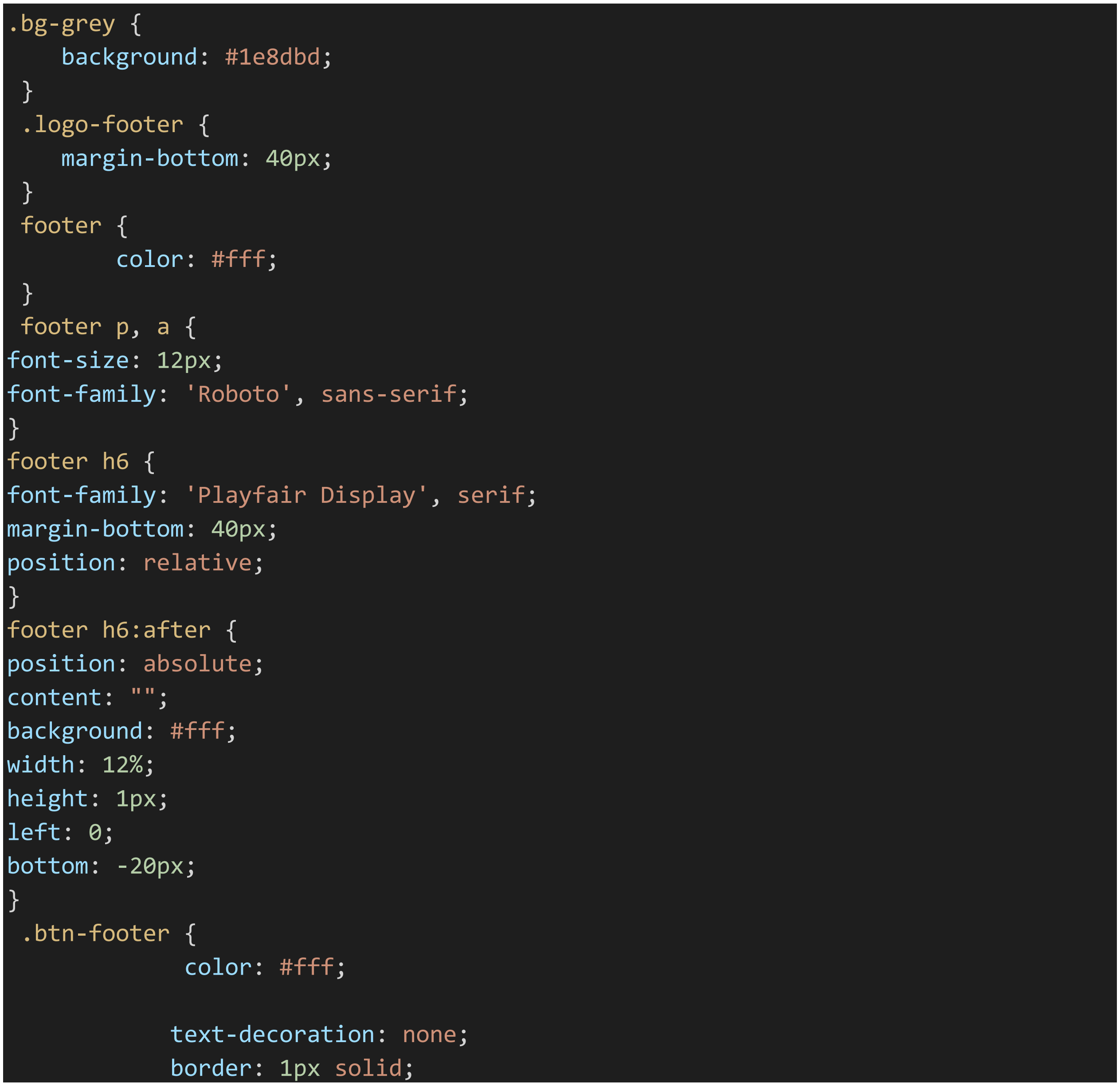
Milestone 3 -CODE **Matriculation: 1310726**

