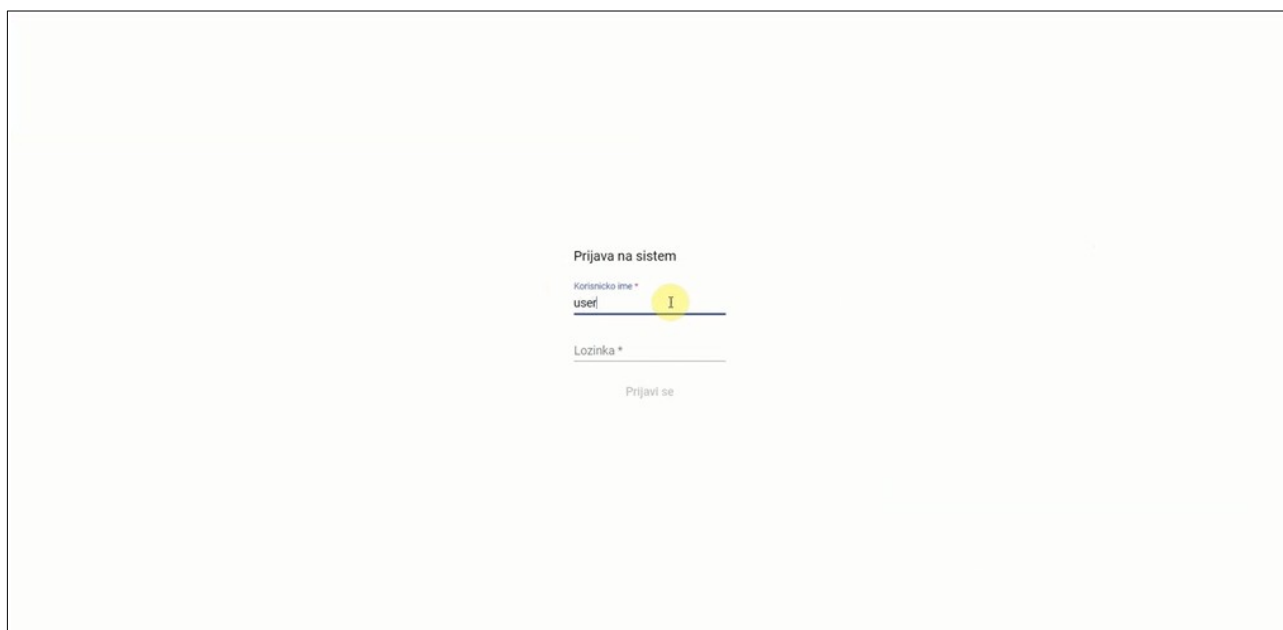
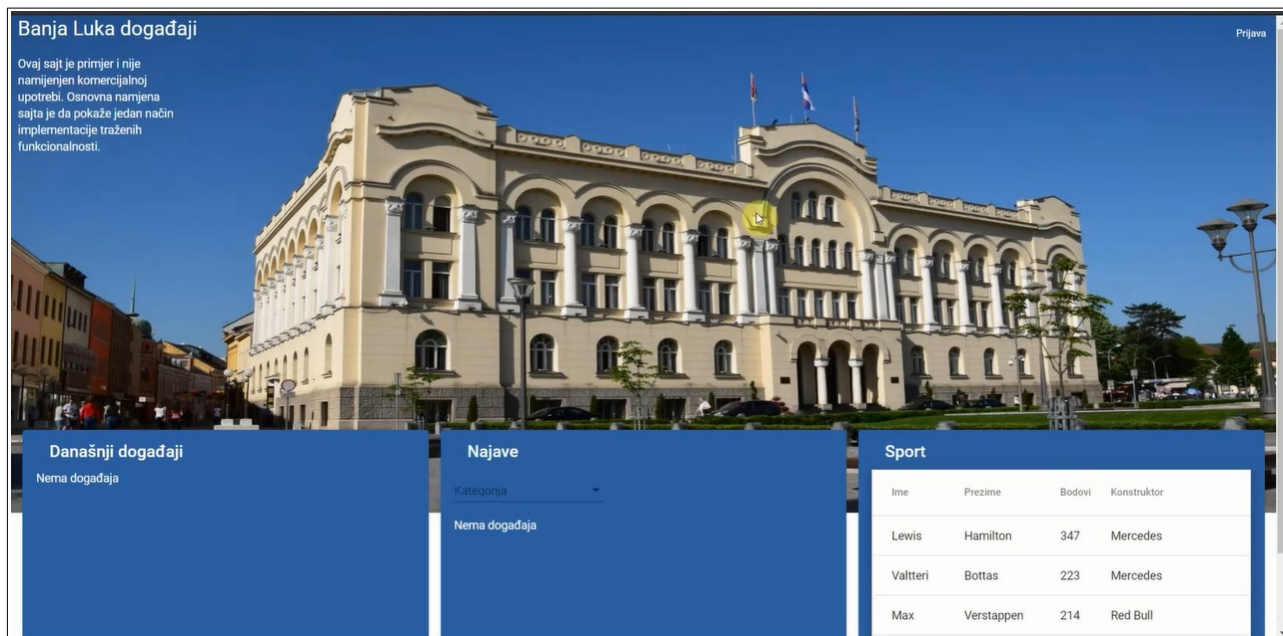
