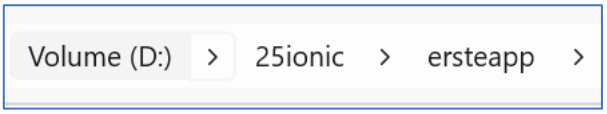
Ionic – Formular erstellen - Seite „Login“ und „Registrieren“



1. Login-Seite erstellen
2. (click)-Funktion für Login
3. Login Seite als redirect einstellen, damit es als Startseite kommt
4. Registrieren-Seite erstellen
5. Cards erstellen inkl. Input-Felder, Radio-Group
6. Button von Login-Seite auf Registrieren weiterleiten – click()-Funktion
7. Login in das Side-Menü aufnehmen

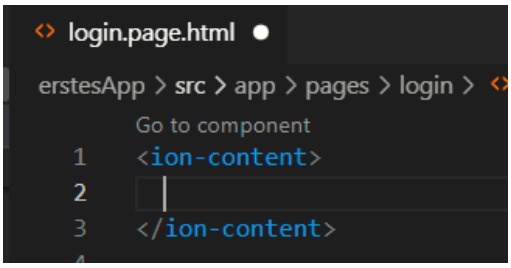
1)Login Site erstellen und anlegen

Erstelle eine neue Seite mit dem Namen „login“. Diese soll ebenfalls im Ordner „pages“ angelegt werden.

```
PS D:\ionic_start> cd erstesApp
PS D:\ionic_start\erstesApp> ionic g page pages/login
```

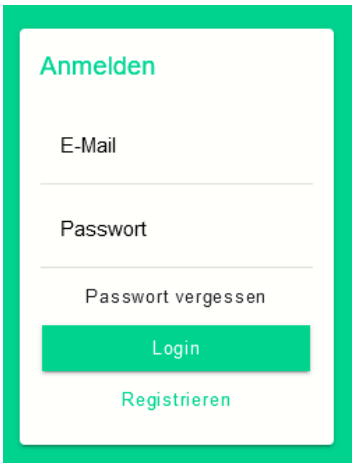
Öffne die login.page.html.

Entferne den Header, den braucht man hier nicht. Somit bleibt nur der content leer zurück:



Ziel:

Titel, E-Mail-Feld, Passwort-Feld, Passwort-Vergessen-Link, 2 Buttons: Login und Registrieren; grüner Hintergrund – alles zentriert



- Zentrieren: wie bei der Loader-Seite. Kopiere die CSS in diese CSS.

login.page.html ● login.page.scss ×

erstesApp > src > app > pages > login > login.page.scss > ...

```scss
.flex-center {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100%;
}
```

- Passend dazu, füge den div-Bereich mit der Verbindung zur CSS-Klasse ein.

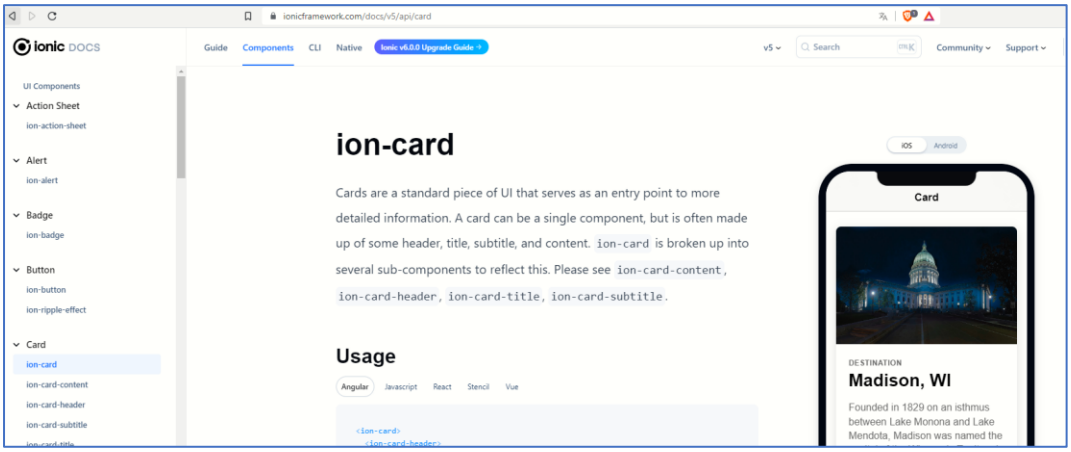
login.page.html ●

erstesApp > src > app > pages > login > login.page.html > ...

Go to component

```html
<ion-content>
  <div class="flex-center">
  </div>
</ion-content>
```

- In der login.page.html erstelle eine „card“ – kennt man aus „bootstrap“, welche die Inputfelder und Buttons aufnehmen soll.
  Dafür holt man sich den Code aus den UI-Components.



