById("cpwd").value.trim();

    const pwdMessage = document.getElementById("pwdMessage");

   const cpwdMessage = document.getElementById("cpwdMessage");

    // Name validation (letters and spaces only)

    const pwdPattern = /^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[!@#$%^&\*()\_+])[A-Za-z\d!@#$%^&\*()\_+]{8,}$/;

    if (pwd==="")  {

        pwdMessage.textContent = "Password required";

        pwdMessage.style.display = "block";

        return false;

     }

    else if(pwd.length <8 || pwd.length >16)

        {

            pwdMessage.textContent="Password should contain minimum 8 and maximum 16 characters";

            pwdMessage.style.display = "block";

            return false;

        }

        else if (!pwdPattern.test(pwd)) {

            pwdMessage.textContent = "Password should contain uppercase,lowercase,number and special character";

           pwdMessage.style.display = "block";

            return false;

        }

        else{

                  pwdMessage.style.display ="none";

              }

     if(cpwd==="") {

        cpwdMessage.textContent = "Confirm password required";

        cpwdMessage.style.display = "block";

        return false;

    }

  else if(pwd!==cpwd)

        {

            cpwdMessage.textContent="Password does not match with confirm password";

            cpwdMessage.style.display = "block";

            return false;

        }

        else{

            cpwdMessage.style.display = "none";

            }

}

// Password strength validation

function validatepwdStrength() {

    const pwd = document.getElementById("pwd").value.trim();

    const meterbar = document.getElementById("meterbar");

    const strengthPwd = document.getElementById("strengthPwd");

    let strength = 0;

    if (pwd.length >= 8 ) strength += 1;

                if (/[a-z]/.test(pwd)) strength += 1;

                if (/[A-Z]/.test(pwd)) strength += 1;

                 if (/\d/.test(pwd)) strength += 1;

               if (/[^a-zA-Z\d]/.test(pwd)) strength += 1;

    let meterWidth = "0%";

    let meterColor = "red";

    let strengthMsg = "week";

    // Adjust based on the score

    if (strength === 1) {

        meterWidth = "20%";

        meterColor = "red";

        strengthMsg = "Strength: Very Weak";

    } else if (strength === 2) {

        meterWidth = "40%";

        meterColor = "orange";

        strengthMsg = "Strength: Weak";

    } else if (strength === 3) {

        meterWidth = "60%";

        meterColor = "yellow";

        strengthMsg = "Strength: Moderate";

    } else if (strength === 4) {

        meterWidth = "80%";

        meterColor = "lightgreen";

        strengthMsg = "Strength: Strong";

    } else if (strength === 5) {

        meterWidth = "100%";

        meterColor = "green";

        strengthMsg = "Strength: Very Strong";

    }

    console.log("Strength:", strength);

    console.log("Meter Width:", meterWidth);

    console.log("Meter Color:", meterColor);

    console.log("Strength Message:", strengthMsg);

    // Update the meter bar and text

    meterbar.style.width = meterWidth;

    meterbar.style.backgroundColor = meterColor;

    strengthPwd.textContent =  strengthMsg;

