login.component.html

```
<div class="form-container">
  <h2>Login</h2>
  <form [formGroup]="loginForm" (ngSubmit)="onSubmit()">
   <div class="form-group">
    <label for="email">Email</label>
    <input
     id="email"
     type="email"
     formControlName="email"
     [class.invalid]="loginForm.get('email')?.invalid && loginForm.get('email')?.touched"
    >
   </div>
   <div class="form-group">
    <label for="password">Password</label>
    <input
     id="password"
     type="password"
     formControlName="password"
     [class.invalid]="loginForm.get('password')?.invalid &&
loginForm.get('password')?.touched"
    >
   </div>
   <button type="submit" [disabled]="!loginForm.valid">Login</button>
  </form>
 </div>
```

login.component.ts

```
import { Component } from '@angular/core';
import { FormBuilder, FormGroup, Validators, ReactiveFormsModule } from '@angular/forms';
import { Router } from '@angular/router';
import { AuthService } from '../../services/auth.service';
import { CommonModule } from '@angular/common';
@Component({
 selector: 'app-login',
 standalone: true,
 imports: [ReactiveFormsModule, CommonModule],
 templateUrl: './login.component.html',
 styleUrls: ['./login.component.css']
})
export class LoginComponent {
 loginForm!: FormGroup;
 constructor(
  private fb: FormBuilder,
  private authService: AuthService,
  private router: Router
 ) {
  this.loginForm = this.fb.group({
   email: ['', [Validators.required, Validators.email]],
   password: ['', Validators.required]
  });
 }
 onSubmit(): void {
  if (this.loginForm.valid) {
   const { email, password } = this.loginForm.value;
   if (this.authService.login(email, password)) {
    this.router.navigate(['/profile']);
   } else {
    alert('Invalid credentials');
   }
  }
 }
}
```

profile.component.html

```
<div class="profile-container">
  <h2>Profile</h2>
  <div *ngIf="user" class="user-info">
   <p><strong>Name:</strong> {{ user.name }}</p>
   <p><strong>Email:</strong> {{ user.email }}</p>
   <button (click)="logout()" class="logout-btn">Logout</button>
  </div>
 </div>
```

profile.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { AuthService } from '../../services/auth.service';
import { CommonModule } from '@angular/common';
@Component({
 selector: 'app-profile',
 standalone: true,
 imports: [CommonModule],
 templateUrl: './profile.component.html',
 styleUrl: './profile.component.css'
})
export class ProfileComponent implements OnInit {
 user: any = null;
 constructor(
  private authService: AuthService,
  private router: Router
 ) {}
 ngOnInit(): void {
  this.authService.currentUser$.subscribe(user => {
   this.user = user;
  });
 }
 logout(): void {
  this.authService.logout();
  this.router.navigate(['/login']);
 }
}
```

register.component.html

```
<div class="form-container">
  <h2>Register</h2>
  <form [formGroup]="registerForm" (ngSubmit)="onSubmit()">
   <div class="form-group">
    <label for="name">Name</label>
    <input
     id="name"
     type="text"
     formControlName="name"
     [class.invalid]="registerForm.get('name')?.invalid && registerForm.get('name')?.touched"
    >
   </div>
   <div class="form-group">
    <label for="email">Email</label>
    <input
     id="email"
     type="email"
     formControlName="email"
     [class.invalid]="registerForm.get('email')?.invalid && registerForm.get('email')?.touched"
    >
   </div>
   <div class="form-group">
    <label for="password">Password</label>
    <input
     id="password"
     type="password"
     formControlName="password"
     [class.invalid]="registerForm.get('password')?.invalid &&
registerForm.get('password')?.touched"
    >
   </div>
   <button type="submit" [disabled]="!registerForm.valid">Register</button>
  </form>
 </div>
```

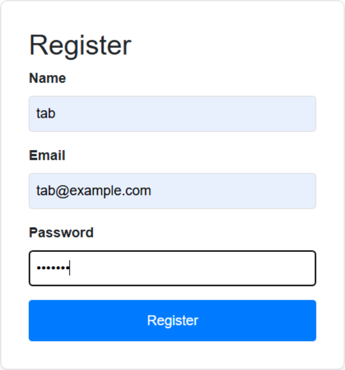
register.component.ts

```
import { Component } from '@angular/core';
import { FormBuilder, FormGroup, Validators, ReactiveFormsModule } from '@angular/forms';
import { Router } from '@angular/router';
import { AuthService } from '../../services/auth.service';
import { CommonModule } from '@angular/common';
@Component({
 selector: 'app-register',
 standalone: true,
 imports: [ReactiveFormsModule, CommonModule],
 templateUrl: './register.component.html',
 styleUrls: ['./register.component.css']
})
```

```
export class RegisterComponent {
 registerForm!: FormGroup;
 constructor(
  private fb: FormBuilder,
  private authService: AuthService,
  private router: Router
 ) {
  this.registerForm = this.fb.group({
   name: ['', [Validators.required, Validators.minLength(3)]],
   email: ['', [Validators.required, Validators.email]],
   password: ['', [Validators.required, Validators.minLength(6)]]
  });
 }
 onSubmit(): void {
  if (this.registerForm.valid) {
   this.authService.register(this.registerForm.value);
   this.router.navigate(['/login']);
  }
 }
}
```

app.routes.ts

```
import { Routes } from '@angular/router';
import { authGuard } from './guards/auth.guard';
export const routes: Routes = [
 {
  path: 'register',
  loadComponent: () => import('./components/register/register.component').then(m =>
m.RegisterComponent)
 },
 {
  path: 'login',
  loadComponent: () => import('./components/login/login.component').then(m =>
m.LoginComponent)
 },
 {
  path: 'profile',
  loadComponent: () => import('./components/profile/profile.component').then(m =>
m.ProfileComponent),
  canActivate: [authGuard]
 },
 {
  path: '',
  redirectTo: '/login',
  pathMatch: 'full'
 }
];
```



Register

Name

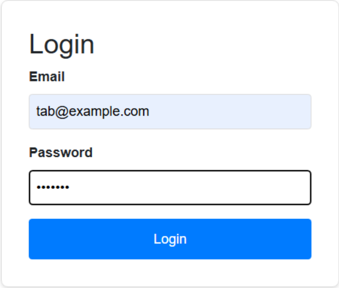
tab

Email

tab@example.com

Password

•••••••



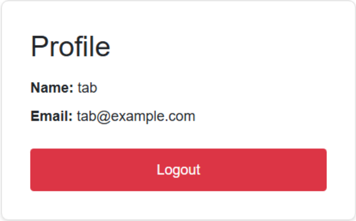
Login

Email

tab@example.com

Password

•••••••



Profile

Name: tab

Email: tab@example.com