

Pemrograman Mobile Cross Platform

Styling and Custom Component

OVERVIEW

Finishing this module, students will be able to:

- Able to modify app styling and theming configuration
- Able to create custom components

This module is divided into several parts. OVERVIEW explains the overall objective of this module and how to understand its contents. MATERIALS NEEDED and SOFTWARE REQUIREMENTS shows the requirements. ACTIVITIES section explains how to do various things, as explained in class. Finally, the TASK describes the tasks that students need to finish and get graded in the corresponding module.

MATERIALS NEEDED / REFERENCES

- This module
- Materials from the class (PowerPoint slides, etc.) – *dari kelas teori*
- Ionic Framework Official Documentation

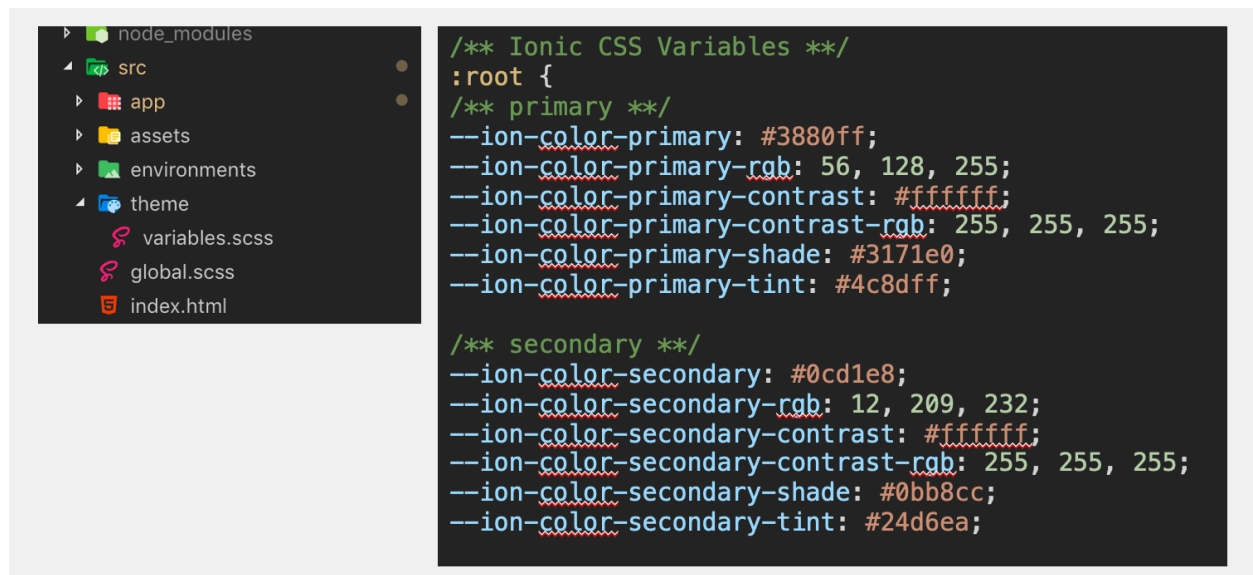
SOFTWARE REQUIREMENTS

- Visual Studio Code
- Android Studio or XCode
- Google Chrome

ACTIVITIES

Changing the global colors variables using Ionic Color Generator

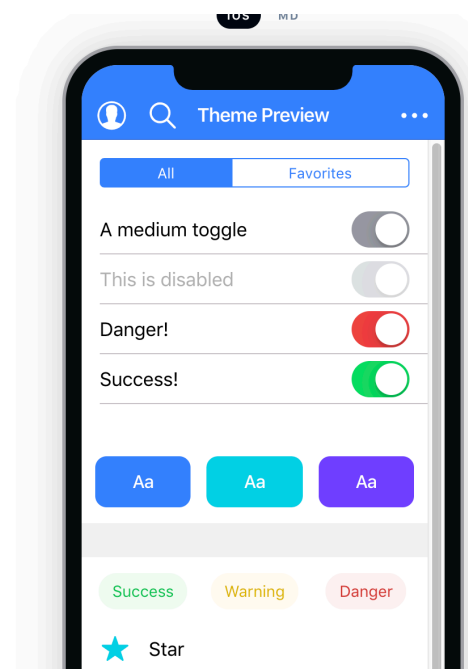
<https://ionicframework.com/docs/theming/color-generator>



Color Generator

Create custom color palettes for your app's UI. Update a color's hex values, check the demo app on the right to confirm, then copy and paste the generated code directly into your Ionic project.

