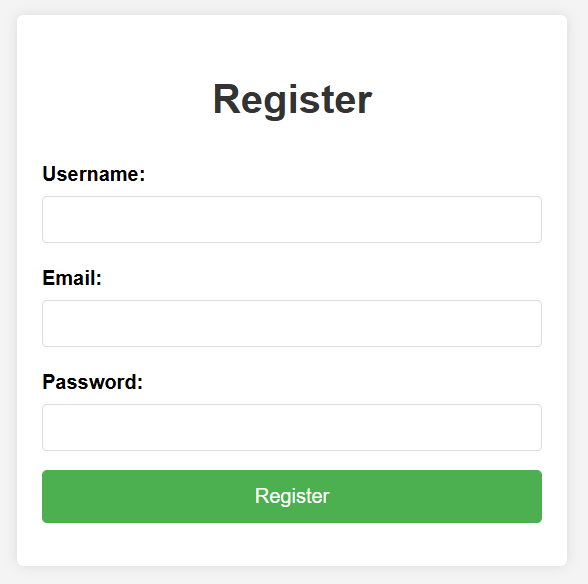
**Q1: Can you create a simple HTML registration form with fields for username, email, and password?**

****

<!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>

<style>

body {

font-family: Arial, sans-serif;

line-height: 1.6;

margin: 0;

padding: 20px;

background-color: #f4f4f4;

}

.container {

max-width: 400px;

margin: 0 auto;

background: #fff;

padding: 20px;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h1 {

text-align: center;

color: #333;

}

.form-group {

margin-bottom: 15px;

}

label {

display: block;

margin-bottom: 5px;

font-weight: bold;

}

input[type="text"],

input[type="email"],

input[type="password"] {

width: 100%;

padding: 10px;

border: 1px solid #ddd;

border-radius: 4px;

box-sizing: border-box;

}

input[type="submit"] {

background: #4CAF50;

color: white;

padding: 12px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

width: 100%;

font-size: 16px;

}

input[type="submit"]:hover {

background: #45a049;

}

</style>

</head>

<body>

<div class="container">

<h1>Register</h1>

<form action="/register" method="POST">

<div class="form-group">

<label for="username">Username:</label>

<input type="text" id="username" name="username" required>

</div>

<div class="form-group">

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

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

</div>

<div class="form-group">

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

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

</div>

<div class="form-group">

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

</div>

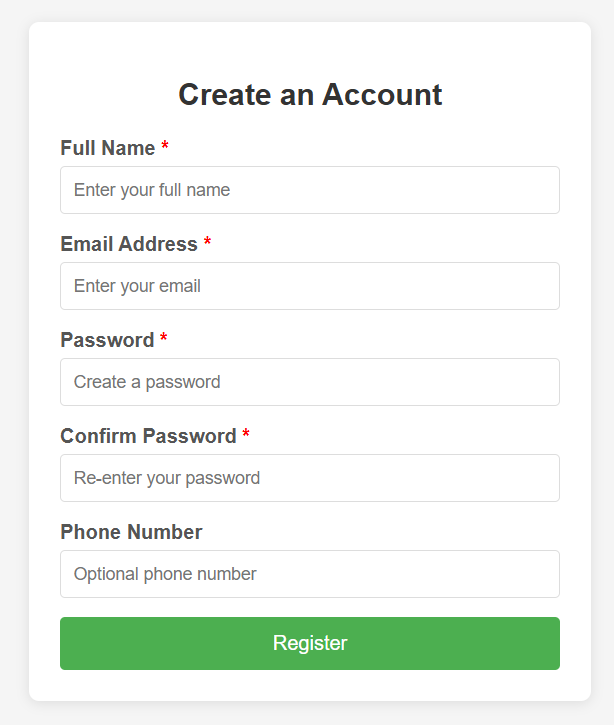
</form>

</div>

</body>

</html>

**Q2: Generate a user registration form with proper input types, required fields, and basic styling using inline CSS or basic classes.**

****

<!DOCTYPE html>

<html>

<head>

<title>User Registration</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f5f5f5;

margin: 0;

padding: 20px;

}

.registration-form {

max-width: 400px;

margin: 0 auto;

background: white;

padding: 25px;

border-radius: 8px;

box-shadow: 0 2px 10px rgba(0,0,0,0.1);

}

.form-title {

text-align: center;

color: #333;

margin-bottom: 20px;

