

Step-by-Step Enhanced Solution

Step 1: Adding Form Validation

Update the form templates to include validation messages.

****Register Component Template**
(`register.component.html`)**:

```
``html
<h2>Register</h2>
<form #registerForm="ngForm"
      (ngSubmit)="onRegister(registerForm)">
  <div>
    <label for="username">Username:</label>
    <input type="text" id="username"
      name="username" ngModel required>
    <div *ngIf="registerForm.submitted &&
      registerForm.controls.username?.invalid">
      <small class="error">Username is
      required.</small>
```

</div>

</div>

<div>

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

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

<div *ngIf="registerForm.submitted &&
registerForm.controls.email?.invalid">

<small class="error"
*ngIf="registerForm.controls.email?.errors?.required
>Email is required.</small>

<small class="error"
*ngIf="registerForm.controls.email?.errors?.email">I
nvalid email address.</small>

</div>

</div>

<div>

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

<input type="password" id="password"
name="password" ngModel required
minlength="6">

<div *ngIf="registerForm.submitted &&
registerForm.controls.password?.invalid">

```

    <small class="error"
*ngIf="registerForm.controls.password?.errors?.req
uired">Password is required.</small>

    <small class="error"
*ngIf="registerForm.controls.password?.errors?.min
length">Password must be at least 6 characters
long.</small>

    </div>

</div>

<button type="submit"
[disabled]="registerForm.invalid">Register</button
>

</form>
```

```

**\*\*Login Component Template**  
(`login.component.html`)\*\*:

```

```html
<h2>Login</h2>

<form #loginForm="ngForm"
(ngSubmit)="onLogin(loginForm)">
  <div>

```

```
<label for="username">Username:</label>
  <input type="text" id="username"
name="username" ngModel required>
  <div *ngIf="loginForm.submitted &&
loginForm.controls.username?.invalid">
    <small class="error">Username is
required.</small>
  </div>
</div>
<div>
  <label for="password">Password:</label>
  <input type="password" id="password"
name="password" ngModel required>
  <div *ngIf="loginForm.submitted &&
loginForm.controls.password?.invalid">
    <small class="error">Password is
required.</small>
  </div>
</div>
  <button type="submit"
[disabled]="loginForm.invalid">Login</button>
</form>
```

```

## #### Step 2: Adding Angular Services for Mock Backend

Create a new service to handle user data.

```bash

ng generate service user

```

**\*\*User Service (`user.service.ts`)\*\*:**

```typescript

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

@Injectable({
 providedIn: 'root'

})

export class UserService {
 private users = [];

```
register(user: any) {  
  this.users.push(user);  
  console.log('User Registered:', user);  
}
```

```
login(credentials: any) {  
  const user = this.users.find(u => u.username ===  
credentials.username && u.password ===  
credentials.password);  
  if (user) {  
    console.log('User Logged In:', user);  
    return true;  
  } else {  
    console.log('Login Failed');  
    return false;  
  }  
}  
}  
}
```

Update the `RegisterComponent` and
`LoginComponent` to use the `UserService`.

****Register Component (`register.component.ts`)**:**

```
``typescript
```

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

```
import { UserService } from '../user.service';
```

```
@Component({
```

```
  selector: 'app-register',
```

```
  templateUrl: './register.component.html'
```

```
})
```

```
export class RegisterComponent {
```

```
  constructor(private userService: UserService) {}
```

```
  onRegister(form: any) {
```

```
    if (form.valid) {
```

```
      this.userService.register(form.value);
```

```
      form.reset();
```

```
    }
```

```
}  
}  
``
```

****Login Component (`login.component.ts`)**:**

```
``typescript
```

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

```
import { UserService } from '../user.service';
```

```
@Component({
```

```
  selector: 'app-login',
```

```
  templateUrl: './login.component.html'
```

```
})
```

```
export class LoginComponent {
```

```
  loginFailed = false;
```

```
  constructor(private userService: UserService) {}
```

```
  onLogin(form: any) {
```

```
    if (form.valid) {
```



```
<label for="username">Username:</label>
  <input type="text" id="username"
name="username" ngModel required>
  <div *ngIf="loginForm.submitted &&
loginForm.controls.username?.invalid">
    <small class="error">Username is
required.</small>
  </div>
</div>
<div>
  <label for="password">Password:</label>
  <input type="password" id="password"
name="password" ngModel required>
  <div *ngIf="loginForm.submitted &&
loginForm.controls.password?.invalid">
    <small class="error">Password is
required.</small>
  </div>
</div>
<button type="submit"
[disabled]="loginForm.invalid">Login</button>
<div *ngIf="loginFailed">
```

```
<small class="error">Login failed. Please check  
your username and password.</small>
```

```
</div>
```

```
</form>
```

```
```
```

### #### Step 3: Styling the Forms

Add basic CSS to make the forms look nicer.

**\*\*styles.css\*\*** (global styles):

```
```css
```

```
body {
```

```
  font-family: Arial, sans-serif;
```

```
  margin: 20px;
```

```
}
```

```
form {
```

```
  max-width: 300px;
```

```
  margin: 0 auto;
```

```
padding: 1em;  
border: 1px solid #ccc;  
border-radius: 1em;  
}
```

```
div {  
  margin-bottom: 1em;  
}
```

```
label {  
  margin-bottom: .5em;  
  color: #333333;  
  display: block;  
}
```

```
input {  
  border: 1px solid #CCCCCC;  
  padding: .5em;  
  font-size: 1em;  
  border-radius: 4px;  
  width: 100%;
```

```
    box-sizing: border-box;
}
```

```
button {
    padding: 0.7em;
    color: #fff;
    background-color: #007BFF;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    width: 100%;
    font-size: 1em;
}
```

```
button:disabled {
    background-color: #CCCCCC;
}
```

```
.error {
    color: red;
    font-size: 0.8em;
```

```
}  
...
```

Final Project Files

- **app-routing.module.ts**
- **app.module.ts**
- **home.component.html**
- **home.component.ts**
- **register.component.html**
- **register.component.ts**
- **login.component.html**
- **login.component.ts**
- **user.service.ts**
- **styles.css**
- **app.component.html**
- **app.component.ts**