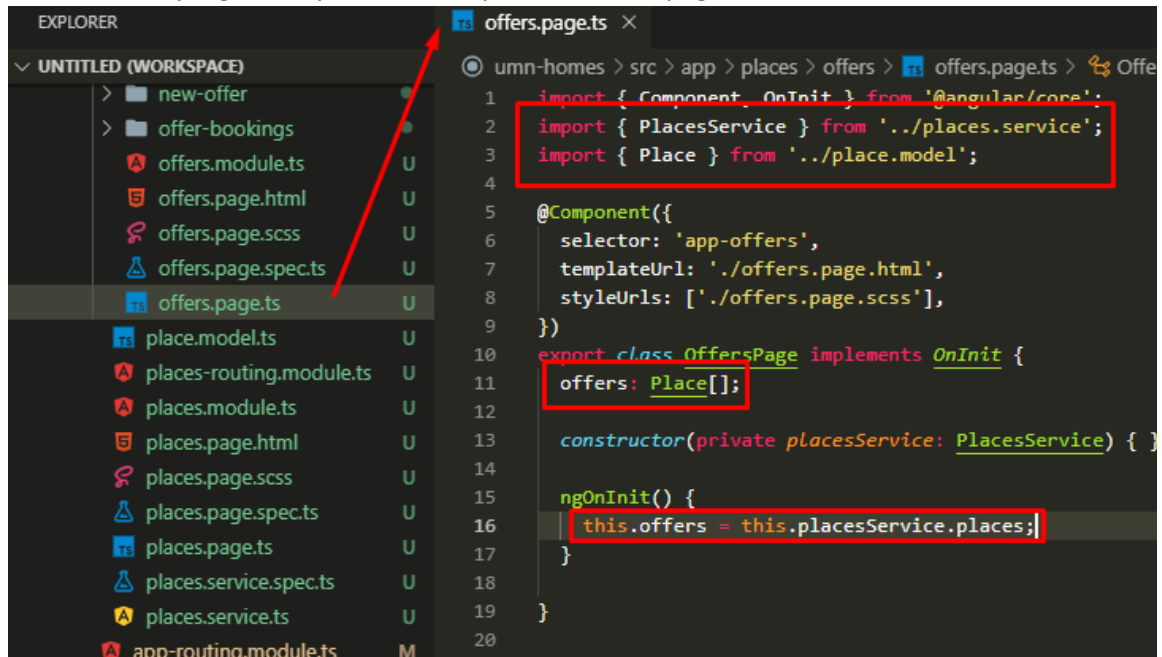


## Modul Week 05: Mobile Cross-Platform Programming

### Membuat Halaman Offers

Kita akan melanjutkan dengan membuat halaman Offers. Untuk sementara kita akan menggunakan data places sebagai offers.

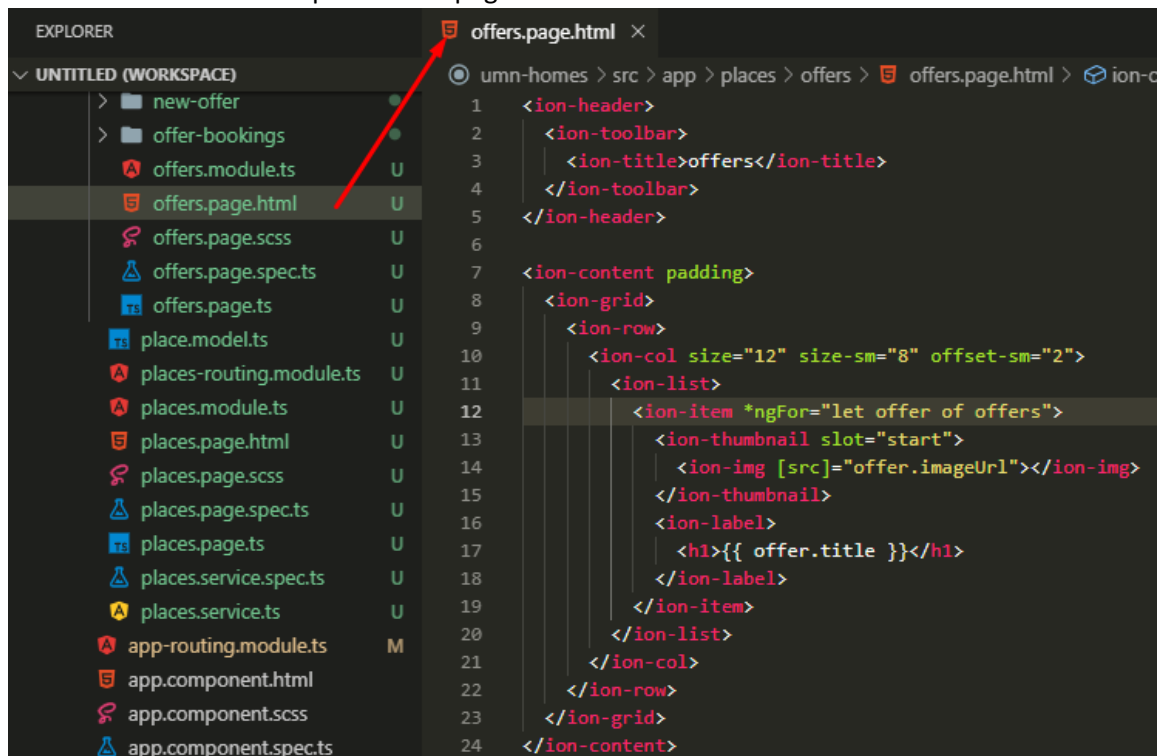
1. Tambahkan kode berikut pada offers.page.ts. Kode ini adalah untuk mengambil data Place dari PlacesService yang nantinya akan ditampilkan di offers.page.html



```
EXPLORER
  UNTITLED (WORKSPACE)
    new-offer
    offer-bookings
    offers.module.ts
    offers.page.html
    offers.page.scss
    offers.page.spec.ts
    offers.page.ts
    place.model.ts
    places-routing.module.ts
    places.module.ts
    places.page.html
    places.page.scss
    places.page.spec.ts
    places.page.ts
    places.service.spec.ts
    places.service.ts
    app-routing.module.ts

  offers.page.ts
    umn-homes > src > app > places > offers > offers.page.ts > OffersPage
    1 import { Component, OnInit } from '@angular/core';
    2 import { PlacesService } from '../places.service';
    3 import { Place } from '../place.model';
    4
    5 @Component({
    6   selector: 'app-offers',
    7   templateUrl: './offers.page.html',
    8   styleUrls: ['./offers.page.scss'],
    9 })
    10 export class OffersPage implements OnInit {
    11   offers: Place[];
    12
    13   constructor(private placesService: PlacesService) { }
    14
    15   ngOnInit() {
    16     this.offers = this.placesService.places;
    17   }
    18
    19 }
    20
```

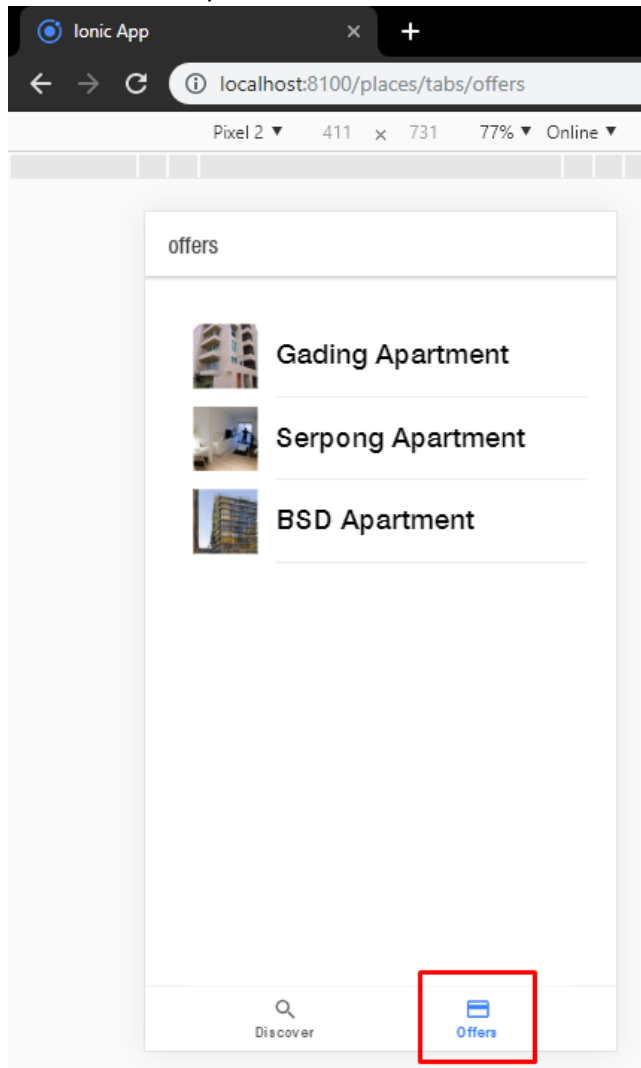
2. Tambahkan kode berikut pada offers.page.html



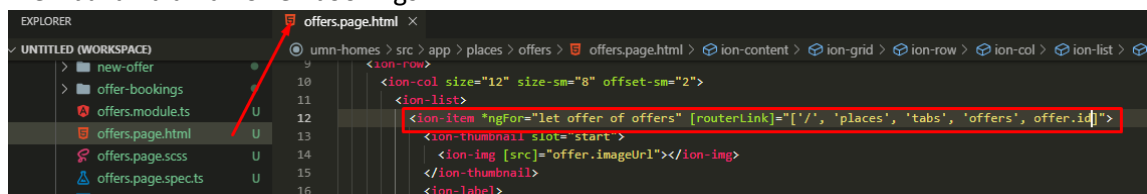
```
EXPLORER
  UNTITLED (WORKSPACE)
    new-offer
    offer-bookings
    offers.module.ts
    offers.page.html
    offers.page.scss
    offers.page.spec.ts
    offers.page.ts
    place.model.ts
    places-routing.module.ts
    places.module.ts
    places.page.html
    places.page.scss
    places.page.spec.ts
    places.page.ts
    places.service.spec.ts
    places.service.ts
    app-routing.module.ts
    app.component.html
    app.component.scss
    app.component.spec.ts

  offers.page.html
    umn-homes > src > app > places > offers > offers.page.html > OffersPage
    1 <ion-header>
    2   <ion-toolbar>
    3     <ion-title>offers</ion-title>
    4   </ion-toolbar>
    5 </ion-header>
    6
    7 <ion-content padding>
    8   <ion-grid>
    9     <ion-row>
    10       <ion-col size="12" size-sm="8" offset-sm="2">
    11         <ion-list>
    12           <ion-item *ngFor="let offer of offers">
    13             <ion-thumbnail slot="start">
    14               <ion-img [src]="offer.imageUrl"></ion-img>
    15             </ion-thumbnail>
    16             <ion-label>
    17               <h1>{{ offer.title }}</h1>
    18             </ion-label>
    19           </ion-item>
    20         </ion-list>
    21       </ion-col>
    22     </ion-row>
    23   </ion-grid>
    24 </ion-content>
```

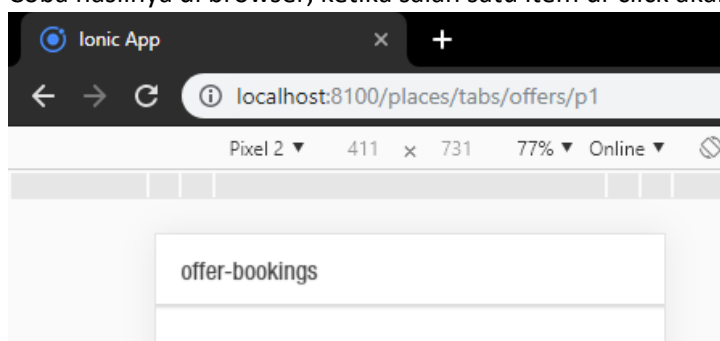
3. Coba lihat hasilnya di browser



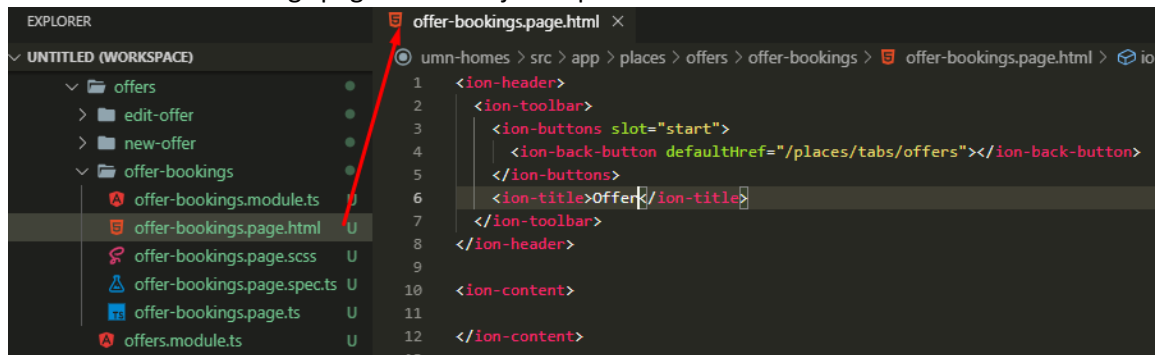
4. Tambahkan routerLink pada ion-item di offers.page.html sehingga setiap item bias di-click dan membuka halaman offer-bookings