Füge den Code in die HTML ein:

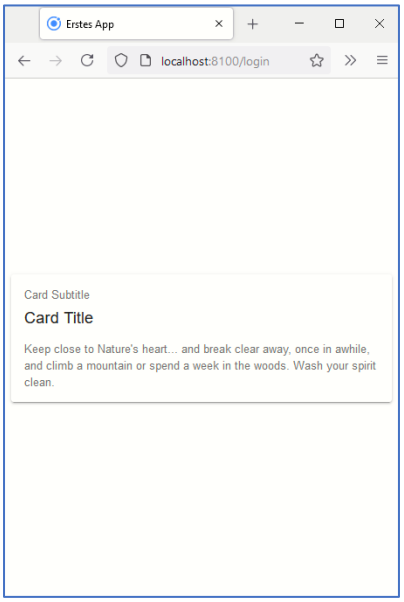
login.page.html

erstesApp > src > app > pages > login > login.page.html > ion-content > div.flex-center

Go to component

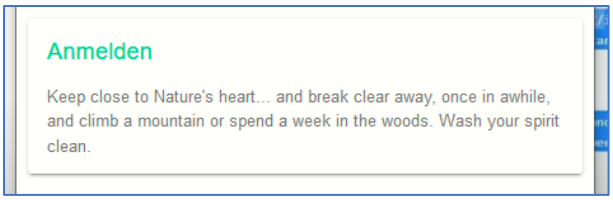
```
<ion-content>
  <div class="flex-center">
    <ion-card>
      <ion-card-header>
        <ion-card-subtitle>Card Subtitle</ion-card-subtitle>
        <ion-card-title>Card Title</ion-card-title>
      </ion-card-header>

      <ion-card-content>
        Keep close to Nature's heart... and break clear away, once in awhile,
        and climb a mountain or spend a week in the woods. Wash your spirit clean.
      </ion-card-content>
    </ion-card>

  </div>
</ion-content>
```

Ergebnis:



- Ändere den Inhalt der „card“:
  Zuerst entferne den Subtitel und ändere den titel auf „Anmelden“
  Farbe auf „success“

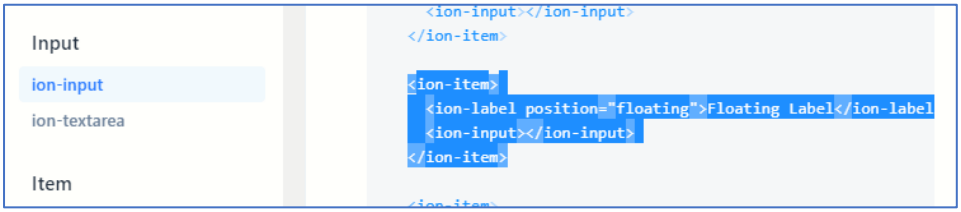
```
<ion-content>
  <div class="flex-center">
    <ion-card>
      <ion-card-header>
        <ion-card-title color="success">Anmelden</ion-card-title>
      </ion-card-header>
```



- **Input-Felder**

  Man kann auf der Ionic – doc Seite danach suchen und es aus „ion-input“ kopieren, oder man schreibt es gleich selbst.

  Es soll das mit dem „floating text“ verwendet werden:



  „floating“ erzeugt ein dynamisches Hochklappen des Labels bei der Eingabe

  Das kommt in den Content der card:

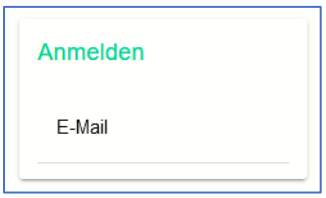
```
8   <ion-card-content>
9     <ion-item>
10      <ion-label position="floating">Floating Label</ion-label>
11      <ion-input></ion-input>
12    </ion-item>
```

```
<ion-item>
  <ion-label position="floating">E-Mail</ion-label>
  <ion-input type="email"></ion-input>
 </ion-item>
```

  Ändere dann den Namen und füge den Typ „email“ ein.

```
10        <ion-label position="floating">E-Mail</ion-label>
11        <ion-input type="email"></ion-input>
```

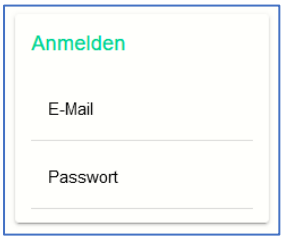
  Ergebnis:



- **Passwort-Feld erstellen**

  Kopiere den E-Mail Bereich, füge ihn ein und ändere es passend auf Passwort:

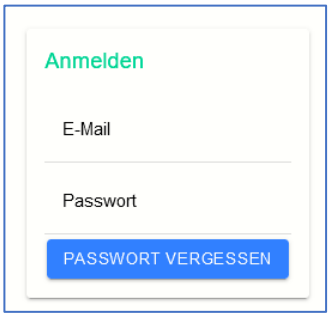
```
12          </ion-item>
13          <ion-item>
14            <ion-label position="floating">Passwort</ion-label>
15            <ion-input type="password"></ion-input>
16          </ion-item>
17
```