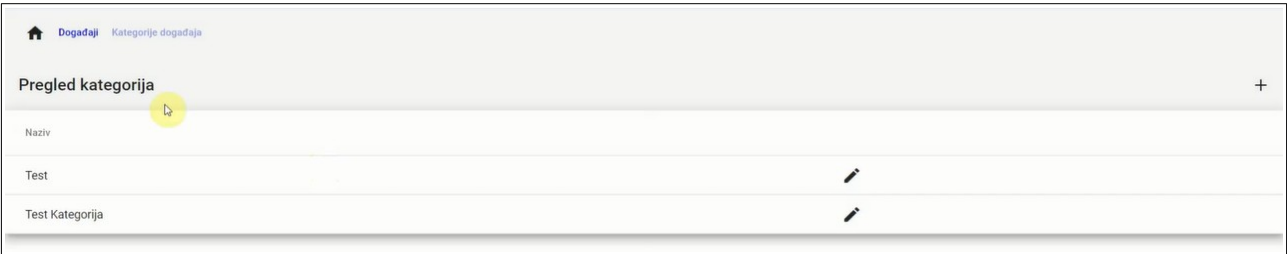
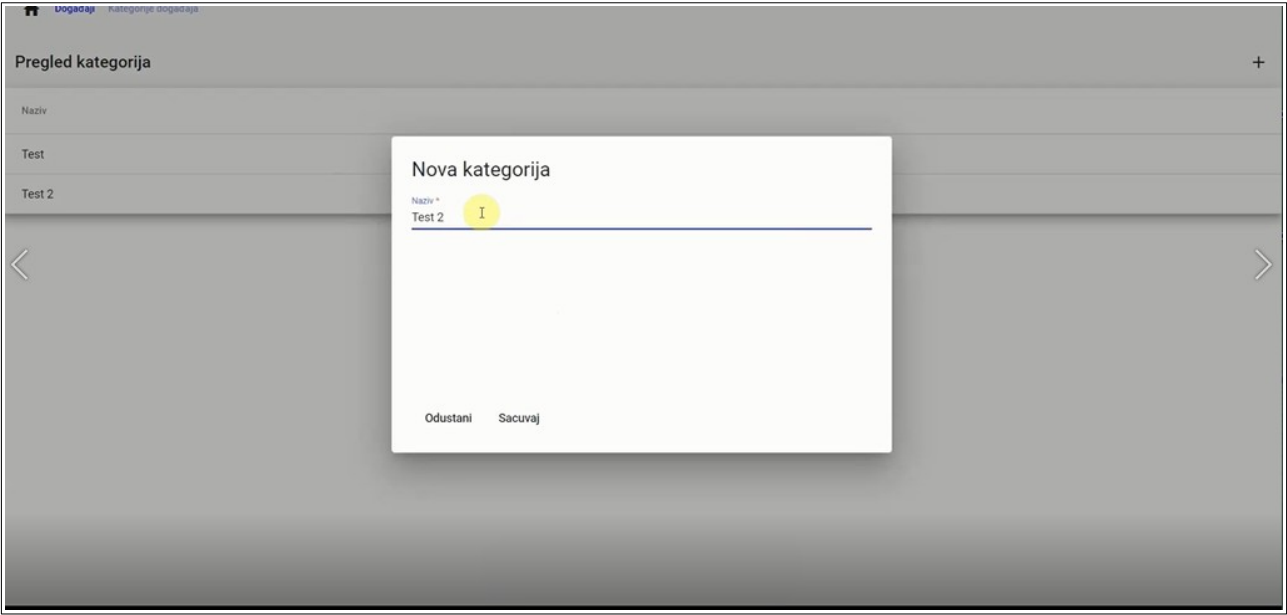
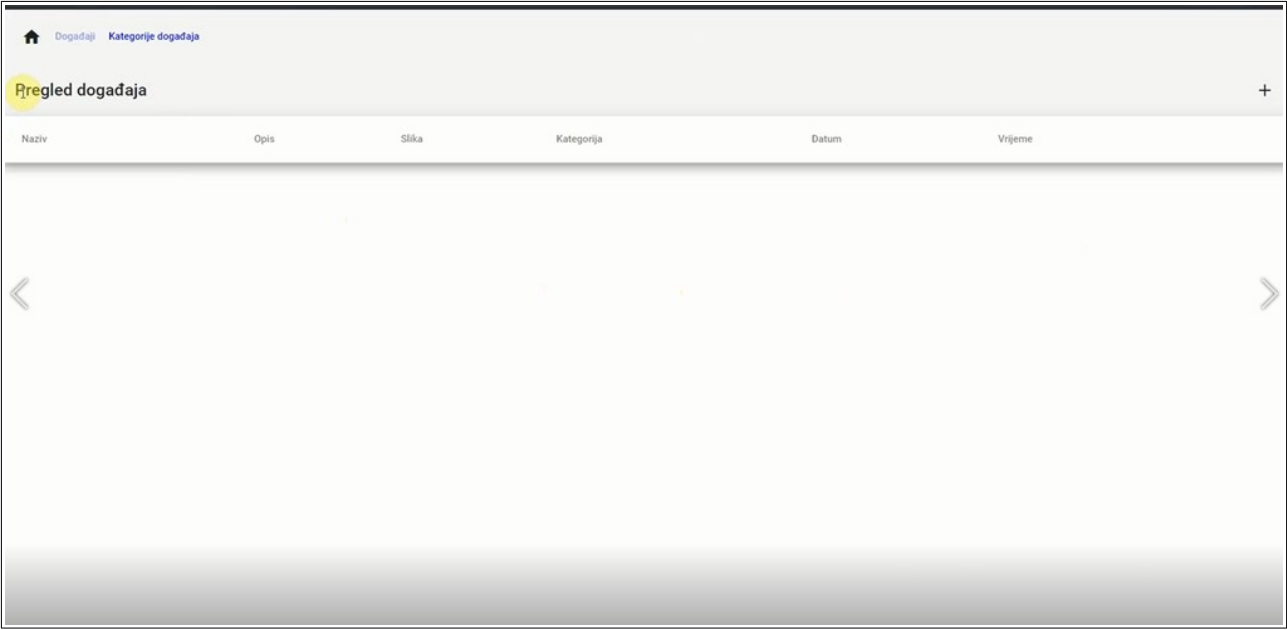