5. Coba hasilnya di browser, ketika salah satu item di-click akan membuka halaman offers-booking.

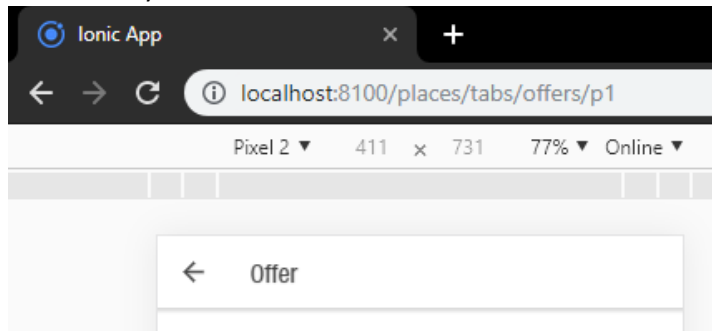


6. Modifikasi offer-bookings.page.html menjadi seperti berikut.



```
1 <ion-header>
2   <ion-toolbar>
3     <ion-buttons slot="start">
4       <ion-back-button defaultHref="/places/tabs/offers"></ion-back-button>
5     </ion-buttons>
6     <ion-title>Offer</ion-title>
7   </ion-toolbar>
8 </ion-header>
9
10 <ion-content>
11
12 </ion-content>
```

7. Lihat hasilnya di browser.

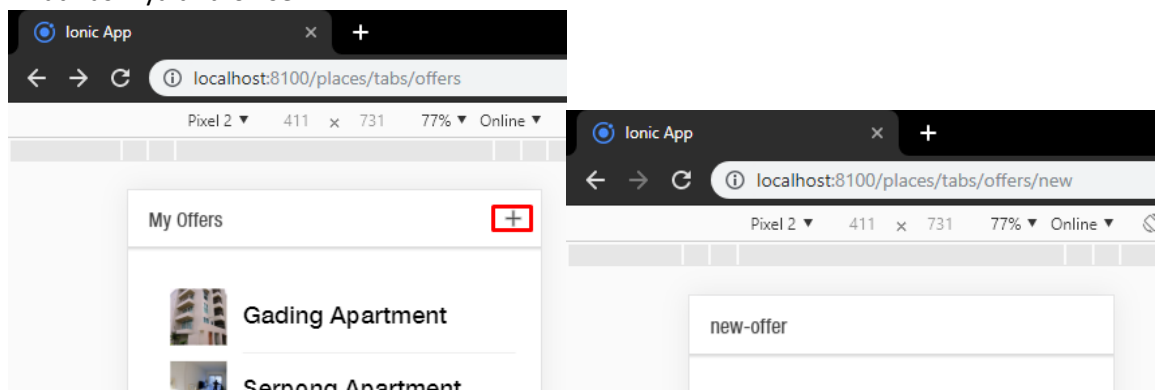


8. Tambahkan kode berikut pada offers.page.html untuk menambah sebuah tombol tambah di halaman offers yang ketika di-click akan membuka halaman new-offer

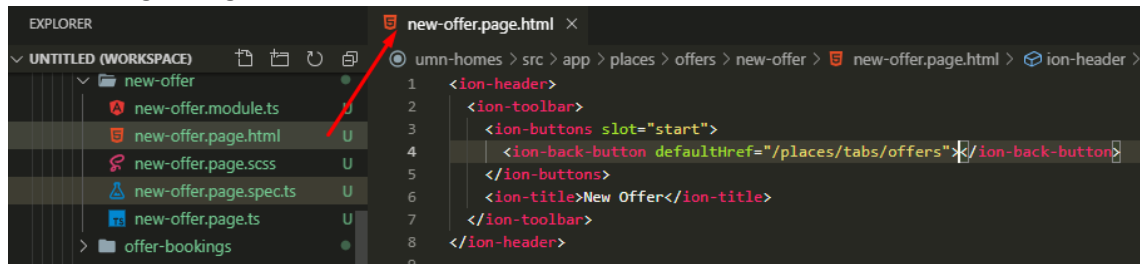


```
1 <ion-header>
2   <ion-toolbar>
3     <ion-title>My Offers</ion-title>
4     <ion-buttons slot="primary">
5       <ion-button routerLink="/places/tabs/offers/new">
6         <ion-icon name="add" slot="icon-only"></ion-icon>
7       </ion-button>
8     </ion-buttons>
9   </ion-toolbar>
10 </ion-header>
```

9. Lihat hasilnya di browser

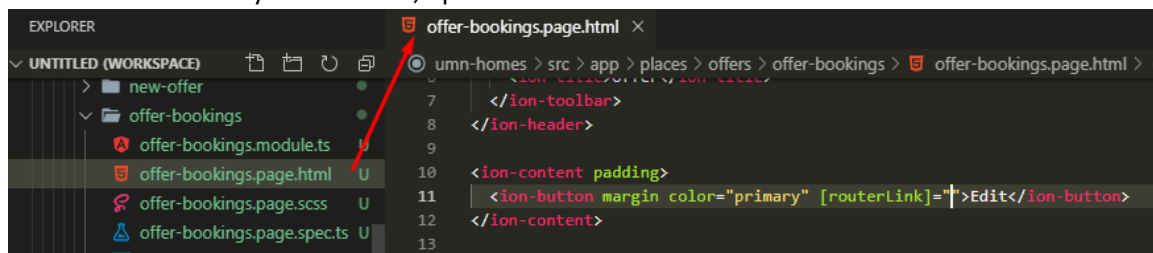


10. Tambahkan kode berikut pada new-offer.page.html untuk menambahkan tombol back. Kemudian coba lihat hasilnya di browser, apakah tombol back pada halaman new offer muncul dan berfungsi dengan baik?



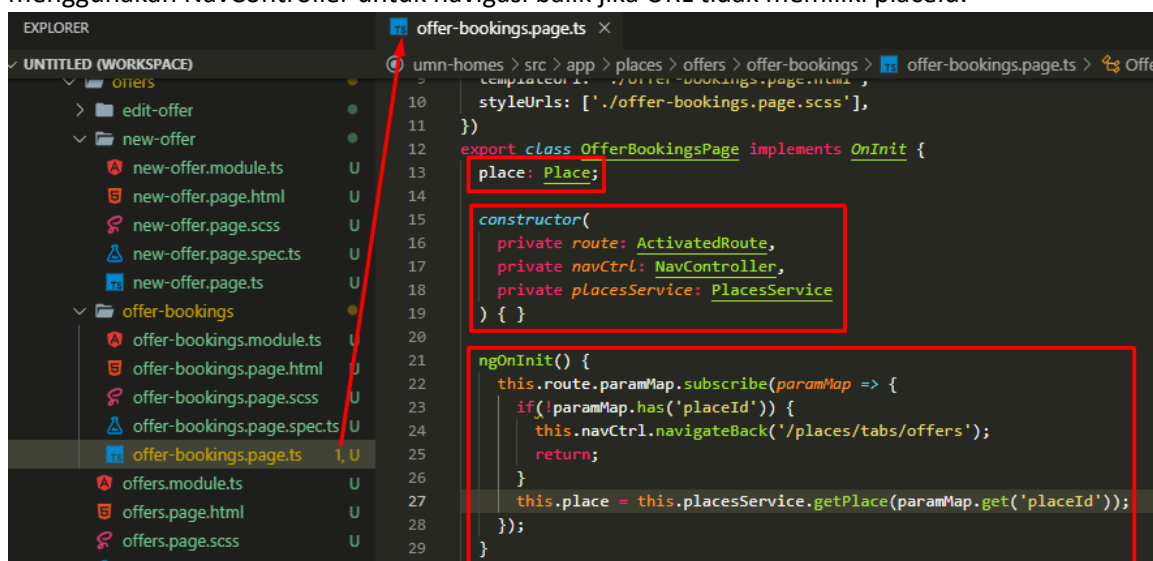
```
1 <ion-header>
2 <ion-toolbar>
3   <ion-buttons slot="start">
4     <ion-back-button defaultHref="/places/tabs/offers"></ion-back-button>
5   </ion-buttons>
6   <ion-title>New Offer</ion-title>
7 </ion-toolbar>
8 </ion-header>
```

11. Tambahkan kode berikut pada halaman offer-bookings.page.html untuk menambahkan tombol edit. Dan lihat hasilnya di browser, apakah tombol edit muncul?



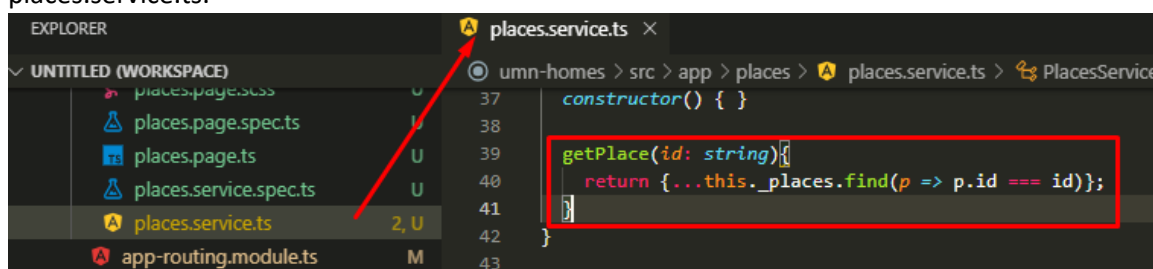
```
7 </ion-toolbar>
8 </ion-header>
9
10 <ion-content padding>
11   <ion-button margin color="primary" [routerLink]=">Edit</ion-button>
12 </ion-content>
13
```

12. Tambahkan kode berikut pada offer-bookings.page.ts. Kita menggunakan ActivatedRoute untuk mendapatkan placeId secara dinamis dengan melakukan subscribe ke paramMap dan menggunakan NavController untuk navigasi balik jika URL tidak memiliki placeId.