- **<u>Darunter soll ein Button erstellt werden für „Passwort vergessen“</u>**

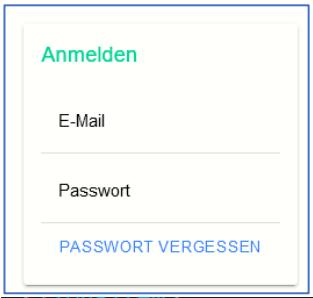
  Erstelle einen <ion-button> Bereich

```
15            <ion-input type="password"></ion-input>
16          </ion-item>
17          <ion-button>Passwort vergessen</ion-button>
18        </ion-card-content>
19      </ion-card>
```



Anpassungen::

- Der Hintergrund soll keine Farbe aufweisen, daher gib ein
  - fill=“clear“ – entfernt den Hintergrund



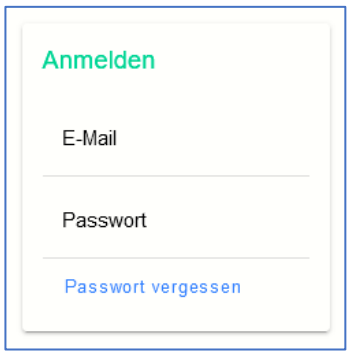

```
<ion-button fill="clear">Passwort vergessen</ion-button>
```

- In einem Button werden die Buchstaben großgeschrieben. Das soll hier verändert werden.

Dafür muss in der login.page.scss der Button umgestellt werden, nämlich auf transform „initial“ und nicht das automatisch eingestellte „up“ (uppercase). Info: „initial“ übernimmt den Text genauso wie er geschrieben ist.

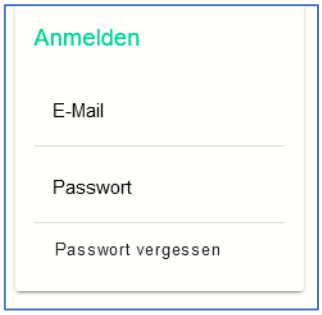
text-transform: initial;

```
ion-button{
  text-transform: initial;
}
```



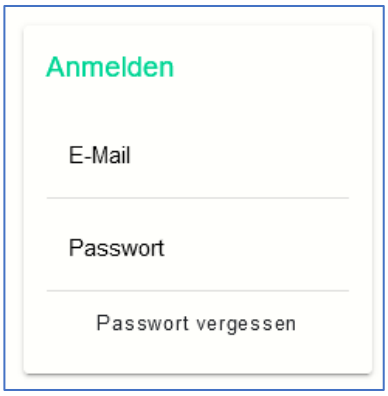
- Farbe soll nicht blau sein, sondern ebenfalls schwarz, daher ändere die color auf „dark“

```
<ion-button color="dark" fill="clear">Passwort vergessen</ion-button>
</ion-card-content>
```



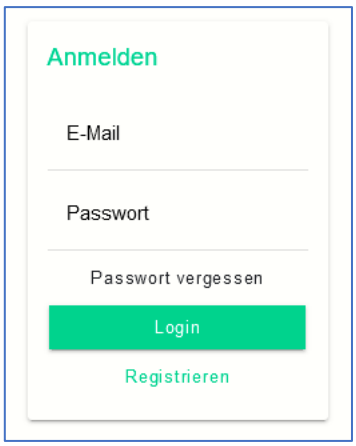
- Der Text soll zentriert erscheinen, daher füge ein:
  - size="full"

```
<ion-button color="dark" size="full" fill="clear">Passwort verges
```



- **Login Button und Register Button erstellen**

```
<ion-button color="dark" size="full" fill="clear">Passwort vergessen</ion-button>
<ion-button color="success" size="full">Login</ion-button>
<ion-button color="success" size="full" fill="clear">Registrieren</ion-button>
</ion-card-content>
</ion-card>
```



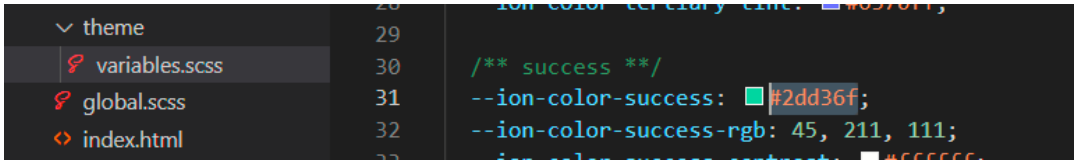
Die Schreibweise ist hier ebenfalls wieder in Original und nicht in Großbuchstaben, weil in der CSS immer noch das „initial“ gilt. Wenn man das ändern will, kann man das gerne tun. Z.B. indem man dem oberen Button (color=“dark“) den Namen mit in die CSS gibt.