Primary	<input type="text" value="#3880ff"/>
Secondary	<input type="text" value="#0cd1e8"/>
Tertiary	<input type="text" value="#7044ff"/>
Success	<input type="text" value="#10dc60"/>
Warning	<input type="text" value="#ffce00"/>



CSS Variables

[Copy](#)

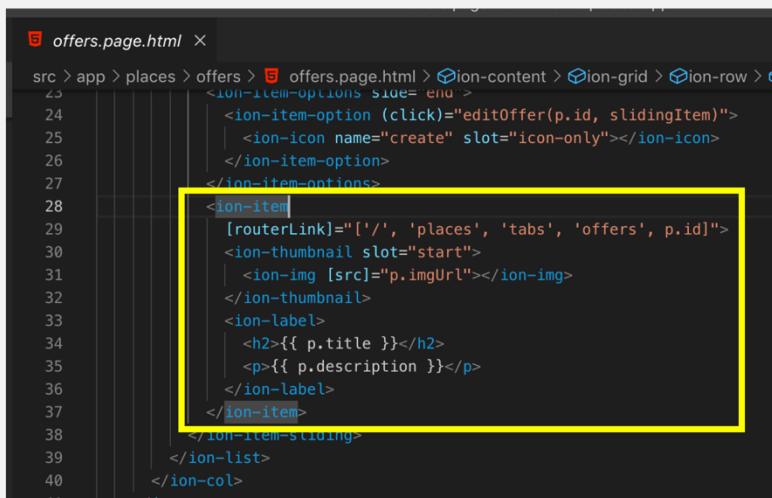
```
:root {
  --ion-color-primary: #666633;
  --ion-color-primary-rgb: 102,102,51;
  --ion-color-primary-contrast: #ffffff;
  --ion-color-primary-contrast-rgb: 255,255,255;
  --ion-color-primary-shade: #5a5a2d;
  --ion-color-primary-tint: #757547;

  --ion-color-secondary: #0cd1e8;
  --ion-color-secondary-rgb: 12,209,232;
  --ion-color-secondary-contrast: #ffffff;
  --ion-color-secondary-contrast-rgb: 255,255,255;
  --ion-color-secondary-shade: #0bb8cc;
  --ion-color-secondary-tint: #24d6ea;

  --ion-color-tertiary: #7044ff;
  --ion-color-tertiary-rgb: 112,68,255;
  --ion-color-tertiary-contrast: #ffffff;
  --ion-color-tertiary-contrast-rgb: 255,255,255;
  --ion-color-tertiary-shade: #633ce0;
  --ion-color-tertiary-tint: #7e57ff;
}
```

Custom Components

Creating a Custom Component

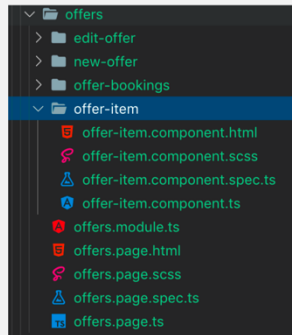


```
src > app > places > offers > offers.page.html > ion-content > ion-grid > ion-row >
24 <ion-item-option (click)="editOffer(p.id, slidingItem)">
25   <ion-icon name="create" slot="icon-only"></ion-icon>
26 </ion-item-option>
27 </ion-item-options>
28 <ion-item
29   [routerLink]="['/', 'places', 'tabs', 'offers', p.id]">
30   <ion-thumbnail slot="start">
31     <ion-img [src]="p.imageUrl"></ion-img>
32   </ion-thumbnail>
33   <ion-label>
34     <h2>{{ p.title }}</h2>
35     <p>{{ p.description }}</p>
36   </ion-label>
37 </ion-item>
38 </ion-item-sliding>
39 </ion-list>
40 </ion-col>
41 </ion-row>
```

Let's move these lines containing the `<ion-item>` and create a new component holding the item.

Generate a New Component

> ionic generate component /places/offers/offer-item



```
offer-item.component.html X
src > app > places > offers > offer-item > offer-item.component.html > ...
1 <ion-item [routerLink]="['/', 'places', 'tabs', 'offers', offer.id]">
2   <ion-thumbnail slot="start">
3     <ion-img [src]="offer.imgUrl"></ion-img>
4   </ion-thumbnail>
5   <ion-label>
6     <h2>{{ offer.title }}</h2>
7   <div>
8     <ion-icon name="calendar"></ion-icon>
9     <ion-text color="primary">{{ getDummyDate() | date }}</ion-text>
10    <span>to</span>
11    <ion-icon name="calendar"></ion-icon>
12    <ion-text color="primary">{{ getDummyDate() | date }}</ion-text>
13  </div>
14 </ion-label>
15 </ion-item>
16
```