Angular Framework Angular Material Angular Forms

Examples-02





Novi događaj

Naziv *
Test 1

Opis *
Nesto

Slika

Datum
2/10/2021




Vrijeme *
15

Kategorija *
Test

Odustani Sacuvaj

[Događaji](#) [Kategorije događaja](#)

Pregled događaja

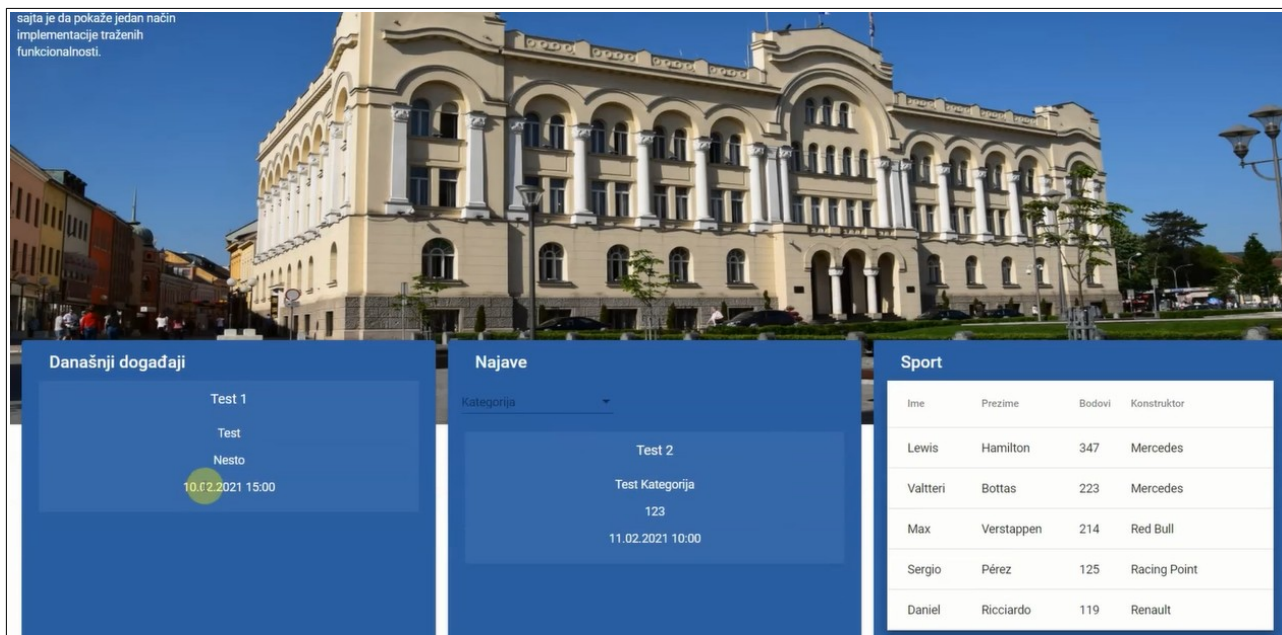
Naziv	Opis	Slika	Kategorija	Datum	Vrijeme	
Test 1	Nesto		Test	10.02.2021	15:00	
Test 2	123		Test Kategorija	11.02.2021	10:00	
asd	asd		Test Kategorija	25.02.2021	asd	

Potvrda

Da li ste sigurni da želite obrisati podatak?

Ne Da

Podaci su sacuvani



npm install @angular/material

```
E:\Enviroment\Y27-Angular\angular-primjer>npm install @angular/material
> core-js@3.8.2 postinstall E:\Enviroment\Y27-Angular\angular-primjer\node_modules\core-js
> node -e "try{require('./postinstall')}catch(e){}"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)

> @angular/cli@11.1.4 postinstall E:\Enviroment\Y27-Angular\angular-primjer\node_modules\@angular\cli
> node ./bin/postinstall/script.js

npm WARN @angular/material@11.2.13 requires a peer of @angular/cdk@11.2.13 but none is installed. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\webpack-dev-server\node_modules\fsevents):npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\watchpack-chokidar2\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.3 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\chokidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ @angular/material@11.2.13
added 522 packages from 316 contributors, removed 234 packages, updated 344 packages and audited 1621 packages in 207.898s

81 packages are looking for funding
  run `npm fund` for details

Found 476 vulnerabilities (70 moderate, 405 high, 1 critical)
  run `npm audit fix` to fix them, or `npm audit` for details
```