Aber hier ist das schon gut und passend, so wie es ist.

- **Hintergrundfarbe einstellen**

  Die Hintergrundfarbe soll folgenden Code erhalten, nämlich der, den auch der Button mit „success“ aufweist, damit es zusammenpasst.

  Den Farbcode findet man im Ordner „theme“ in der Datei „variables.scss“. Kopiere den Hexadezimalcode von dort und verwende ihn dann:



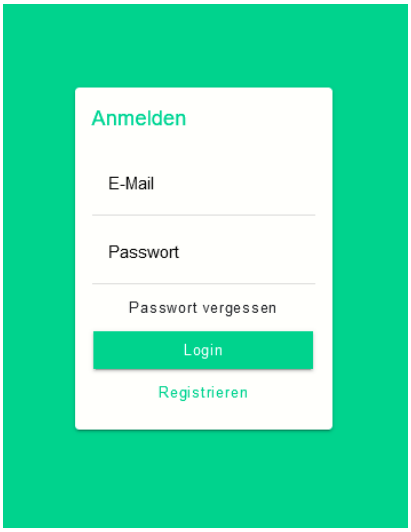
#2dd36f

In die Datei „login.page.scss“ füge nun diesen CSS Code ein, wobei man den Content aus der HTML anspricht.

Beachte: vor dem „background“ müssen zwei – stehen!

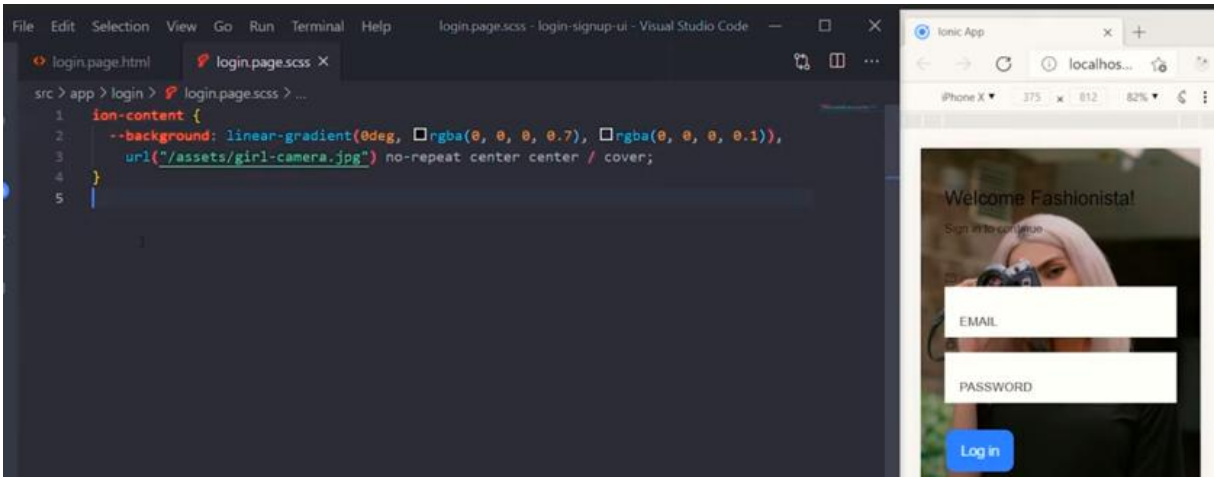
```
11
12   ion-content{
13     --background:  #2dd36f;
14   }
15
```

Ergebnis:



- **Hintergrundbild einfügen**

Nur zur Info: https://www.youtube.com/watch?v=aLA5nYCN6kU ca. 5:00



Oder:
Es soll nun noch ein Hintergrundbild eingefügt werden, ebenfalls aus dem „asset“-Ordner.

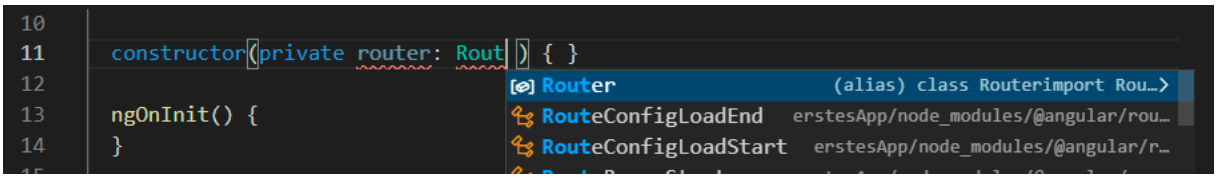
```
<ion-content padding style="background-image:url('../../assets/imgs/hak.jpg');
            background-repeat: no-repeat; background-size: 100%">
    <br>
    <h2>Willkommen zu unserem ersten Login App der Digbiz-Hak Mistelbach</h2>
```

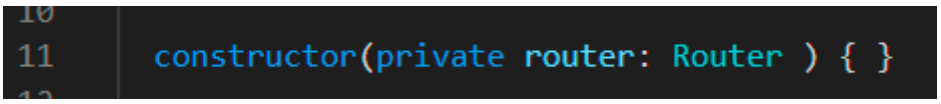
2.)Klick auf LOGIN soll später eine Funktion ausführen

2a)Login-Funktion erstellen

Öffne die „login.page.ts“, die ja zur Login-Seite dazugehört. Von dort aus soll nun diese Weiterleitung organisiert werden.