Generate a New Component











> ionic generate component /places/offers/offer-item

```
offers.module.ts X
src > app > places > offers > offers.module.ts > OffersPageModule
18 @NgModule({
19   imports: [
20     CommonModule,
21     FormsModule,
22     IonicModule,
23     RouterModule.forChild(routes)
24   ],
25   declarations: [OffersPage, OfferItemComponent]
26 })
27 export class OffersPageModule {}
```

```
offer-item.component.ts X
src > app > places > offers > offer-item > offer-item.component.ts > OfferItemComponent
1 import { Component, OnInit, Input } from '@angular/core';
2 import { Place } from '../place.model';
3
4 @Component({
5   selector: 'app-offer-item',
6   templateUrl: './offer-item.component.html',
7   styleUrls: ['./offer-item.component.scss'],
8 })
9 export class OfferItemComponent implements OnInit {
10   @Input() offer: Place;
11
12   constructor() {}
13
14   ngOnInit() {}
15}
```

```
offers.module.ts X offers.page.html X
src > app > places > offers > offers.page.html > ion-content > ion-grid > ion-row > ion-col
15 </ion-header>
16
17 <ion-content>
18   <ion-grid>
19     <ion-row>
20       <ion-col size="12" size-sm="8" offset-sm="2" text-center>
21         <ion-list>
22           <ion-item-sliding *ngFor="let offer of loadedPlaces" #slidingItem>
23             <ion-item-options side="end">
24               <ion-item-option (click)="editOffer(offer.id, slidingItem)">
25                 <ion-icon name="create" slot="icon-only"></ion-icon>
26               </ion-item-option>
27             </ion-item-options>
28             <app-offer-item [offer]="offer"></app-offer-item>
29           </ion-item-sliding>
30         </ion-list>
31       </ion-col>
32     </ion-row>
```

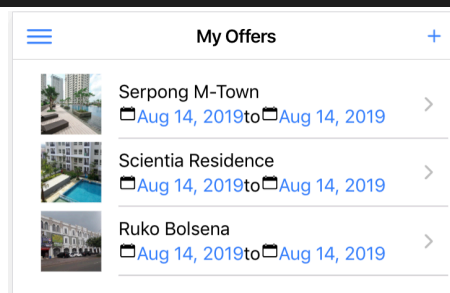
Resulting component

	My Offers	+
	<p>Serpong M-Town</p> <p> Aug 14, 2019 to  Aug 14, 2019</p>	>
	<p>Scientia Residence</p> <p> Aug 14, 2019 to  Aug 14, 2019</p>	>
	<p>Ruko Bolsena</p> <p> Aug 14, 2019 to  Aug 14, 2019</p>	>

TASKS (deadline today)

1. Continue working on the Ionic Places App from last week.
2. Change all the default color variables with new colors of your likings in the **variables.scss** file, using the Ionic Color Generator
3. Create a custom component for the Offer Items as explained in the previous pages of this module.


```
offers.module.ts  offers.page.html x
src > app > places > offers > offers.page.html > ion-content > ion-grid > ion-row > ion-
15  </ion-header>
16
17  <ion-content>
18    <ion-grid>
19      <ion-row>
20        <ion-col size="12" size-sm="8" offset-sm="2" text-center>
21          <ion-list>
22            <ion-item-sliding *ngFor="let offer of loadedPlaces" #slidingItem>
23              <ion-item-options side="end">
24                <ion-item-option (click)="editOffer(offer.id, slidingItem)">
25                  <ion-icon name="create" slot="icon-only"></ion-icon>
26                </ion-item-option>
27              </ion-item-options>
28              <app-offer-item [offer]="offer"></app-offer-item>
29            </ion-item-sliding>
30          </ion-list>
31        </ion-col>
32      </ion-row>
```



4. Create a custom component for the Your Bookings component in the bookings.page.html, so it will look like this:

```
<ion-content>
  <ion-grid>
    <ion-row>
      <ion-col size-md="6" offset-md="3">
        <ion-list>
          <ion-item-sliding *ngFor="let b of bookings" #slidingBooking>
            <app-your-booking-item [booking]="b"></app-your-booking-item>
            <ion-item-options>
              <ion-item-option color="danger">
                <ion-icon name="trash" slot="icon-only" (click)="onCancelBooking(b.id, slidingBooking)"></ion-icon>
              </ion-item-option>
            </ion-item-options>
          </ion-item-sliding>
        </ion-list>
      </ion-col>
    </ion-row>
  </ion-grid>
</ion-content>
```

With the same view as follows:

	My Bookings
Ruko Bolsena	
Guests: 2	
Scientia Residence	
Guests: 8	