ng install @angular/cdk

```
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

^CTerminate batch job (Y/N)? y

E:\Enviroment\Y27-Angular\angular-primjer>npm install @angular/cdk
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\chokidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.3 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\watchpack-chokidar2\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\webpack-dev-server\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ @angular/cdk@11.2.13
removed 1 package, updated 1 package and audited 1621 packages in 22.328s

84 packages are looking for funding
  run `npm fund` for details

found 476 vulnerabilities (70 moderate, 405 high, 1 critical)
  run `npm audit fix` to fix them, or `npm audit` for details

E:\Enviroment\Y27-Angular\angular-primjer>ng serve
Your global Angular CLI version (12.0.5) is greater than your local version (11.1.4). The local Angular CLI version is used.

To disable this warning use "ng config -g cli.warnings.versionMismatch false".
Compiling @angular/cdk/platform : es2015 as esm2015
Compiling @angular/cdk/collections : es2015 as esm2015
Compiling @angular/cdk/bidi : es2015 as esm2015
Compiling @angular/cdk/keycodes : es2015 as esm2015
Compiling @angular/cdk/observers : es2015 as esm2015
Compiling @angular/cdk/portal : es2015 as esm2015
Compiling @angular/cdk/scrolling : es2015 as esm2015
Compiling @angular/cdk/a11y : es2015 as esm2015
Compiling @angular/cdk/layout : es2015 as esm2015
Compiling @angular/cdk/overlay : es2015 as esm2015
Compiling @angular/cdk/accordion : es2015 as esm2015
Compiling @angular/cdk/table : es2015 as esm2015
Compiling @angular/cdk/tree : es2015 as esm2015
Compiling @angular/cdk/text-field : es2015 as esm2015
✓ Browser application bundle generation complete.
```

ng serve

```
E:\Enviroment\Y27-Angular\angular-primjer>ng serve
Your global Angular CLI version (12.0.5) is greater than your local version (11.1.4). The local Angular CLI version is used.

To disable this warning use "ng config -g cli.warnings.versionMismatch false".
✓ Browser application bundle generation complete.

Initial Chunk Files | Names | Size
vendor.js           | vendor | 5.94 MB
polyfills.js        | polyfills | 472.80 kB
styles.css, styles.js | styles | 420.77 kB
main.js             | main | 34.98 kB
runtime.js          | runtime | 9.17 kB

| Initial Total | 6.85 MB

Lazy Chunk Files | Names | Size
event-event-module.js | event-event-module | 42.58 kB
home-home-module.js | home-home-module | 40.99 kB
event-category-event-category-module.js | event-category-event-category-module | 31.23 kB
auth-auth-module.js | auth-auth-module | 13.44 kB
common.js | common | 5.08 kB

Build at: 2021-08-11T12:14:53.574Z - Hash: 0a3cb6d836a20177fb61 - Time: 16588ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

✓ Compiled successfully.
```

Banja Luka događaji

[Prijava](#)

Ovaj sajt je primjer i nije namijenjen komercijalnoj upotrebi. Osnovna namjena sajta je da pokaže jedan način implementacije traženih funkcionalnosti.



Današnji događaji

Nema događaja

Najave

Kategorija

Nema događaja

Sport

Ime	Prezime	Bodovi	Konstruktor
Lewis	Hamilton	347	Mercedes
Valtteri	Bottas	223	Mercedes
Max	Verstappen	214	Red Bull
Sergio	Pérez	125	Racing Point
Daniel	Bicciardi	110	Renault

Activate Windows

Prijava na sistem

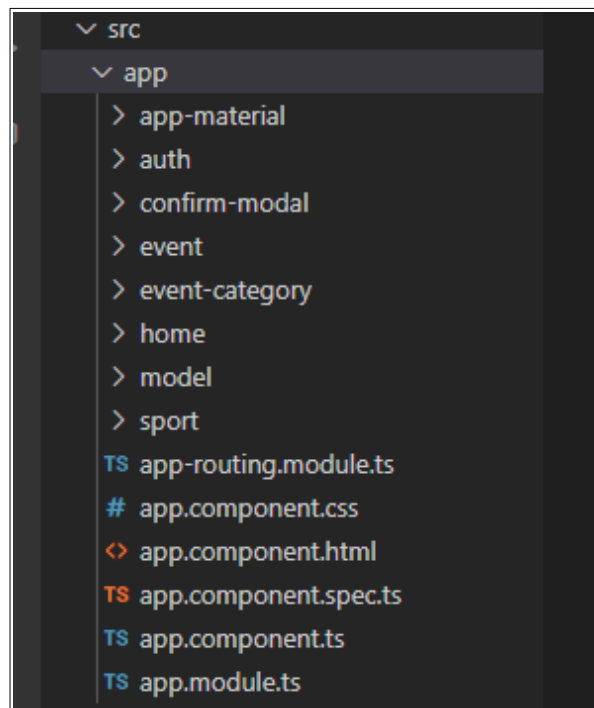
Korisnicko ime *

Lozinka *

[Prijavi se](#)

Activate Windows

- > dist
- > e2e
- > node_modules
- ▼ src
 - > app
 - > assets
 - > environments
 - ★ favicon.ico
 - <> index.html
 - TS main.ts
 - TS polyfills.ts
 - # styles.css
 - TS test.ts
- ≡ .browserslistrc
- ⚙ .editorconfig
- 💎 .gitignore
- { } angular.json
- ⚙ karma.conf.js
- { } package-lock.json
- { } package.json
- ⓘ README.md
- { } tsconfig.app.json
- ts tsconfig.json
- { } tsconfig.spec.json
- { } tslint.json



app.modules.ts

```
1  import { NgModule } from '@angular/core';
2  import { BrowserModule } from '@angular/platform-browser';
3
4  import { AppRoutingModule } from './app-routing.module';
5  import { AppComponent } from './app.component';
6  import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
7  import { ConfirmModalComponent } from './confirm-modal/confirm-modal.component';
8  import { AppMaterialModule } from './app-material/app-material.module';
9  import { HttpClientModule } from '@angular/common/http';
10
11  @NgModule({
12    declarations: [
13      AppComponent,
14      ConfirmModalComponent
15    ],
16    imports: [
17      BrowserModule,
18      AppRoutingModule,
19      AppMaterialModule,
20      BrowserAnimationsModule,
21      HttpClientModule
22    ],
23    providers: [],
24    bootstrap: [AppComponent],
25    entryComponents: [
26      ConfirmModalComponent
27    ]
28  })
29  export class AppModule { }
```



```

1  <!doctype html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <title>AngularPrimjer</title>
6    <base href="/">
7    <meta name="viewport" content="width=device-width, initial-scale=1">
8    <link rel="icon" type="image/x-icon" href="favicon.ico">
9    <link rel="preconnect" href="https://fonts.gstatic.com">
10   <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500&display=swap" rel="styl
11   <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
12 </head>
13 <body class="mat-typography">
14   <app-root></app-root>
15 </body>
16 </html>

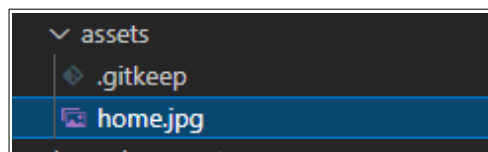
```

style.css

```

3  html, body { height: 100%; }
4  body { margin: 0; font-family: Roboto, "Helvetica Neue", sans-serif;
5
6  .primary {
7    color: rgb(150, 156, 230);
8  }

```



app.component.ts

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = 'Angular-primer';
10 }
11
```

app.component.html

```
1 <router-outlet></router-outlet>
```

app-routing.module.ts

```
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { GuardService } from './auth/services/guard.service';
4
5 const routes: Routes = [
6   {
7     path: 'login',
8     loadChildren: () => import('./auth/auth.module').then(mod => mod.AuthModule)
9   },
10  {
11    path: 'events',
12    loadChildren: () => import('./event/event.module').then(mod => mod.EventModule),
13    canActivate: [GuardService]
14  },
15  {
16    path: 'event-categories',
17    loadChildren: () => import('./event-category/event-category.module').then(mod => mod.EventCategory),
18    canActivate: [GuardService]
19  },
20  {
21    path: '',
22    loadChildren: () => import('./home/home.module').then(mod => mod.HomeModule)
23  },
24  {
25    path: '**',
26    redirectTo: '/',
27    pathMatch: 'full'
28  }
29 ];
30 @NgModule({
31   imports: [RouterModule.forRoot(routes)],
32   exports: [RouterModule]
33 })
34 export class AppRoutingModule { }
35
```

styles.css

```
1  /* You can add global styles to this file, and also import other style files */
2
3  html, body { height: 100%; }
4  body { margin: 0; font-family: Roboto, "Helvetica Neue", sans-serif; }
5
6  .primary {
7    color: rgb(150, 156, 230);
8  }
```

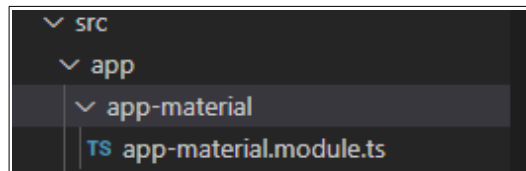
app.module.css

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

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { ConfirmModalComponent } from './confirm-modal/confirm-modal.component';
import { AppMaterialModule } from './app-material/app-material.module';
import { HttpClientModule } from '@angular/common/http';

@NgModule({
  declarations: [
    AppComponent,
    ConfirmModalComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    AppMaterialModule,
    BrowserAnimationsModule,
    HttpClientModule
  ],
  providers: [],
  bootstrap: [AppComponent],
  entryComponents: [
    ConfirmModalComponent
  ]
})
export class AppModule { }
```

