Code

1. register-user.component.html

```html
<div class="login-page">
    <mat-card class="mat-elevation-z12">
      <mat-card-title style="text-align:center;">SignIn/SignUp</mat-card-title>
      <mat-tab-group mat-stretch-tabs class="example-stretched-tabs">
        <mat-tab label="Sign In">
            <mat-card-content>
              <form action="">
                <mat-form-field>
                  <input
                    matInput
                    placeholder="Email"
                    required
                    name="email"
                    ngModel
                    #email="ngModel">
                    <mat-error *ngIf="email.invalid">Email is required</mat-error>
                  <mat-icon matSuffix>email</mat-icon>
                </mat-form-field>
                <mat-form-field>
                  <input
                    matInput
                    placeholder="Password"
                    type="password"
                    required
                    minlength="6"
                    name="password"
                    ngModel
                    #password="ngModel">
                    <mat-error
                      *ngIf="password.errors">
                      Password is required
                    </mat-error>
                    <mat-error
                      *ngIf="password.errors">
                      Password must be of length 6
                    </mat-error>
                  <mat-icon matSuffix>lock</mat-icon>
                </mat-form-field>
                <mat-checkbox>Remember Me!</mat-checkbox>
                <button
                  mat-raised-button
                  color="primary"
                  class="login-button">
                  Sign In
                </button>

                <button
                  mat-raised-button
```

```html
          class="login-button">
          Login with Google
        </button>
      </form>
    </mat-card-content>
  </mat-tab>
  <mat-tab label="Sign Up">
    <mat-card-content>
      <form action="">
        <mat-form-field>
          <input
            matInput
            placeholder="User Name"
            required
            name="username"
            ngModel
            #username="ngModel">
          <mat-icon matSuffix>perm_identity</mat-icon>
          <mat-error
            *ngIf="username.invalid">
            Username is required
          </mat-error>

        </mat-form-field>
        <mat-form-field>
          <input
            matInput
            placeholder="Email"
            required
            name="email"
            ngModel
            #email="ngModel">
          <mat-error *ngIf="email.invalid">Email is required</mat-error>
          <mat-icon matSuffix>email</mat-icon>
        </mat-form-field>
        <mat-form-field>
          <input
            matInput
            placeholder="Password"
            type="password"
            required
            minlength="6"
            name="password"
            ngModel
            #password="ngModel">
          <mat-error
            *ngIf="password.errors">
            Password is required
          </mat-error>
          <mat-error
            *ngIf="password.errors">
```

```
                    Password must be of length 6
                </mat-error>
            <mat-icon matSuffix>lock</mat-icon>

        </mat-form-field>
        <mat-form-field>
            <input
            matInput
            placeholder="Password"
            type="password"
            required
            minlength="6"
            name="repassword"
            ngModel
            #repassword="ngModel">
            <mat-error
              *ngIf="repassword.errors">
                Re-Enter Password is required
            </mat-error>
          <mat-icon matSuffix>lock</mat-icon>
        </mat-form-field>
        <button
          mat-raised-button
          color="primary"
          class="login-button">
            Sign Up
        </button>
      </form>
    </mat-card-content>
  </mat-tab>
</mat-tab-group>
</mat-card>
</div>
```

2. register-user.component.css

```css
.login-page {
    top: 0;
    margin: 0;
    padding: 0;
    width: 100%;
    height: 100%;
    display: block;
    background-color: #303F9F;
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
    overflow: hidden;
}

mat-card {
```

```css
    width: 30%;
    margin: 10% auto;
    border-radius: 8px;
  }
  mat-form-field, mat-checkbox {
    width: 80%;
    display: block;
    margin: auto;
  }
  mat-card-content {
    margin-top: 20px;
  }

  .login-button {
    margin-top: 20px;
    display: block;
    width: 80%;
    margin: 20px auto;
  }
```

3. app.module.ts

```typescript
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { FormsModule } from '@angular/forms';
import {MatButtonModule} from '@angular/material/button';
import {MatFormFieldModule} from '@angular/material/form-field';
import {MatCardModule} from '@angular/material/card';
import {MatTabsModule} from '@angular/material/tabs';
import {MatInputModule} from '@angular/material/input';
import {MatCheckboxModule} from '@angular/material/checkbox';
import {MatIconModule} from '@angular/material/icon';
import { AppComponent } from './app.component';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { RegisterUserComponent } from './register-user/register-user.component';
import { ɵAnimationGroupPlayer } from '@angular/animations';
@NgModule({
  declarations: [
    AppComponent,
    RegisterUserComponent
  ],
  imports: [
    BrowserModule,
    BrowserAnimationsModule,
    FormsModule,
    MatInputModule,
    MatCardModule,
    MatTabsModule,
    MatFormFieldModule,
    MatButtonModule,
```

```
    MatCheckboxModule,
    MatIconModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

4. app.component.spec.ts

```typescript
import { TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { AppComponent } from './app.component';

describe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [
        RouterTestingModule
      ],
      declarations: [
        AppComponent
      ],
    }).compileComponents();
  });

  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have as title 'angular-login'`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('angular-login');
  });

  it('should render title', () => {
    const fixture = TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement as HTMLElement;
    expect(compiled.querySelector('.content span')?.textContent).toContain('angular-
login app is running!');
  });
});
```

Output