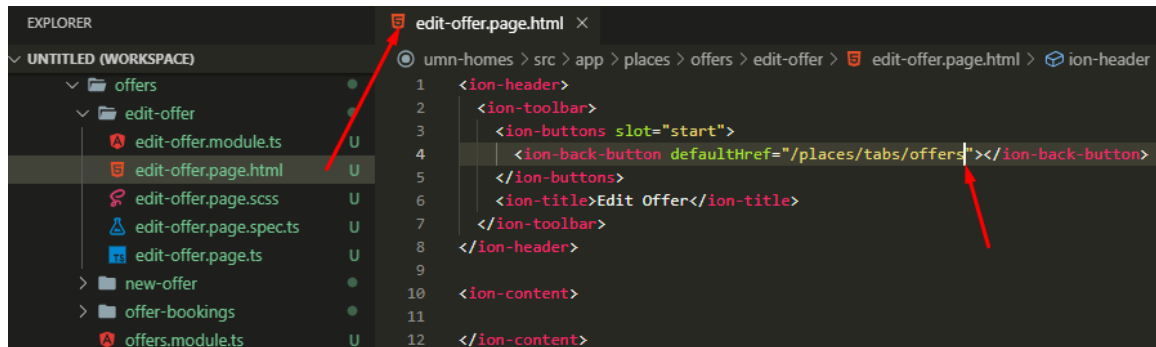
```
10
11
12 export class OfferBookingsPage implements OnInit {
13   place: Place;
14
15   constructor(
16     private route: ActivatedRoute,
17     private navCtrl: NavController,
18     private placesService: PlacesService
19   ) {}
20
21   ngOnInit() {
22     this.route.paramMap.subscribe(paramMap => {
23       if (!paramMap.has('placeId')) {
24         this.navCtrl.navigateBack('/places/tabs/offers');
25         return;
26       }
27       this.place = this.placesService.getPlace(paramMap.get('placeId'));
28     });
29   }
30 }
```

13. Selanjutnya kita tambahkan kode berikut untuk membuat method getPlace() pada places.service.ts.



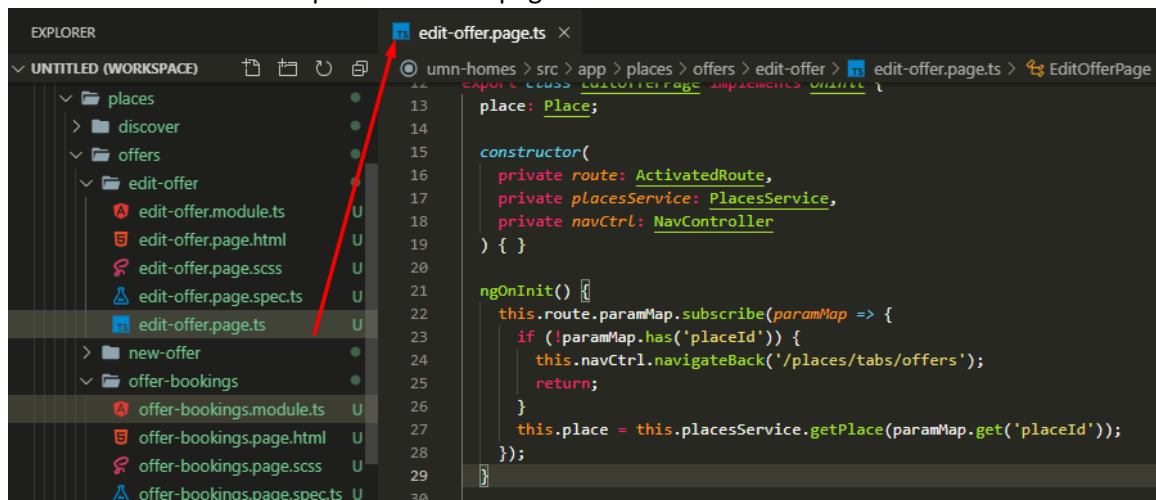
```
37 constructor() {}
38
39 getPlace(id: string) {
40   return {...this._places.find(p => p.id === id)};
41 }
42
43
```

14. Tambahkan kode berikut pada edit-offer.page.ts untuk menambahkan back button. Perhatikan bahwa defaultHref akan berbeda-beda, tergantung pada place yang di-edit, sehingga defaultHref harus dinamis.



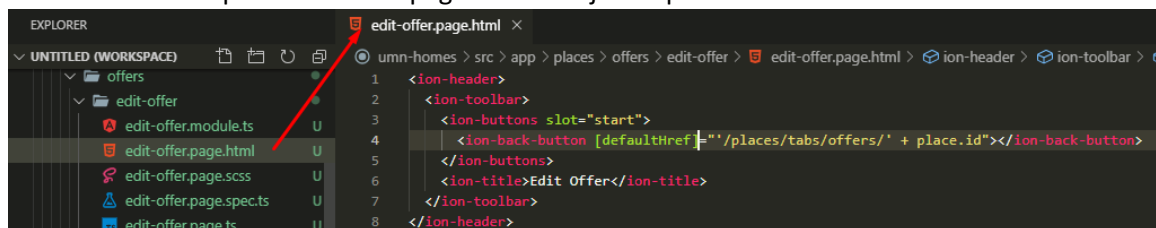
```
1 <ion-header>
2   <ion-toolbar>
3     <ion-buttons slot="start">
4       <ion-back-button defaultHref="/places/tabs/offers/"></ion-back-button>
5     </ion-buttons>
6     <ion-title>Edit Offer</ion-title>
7   </ion-toolbar>
8 </ion-header>
9
10 <ion-content>
11
12 </ion-content>
```

15. Tambahkan kode berikut pada edit-offer.page.ts



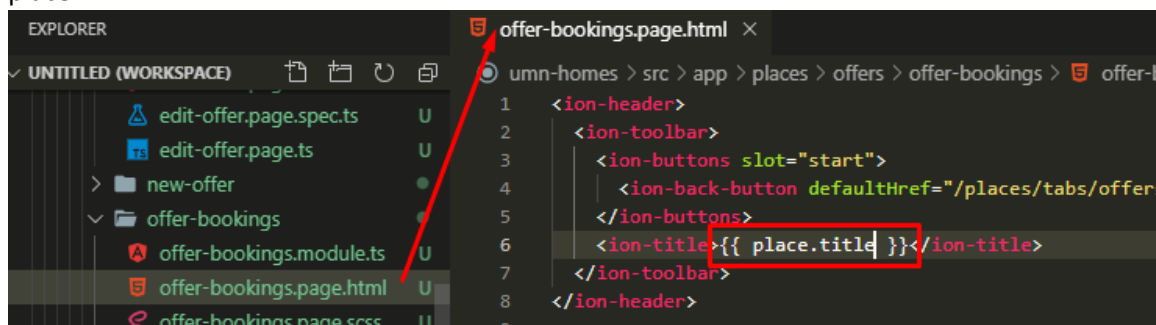
```
12 export class EditOfferPage implements OnInit {
13   place: Place;
14
15   constructor(
16     private route: ActivatedRoute,
17     private placesService: PlacesService,
18     private navCtrl: NavController
19   ) {}
20
21   ngOnInit() {
22     this.route.paramMap.subscribe(paramMap => {
23       if (!paramMap.has('placeId')) {
24         this.navCtrl.navigateBack('/places/tabs/offers');
25         return;
26       }
27       this.place = this.placesService.getPlace(paramMap.get('placeId'));
28     });
29   }
30 }
```

16. Ubah defaultHref pada edit-offer.page.html menjadi seperti berikut.



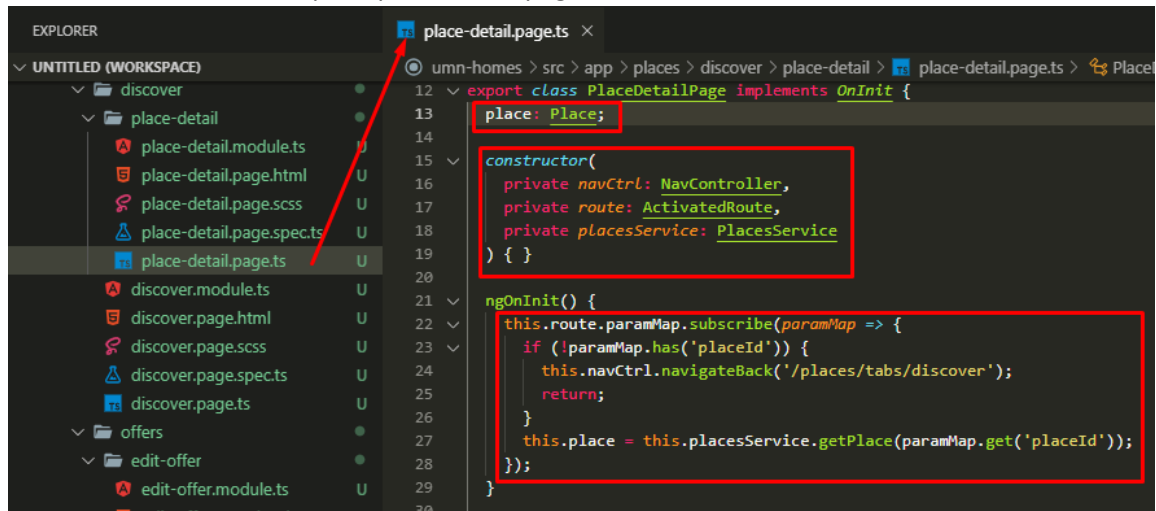
```
1 <ion-header>
2   <ion-toolbar>
3     <ion-buttons slot="start">
4       <ion-back-button [defaultHref]="'/places/tabs/offers/' + place.id"></ion-back-button>
5     </ion-buttons>
6     <ion-title>Edit Offer</ion-title>
7   </ion-toolbar>
8 </ion-header>
```

17. Ubah kode pada offer-bookings.page.html menjadi seperti berikut untuk menampilkan nama place



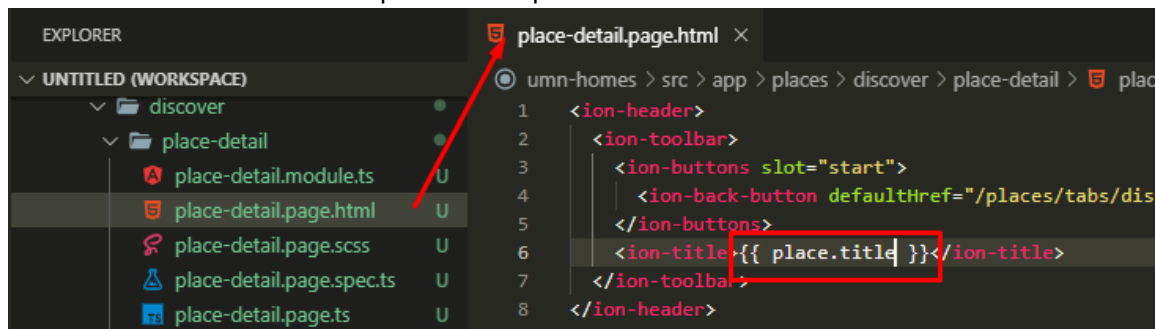
```
1 <ion-header>
2   <ion-toolbar>
3     <ion-buttons slot="start">
4       <ion-back-button defaultHref="/places/tabs/offers/"></ion-back-button>
5     </ion-buttons>
6     <ion-title>{{ place.title }}</ion-title>
7   </ion-toolbar>
8 </ion-header>
```

18. Coba di browser, ketika masuk ke halaman edit, kemudian refresh halaman edit, ketika tombol back di klik, apakah kembali ke place yang sebelumnya di buka?
19. Tambahkan kode berikut pada place-detail.page.ts



```
12 export class PlaceDetailPage implements OnInit {
13   place: Place;
14
15   constructor(
16     private navCtrl: NavController,
17     private route: ActivatedRoute,
18     private placesService: PlacesService
19   ) { }
20
21   ngOnInit() {
22     this.route.paramMap.subscribe(paramMap => {
23       if (!paramMap.has('placeId')) {
24         this.navCtrl.navigateBack('/places/tabs/discover');
25         return;
26       }
27       this.place = this.placesService.getPlace(paramMap.get('placeId'));
28     });
29   }
30 }
```

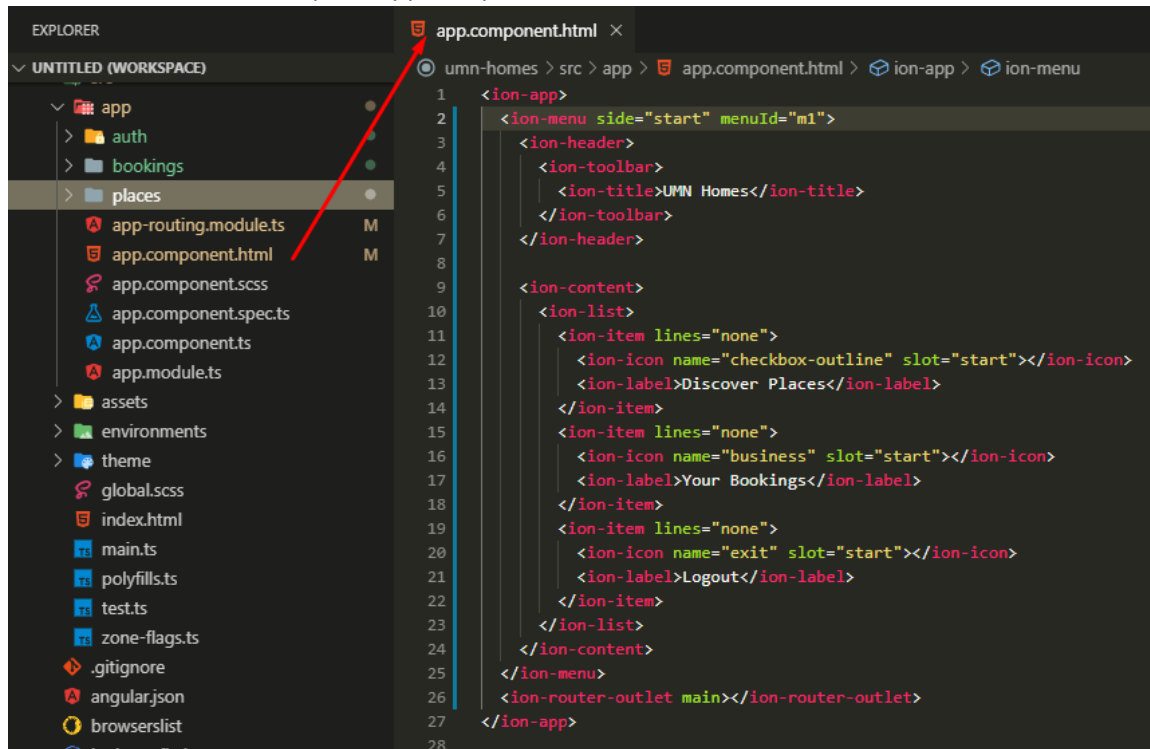
20. Tambahkan kode berikut pada place-detail.page.html, dan coba buka halaman place-detail melalui tab discover dan lihat apakah nama place muncul?



```
1 <ion-header>
2   <ion-toolbar>
3     <ion-buttons slot="start">
4       <ion-back-button defaultHref="/places/tabs/discover">
5     </ion-buttons>
6     <ion-title>{{ place.title }}</ion-title>
7   </ion-toolbar>
8 </ion-header>
```

