

## Phase – 4 Practice Project: Assisted Practice

### 16. Form Validations.

- **Code**

- ✓ **Creating a new angular project**

# ng new FormValidations

- ✓ **app.component.html**

```
<form [formGroup]="signupForm" (ngSubmit)="submitForm()">
  <div>
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" FormControlName="username" required>
    <div *ngIf="signupForm?.get('username')?.errors?.['required']">Username is
      required.</div>
    <div *ngIf="signupForm?.get('username')?.errors?.['minlength']">Username must be at
      least 5 characters long.</div>
  </div>
  <div>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" FormControlName="email" required>
    <div *ngIf="signupForm?.get('email')?.errors?.['required']">Email is required.</div>
    <div *ngIf="signupForm?.get('email')?.errors?.['email']">Invalid email format.</div>
  </div>
  <div>
    <label for="password">Password:</label>
    <input type="password" id="password" name="password" FormControlName="password"
required>
    <div *ngIf="signupForm?.get('password')?.errors?.['required']">Password is
      required.</div>
    <div *ngIf="signupForm?.get('password')?.errors?.['minlength']">Password must be at
      least 8 characters long.</div>
  </div>
  <button type="submit">Sign Up</button>
</form>
```

- ✓ **app.component.css**

```
form {
  max-width: 400px;
  margin: 0 auto;
}

div {
  margin-bottom: 10px;
}

label {
  display: block;
  font-weight: bold;
}
```

```

input[type="text"],
input[type="email"],
input[type="password"] {
    width: 100%;
    padding: 5px;
}

button[type="submit"] {
    background-color: #007bff;
    color: #fff;
    padding: 10px 20px;
    border: none;
    cursor: pointer;
}

```

### ✓ app.component.ts

```

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

@Component({
    selector: 'app-root',
    templateUrl: './app.component.html',
    styleUrls: ['./app.component.css']
})
export class AppComponent {
    title(title: any) {
        throw new Error('Method not implemented.');
```

```

    }
    signupForm: FormGroup;

    constructor(private formBuilder: FormBuilder) {
        this.signupForm = this.formBuilder.group({
            username: ['', [Validators.required, Validators.minLength(5)]],
            email: ['', [Validators.required, Validators.email]],
            password: ['', [Validators.required, Validators.minLength(8)]]
        });
    }

    submitForm() {
        if (this.signupForm.valid) {
            console.log(this.signupForm.value); // Log the form values to the console
        }
    }
}

```

### ✓ app.module.ts

```

import { NgModule } from '@angular/core';
import { ReactiveFormsModule } from '@angular/forms'
import { BrowserModule } from '@angular/platform-browser';

```

```
import { AppRoutingModuleModule } from './app-routing.module';
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    ReactiveFormsModule,
    AppRoutingModuleModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

- **Output**

**Username:**

Username is required.

**Email:**

Email is required.

**Password:**

Password is required.

Sign Up