}

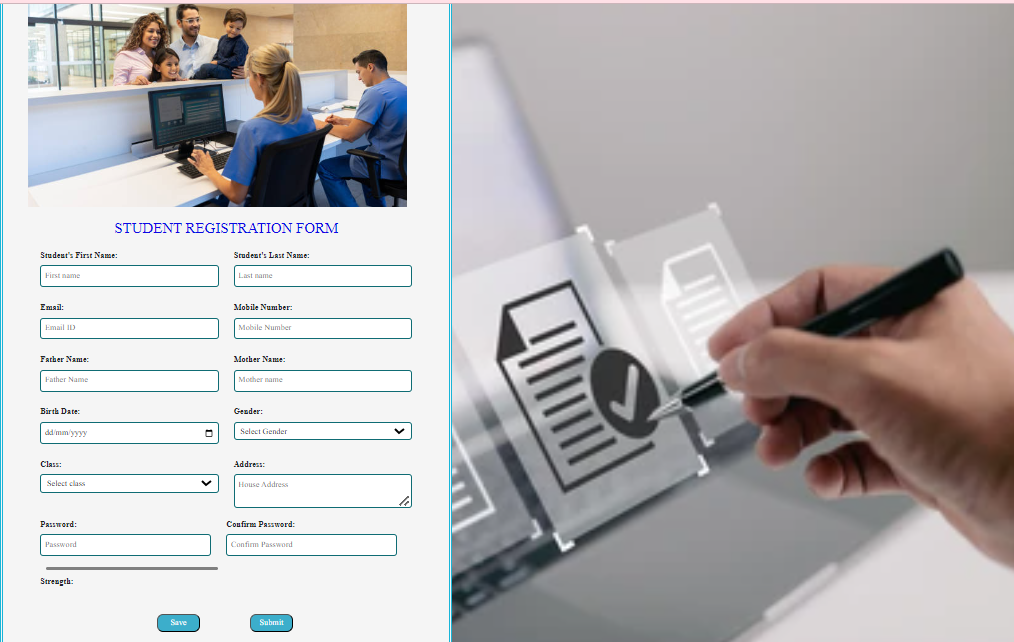
//Save Form

function printform() {

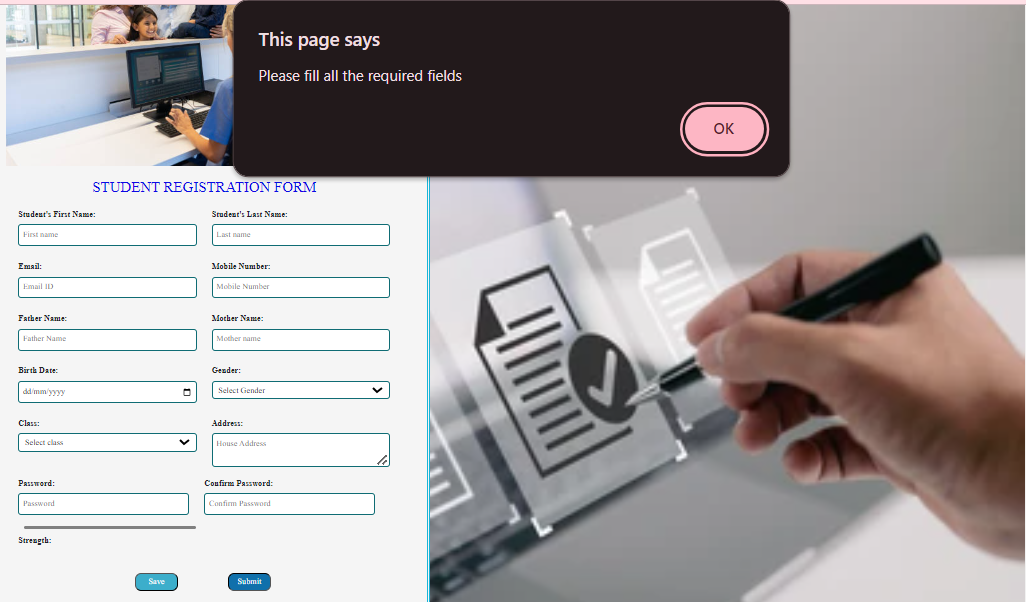
    window.print();