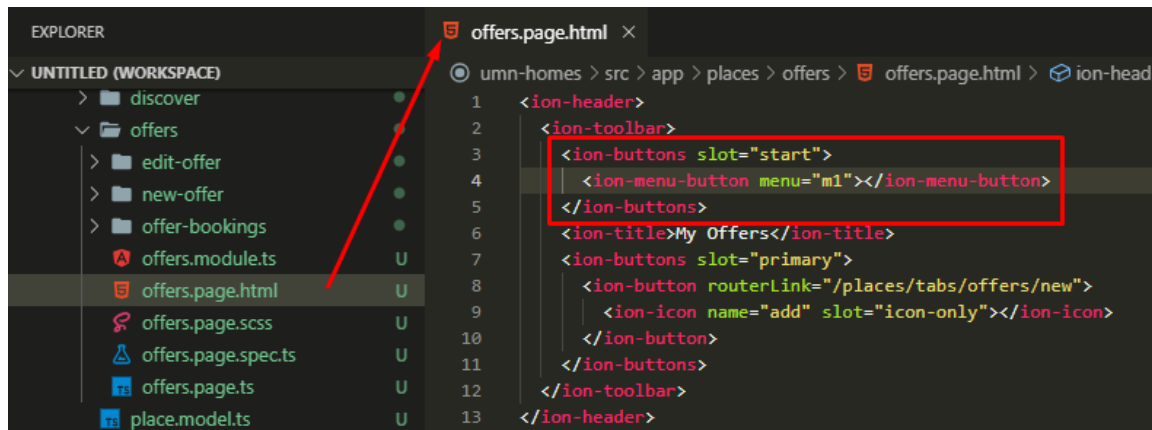
## Side Drawer

1. Tambahkan kode berikut pada app.component.html



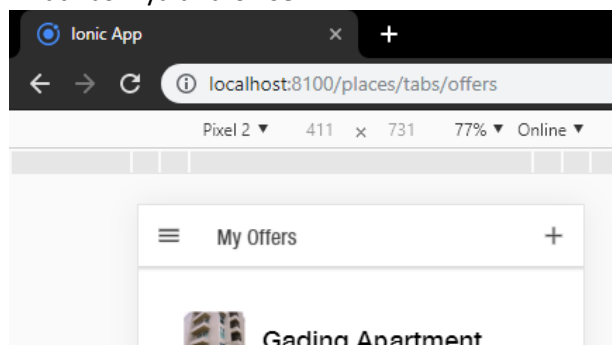
```
1 <ion-app>
2 <ion-menu side="start" menuId="m1">
3
4 <ion-header>
5   <ion-toolbar>
6     <ion-title>UMN Homes</ion-title>
7   </ion-toolbar>
8 </ion-header>
9
10 <ion-content>
11   <ion-list>
12     <ion-item lines="none">
13       <ion-icon name="checkbox-outline" slot="start"></ion-icon>
14       <ion-label>Discover Places</ion-label>
15     </ion-item>
16     <ion-item lines="none">
17       <ion-icon name="business" slot="start"></ion-icon>
18       <ion-label>Your Bookings</ion-label>
19     </ion-item>
20     <ion-item lines="none">
21       <ion-icon name="exit" slot="start"></ion-icon>
22       <ion-label>Logout</ion-label>
23     </ion-item>
24   </ion-list>
25 </ion-content>
26 </ion-menu>
27 <ion-router-outlet main></ion-router-outlet>
28 </ion-app>
```

2. Tambahkan kode berikut pada offers.page.html untuk menmbahkan menu side drawer pada halaman offers.

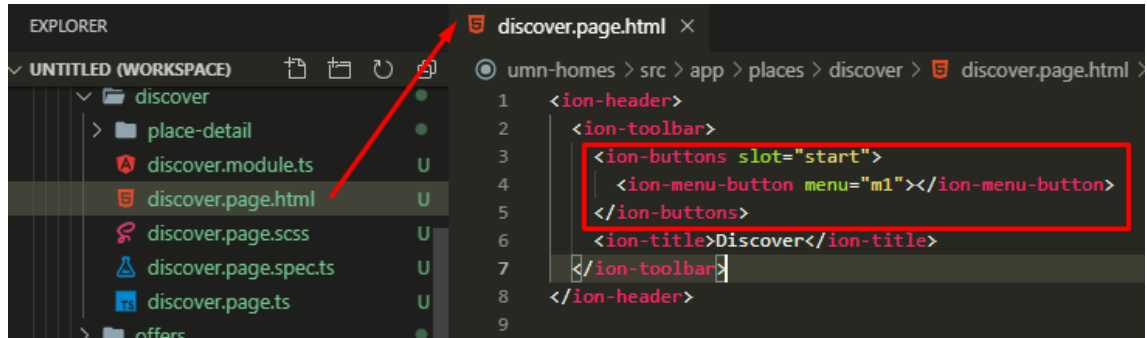


```
1 <ion-header>
2   <ion-toolbar>
3     <ion-buttons slot="start">
4       <ion-menu-button menu="m1"></ion-menu-button>
5     </ion-buttons>
6   <ion-title>My Offers</ion-title>
7   <ion-buttons slot="primary">
8     <ion-button routerLink="/places/tabs/offers/new">
9       <ion-icon name="add" slot="icon-only"></ion-icon>
10     </ion-button>
11 </ion-buttons>
12 </ion-toolbar>
13 </ion-header>
```

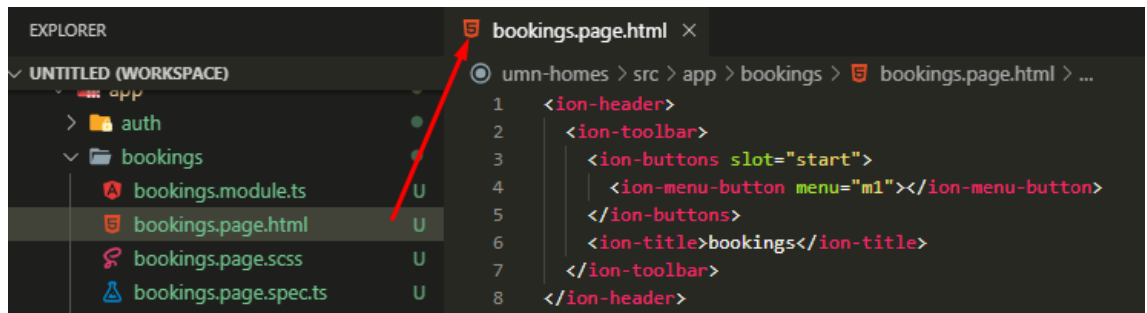
3. Lihat hasilnya di browser



4. Lakukan hal yang sama pada halaman discover dan bookings.

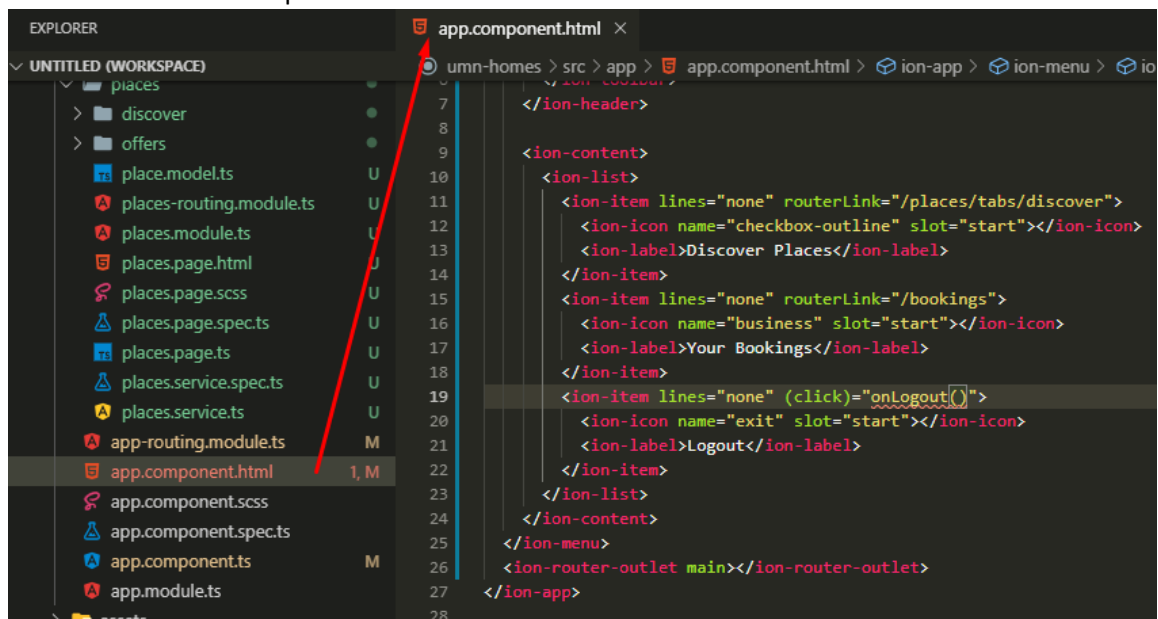


```
1 <ion-header>
2   <ion-toolbar>
3     <ion-buttons slot="start">
4       <ion-menu-button menu="m1"></ion-menu-button>
5     </ion-buttons>
6     <ion-title>Discover</ion-title>
7   </ion-toolbar>
8 </ion-header>
9
```



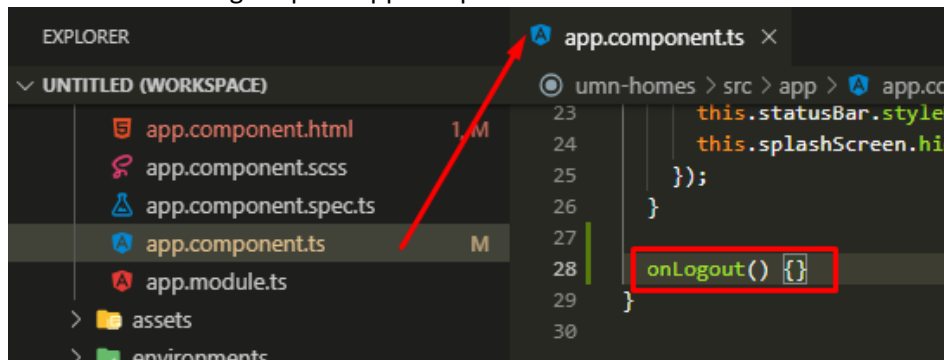
```
1 <ion-header>
2   <ion-toolbar>
3     <ion-buttons slot="start">
4       <ion-menu-button menu="m1"></ion-menu-button>
5     </ion-buttons>
6     <ion-title>bookings</ion-title>
7   </ion-toolbar>
8 </ion-header>
9
```

5. Tambahkan routerLink pada side menu tadi.



```
1 <ion-app>
2   <ion-header>
3     <ion-title>Umn Homes</ion-title>
4   </ion-header>
5   <ion-content>
6     <ion-list>
7       <ion-item lines="none" routerLink="/places/tabs/discover">
8         <ion-icon name="checkbox-outline" slot="start"></ion-icon>
9         <ion-label>Discover Places</ion-label>
10      </ion-item>
11      <ion-item lines="none" routerLink="/bookings">
12        <ion-icon name="business" slot="start"></ion-icon>
13        <ion-label>Your Bookings</ion-label>
14      </ion-item>
15      <ion-item lines="none" (click)="onLogout()">
16        <ion-icon name="exit" slot="start"></ion-icon>
17        <ion-label>Logout</ion-label>
18      </ion-item>
19    </ion-list>
20  </ion-content>
21 </ion-app>
22
```

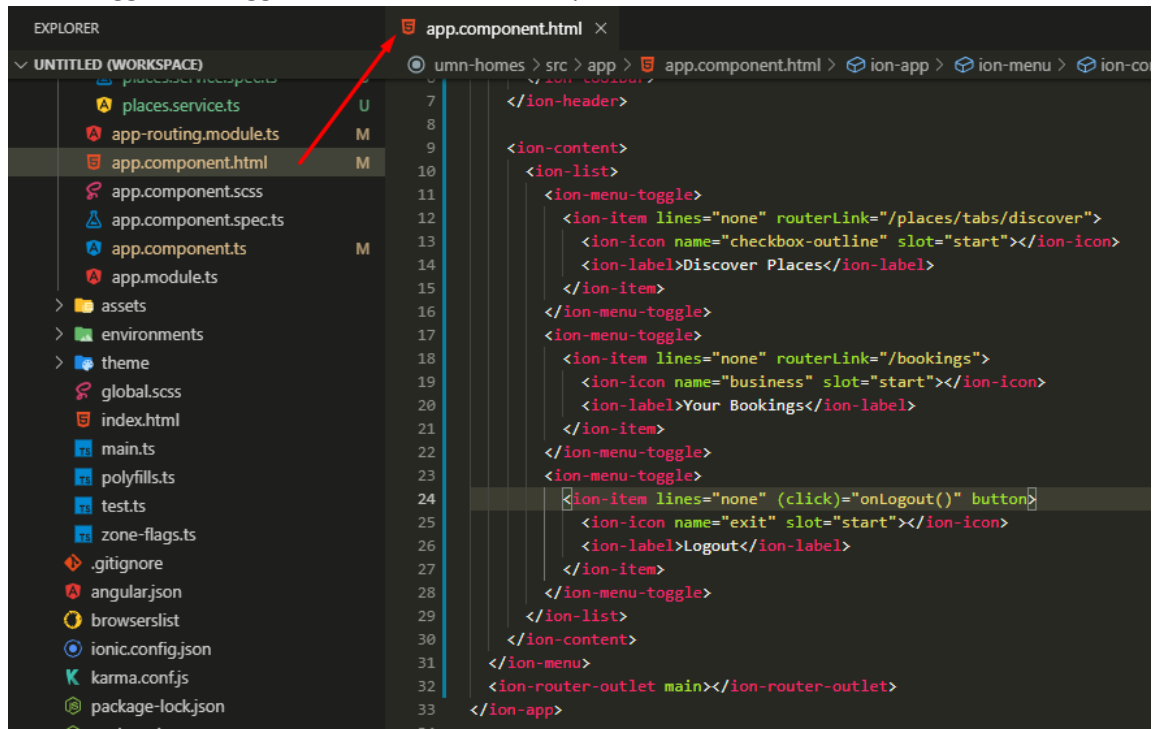
6. Buat method onLogout pada app.component.ts



```
1 import { Component } from '@angular/core';
2 import { Router } from '@angular/router';
3
4 @Component({
5   selector: 'app-root',
6   templateUrl: './app.component.html',
7   styleUrls: ['./app.component.scss'],
8 })
9 export class AppComponent {
10   title = 'Umn Homes';
11
12   constructor(private router: Router) {}
13
14   ngOnInit(): void {}
15
16   onLogout() {}
17 }
18
```



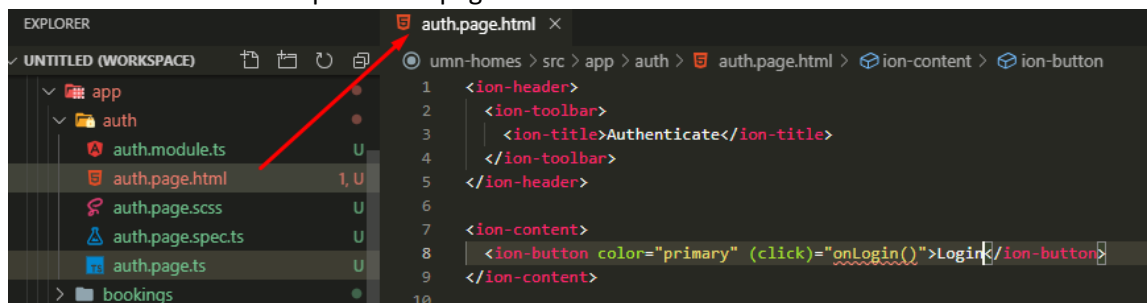
- Ubah kode pada app.component.html menjadi seperti berikut, dengan menambahkan <ion-menu-toggle> sehingga side menu akan tertutup otomatis ketika salah satu menu di-click.



```
7 </ion-header>
8
9 <ion-content>
10   <ion-list>
11     <ion-menu-toggle>
12       <ion-item lines="none" routerLink="/places/tabs/discover">
13         <ion-icon name="checkbox-outline" slot="start"></ion-icon>
14         <ion-label>Discover Places</ion-label>
15       </ion-item>
16     </ion-menu-toggle>
17     <ion-menu-toggle>
18       <ion-item lines="none" routerLink="/bookings">
19         <ion-icon name="business" slot="start"></ion-icon>
20         <ion-label>Your Bookings</ion-label>
21       </ion-item>
22     </ion-menu-toggle>
23     <ion-menu-toggle>
24       <ion-item lines="none" (click)="onLogout()" button>
25         <ion-icon name="exit" slot="start"></ion-icon>
26         <ion-label>Logout</ion-label>
27       </ion-item>
28     </ion-menu-toggle>
29   </ion-list>
30 </ion-content>
31 </ion-menu>
32 <ion-router-outlet main></ion-router-outlet>
33 </ion-app>
```


## Authenticate

- Tambahkan kode berikut pada auth.page.html



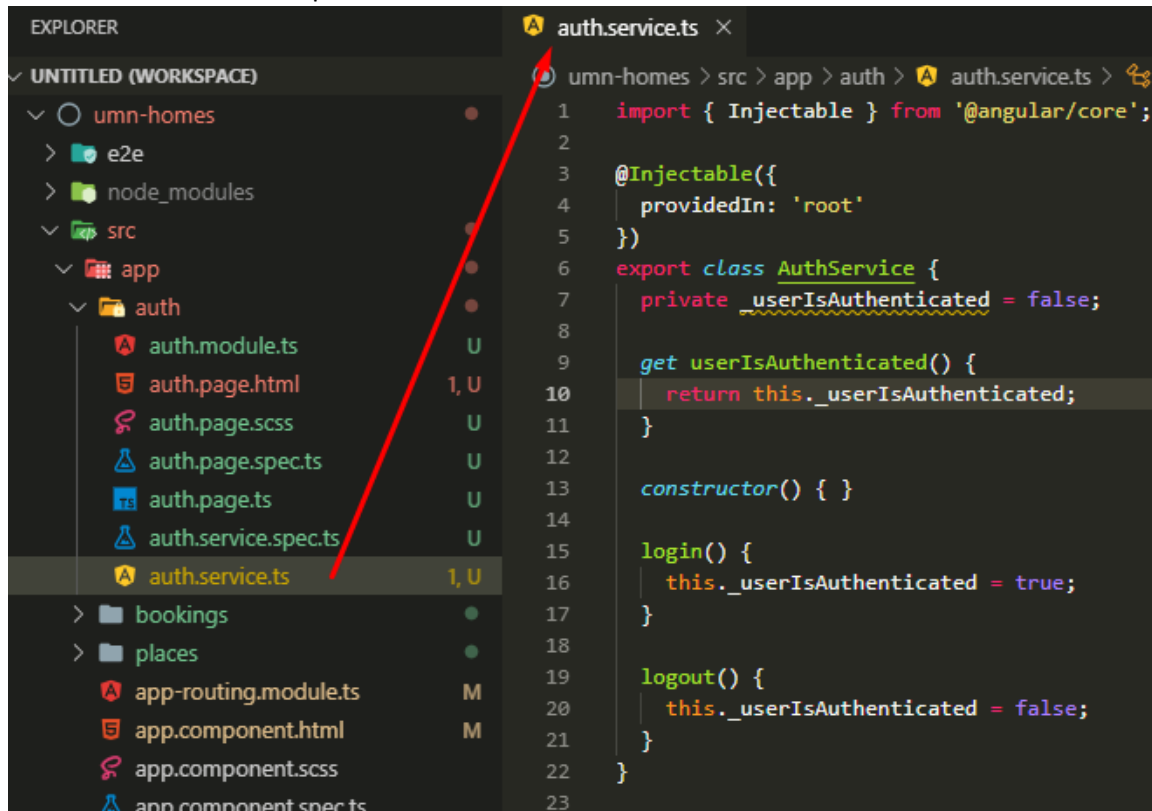
```
1 <ion-header>
2   <ion-toolbar>
3     <ion-title>Authenticate</ion-title>
4   </ion-toolbar>
5 </ion-header>
6
7 <ion-content>
8   <ion-button color="primary" (click)="onLogin()">Login</ion-button>
9 </ion-content>
```

- Buat sebuah auth service dengan mengetikkan command berikut

 Command Prompt

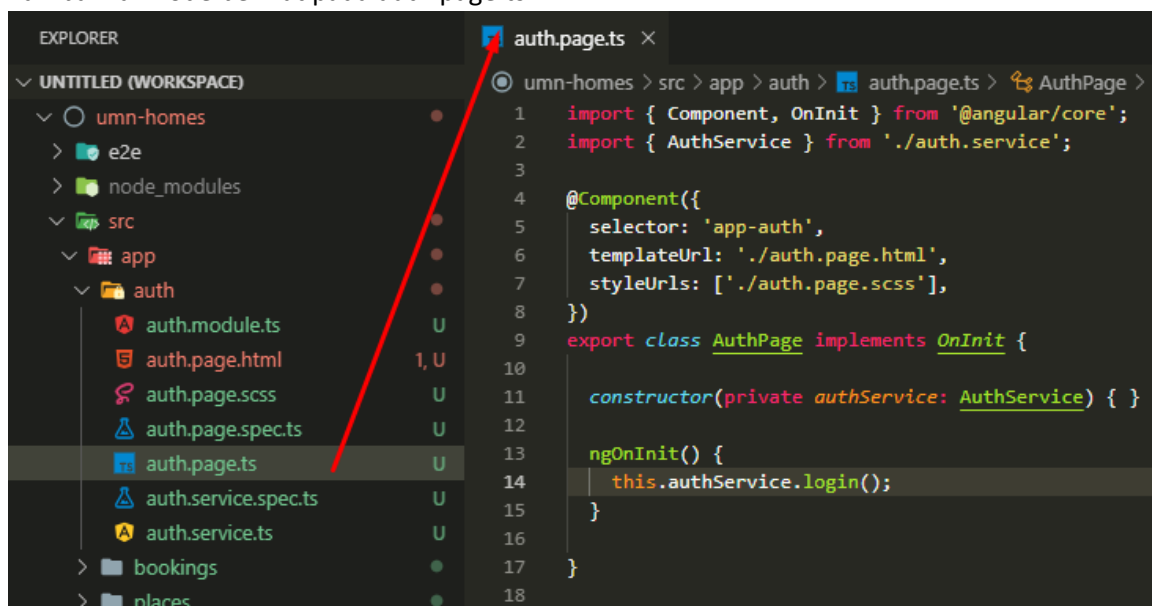
```
D:\ionic-course-umn\umn-homes>ionic generate service auth/auth_
```

3. Tambahkan kode berikut pada auth.service.ts




```
1 import { Injectable } from '@angular/core';
2
3 @Injectable({
4   providedIn: 'root'
5 })
6 export class AuthService {
7   private _userIsAuthenticated = false;
8
9   get userIsAuthenticated() {
10     return this._userIsAuthenticated;
11   }
12
13   constructor() { }
14
15   login() {
16     this._userIsAuthenticated = true;
17   }
18
19   logout() {
20     this._userIsAuthenticated = false;
21   }
22 }
23
```

4. Tambahkan kode berikut pada auth.page.ts



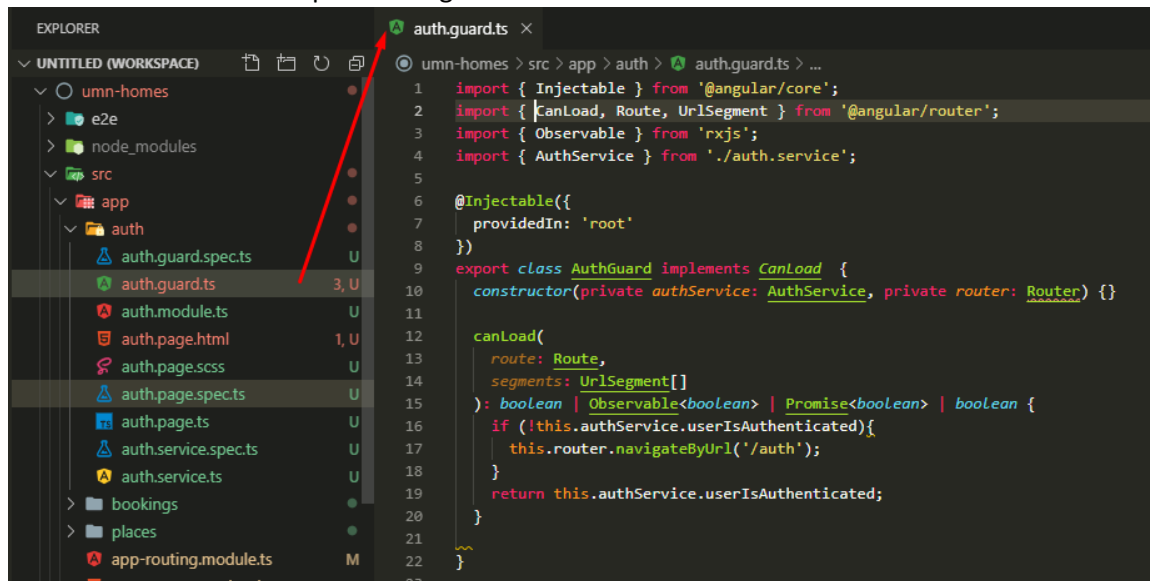
```
1 import { Component, OnInit } from '@angular/core';
2 import { AuthService } from './auth.service';
3
4 @Component({
5   selector: 'app-auth',
6   templateUrl: './auth.page.html',
7   styleUrls: ['./auth.page.scss'],
8 })
9 export class AuthPage implements OnInit {
10
11   constructor(private authService: AuthService) { }
12
13   ngOnInit() {
14     this.authService.login();
15   }
16
17 }
18
```

5. Sampai dengan saat ini, auth page tidak pernah ditampilkan. Kita akan membuat auth guard sehingga nantinya halaman your places dan bookings hanya dapat ditampilkan jika sudah login. Buat guard dengan command berikut

 Command Prompt

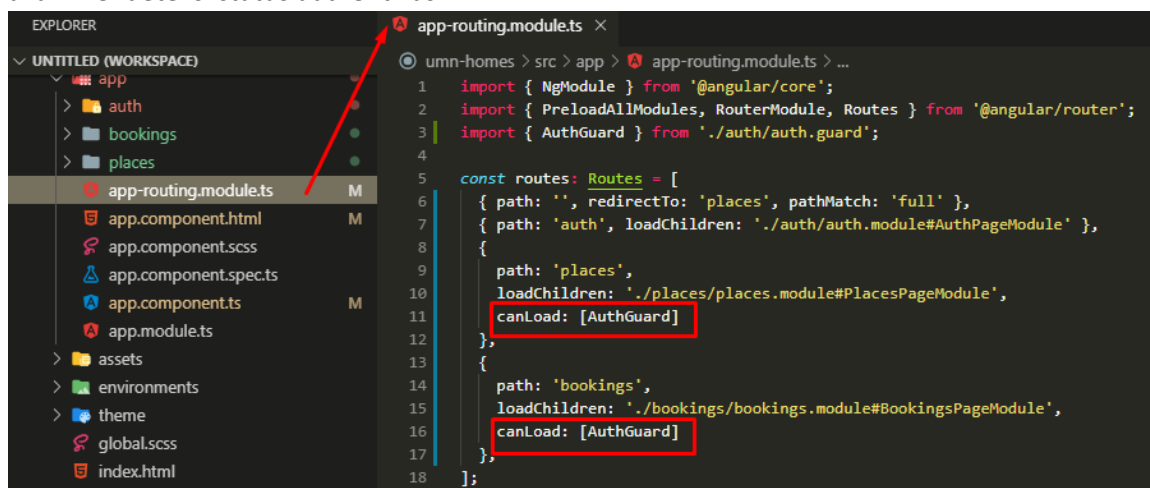
```
D:\ionic-course-umn\umn-homes>ionic generate guard auth/auth
```

6. Tambahkan kode berikut pada auth.guard.ts



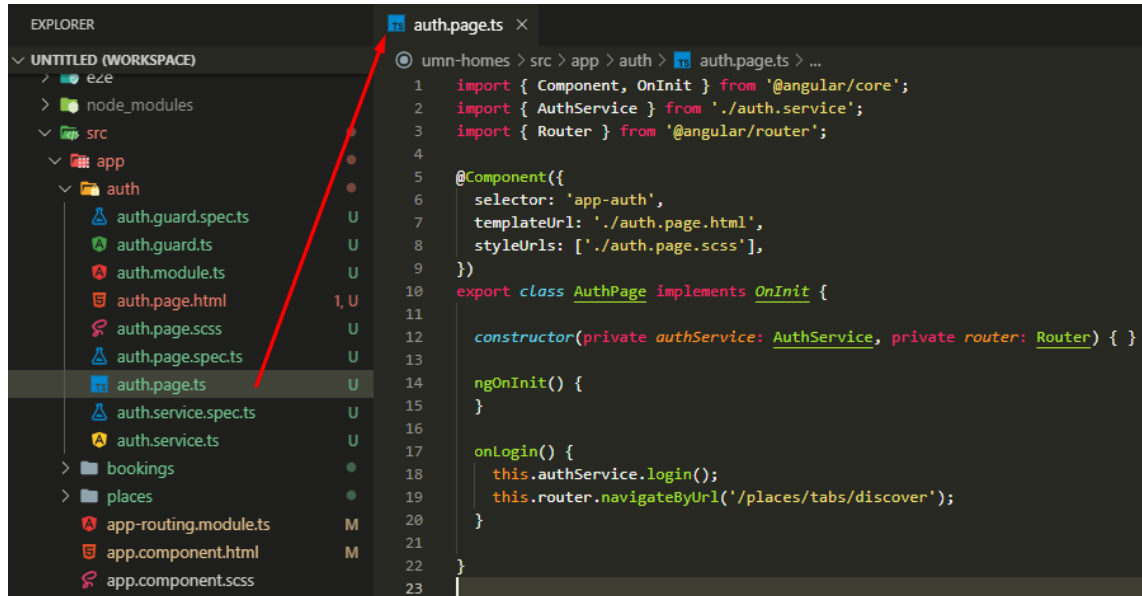
```
1 import { Injectable } from '@angular/core';
2 import { CanLoad, Route, UrlSegment } from '@angular/router';
3 import { Observable } from 'rxjs';
4 import { AuthService } from './auth.service';
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class AuthGuard implements CanLoad {
10   constructor(private authService: AuthService, private router: Router) {}
11
12   canLoad(
13     route: Route,
14     segments: UrlSegment[]
15   ): boolean | Observable<boolean> | Promise<boolean> | boolean {
16     if (!this.authService.isAuthenticated){
17       this.router.navigateByUrl('/auth');
18     }
19     return this.authService.isAuthenticated;
20   }
21 }
22
23
```

7. Tambahkan kode berikut pada app-routing.module.ts sehingga halaman places dan bookings akan mendeteksi status autentikasi.



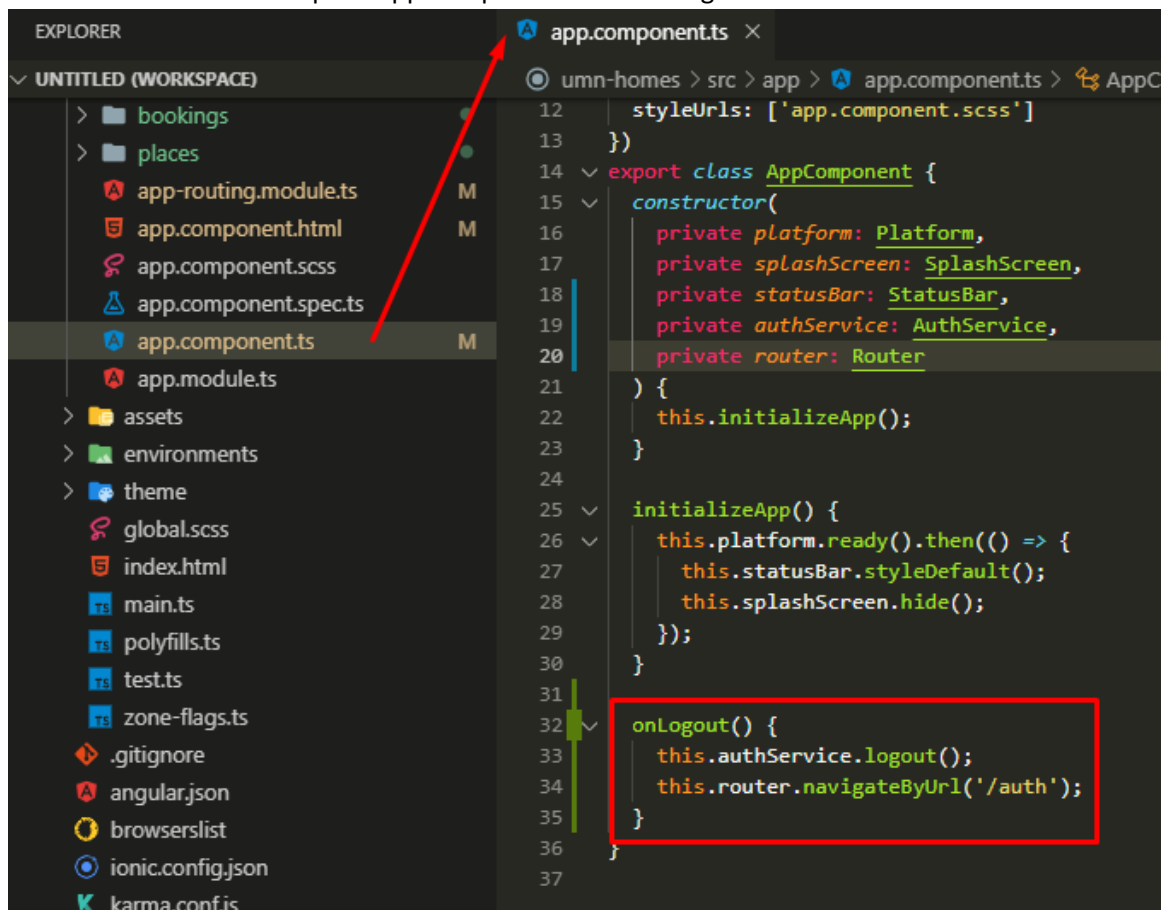
```
1 import { NgModule } from '@angular/core';
2 import { PreloadAllModules, RouterModule, Routes } from '@angular/router';
3 import { AuthGuard } from './auth/auth.guard';
4
5 const routes: Routes = [
6   { path: '', redirectTo: 'places', pathMatch: 'full' },
7   { path: 'auth', loadChildren: './auth/auth.module#AuthPageModule' },
8   {
9     path: 'places',
10    loadChildren: './places/places.module#PlacesPageModule',
11    canLoad: [AuthGuard]
12  },
13   {
14     path: 'bookings',
15    loadChildren: './bookings/bookings.module#BookingsPageModule',
16    canLoad: [AuthGuard]
17  },
18 ];
19
20
```

8. Tambahkan kode berikut pada auth.page.ts



```
1 import { Component, OnInit } from '@angular/core';
2 import { AuthService } from '../auth.service';
3 import { Router } from '@angular/router';
4
5 @Component({
6   selector: 'app-auth',
7   templateUrl: './auth.page.html',
8   styleUrls: ['./auth.page.scss'],
9 })
10 export class AuthPage implements OnInit {
11
12   constructor(private authService: AuthService, private router: Router) {}
13
14   ngOnInit() {}
15
16   onLogin() {
17     this.authService.login();
18     this.router.navigateByUrl('/places/tabs/discover');
19   }
20
21 }
22
23
```

9. Tambahkan kode berikut pada app.component.ts untuk logout



```
12 styleUrls: ['app.component.scss']
13 })
14 export class AppComponent {
15   constructor(
16     private platform: Platform,
17     private splashScreen: SplashScreen,
18     private statusBar: StatusBar,
19     private authService: AuthService,
20     private router: Router
21   ) {
22     this.initializeApp();
23   }
24
25   initializeApp() {
26     this.platform.ready().then(() => {
27       this.statusBar.styleDefault();
28       this.splashScreen.hide();
29     });
30   }
31
32   onLogout() {
33     this.authService.logout();
34     this.router.navigateByUrl('/auth');
35   }
36
37 }
```

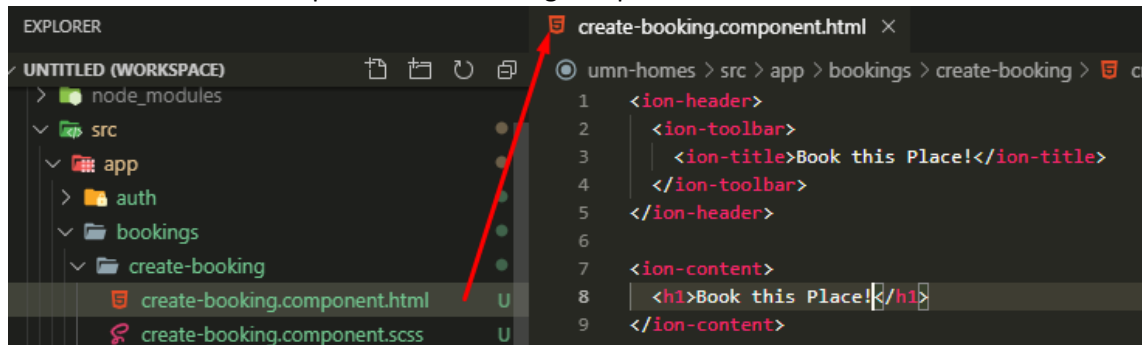
## Opening a Modal

1. Buat component bookings/create-booking dengan command berikut

```
Command Prompt

D:\ionic-course-umn\umn-homes>ionic generate component bookings/create-booking
> ng.cmd generate component bookings/create-booking
CREATE src/app/bookings/create-booking/create-booking.component.html (33 bytes)
CREATE src/app/bookings/create-booking/create-booking.component.spec.ts (776 bytes)
CREATE src/app/bookings/create-booking/create-booking.component.ts (299 bytes)
CREATE src/app/bookings/create-booking/create-booking.component.scss (0 bytes)
[OK] Generated component!
```

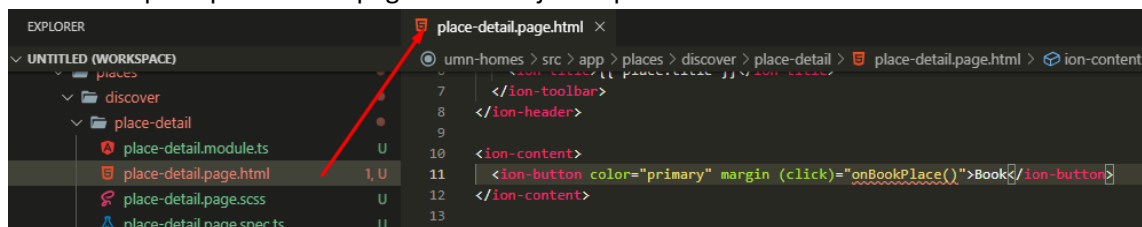
2. Tambahkan kode berikut pada create-booking.component.html



```
EXPLORER
node_modules
src
  app
    auth
    bookings
      create-booking
        create-booking.component.html
        create-booking.component.scss

create-booking.component.html x
1 <ion-header>
2 <ion-toolbar>
3   <ion-title>Book this Place!</ion-title>
4 </ion-toolbar>
5 </ion-header>
6
7 <ion-content>
8   <h1>Book this Place!</h1>
9 </ion-content>
```

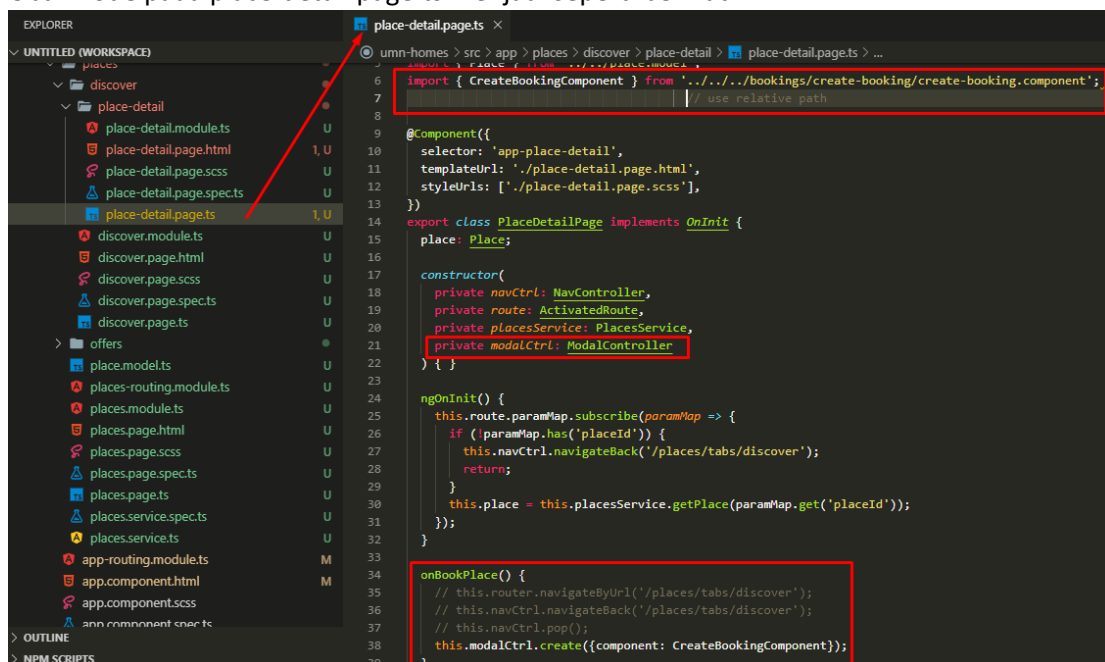
3. Ubah kode pada place-detail.page.html menjadi seperti berikut



```
EXPLORER
node_modules
src
  app
    places
      discover
        place-detail
          place-detail.module.ts
          place-detail.page.html
          place-detail.page.scss
          place-detail.page.spec.ts

place-detail.page.html x
7 </ion-toolbar>
8 </ion-header>
9
10 <ion-content>
11   <ion-button color="primary" margin (click)="onBookPlace()">Book</ion-button>
12 </ion-content>
13
```

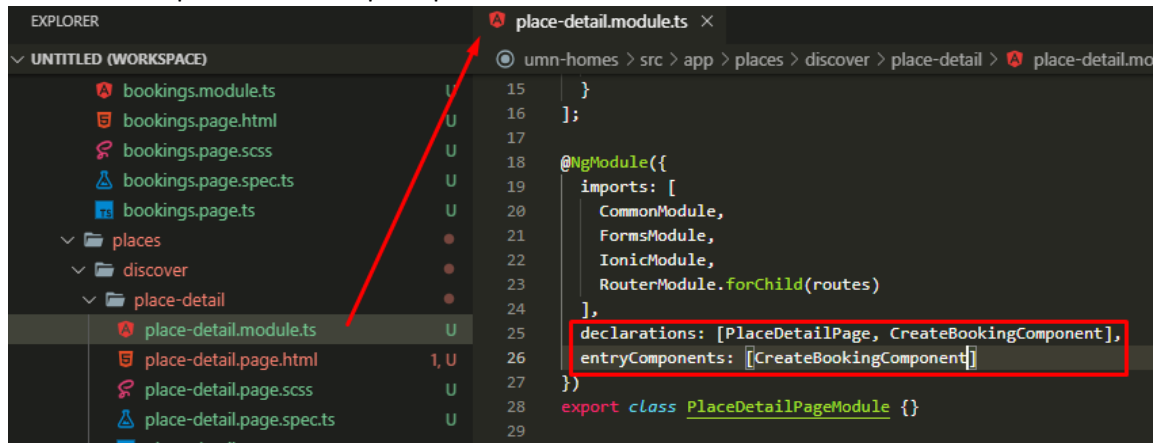
4. Ubah kode pada place-detail.page.ts menjadi seperti berikut



```
EXPLORER
node_modules
src
  app
    places
      discover
        place-detail
          place-detail.module.ts
          place-detail.page.html
          place-detail.page.scss
          place-detail.page.spec.ts
          place-detail.page.ts
        discover.module.ts
        discover.page.html
        discover.page.scss
        discover.page.spec.ts
        discover.page.ts
      offers
        place.model.ts
        places-routing.module.ts
        places.module.ts
        places.page.html
        places.page.scss
        places.page.spec.ts
        places.page.ts
        places.service.spec.ts
        places.service.ts
      app-routing.module.ts
      app.component.html
      app.component.scss
      app.component.spec.ts

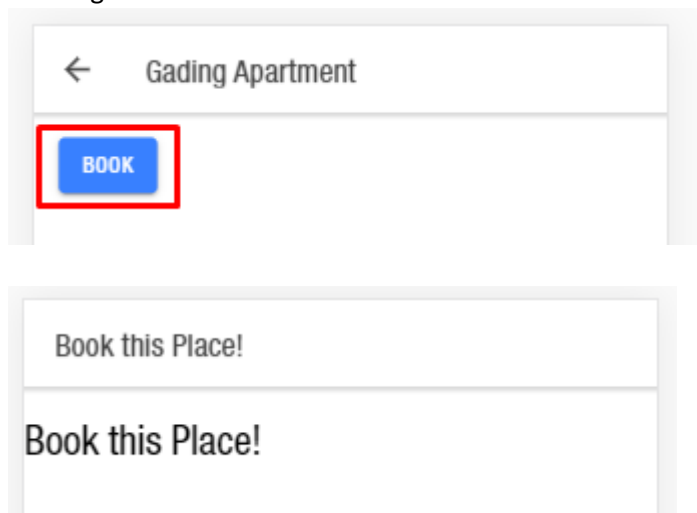
place-detail.page.ts x
6 import { CreateBookingComponent } from '../bookings/create-booking/create-booking.component';
7 // use relative path
8
9 @Component({
10   selector: 'app-place-detail',
11   templateUrl: './place-detail.page.html',
12   styleUrls: ['./place-detail.page.scss'],
13 })
14 export class PlaceDetailPage implements OnInit {
15   place: Place;
16
17   constructor(
18     private navCtrl: NavController,
19     private route: ActivatedRoute,
20     private placesService: PlacesService,
21     private modalCtrl: ModalController
22   ) {}
23
24   ngOnInit() {
25     this.route.paramMap.subscribe(paramMap => {
26       if (!paramMap.has('placeId')) {
27         this.navCtrl.navigateBack('/places/tabs/discover');
28         return;
29       }
30       this.place = this.placesService.getPlace(paramMap.get('placeId'));
31     });
32
33     onBookPlace() {
34       // this.router.navigateByUrl('/places/tabs/discover');
35       // this.navCtrl.navigateBack('/places/tabs/discover');
36       // this.navCtrl.pop();
37       this.modalCtrl.create({component: CreateBookingComponent});
38     }
39 }
```

5. Deklarasi komponen tersebut pada place-detail.module.ts

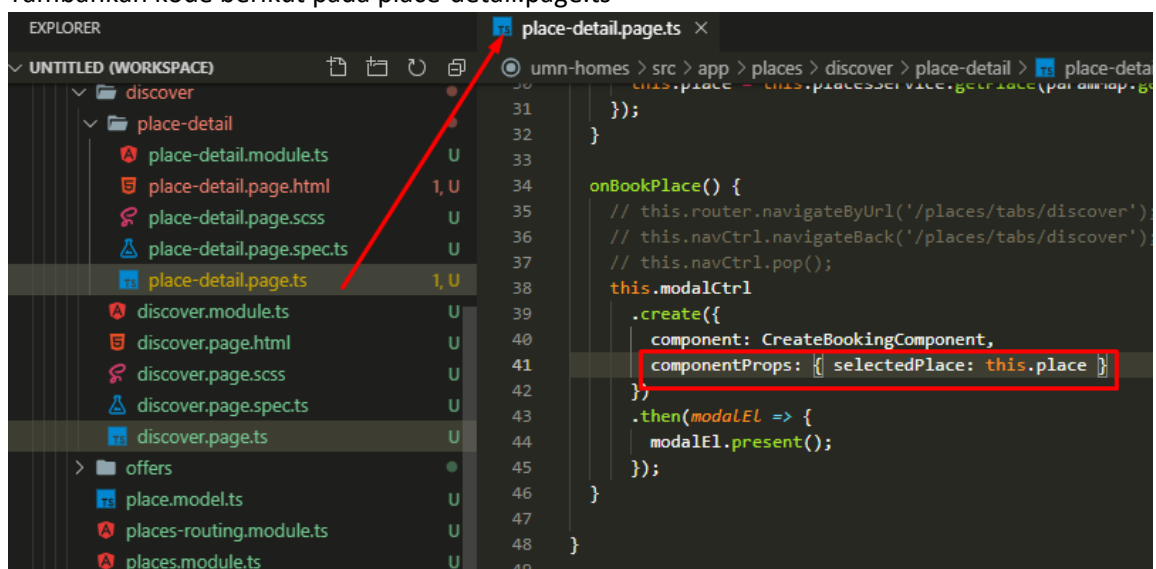


```
15 }
16 };
17
18 @NgModule({
19   imports: [
20     CommonModule,
21     FormsModule,
22     IonicModule,
23     RouterModule.forChild(routes)
24   ],
25   declarations: [PlaceDetailPage, CreateBookingComponent],
26   entryComponents: [CreateBookingComponent]
27 })
28 export class PlaceDetailPageModule {}
```

6. Lihat hasilnya di browser, ketika tombol Book di-click, maka akan tampil halaman modal create booking



7. Tambahkan kode berikut pada place-detail.page.ts



```
30 this.place = this.placeService.getPlace(paramMap.get('id'));
31 });
32 }
33
34 onBookPlace() {
35   // this.router.navigateByUrl('/places/tabs/discover');
36   // this.navCtrl.navigateBack('/places/tabs/discover');
37   // this.navCtrl.pop();
38   this.modalCtrl
39     .create({
40       component: CreateBookingComponent,
41       componentProps: { selectedPlace: this.place }
42     })
43     .then(modalEl => {
44       modalEl.present();
45     });
46 }
47
48 }
```

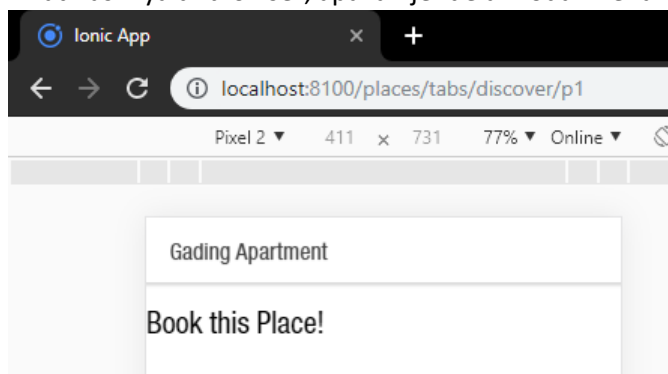
8. Tambahkan kode berikut pada create-booking.components.ts

```
1 import { Component, OnInit, Input } from '@angular/core';
2 import { Place } from 'src/app/places/place.model';
3
4 @Component({
5   selector: 'app-create-booking',
6   templateUrl: './create-booking.component.html',
7   styleUrls: ['./create-booking.component.scss'],
8 })
9 export class CreateBookingComponent implements OnInit {
10   @Input() selectedPlace: Place;
11
12   constructor() { }
13
14   ngOnInit() { }
15
16 }
```

9. Ubah kode pada create-booking.component.html menjadi seperti berikut.

```
1 <ion-header>
2   <ion-toolbar>
3     <ion-title>{{ selectedPlace.title }}</ion-title>
4   </ion-toolbar>
5 </ion-header>
6
7 <ion-content>
8   <h1>Book this Place!</h1>
9 </ion-content>
```

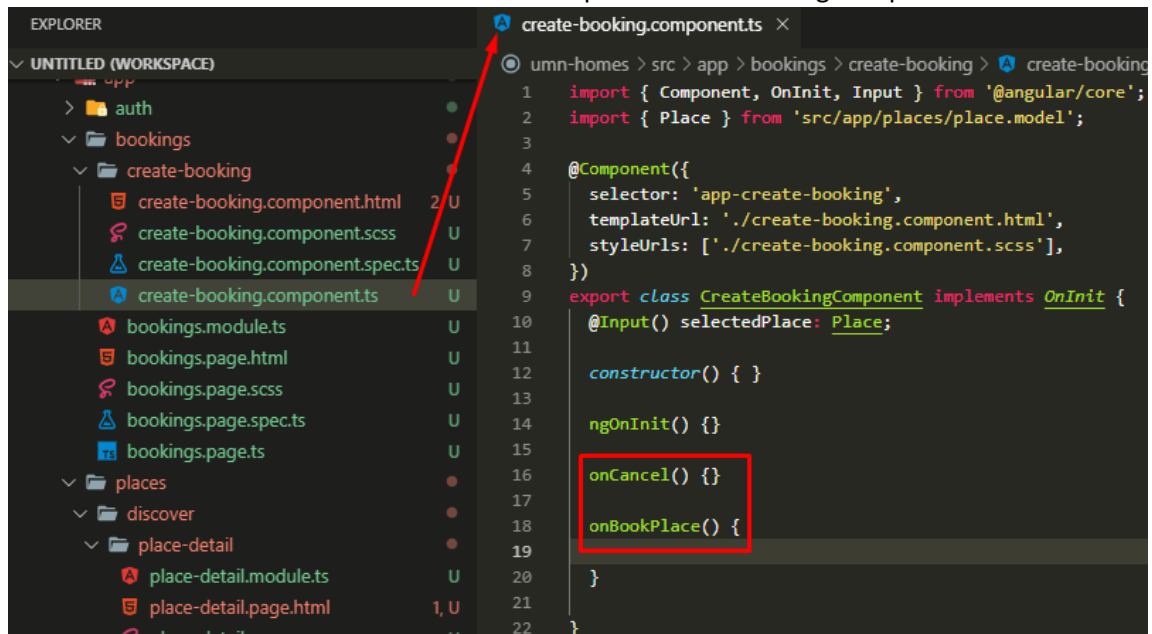
10. Lihat hasilnya di browser, apakah jendela modal menampilkan nama place?



11. Tambahkan kode berikut pada create-booking.component.html

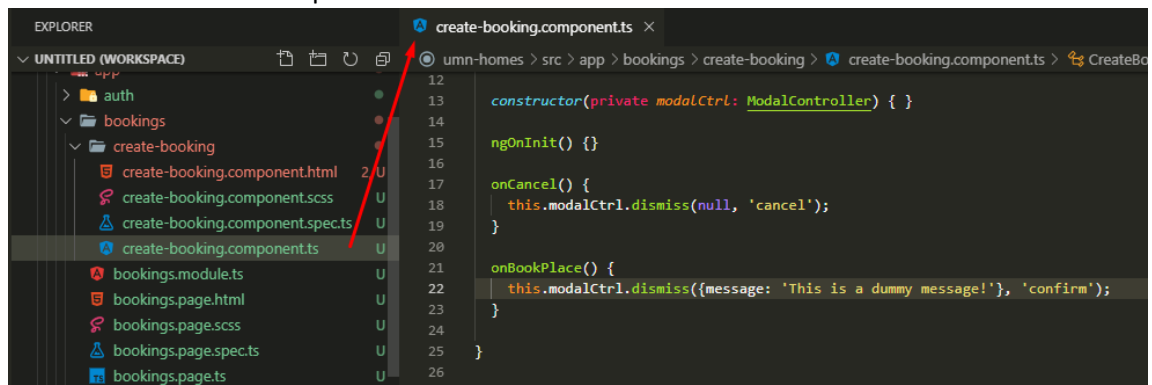
```
1 <ion-header>
2   <ion-toolbar>
3     <ion-title>{{ selectedPlace.title }}</ion-title>
4     <ion-buttons slot="primary">
5       <ion-button (click)="onCancel()">
6         <ion-icon name="close"></ion-icon>
7       </ion-button>
8     </ion-buttons>
9   </ion-toolbar>
10 </ion-header>
11
12 <ion-content text-center padding>
13   <p>{{ selectedPlace.description }}</p>
14   <ion-button color="primary" (click)="onBookPlace()">Book!</ion-button>
15 </ion-content>
```

12. Tambahkan method onCancel dan onBookPlace pada create-booking.component.ts



```
1 import { Component, OnInit, Input } from '@angular/core';
2 import { Place } from 'src/app/places/place.model';
3
4 @Component({
5   selector: 'app-create-booking',
6   templateUrl: './create-booking.component.html',
7   styleUrls: ['./create-booking.component.scss'],
8 })
9 export class CreateBookingComponent implements OnInit {
10   @Input() selectedPlace: Place;
11
12   constructor() { }
13
14   ngOnInit() {}
15
16   onCancel() {}
17
18   onBookPlace() {}
19
20 }
21
22 }
```

13. Tambahkan kode berikut pada kedua method tersebut.

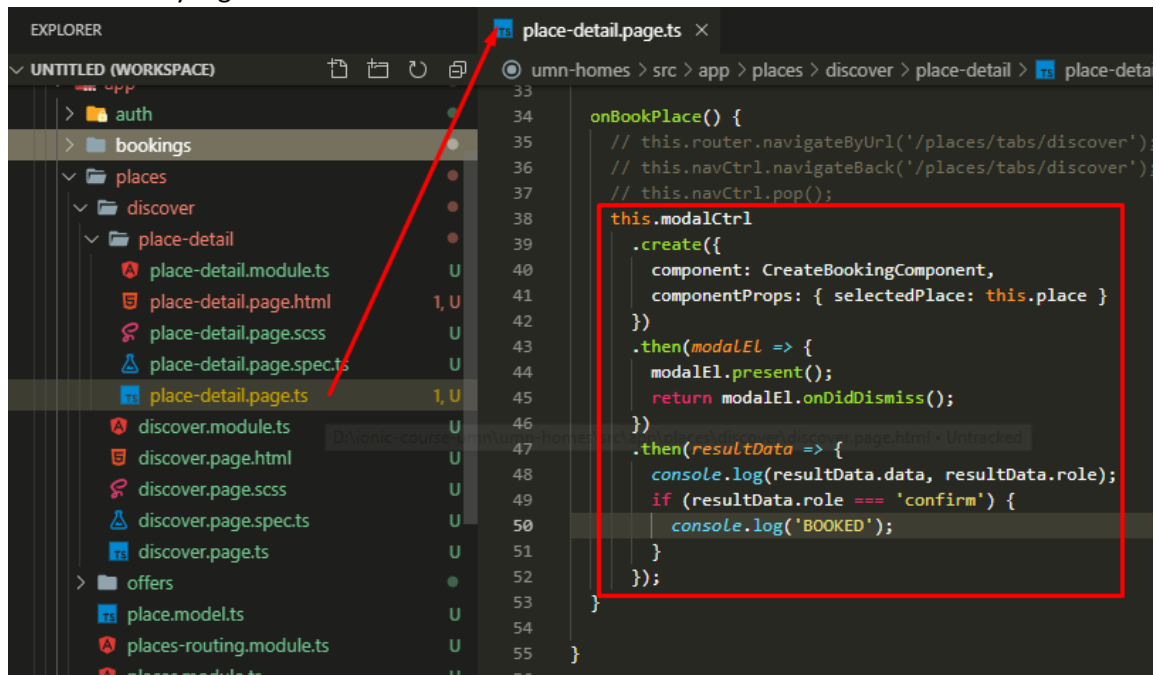


```
12
13 constructor(private modalCtrl: ModalController) { }
14
15 ngOnInit() {}
16
17 onCancel() {
18   this.modalCtrl.dismiss(null, 'cancel');
19 }
20
21 onBookPlace() {
22   this.modalCtrl.dismiss({message: 'This is a dummy message!'}, 'confirm');
23 }
24
25 }
26
```

14. Coba pada browser, apakah halaman modal tertutup ketika tombol close atau Book di-click?



15. Modifikasi kode pada place-detail.page.ts menjadi seperti berikut. Ini untuk mendapatkan data tombol mana yang di click.



```
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

onBookPlace() {
  // this.router.navigateByUrl('/places/tabs/discover');
  // this.navCtrl.navigateBack('/places/tabs/discover');
  // this.navCtrl.pop();
  this.modalCtrl
    .create({
      component: CreateBookingComponent,
      componentProps: { selectedPlace: this.place }
    })
    .then(modalEl => {
      modalEl.present();
      return modalEl.onDidDismiss();
    })
    .then(resultData => {
      console.log(resultData.data, resultData.role);
      if (resultData.role === 'confirm') {
        console.log('BOOKED');
      }
    });
}
```

16. Lihat hasilnya di browser, ketika tombol cancel atau Book di-click, apa yang tampil pada jendela console?

