Step 1: Set Up Angular Project

1. **Install Angular CLI (if not already installed):**

```
npm install -g @angular/cli
```

2. **Create a new Angular project:**

```
ng new angular-auth-app
cd angular-auth-app
```

3. **Install Bootstrap 5:**

```
npm install bootstrap
```

4. **Include Bootstrap in `angular.json`:**

```
"styles": [
"node_modules/bootstrap/dist/css/bootstrap.min.css",
"src/styles.css"
]
```

Step 2: Create Standalone Components

1. **Generate Login and Register Components:**

```
ng generate component login
ng generate component register
```

2. **Modify `login.component.ts`:**

```
import { Component } from '@angular/core';
import { FormBuilder, FormGroup, Validators, ReactiveFormsModule } from '@angular/forms';
import { CommonModule } from '@angular/common';
@Component({
selector: 'app-login',
templateUrl: './login.component.html',
styleUrls: ['./login.component.css'],
standalone: true,
imports: [CommonModule, ReactiveFormsModule]
})
export class LoginComponent {
loginForm: FormGroup;
constructor(private fb: FormBuilder) {
this.loginForm = this.fb.group({
```

```
email: ['', [Validators.required, Validators.email]],
password: ['', [Validators.required, Validators.minLength(6)]],
});
}
onBlur(field: string) {
this.loginForm.get(field)!.markAsTouched();
}
onSubmit() {>
if (this.loginForm.valid) {
console.log(this.loginForm.value);
}
}
}
```

3. **Modify `login.component.html`:**

```
<div class="container mt-5">
<div class="row justify-content-center">
<div class="col-md-6">
<div class="card">
<div class="card-header text-center">
<h2>Login</h2>
</div>
<div class="card-body">
<form [formGroup]="loginForm" (ngSubmit)="onSubmit()">
<div class="mb-3">
<label for="email" class="form-label">Email address</label>
<input type="email" class="form-control" id="email" formControlName="email"
(blur)="onBlur('email')">
<div *ngIf="loginForm.get('email')!.touched && loginForm.get('email')!.invalid"
class="text-danger">
<small *ngIf="loginForm.get('email')!.errors?.['required']">Email is required.</small>
<small *ngIf="loginForm.get('email')!.errors?.['email']">Email is invalid.</small>
</div>
</div>
<div class="mb-3">
<label for="password" class="form-label">Password</label>
<input type="password" class="form-control" id="password" formControlName="password"
(blur)="onBlur('password')">
<div *ngIf="loginForm.get('password')!.touched && loginForm.get('password')!.invalid"
class="text-danger">
<small *ngIf="loginForm.get('password')!.errors?.['required']">Password is required.</small>
<small *ngIf="loginForm.get('password')!.errors?.['minlength']">Password must be at least 6
characters long.</small>
```

```
</div>
</div>
<div class="d-grid">
<button type="submit" class="btn btn-primary" [disabled]="loginForm.invalid">Login</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
```

4. **Modify `register.component.ts`:**

```
import { Component } from '@angular/core';
import { FormBuilder, FormGroup, Validators, ReactiveFormsModule } from '@angular/forms';
import { CommonModule } from '@angular/common';
@Component({
selector: 'app-register',
templateUrl: './register.component.html',
styleUrls: ['./register.component.css'],
standalone: true,
imports: [CommonModule, ReactiveFormsModule]
})
export class RegisterComponent {
registerForm: FormGroup;
constructor(private fb: FormBuilder) {
this.registerForm = this.fb.group({
email: ['', [Validators.required, Validators.email]],
password: ['', [Validators.required, Validators.minLength(6)]],
confirmPassword: ['', Validators.required]
}, { validator: this.passwordMatchValidator });
}
passwordMatchValidator(form: FormGroup) {
return form.get('password')!.value === form.get('confirmPassword')!.value
? null : { 'mismatch': true };
}
onBlur(field: string) {
this.registerForm.get(field)!.markAsTouched();
}
onSubmit() {
if (this.registerForm.valid) {
console.log(this.registerForm.value);
}
}
}
```

5. **Modify `register.component.html`:**

```
<div class="container mt-5">
<div class="row justify-content-center">
<div class="col-md-6">
<div class="card">
<div class="card-header text-center">
<h2>Register</h2>
</div>
<div class="card-body">
<form [formGroup]="registerForm" (ngSubmit)="onSubmit()">
<div class="mb-3">
<label for="email" class="form-label">Email address</label>
<input type="email" class="form-control" id="email" formControlName="email"
(blur)="onBlur('email')">
<div *ngIf="registerForm.get('email')!.touched && registerForm.get('email')!.invalid"
class="text-danger">
<small *ngIf="registerForm.get('email')!.errors?.['required']">Email is required.</small>
<small *ngIf="registerForm.get('email')!.errors?.['email']">Email is invalid.</small>
</div>
</div>
<div class="mb-3">
<label for="password" class="form-label">Password</label>
<input type="password" class="form-control" id="password" formControlName="password"
(blur)="onBlur('password')">
<div *ngIf="registerForm.get('password')!.touched && registerForm.get('password')!.invalid"
class="text-danger">
<small *ngIf="registerForm.get('password')!.errors?.['required']">Password is required.</small>
<small *ngIf="registerForm.get('password')!.errors?.['minlength']">Password must be at least 6
characters long.</small>
</div>
</div>
<div class="mb-3">
<label for="confirmPassword" class="form-label">Confirm Password</label>
<input type="password" class="form-control" id="confirmPassword"
formControlName="confirmPassword" (blur)="onBlur('confirmPassword')">
<div *ngIf="registerForm.get('confirmPassword')!.touched &&
registerForm.get('confirmPassword')!.invalid" class="text-danger">
<small *ngIf="registerForm.get('confirmPassword')!.errors?.['required']">Confirm Password is
required.</small>
<small *ngIf="registerForm.errors?.['mismatch']">Passwords do not match.</small>
```

```
</div>
</div>
<div class="d-grid">
<button type="submit" class="btn btn-primary"
[disabled]="registerForm.invalid">Register</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
```

Step 3: Add Routing

1. **Modify `app-routs.ts`:**

```
import { Routes } from '@angular/router';
import { LoginComponent } from './login/login.component';
import { RegisterComponent } from './register/register.component';
export const routes: Routes = [
{ path: 'login', component: LoginComponent },
{ path: 'register', component: RegisterComponent },
{ path: '', redirectTo: '/login', pathMatch: 'full' }
];
```

2. **Modify `app.component.html`:**

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
<div class="container-fluid">
<a class="navbar-brand" href="#">AuthApp</a>
<div class="collapse navbar-collapse" id="navbarNav">
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link" routerLink="/login">Login</a>
</li>
<li class="nav-item">
<a class="nav-link" routerLink="/register">Register</a>
</li>
</ul>
</div>
</div>
</nav>
<router-outlet></router-outlet>
```

3. **Modify `app.component.ts`:**

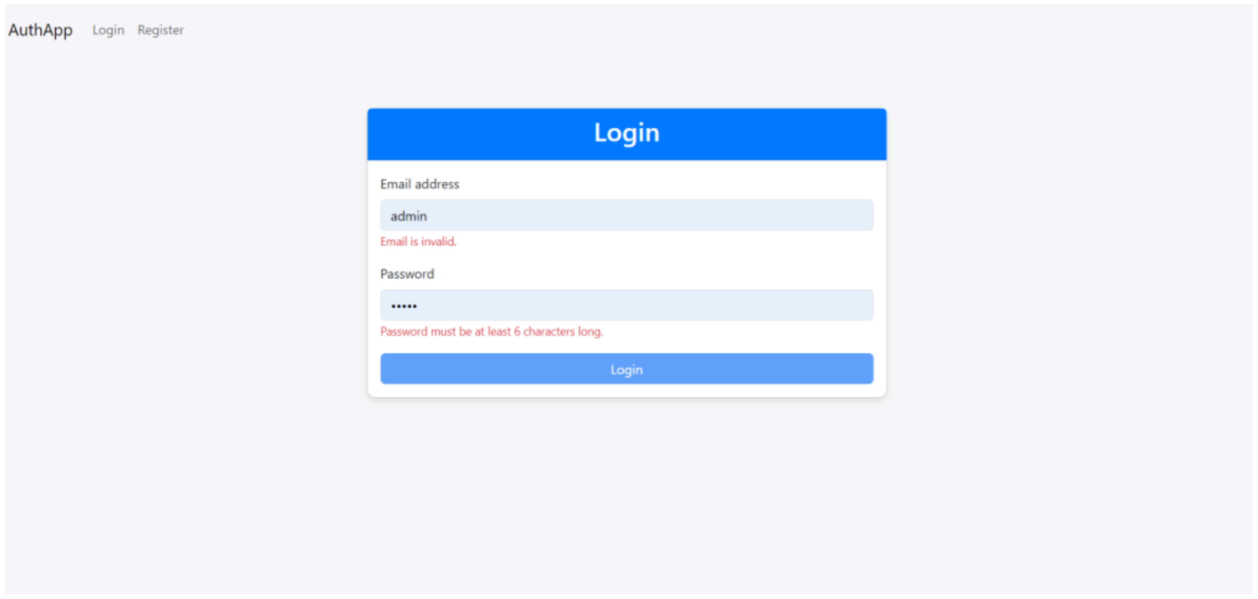
```
import { Component } from '@angular/core';
import { CommonModule } from '@angular/common';
import { RouterOutlet, RouterLink } from '@angular/router';
@Component({
selector: 'app-root',
standalone: true,
imports: [CommonModule, RouterOutlet, RouterLink],
templateUrl: './app.component.html',
styleUrl: './app.component.css'
})
export class AppComponent {
title = 'angular-auth-app';
}
```

4. Modify styles.css file for custom form styles:

```
/* You can add global styles to this file, and also import other style files */
/* src/styles.css */
body {
background-color: #f8f9fa;
}
.card {
margin-top: 20px;
border-radius: 10px;
box-shadow: 0 4px 8px rgba(0,0,0,0.1);
}
.card-header {
background-color: #007bff;
color: white;
border-radius: 10px 10px 0 0;
}
```

Step 4: Run the Application

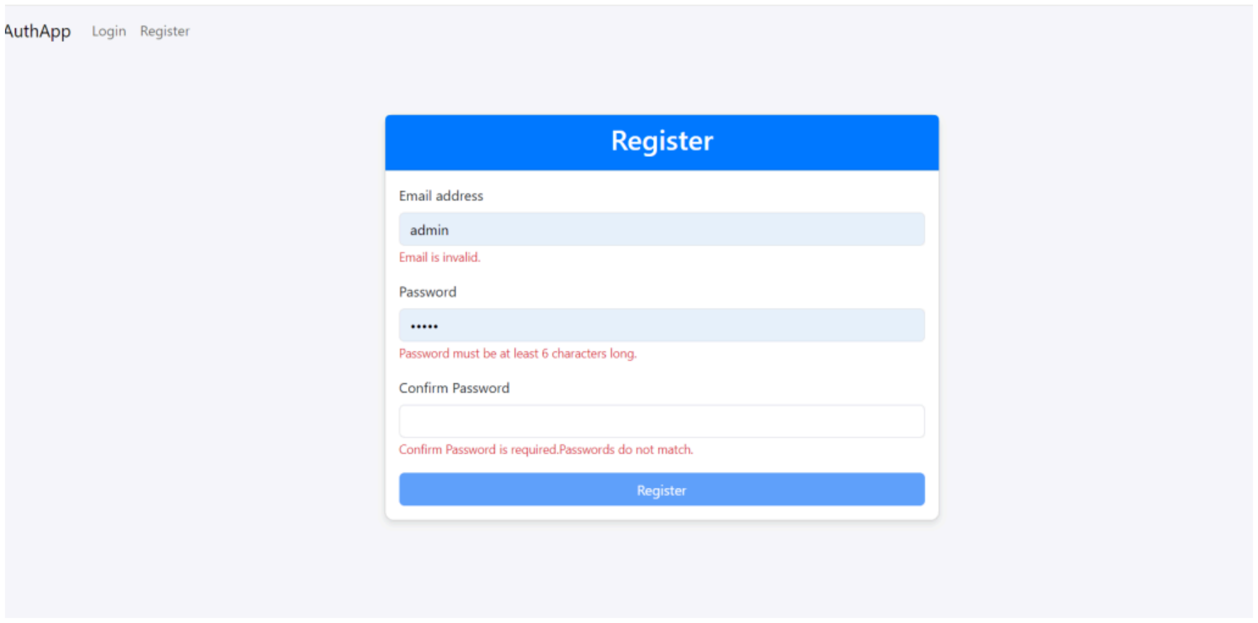
```
ng serve
```



Email is invalid.

Password must be at least 6 characters long.

Create a User Login and Registration forms with validations using Angular 18



Email is invalid.

Password must be at least 6 characters long.

Confirm Password is required.Passwords do not match.

Register form

You now have a basic Angular application with standalone login and register components, using Bootstrap 5 for styling and form validation.