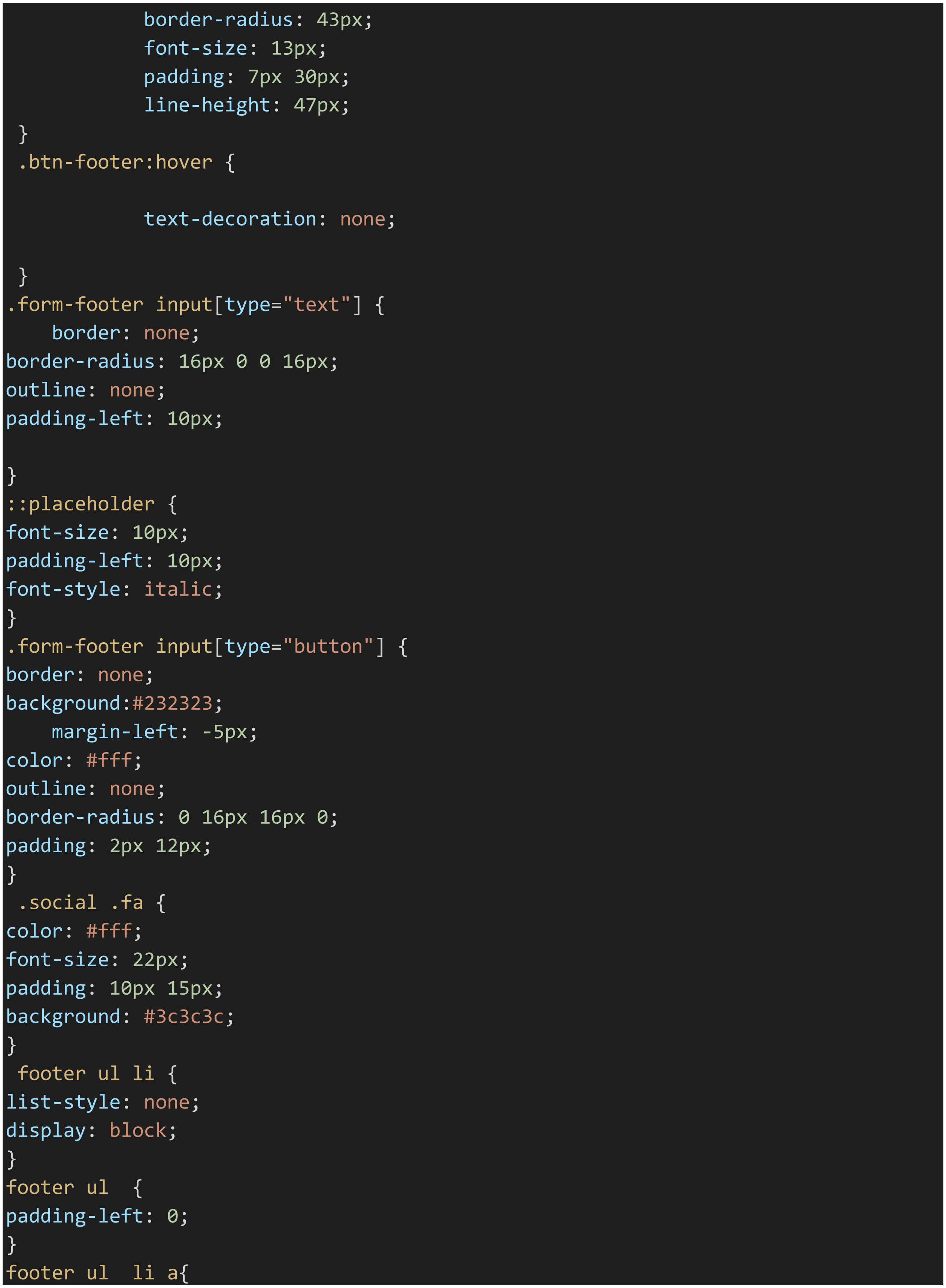
Name: Muhammad Bilal Mehdi

Front-End

Footer.component.css



```
.bg-grey {
    background: #1e8dbd;
 }
 .logo-footer {
    margin-bottom: 40px;
 }
 footer {
        color: #fff;
 }
 footer p, a {
font-size: 12px;
font-family: 'Roboto', sans-serif;
}
footer h6 {
font-family: 'Playfair Display', serif;
margin-bottom: 40px;
position: relative;
}
footer h6:after {
position: absolute;
content: "";
background: #fff;
width: 12%;
height: 1px;
left: 0;
bottom: -20px;
}
 .btn-footer {
             color: #fff;

            text-decoration: none;
            border: 1px solid;
```



```
            border-radius: 43px;
            font-size: 13px;
            padding: 7px 30px;
            line-height: 47px;
 }
 .btn-footer:hover {

            text-decoration: none;

 }
.form-footer input[type="text"] {
    border: none;
border-radius: 16px 0 0 16px;
outline: none;
padding-left: 10px;

}
::placeholder {
font-size: 10px;
padding-left: 10px;
font-style: italic;
}
.form-footer input[type="button"] {
border: none;
background:#232323;
    margin-left: -5px;
color: #fff;
outline: none;
border-radius: 0 16px 16px 0;
padding: 2px 12px;
}
 .social .fa {
color: #fff;
font-size: 22px;
padding: 10px 15px;
background: #3c3c3c;
}
 footer ul li {
list-style: none;
display: block;
}
footer ul  {
padding-left: 0;
}
footer ul  li a{
```

```
text-decoration: none;
color: #fff;
text-decoration:none;
}
a:hover {
text-decoration: none;
color: #f5f5f5 !important;

}
.logo-part {
border-right: 1px solid #fff;
height: 100%;
}
```

Footer.component.html

```

    <div class="footer">
      <footer class="text-muted py-5">
        <div class="container">
          <p class="float-end mb-1">
            <a href="#">Back to top</a>
          </p>
          <p class="mb-1">Test</p>
          <p class="mb-0">Test</p>
        </div>
      </footer>
```

Footer.component.ts

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

@Component({
  selector: 'app-footer',
  templateUrl: './footer.component.html',
  styleUrls: ['./footer.component.css']
})

export class FooterComponent implements OnInit {

  constructor() { }
  ngOnInit(): void {
  }

}
```

Register.component.css

```
.global-container{
    height:100%;

    display: flex;
    align-items: center;
    justify-content: center;
    background-color: #f7f2e3;
}