}

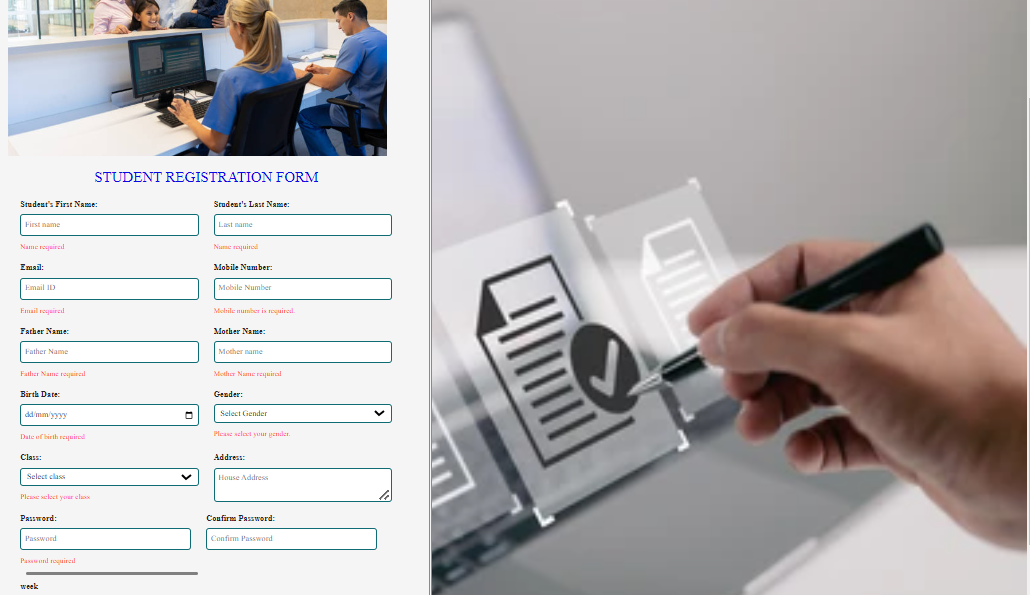
Home Page



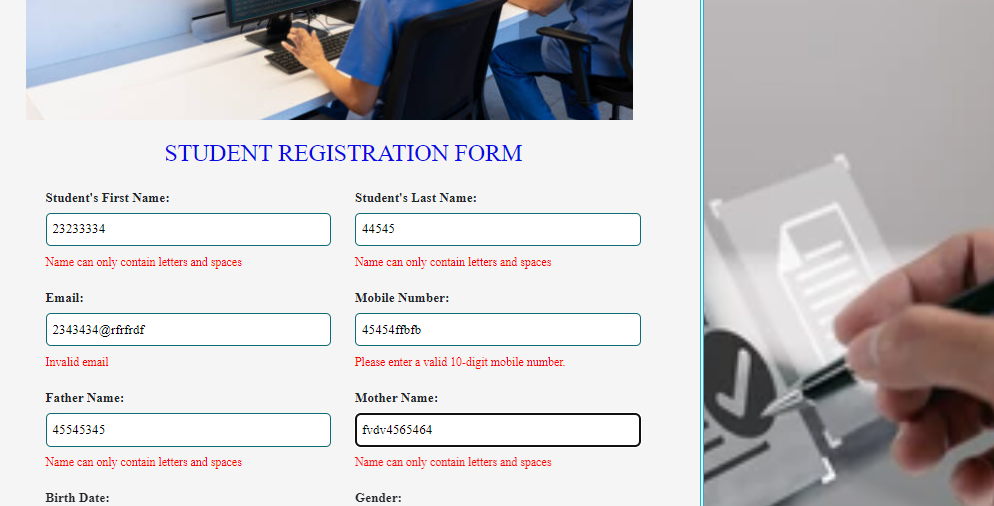
Validation if submit blank details.



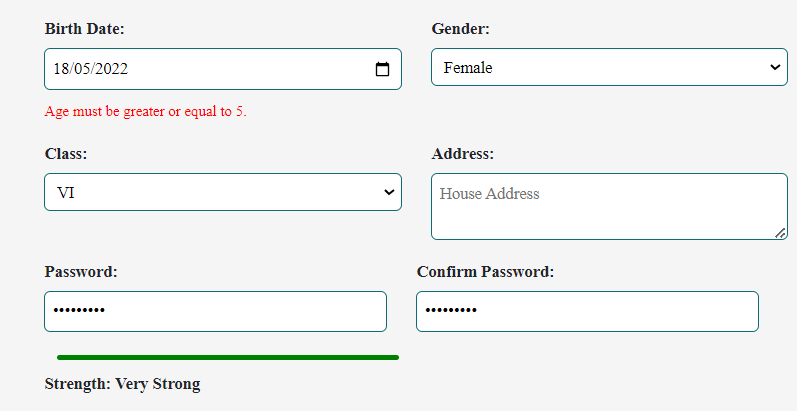
Message to enter each input field correctly.



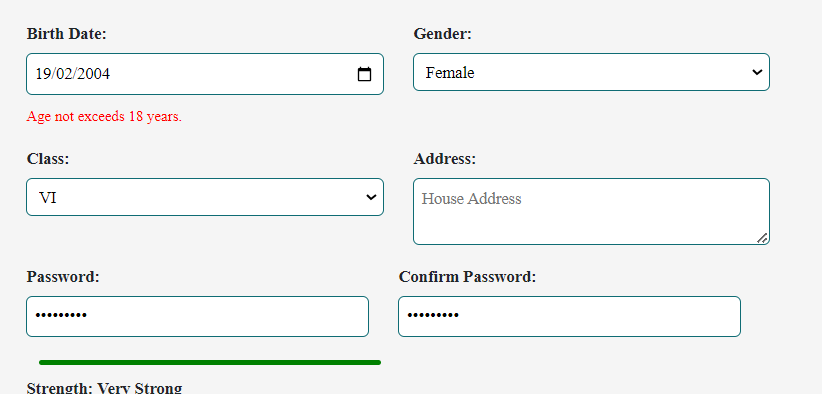
1. Student name not allowed numbers and special characters in the name field. If entered “name can only contain letters and spaces message displays”.
2. Email validation as to include @, .com and .in in the input field. No space allowed at first.
3. Mobile input field allows only 10 digits and no characters and special characters allowed.
4. Father name and mother name not allowed numbers and special characters in the name field. If entered “name can only contain letters and spaces message displays”.