Im „constructor“ muss man zuerst die private Route anlegen, die, wenn man mit der Maus den Vorschlag annimmt, auch gleichzeitig den IMPORT in Zeile 2 vornimmt. Dafür klick bevor das Wort „Router“ ausgeschrieben ist, auf den passenden Vorschlag und übernimm in dann mit „Enter“.

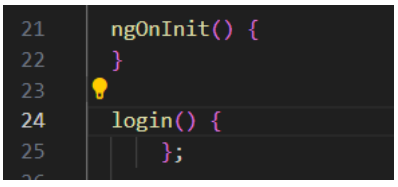




Nur dann legt sich automatisch der Import dazu passend auch an. Ansonsten muss man den in der Zeile 2 selbst schreiben.

Danach muss man unterhalb der Funktion „ngOnInit()“ die neue Funktion „login()“ erstellen.

Diese wird später in der login.ts wichtige Funktionen ausführen.



Das ist nun vorbereitet.

Klick-Funktion beim Button herstellen

Wechsle in die „login.page.html“ zum Button. Dort muss der Click-Event nun auf diese neue Funktion erfolgen:

- click in Klammern
- in Anführungszeichen der Name der Funktion, die vorher angelegt wurde und die Navigation vornimmt, inkl. Funktionsklammern danach

(click)="login()"

```
23    <ion-button fill="clear" size="full" color="dark">Passwort vergessen</ion-button>
24    <ion-button (click)="login()" color="success" size="full" >Login</ion-button>
25    <ion-button fill="clear" size="full" color="success">Registrieren</ion-button>
```

Speichern und testen.

3.)Login Seite als Startseite einstellen – redirect

Wenn man die Seite mit der einfachen URL aufruft, soll nun nicht mehr das „Team“ angezeigt werden, sondern die „login“-Seite. Das ist sinnvoll, um sich sofort anmelden zu können.

Öffne die Datei „loader.page.ts“ und ändere die Weiterleitung von „team“ auf „login“. Das Timeout soll natürlich erhalten bleiben.

```
13    ngOnInit() {
14      setTimeout(() => {
15      this.router.navigate(['login']);
16      }, 2000);  }
```

```
setTimeout(() => {
 this.router.navigate(['login']);
 }, 2000); }
```

4.)Erstelle eine neue Seite „registrieren“

```
[OK] Generated page!
PS D:\ionic_start\erstesApp> ionic g page pages/registrieren
```

Für die grüne Hintergrundfarbe kopiere den CSS-Code aus der login.scss in die neu geschaffene registrieren.page.scss.

registrieren.page.scss × registrieren.pa

erstesApp > src > app > pages > registrieren >

```
1   ion-content{
2     --background: #2dd36f;
3   }
```

Öffne die registrieren.page.html.

- Die Überschrift soll mit einem großen „R“ beginnen.
- Dieses Mal soll der Header oben bestehen bleiben. Aber dessen Hintergrund soll ebenfalls grün werden.

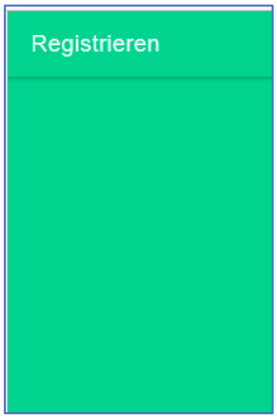
registrieren.page.html registrieren.page.scss

erstesApp > src > app > pages > registrieren > registrieren.p

Go to component

```
1  <ion-header>
2    <ion-toolbar color="success">
3      <ion-title>Registrieren</ion-title>
4    </ion-toolbar>
5  </ion-header>
```

Ergebnis:



5.)Cards erstellen inkl. Input-Felder, Radio-Group

Bereich für Kontaktdaten erstellen

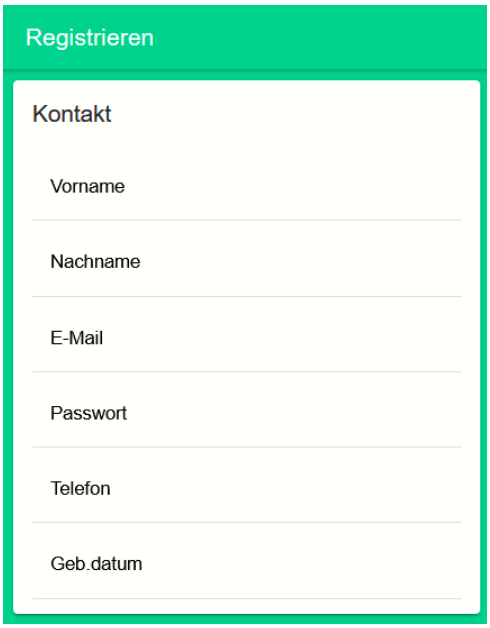
Erstelle eine Card mit Titel „Kontakt“ und einzelnen Input-Feldern.

```
7   <ion-content>
8   <ion-card>
9     <ion-card-header>
10      <ion-card-title>
11        Kontakt
12      </ion-card-title>
13    </ion-card-header>
14  </ion-card>
15  </ion-content>
```

Unter dem ion-card-header soll dann der ion-card-content beginnen, der einige Input-Felder aufweisen soll. Für die Input-Felder kann man diese aus dem Login kopieren und abändern:

- Vorname
- Nachname
- E-Mail
- Passwort
- Telefon
- Geb.Datum

```
13      </ion-card-header>
14      <ion-card-content>
15        <ion-item>
16          <ion-label position="floating">Name</ion-label>
17          <ion-input type="text"></ion-input>
18        </ion-item>
```



Radio-group: männlich/weiblich

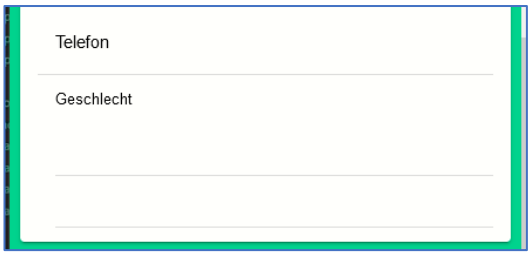
Man kann auch mit Hilfe von Radio-group das Geschlecht wählen lassen:

Darunter soll auch eine Auswahlmöglichkeit erstellt werden, um das Geschlecht anzugeben.

Folgende Elemente sind bedeutend:

- ion-radio-group
- ion-list-header
- ion-item …. für die einzelnen Elemente

```
33          <ion-input type="tel"></ion-input>
34        </ion-item>
35        <ion-radio-group>
36          <ion-list-header>Geschlecht</ion-list-header>
37          <ion-item></ion-item>
38          <ion-item></ion-item>
39        </ion-radio-group>
40      </ion-card-content>
```

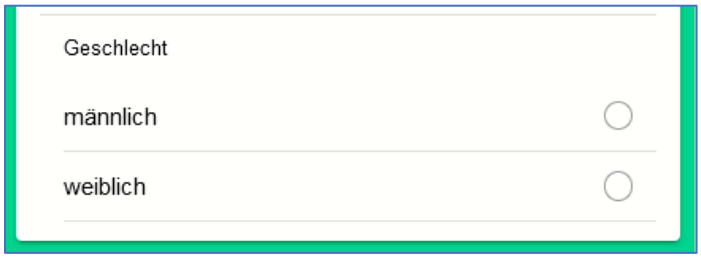
Ergebnis: 

Nun folgen noch die einzelnen Element:

- männlich
- weiblich

<ion-label>männlich</ion-label>
<ion-radio value="m"></ion-radio>

```
36      <ion-list-header>Geschlecht</ion-list-header>
37      <ion-item>
38        <ion-label>männlich</ion-label>
39        <ion-radio value="m"></ion-radio>
40      </ion-item>
41      <ion-item>
42        <ion-label>weiblich</ion-label>
43        <ion-radio value="w"></ion-radio>
44      </ion-item>
45    </ion-radio-group>
```

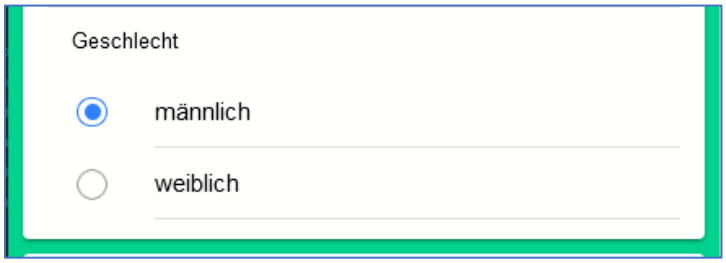


Änderung:

Der Auswahlring soll nicht rechts, sondern links vom Wort stehen:

Dazu verwendet man „slot=“start“ ….wobei „start“ für den Beginn steht. Abgeleitet vom Beginn der Zeile bei der Schrift.

```
38        <ion-label>männlich</ion-label>
39        <ion-radio value="m" slot="start"></ion-radio>
40      </ion-item>
```