.card-title{ font-weight:300; }

.btn{
    font-size: 14px;
    margin-top:20px;
}


.login-form{
    margin:20px;
}

.sign-up{
    text-align:center;
    padding:20px 0 0;
}

.alert{
    margin-bottom:-30px;
    font-size: 13px;
    margin-top:20px;
}

.formCustomRegister div{
    margin-bottom:20px;
}
```

Register.component.html

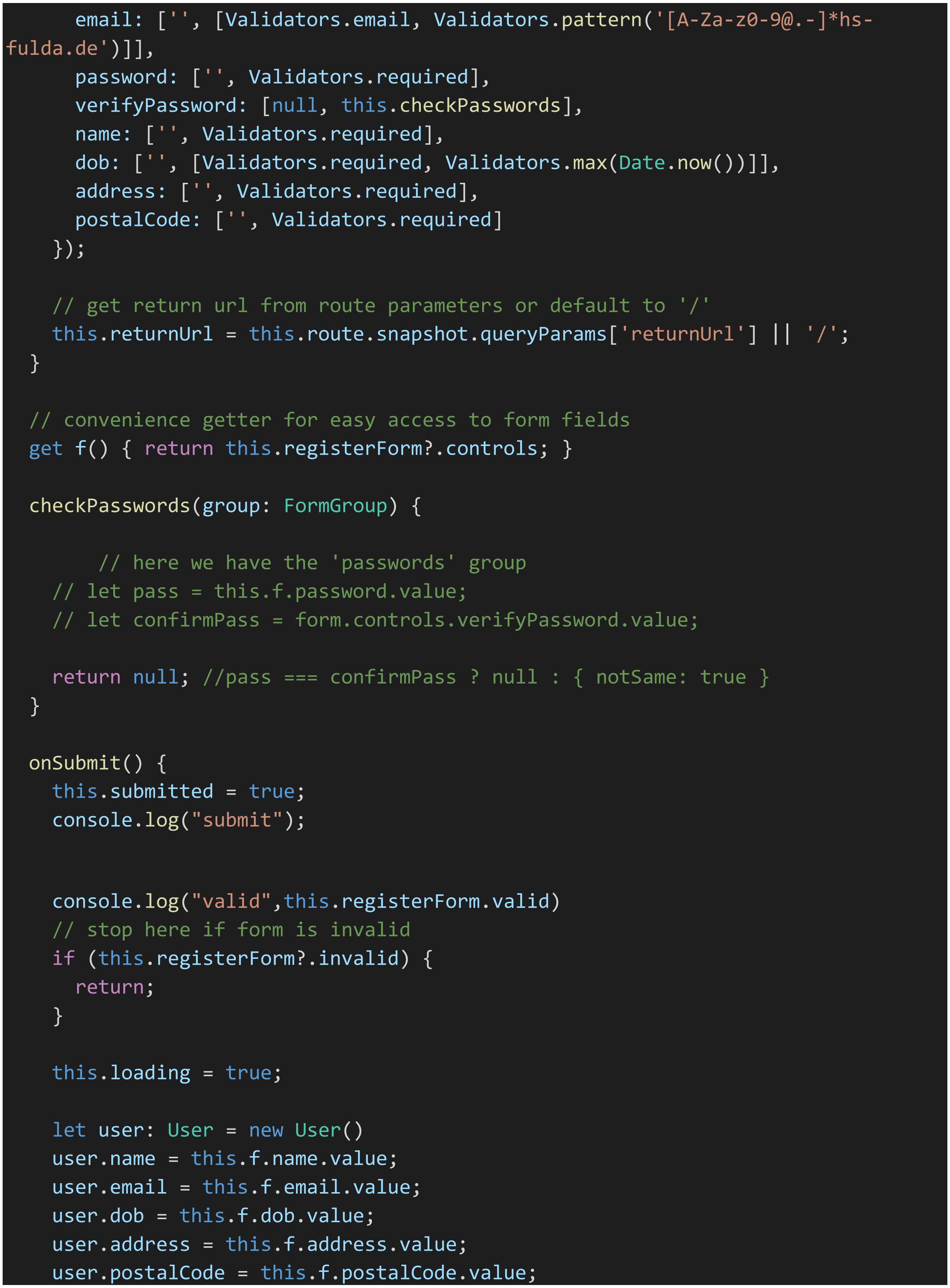
```
<div class="global-container" *ngIf="!successSubmitted">
    <div class="card login-form w-75">
    <div class="card-body">
        <h3 class="card-title text-center">Register to E-Market Fulda</h3>
        <div class="card-text">
            <!--
            <div class="alert alert-danger alert-dismissible fade show" role="alert">Incorrect username or password.</div> -->
            <form class="form-group formCustomRegister"  [formGroup]="registerForm" (ngSubmit)="onSubmit()">
                <!-- to error: add class "has-danger" -->
                <div>
                    <label for="name">Full Name</label>
                    <input type="text" name="name" class="form-control form-control-sm" id="name" aria-describedby="name" formControlName="name" [ngClass]="{ 'is-invalid': submitted && f.name.errors }"/>
                    <div *ngIf="submitted && f.name.errors" class="invalid-feedback">
                        <div *ngIf="f.name.errors.required">Username is required</div>
                    </div>
                </div>

                <div >
                    <label for="email">Email</label>