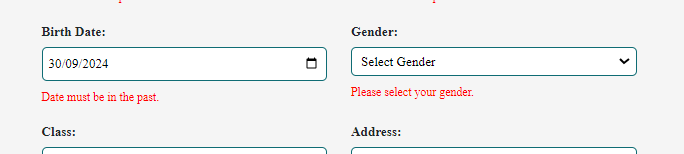
1. Birth date field validation as not to allow the age below 5 years. If entered message shows as “Age must be greater or equal to 5” message shows.



1. Birth date field validation as not to allow the age above 18 years. If entered message shows as “Age not exceeds 18 years” message shows.

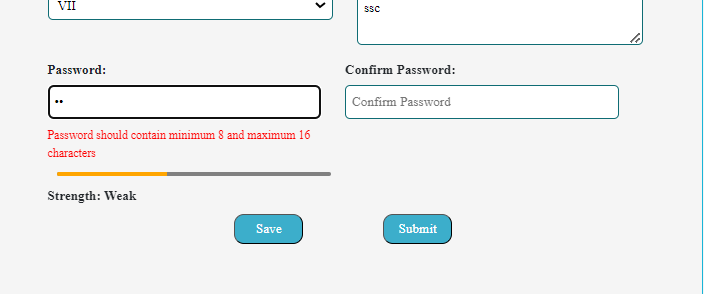


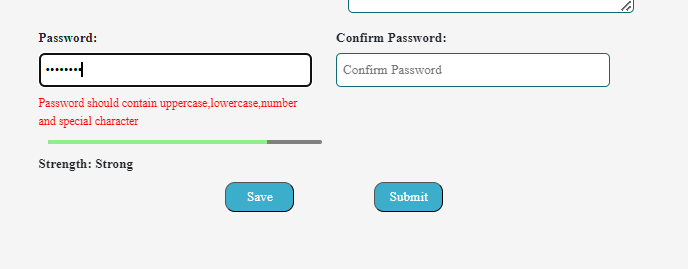
1. Birth date field validation as not to allow the future date. If entered message shows as “Date must be in the past” message shows.

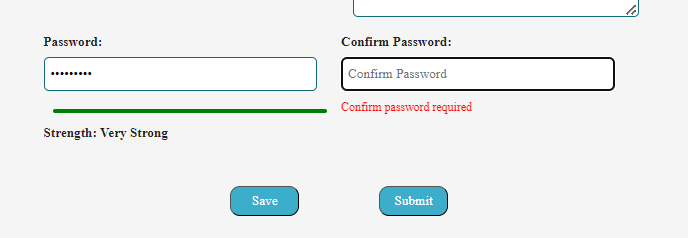


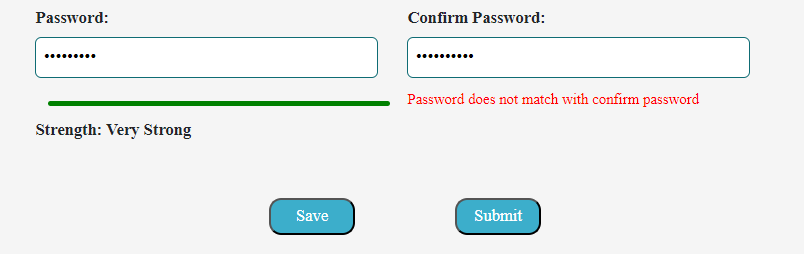
1. Password validation:-

* A meter bar is inserted to check the strength of the password.
* “Password should contain minimum 8 and maximum 16 characters” message shows if the password length is below 8 digit.
* “Password should contain uppercase, lowercase, number and special characters” message shows if we not entered any one of the above.
* According to the entry strength-meter is updating as very weak, weak, moderate, strong and very strong.
* Confirm password message shows as “Password not match with confirm password” if the entered confirm password is wrong.









Submit the form:

Registration completed successfully message shows at the time of submit , if the input fields are correct.

