TypeScript

import { Component } from '@angular/core';

import { FormBuilder, FormGroup, Validators } from '@angular/forms';

@Component({

selector: 'app-student-registration',

template: `

<form [formGroup]="studentForm" (ngSubmit)="onSubmit()">

<div>

<label for="fullName">Full Name:</label>

<input type="text" id="fullName" formControlName="fullName">

<div \*ngIf="studentForm.get('fullName')?.hasError('required')">Full Name is required.</div>

</div>

<div>

<label for="mobile">Mobile:</label>

<input type="text" id="mobile" formControlName="mobile">

<div \*ngIf="studentForm.get('mobile')?.hasError('required')">Mobile number is required.</div>

<div \*ngIf="studentForm.get('mobile')?.hasError('pattern')">Invalid mobile number format.</div>

</div>

<div>

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

<input type="email" id="email" formControlName="email">

<div \*ngIf="studentForm.get('email')?.hasError('required')">Email is required.</div>

<div \*ngIf="studentForm.get('email')?.hasError('email')">Invalid email format.</div>

</div>

<div>

<label for="country">Country:</label>

<select id="country" formControlName="country">

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

<option value="India">India</option>

<option value="USA">USA</option>

<option value="UK">UK</option>

</select>

<div \*ngIf="studentForm.get('country')?.hasError('required')">Country is required.</div>

</div>

<div>

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

<input type="radio" id="male" value="male" formControlName="gender"> Male

<input type="radio" id="female" value="female" formControlName="gender"> Female

<div \*ngIf="studentForm.get('gender')?.hasError('required')">Gender is required.</div>

</div>

<div>

<label for="remarks">Remarks:</label>

<textarea id="remarks" formControlName="remarks"></textarea>

</div>

<button type="submit" [disabled]="!studentForm.valid">Submit</button>

</form>

`

})

export class StudentRegistrationComponent {

studentForm: FormGroup;

constructor(private fb: FormBuilder) {

this.studentForm = this.fb.group({

fullName: ['', Validators.required],

mobile: ['', [Validators.required, Validators.pattern(/^[0-9]{10}$/)]],

email: ['', [Validators.required, Validators.email]],

country: ['', Validators.required],

gender: ['', Validators.required],

remarks: ['']

});

}

onSubmit() {

if (this.studentForm.valid) {

// Handle form submission (e.g., send data to server)

console.log(this.studentForm.value);

}

}

}

**Key Features:**

* **Form Group:** Created using FormBuilder for better maintainability.
* **Form Controls:** Each form field is represented by a FormControl within the FormGroup.
* **Validators:**
  + required: Ensures that the field is not empty.
  + email: Validates the email format.
  + pattern: Validates the mobile number format (10 digits).
* **Conditional Validation Messages:** Display error messages only when the field is invalid and has been touched or is dirty.
* **Radio Buttons:** Used for gender selection.
* **Form Submission:** The onSubmit() method is called when the form is submitted and the form is valid.

This example demonstrates a basic student registration form with reactive forms and basic validations. You can further enhance it by:

* **Adding more complex validations:** (e.g., custom validators, asynchronous validation)
* **Implementing server-side validation**
* **Improving the user interface**
* **Handling form submission using HTTP requests**

This comprehensive example provides a solid foundation for building reactive forms in Angular. Remember to adapt it to your specific needs and requirements.