                    <input type="email" name="email" class="form-control form-control-sm" id="email" aria-describedby="email" formControlName="email" [ngClass]="{ 'is-invalid': submitted && f.email.errors }"/>
                    <div *ngIf="submitted && f.email.errors" class="invalid-feedback">
                        <div *ngIf="f.email.errors.email">Email is invalid</div>
                        <div *ngIf="!f.email.errors.email && f.email.errors.pattern">Please enter your university email address.</div>
                    </div>
                </div>
        <div class="form-group">
          <label for="password">Password</label>
          <input type="password" name="password" class="form-control form-control-sm" id="password" formControlName="password" [ngClass]="{ 'is-invalid': submitted && f.password.errors }"/>
          <div *ngIf="submitted && f.password.errors" class="invalid-feedback">
            <div *ngIf="f.password.errors.required">Password is required</div>
          </div>
        </div>

        <div class="form-group">
          <label for="verifyPassword">Confirm Password</label>
          <input type="password" name="verifyPassword" class="form-control form-control-sm" id="verifyPassword" formControlName="verifyPassword" [ngClass]="{ 'is-invalid': submitted && f.verifyPassword.errors }"/>
          <div *ngIf="submitted && passwordMismatch" class="invalid-feedback" style="display: block;">
            <div *ngIf="passwordMismatch">Confirm Password not same</div>
          </div>
        </div>