Ergebnis:

```
    </ion-item>
    <ion-radio-group>
      <ion-list-header>Geschlecht</ion-list-header>
      <ion-item>
        <ion-label>männlich</ion-label>
        <ion-radio value="m" slot="start"></ion-radio>
      </ion-item>
      <ion-item>
        <ion-label>weiblich</ion-label>
        <ion-radio value="w" slot="start"></ion-radio>
      </ion-item>
    </ion-radio-group>
  </ion-card-content>
```

```
<ion-radio-group>
    <ion-list-header>Geschlecht</ion-list-header>
    <ion-item>
      <ion-label>männlich</ion-label>
      <ion-radio value="m" slot="start"></ion-radio>
    </ion-item>
    <ion-item>
      <ion-label>weiblich</ion-label>
      <ion-radio value="w" slot="start"></ion-radio>
    </ion-item>
  </ion-radio-group>
```

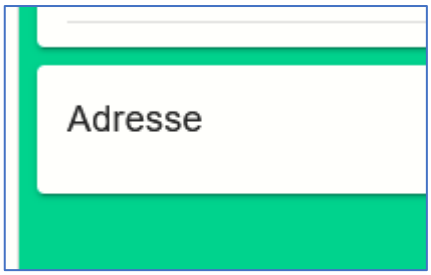
Zweite Card mit Adresse usw.

Erstelle eine weitere Card, welche die Adresse usw. aufnehmen soll.

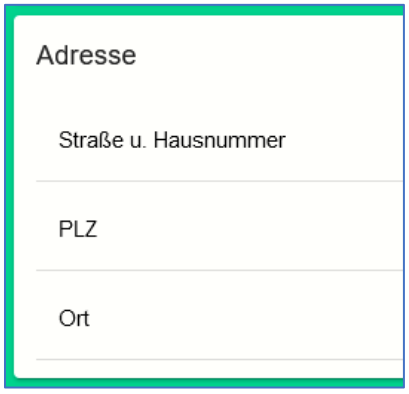
```
  </ion-card>
  <ion-card>
    <ion-card-header>
      <ion-card-title>Adresse</ion-card-title>
    </ion-card-header>
    <ion-card-content>

    </ion-card-content>
  </ion-card>
  </ion-content>

```

```
<ion-card>
  <ion-card-header>
    <ion-card-title>Adresse</ion-card-title>
  </ion-card-header>
  <ion-card-content>

  </ion-card-content>
</ion-card>
```

Ergebnis:



Weitere Input-Felder anlegen. Kopiere dafür als Vorlage welche vom Code darüber.




```
<ion-card>
<ion-card-header>
  <ion-card-title>Adresse</ion-card-title>
</ion-card-header>
<ion-card-content>
  <ion-item>
    <ion-label position="floating">Straße u. Hausnummer</ion-label>
    <ion-input type="text" ></ion-input>
  </ion-item>
  <ion-item>
```

- **<u>Abschließender Button</u>**

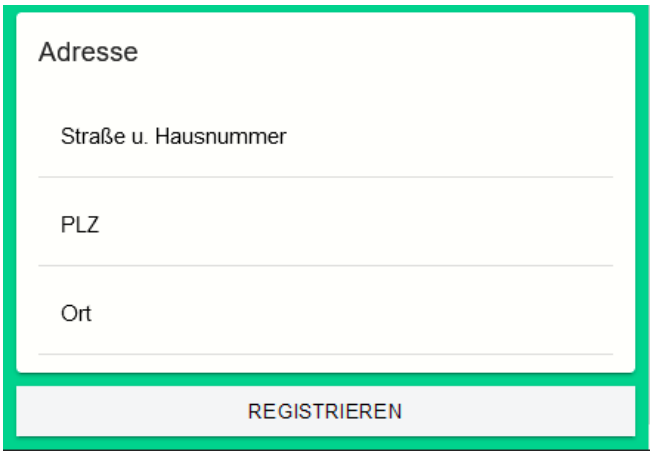
Dieser soll außerhalb der cards stehen und „full-size“ haben, sowie links und rechts passend eingerückt sein.

```
  </ion-card-content>
</ion-card>
<ion-button size="full" color="light">Registrieren</ion-button>
</ion-content>
```

Ergebnis, aber die „margin“ fehlt noch, damit es eingeückt ist:

Dafür lege in der scss-Datei diesen Code an:

```
4
5  ion-button{
6    margin: 10px;
7  }
8
```