}

.form-group {

margin-bottom: 15px;

}

.form-group label {

display: block;

margin-bottom: 5px;

font-weight: 600;

color: #555;

}

.form-control {

width: 100%;

padding: 10px;

border: 1px solid #ddd;

border-radius: 4px;

box-sizing: border-box;

font-size: 14px;

}

.form-control:focus {

border-color: #4CAF50;

outline: none;

}

.submit-btn {

background-color: #4CAF50;

color: white;

border: none;

padding: 12px;

width: 100%;

border-radius: 4px;

cursor: pointer;

font-size: 16px;

transition: background-color 0.3s;

}

.submit-btn:hover {

background-color: #45a049;

}

.required:after {

content: " \*";

color: red;

}

</style>

</head>

<body>

<div class="registration-form">

<h2 class="form-title">Create an Account</h2>

<form action="/register" method="post">

<div class="form-group">

<label for="name" class="required">Full Name</label>

<input type="text" id="name" name="name" class="form-control" required

placeholder="Enter your full name" minlength="2">

</div>

<div class="form-group">

<label for="email" class="required">Email Address</label>

<input type="email" id="email" name="email" class="form-control" required

placeholder="Enter your email">

</div>

<div class="form-group">

<label for="password" class="required">Password</label>

<input type="password" id="password" name="password" class="form-control" required

placeholder="Create a password" minlength="8">

</div>

<div class="form-group">

<label for="confirm-password" class="required">Confirm Password</label>

<input type="password" id="confirm-password" name="confirm-password" class="form-control" required

placeholder="Re-enter your password">

</div>

<div class="form-group">

<label for="phone">Phone Number</label>

<input type="tel" id="phone" name="phone" class="form-control"

placeholder="Optional phone number" pattern="[0-9]{10}">

</div>

<button type="submit" class="submit-btn">Register</button>

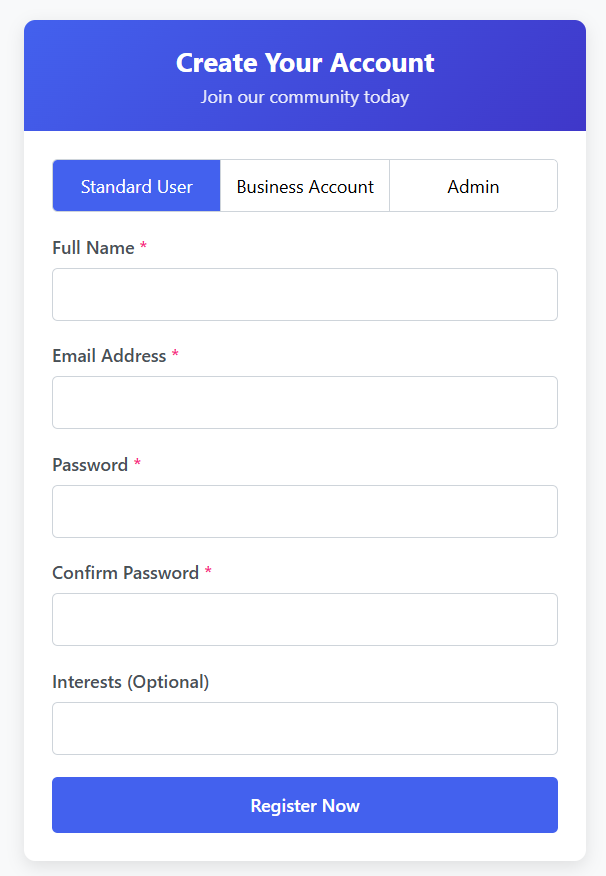
</form>

</div>

</body>

</html>

**Q3: Create a styled user registration interface using HTML, CSS, and JavaScript that includes client-side validation and is easy to reuse for different user types.**

****

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Universal Registration Form</title>

<style>

:root {

--primary-color: #4361ee;

--secondary-color: #3f37c9;

--error-color: #f72585;

--success-color: #4cc9f0;

--light-gray: #f8f9fa;

--dark-gray: #adb5bd;

}

\* {

box-sizing: border-box;

margin: 0;

padding: 0;

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

}

body {

background-color: var(--light-gray);

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

padding: 20px;

}

.registration-container {

width: 100%;

max-width: 500px;

background: white;

border-radius: 10px;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);

overflow: hidden;

}

.form-header {

background: linear-gradient(135deg, var(--primary-color), var(--secondary-color));

color: white;

padding: 20px;

text-align: center;

}

.form-header h2 {

margin-bottom: 5px;

}

.form-header p {

opacity: 0.9;

}

.form-body {

padding: 25px;

}

.form-group {

margin-bottom: 20px;

position: relative;

}

.form-group label {

display: block;

margin-bottom: 8px;

font-weight: 500;

color: #495057;

}

.form-group label.required:after {

content: " \*";

color: var(--error-color);

}

.form-control {

width: 100%;

padding: 12px 15px;

border: 1px solid #ced4da;

border-radius: 5px;

font-size: 16px;

transition: all 0.3s;

}

.form-control:focus {

border-color: var(--primary-color);

box-shadow: 0 0 0 3px rgba(67, 97, 238, 0.2);

outline: none;

}

.error-message {

color: var(--error-color);

font-size: 14px;

margin-top: 5px;

display: none;

}

.form-control.error {

border-color: var(--error-color);

}

.form-control.success {

border-color: var(--success-color);

}

.user-type-selector {

display: flex;

margin-bottom: 20px;

border-radius: 5px;

overflow: hidden;

border: 1px solid #ced4da;

}

.user-type-option {

flex: 1;

text-align: center;

padding: 12px;

cursor: pointer;

transition: all 0.3s;

background-color: white;

}

.user-type-option:not(:last-child) {

border-right: 1px solid #ced4da;

}

.user-type-option.selected {

background-color: var(--primary-color);

color: white;

}

.submit-btn {

width: 100%;

padding: 14px;

background-color: var(--primary-color);

color: white;

border: none;

border-radius: 5px;

font-size: 16px;

font-weight: 500;

cursor: pointer;

transition: background-color 0.3s;

}

.submit-btn:hover {

background-color: var(--secondary-color);

}

.hidden {

display: none;

}

@media (max-width: 576px) {

.user-type-selector {

flex-direction: column;

}

.user-type-option:not(:last-child) {

border-right: none;

border-bottom: 1px solid #ced4da;

}

}

</style>

</head>

<body>

<div class="registration-container">

<div class="form-header">

<h2>Create Your Account</h2>

<p>Join our community today</p>

</div>

<div class="form-body">

<div class="user-type-selector">

<div class="user-type-option selected" data-user-type="standard">Standard User</div>

<div class="user-type-option" data-user-type="business">Business Account</div>

<div class="user-type-option" data-user-type="admin">Admin</div>

</div>

<form id="registrationForm">

<div class="form-group">

<label for="fullName" class="required">Full Name</label>

<input type="text" id="fullName" class="form-control" required>

<div class="error-message" id="nameError">Please enter a valid name (min 2 characters)</div>

</div>

<div class="form-group">

<label for="email" class="required">Email Address</label>

<input type="email" id="email" class="form-control" required>

<div class="error-message" id="emailError">Please enter a valid email address</div>

</div>

<div class="form-group">

<label for="password" class="required">Password</label>

<input type="password" id="password" class="form-control" required>

<div class="error-message" id="passwordError">Password must be at least 8 characters</div>

</div>

<div class="form-group">

<label for="confirmPassword" class="required">Confirm Password</label>

<input type="password" id="confirmPassword" class="form-control" required>

<div class="error-message" id="confirmPasswordError">Passwords do not match</div>

</div>

<!-- Standard User Fields -->

<div class="user-specific-fields" id="standardFields">

<div class="form-group">

<label for="interests">Interests (Optional)</label>

<input type="text" id="interests" class="form-control">

</div>

</div>

<!-- Business User Fields -->

<div class="user-specific-fields hidden" id="businessFields">

<div class="form-group">

<label for="companyName" class="required">Company Name</label>

<input type="text" id="companyName" class="form-control">

</div>

<div class="form-group">

<label for="taxId">Tax ID (Optional)</label>

<input type="text" id="taxId" class="form-control">

</div>

</div>

<!-- Admin User Fields -->

<div class="user-specific-fields hidden" id="adminFields">

<div class="form-group">

<label for="adminKey" class="required">Admin Key</label>

<input type="password" id="adminKey" class="form-control">

<div class="error-message" id="adminKeyError">Admin key is required</div>

</div>

</div>

<button type="submit" class="submit-btn">Register Now</button>

</form>

</div>

</div>

<script>

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

// User type selection

const userTypeOptions = document.querySelectorAll('.user-type-option');

const userSpecificFields = {

standard: document.getElementById('standardFields'),

business: document.getElementById('businessFields'),

admin: document.getElementById('adminFields')

};

userTypeOptions.forEach(option => {

option.addEventListener('click', function() {

// Update UI

userTypeOptions.forEach(opt => opt.classList.remove('selected'));

this.classList.add('selected');

// Hide all specific fields

Object.values(userSpecificFields).forEach(field => {

field.classList.add('hidden');

});

// Show fields for selected user type

const userType = this.getAttribute('data-user-type');

userSpecificFields[userType].classList.remove('hidden');

});

});

// Form validation

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

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

e.preventDefault();

// Reset error states

resetErrors();

// Validate form

const isValid = validateForm();

if (isValid) {

// Form is valid - proceed with submission

const userType = document.querySelector('.user-type-option.selected').getAttribute('data-user-type');

const formData = {

userType: userType,

fullName: document.getElementById('fullName').value,

email: document.getElementById('email').value,

password: document.getElementById('password').value

};

// Add user-specific data

if (userType === 'standard') {

formData.interests = document.getElementById('interests').value;

} else if (userType === 'business') {

formData.companyName = document.getElementById('companyName').value;

formData.taxId = document.getElementById('taxId').value;

} else if (userType === 'admin') {

formData.adminKey = document.getElementById('adminKey').value;

}

console.log('Form data to submit:', formData);

alert('Registration successful! Check console for form data.');

// In a real app, you would send this data to your server here

}

});

function resetErrors() {

document.querySelectorAll('.form-control').forEach(input => {

input.classList.remove('error', 'success');

});

document.querySelectorAll('.error-message').forEach(error => {

error.style.display = 'none';

});

}

function validateForm() {

let isValid = true;

const userType = document.querySelector('.user-type-option.selected').getAttribute('data-user-type');

// Validate full name

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

if (fullName.value.trim().length < 2) {

showError(fullName, 'nameError');

isValid = false;

} else {

showSuccess(fullName);

}

// Validate email

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

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

if (!emailRegex.test(email.value)) {

showError(email, 'emailError');

isValid = false;

} else {

showSuccess(email);

}

// Validate password

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

if (password.value.length < 8) {

showError(password, 'passwordError');

isValid = false;

} else {

showSuccess(password);

}

// Validate confirm password

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

if (confirmPassword.value !== password.value) {

showError(confirmPassword, 'confirmPasswordError');

isValid = false;

} else if (password.value.length >= 8) {

showSuccess(confirmPassword);

}

// Validate user-specific fields

if (userType === 'business') {

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

if (companyName.value.trim().length < 2) {

showError(companyName, 'Please enter a valid company name');

isValid = false;

} else {

showSuccess(companyName);

}

} else if (userType === 'admin') {

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

if (adminKey.value.trim().length === 0) {

showError(adminKey, 'adminKeyError');

isValid = false;

} else {

showSuccess(adminKey);

}

}

return isValid;

}

function showError(inputElement, errorElementId) {

inputElement.classList.add('error');

const errorElement = document.getElementById(errorElementId);

errorElement.style.display = 'block';

}

function showSuccess(inputElement) {

inputElement.classList.add('success');

}

// Real-time validation for better UX

document.getElementById('fullName').addEventListener('input', function() {

if (this.value.trim().length >= 2) {

showSuccess(this);

document.getElementById('nameError').style.display = 'none';

}

});

document.getElementById('email').addEventListener('input', function() {

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

if (emailRegex.test(this.value)) {

showSuccess(this);

document.getElementById('emailError').style.display = 'none';

}

});

document.getElementById('password').addEventListener('input', function() {

if (this.value.length >= 8) {

showSuccess(this);

document.getElementById('passwordError').style.display = 'none';

// Also validate confirm password if it has value

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

if (confirmPassword.value && confirmPassword.value === this.value) {

showSuccess(confirmPassword);

document.getElementById('confirmPasswordError').style.display = 'none';

}

}

});

document.getElementById('confirmPassword').addEventListener('input', function() {

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

if (this.value === password && password.length >= 8) {

showSuccess(this);

document.getElementById('confirmPasswordError').style.display = 'none';

}

});

});

</script>

</body>

</html>