        <div >
          <label for="address">Address</label>
          <input type="text" name="address" class="form-control form-control-sm" id="address" aria-describedby="address" formControlName="address" [ngClass]="{ 'is-invalid': submitted && f.address.errors }"/>
          <div *ngIf="submitted && f.address.errors" class="invalid-feedback">
            <div *ngIf="f.address.errors.required">Address is required</div>
          </div>
        </div>

        <div >
          <label for="postalCode">Postal Code</label>
          <input type="text" name="postalCode" class="form-control form-control-sm" id="postalCode" aria-describedby="postalCode" formControlName="postalCode" [ngClass]="{ 'is-invalid': submitted && f.postalCode.errors }"/>
```

```
                    <div *ngIf="submitted && f.postalCode.errors" class="invalid-
feedback">
                        <div *ngIf="f.postalCode.errors.required">Username is req
uired</div>
                    </div>
                </div>

                <div >
                    <label for="dob">Date of Birth</label>
                    <input type="date" name="dob" class="form-control form-
control-sm" id="dob" aria-
describedby="dob" formControlName="dob" [ngClass]="{ 'is-
invalid': submitted && f.dob.errors }"/>
                    <div *ngIf="submitted && f.dob.errors" class="invalid-
feedback">
                        <div *ngIf="f.dob.errors.required">Date of birth is requi
red</div>
                    </div>
                </div>
                <br><br>
                <button type="submit" class="btn btn-primary btn-
block"> Register </button>
                <div *ngIf="error" class="alert alert-danger mt-3 mb-
0">{{error}}</div>

                <div class="sign-up">
                    Already Registered? <a href="/login">Sign In</a>
                </div>
            </form>
        </div>
    </div>
</div>
</div>


<div *ngIf="successSubmitted">
    <div class="d-flex justify-content-center">
        <img class="img-fluid" src="assets/images/success.png"/>
    </div>
    <div class="d-flex justify-content-center mt-3">
        <h3>User has successfully created</h3>
    </div>
    <div class="d-flex justify-content-center mt-3">
        <h5>Click here to <a href="/login"> login</a></h5>
    </div>
```

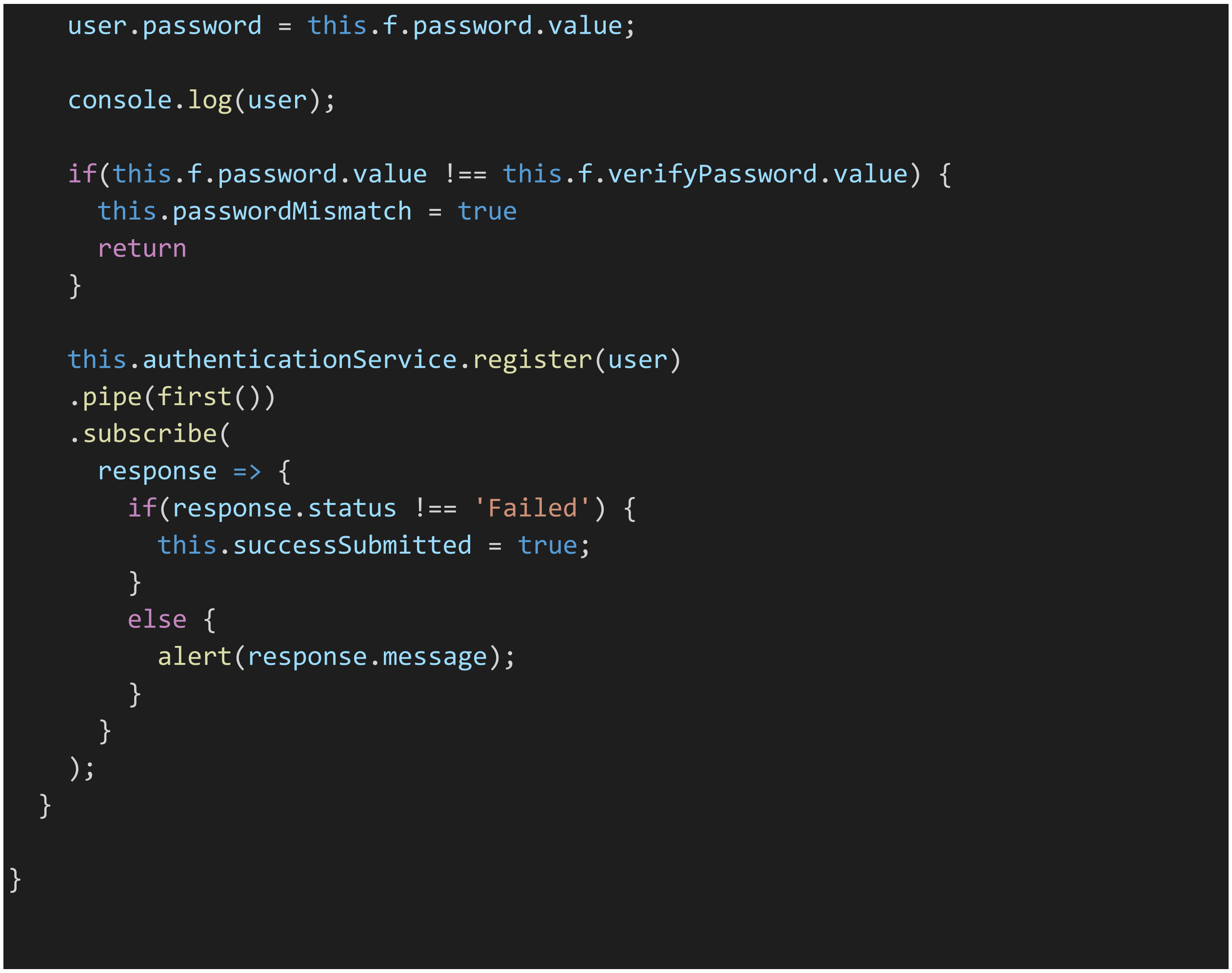
```
</div>
```

Register.component.ts

```
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { ActivatedRoute, Router } from '@angular/router';
import { first } from 'rxjs/operators';
import { User } from '../_models/user-model';
import { AuthenticationService } from '../_services/authentication.service';

@Component({
  selector: 'app-register',
  templateUrl: './register.component.html',
  styleUrls: ['./register.component.css']
})
export class RegisterComponent implements OnInit {

  registerForm?: FormGroup;
  loading = false;
  submitted = false;
  returnUrl?: string;
  error = '';
  passwordMismatch = false;

  successSubmitted = false;

  constructor(
    private formBuilder: FormBuilder,
    private route: ActivatedRoute,
    private router: Router,
    private authenticationService: AuthenticationService
  ) {
    // redirect to home if already logged in
    if (this.authenticationService.currentUser) {
      this.router.navigate(['/']);
    }
  }

  ngOnInit() {
    this.registerForm = this.formBuilder.group({
```



```
      email: ['', [Validators.email, Validators.pattern('[A-Za-z0-9@.-]*hs-fulda.de')]],
      password: ['', Validators.required],
      verifyPassword: [null, this.checkPasswords],
      name: ['', Validators.required],
      dob: ['', [Validators.required, Validators.max(Date.now())]],
      address: ['', Validators.required],
      postalCode: ['', Validators.required]
    });

    // get return url from route parameters or default to '/'
    this.returnUrl = this.route.snapshot.queryParams['returnUrl'] || '/';
  }

  // convenience getter for easy access to form fields
  get f() { return this.registerForm?.controls; }

  checkPasswords(group: FormGroup) {

        // here we have the 'passwords' group
    // let pass = this.f.password.value;
    // let confirmPass = form.controls.verifyPassword.value;

    return null; //pass === confirmPass ? null : { notSame: true }
  }

  onSubmit() {
    this.submitted = true;
    console.log("submit");


    console.log("valid",this.registerForm.valid)
    // stop here if form is invalid
    if (this.registerForm?.invalid) {
      return;
    }

    this.loading = true;

    let user: User = new User()
    user.name = this.f.name.value;
    user.email = this.f.email.value;
    user.dob = this.f.dob.value;
    user.address = this.f.address.value;
    user.postalCode = this.f.postalCode.value;
```



```
    user.password = this.f.password.value;

    console.log(user);

    if(this.f.password.value !== this.f.verifyPassword.value) {
      this.passwordMismatch = true
      return
    }

    this.authenticationService.register(user)
    .pipe(first())
    .subscribe(
      response => {
        if(response.status !== 'Failed') {
          this.successSubmitted = true;
        }
        else {
          alert(response.message);
        }
      }
    );
  }

}
```