6.)Klick auf Registrieren soll auf diese Seite weiterleiten

6a)Registrieren-Funktion erstellen in der „login.page.ts“

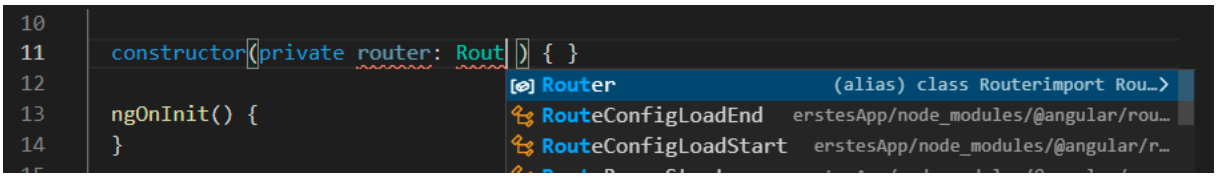
Öffne: login.page.ts

Dazu erstelle eine „registrieren()“-Funktion in der „login.page.ts“.

TS steht für TypeScript und ist für die Logik zuständig, so wie JavaScript.

Öffne die „login.page.ts“, die ja zur Login-Seite dazugehört. Von dort aus soll nun diese Weiterleitung organisiert werden.

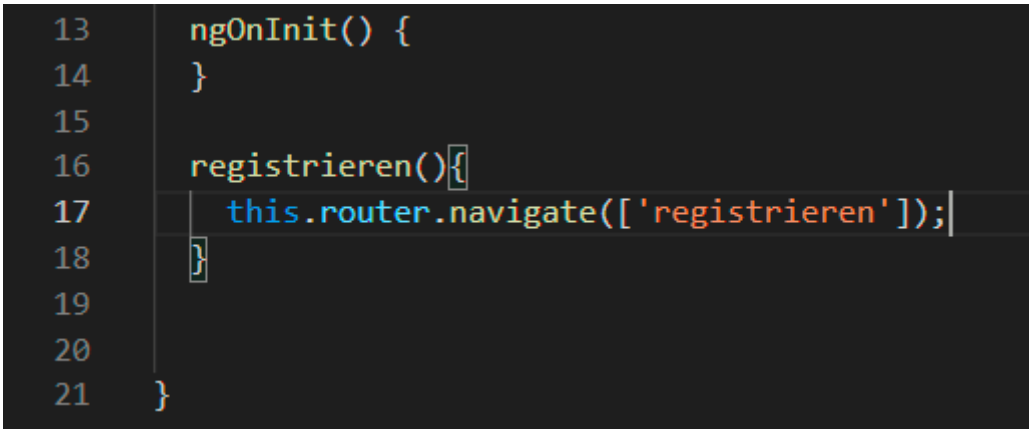
Im „constructor“ muss man zuerst die private Route anlegen, di, wenn man mit der Maus den Vorschlag annimmt, auch gleichzeitig den IMPORT in Zeile 2 vornimmt. Dafür klicke bevor das Wort „Router“ ausgeschrieben ist, auf den passenden Vorschlag und übernimm in dann mit „Enter“.



Nur dann legt sich automatisch der Import dazu passend auch an. Ansonsten muss man den in der Zeile 2 selbst schreiben.

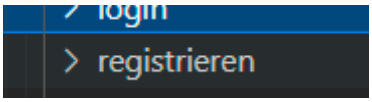
Danach muss man unterhalb der Funktion „ngOnInit()“ die neue Funktion „registrieren()“ erstellen.

Diese zeigt die Navigation auf die Seite „registrieren“ an.




```
registrieren(){
  this.router.navigate(['registrieren']);
}
```

In der viereckigen Klammer muss der Name der Seite richtig geschrieben sein, so wie sie als „page“ auch vorkommt, damit sie auch gefunden wird:



Das ist nun vorbereitet.

6b)Klick-Funktion beim Button herstellen

Wechsle in die „login.page.html“ zum Button. Dort muss der Click-Event nun auf diese neue Funktion erfolgen:

- click in Klammern
- in Anführungszeichen der Name der Funktion, die vorher angelegt wurde und die Navigation vornimmt, inkl. Funktionsklammern danach

(click)="registrieren()"

```
18      <ion-button color="success" size="full">Login</ion-button>
19      <ion-button color="success" size="full" fill="clear" (click)="registrieren()">Registrie
20    </ion-card-content>
```

Speichern und testen.

7)Das Login in das Side-Menü aufnehmen

Damit man das Login auch jederzeit aufrufen kann, soll es auch in das Side-Menü aufgenommen werden.

Öffne dazu die Datei app.component.ts

- kopiere die Zeile 10 und füge diese darunter ein.
- Ändere den titel und die url
- Als Icon wähle „enter“

```
<ion-icon name="enter"></ion-icon>
```

```
8   public appPages = [
9     { title: 'Team', url: '/team', icon: 'man' },
10    { title: 'Standort', url: '/standort', icon: 'home' },
11    { title: 'Login', url: '/login', icon: 'enter' },
12  ];
```

Ergebnis: