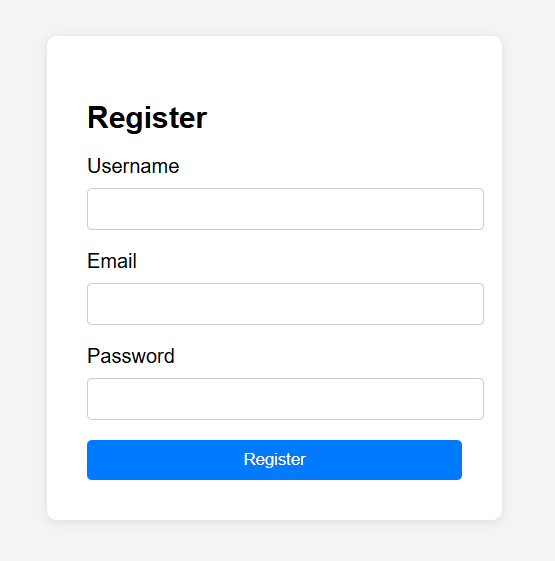
**Q1: Can you create a simple HTML registration form with fields for username, email, and password?**



<!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;

background-color: #f4f4f4;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.registration-form {

background: #fff;

padding: 2rem;

border-radius: 8px;

box-shadow: 0 2px 8px rgba(0,0,0,0.1);

width: 300px;

}

.registration-form h2 {

margin-bottom: 1rem;

}

.registration-form label {

display: block;

margin-bottom: 0.5rem;

}

.registration-form input[type="text"],

.registration-form input[type="email"],

.registration-form input[type="password"] {

width: 100%;

padding: 0.5rem;

margin-bottom: 1rem;

border: 1px solid #ccc;

border-radius: 4px;

}

.registration-form button {

width: 100%;

padding: 0.5rem;

background: #007bff;

color: #fff;

border: none;

border-radius: 4px;

cursor: pointer;

}

.registration-form button:hover {

background: #0056b3;

}

</style>

</head>

<body>

<form class="registration-form">

<h2>Register</h2>

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

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

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

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

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

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

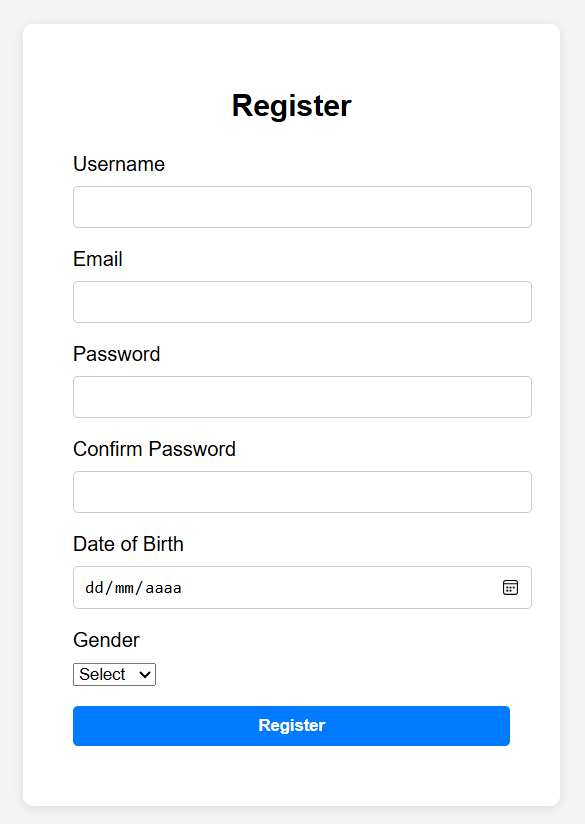
<button type="submit">Register</button>

</form>

</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 lang="en">

<head>

<meta charset="UTF-8">

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

<title>User Registration</title>

<style>

body {

font-family: Arial, sans-serif;

background: #f4f4f4;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.registration-form {

background: #fff;

padding: 2rem 2.5rem;

border-radius: 8px;

box-shadow: 0 2px 8px rgba(0,0,0,0.1);

width: 350px;

}

.registration-form h2 {

margin-bottom: 1.5rem;

text-align: center;

}

.form-group {

margin-bottom: 1rem;

}

.form-group label {

display: block;

margin-bottom: 0.5rem;

}

.form-group input {

width: 100%;

padding: 0.5rem;

border: 1px solid #ccc;

border-radius: 4px;

}

.form-group input[type="submit"] {

background: #007bff;

color: #fff;

border: none;

cursor: pointer;

font-weight: bold;

transition: background 0.2s;

}

.form-group input[type="submit"]:hover {

background: #0056b3;

}

</style>

</head>

<body>

<form class="registration-form">

<h2>Register</h2>

<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 minlength="6">

</div>

<div class="form-group">

<label for="confirm\_password">Confirm Password</label>

<input type="password" id="confirm\_password" name="confirm\_password" required minlength="6">

</div>

<div class="form-group">

<label for="dob">Date of Birth</label>

<input type="date" id="dob" name="dob" required>

</div>

<div class="form-group">

<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>

</select>

</div>

<div class="form-group">

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

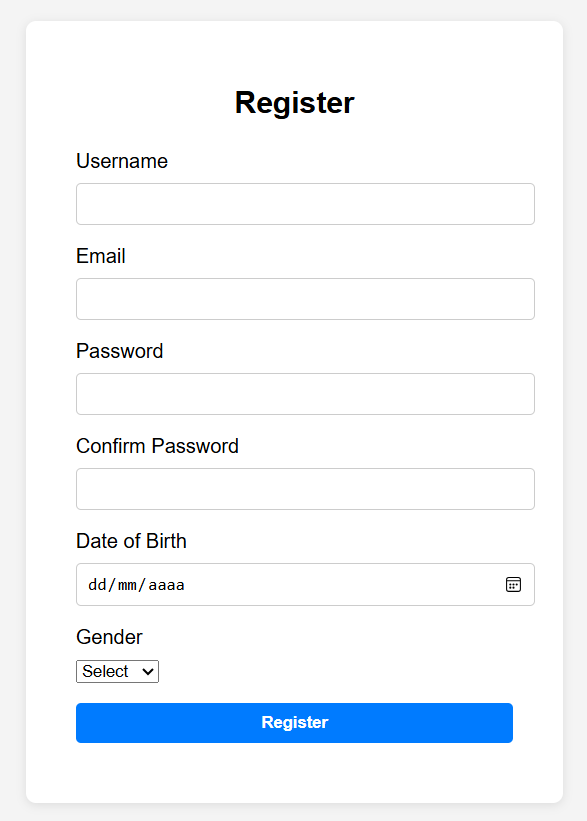
</div>

</form>

</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>User Registration</title>

<style>

body {

font-family: Arial, sans-serif;

background: #f4f4f4;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.registration-form {

background: #fff;

padding: 2rem 2.5rem;

border-radius: 8px;

box-shadow: 0 2px 8px rgba(0,0,0,0.1);

width: 350px;

}

.registration-form h2 {

margin-bottom: 1.5rem;

text-align: center;

}

.form-group {

margin-bottom: 1rem;

}

.form-group label {

display: block;

margin-bottom: 0.5rem;

}

.form-group input {

width: 100%;

padding: 0.5rem;

border: 1px solid #ccc;

border-radius: 4px;

}

.form-group input[type="submit"] {

background: #007bff;

color: #fff;

border: none;

cursor: pointer;

font-weight: bold;

transition: background 0.2s;

}

.form-group input[type="submit"]:hover {

background: #0056b3;

}

</style>

</head>

<body>

<form class="registration-form" id="registrationForm">

<h2 id="formTitle">Register</h2>

<div id="formFields"></div>

<div class="form-group">

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

</div>

</form>

<script>

// Reusable form fields for different user types

const userTypes = {

default: [

{ label: 'Username', id: 'username', type: 'text', required: true },

{ label: 'Email', id: 'email', type: 'email', required: true },

{ label: 'Password', id: 'password', type: 'password', required: true, minlength: 6 },

{ label: 'Confirm Password', id: 'confirm\_password', type: 'password', required: true, minlength: 6 },

{ label: 'Date of Birth', id: 'dob', type: 'date', required: true },

{ label: 'Gender', id: 'gender', type: 'select', required: true, options: [

{ value: '', text: 'Select' },

{ value: 'male', text: 'Male' },

{ value: 'female', text: 'Female' },

{ value: 'other', text: 'Other' }

] }

],

// Example: add more user types here

admin: [

{ label: 'Admin Code', id: 'admin\_code', type: 'text', required: true }

]

};

// Function to render form fields

function renderFormFields(userType = 'default') {

const fields = [...userTypes.default];

if (userType !== 'default' && userTypes[userType]) {

fields.push(...userTypes[userType]);

document.getElementById('formTitle').textContent = userType.charAt(0).toUpperCase() + userType.slice(1) + ' Registration';

} else {

document.getElementById('formTitle').textContent = 'Register';

}

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

formFields.innerHTML = '';

fields.forEach(field => {

const group = document.createElement('div');

group.className = 'form-group';

const label = document.createElement('label');

label.setAttribute('for', field.id);

label.textContent = field.label;

group.appendChild(label);

let input;

if (field.type === 'select') {

input = document.createElement('select');

input.id = field.id;

input.name = field.id;

if (field.required) input.required = true;

field.options.forEach(opt => {

const option = document.createElement('option');

option.value = opt.value;

option.textContent = opt.text;

input.appendChild(option);

});

} else {

input = document.createElement('input');

input.type = field.type;

input.id = field.id;

input.name = field.id;

if (field.required) input.required = true;

if (field.minlength) input.minLength = field.minlength;

}

group.appendChild(input);

group.appendChild(document.createElement('div')).className = 'error-message';

formFields.appendChild(group);

});

}

// Client-side validation

document.getElementById('registrationForm').addEventListener('submit', function(e) {

e.preventDefault();

let valid = true;

const form = e.target;

// Clear previous errors

form.querySelectorAll('.error-message').forEach(div => div.textContent = '');

// Password match validation

const password = form.querySelector('#password');

const confirm = form.querySelector('#confirm\_password');

if (password && confirm && password.value !== confirm.value) {

valid = false;

confirm.parentElement.querySelector('.error-message').textContent = 'Passwords do not match.';

}

// Email format validation

const email = form.querySelector('#email');

if (email && !/^\S+@\S+\.\S+$/.test(email.value)) {

valid = false;

email.parentElement.querySelector('.error-message').textContent = 'Invalid email address.';

}

// Additional validations can be added here

if (valid) {

alert('Registration successful!');

form.reset();

}

});

// Example: render default form, or pass 'admin' to render admin form

renderFormFields();

// To render for admin: renderFormFields('admin');

</script>

</body>

</html>