Моделни дијалози

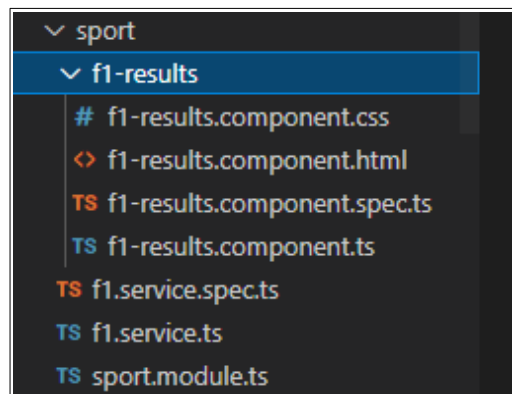


app-material-module.ts

```
import { NgModule } from '@angular/core';
import { OverlayModule } from '@angular/cdk/overlay';
import { CdkTreeModule } from '@angular/cdk/tree';
import { PortalModule } from '@angular/cdk/portal';
import { MatAutocompleteModule } from '@angular/material/autocomplete';
import { MatButtonModule } from '@angular/material/button';
import { MatButtonModuleToggleModule } from '@angular/material/button-toggle';
import { MatCardModule } from '@angular/material/card';
import { MatCheckboxModule } from '@angular/material/checkbox';
import { MatChipsModule } from '@angular/material/chips';
import { MatNativeDateModule, MatRippleModule } from '@angular/material/core';
import { MatDividerModule } from '@angular/material/divider';
import { MatExpansionModule } from '@angular/material/expansion';
import { MatFormFieldModule } from '@angular/material/form-field';
import { MatIconModule } from '@angular/material/icon';
import { MatInputModule } from '@angular/material/input';
import { MatListModule } from '@angular/material/list';
import { MatMenuModule } from '@angular/material/menu';
import { MatPaginatorModule } from '@angular/material/paginator';
import { MatProgressSpinnerModule } from '@angular/material/progress-spinner';
import { MatSelectModule } from '@angular/material/select';
import { MatSidenavModule } from '@angular/material/sidenav';
import { MatSnackBarModule } from '@angular/material/snack-bar';
import { MatSortModule } from '@angular/material/sort';
import { MatTableModule } from '@angular/material/table';
import { MatTabsModule } from '@angular/material/tabs';
import { MatToolbarModule } from '@angular/material/toolbar';
import { MatTreeModule } from '@angular/material/tree';
import { MatDatepickerModule } from '@angular/material/datepicker';
```

```
const materialModules = [  
  CdkTreeModule,  
  MatAutocompleteModule,  
  MatButtonModule,  
  MatCardModule,  
  MatCheckboxModule,  
  MatChipsModule,  
  MatDividerModule,  
  MatExpansionModule,  
  MatIconModule,  
  MatInputModule,  
  MatListModule,  
  MatMenuModule,  
  MatProgressSpinnerModule,  
  MatPaginatorModule,  
  MatRippleModule,  
  MatSelectModule,  
  MatSidenavModule,  
  MatSnackBarModule,  
  MatSortModule,  
  MatTableModule,  
  MatTabsModule,  
  MatToolbarModule,  
  MatFormFieldModule,  
  MatButtonModuleToggleModule,  
  MatTreeModule,  
  OverlayModule,  
  PortalModule,  
  MatDatepickerModule,  
  MatNativeDateModule  
];
```

```
@NgModule({  
  imports: [  
    ...materialModules  
  ],  
  exports: [  
    ...materialModules  
  ],  
})  
export class AppMaterialModule {  
}
```



f1.service.ts

```
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class F1Service {
8
9   constructor(private http: HttpClient) { }
10
11   public getDataForSeason(season: string) {
12     return this.http.get("http://ergast.com/api/f1/" + season + "/driverStandings.json")
13   }
14 }
```

f1-results.component.ts

```
import { Component, EventEmitter, Input, OnChanges, OnInit, Output, SimpleChanges } from '@angular/core';
import { FormulaDriver } from 'src/app/model/formula-driver.model';
import { Team } from 'src/app/model/team.model';
import { F1Service } from '../f1.service';

@Component({
  selector: 'app-f1-results',
  templateUrl: './f1-results.component.html',
  styleUrls: ['./f1-results.component.css']
})
export class F1ResultsComponent implements OnInit, OnChanges {

  @Input("season") public season: string = "2020";
  @Output("onDataLoad") public onDataLoad: EventEmitter<any> = new EventEmitter()

  public dataSource = new Array<FormulaDriver>();
  public displayedColumns: string[] = ['firstName', 'lastName', 'points', 'constructorName'];

  constructor(private service: F1Service) { }

  ngOnChanges(changes: SimpleChanges): void {
    if(changes.season) {
      this.showData();
    }
  }

  ngOnInit(): void {
  }
}
```


f1-result.component.html

```

1 <table mat-table [dataSource]="dataSource" class="mat-elevation-z8">
2   <ng-container matColumnDef="firstName">
3     <th mat-header-cell *matHeaderCellDef> Ime </th>
4     <td mat-cell *matCellDef="let element"> {{element.firstName}} </td>
5   </ng-container>
6
7   <ng-container matColumnDef="lastName">
8     <th mat-header-cell *matHeaderCellDef> Prezime </th>
9     <td mat-cell *matCellDef="let element"> {{element.lastName}} </td>
10  </ng-container>
11
12  <ng-container matColumnDef="points">
13    <th mat-header-cell *matHeaderCellDef> Bodovi </th>
14    <td mat-cell *matCellDef="let element"> {{element.points}} </td>
15  </ng-container>
16
17  <ng-container matColumnDef="constructorName">
18    <th mat-header-cell *matHeaderCellDef> Konstruktor </th>
19    <td mat-cell *matCellDef="let element"> {{element.team?.name}} </td>
20  </ng-container>
21
22  <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
23  <tr mat-row *matRowDef="let row; columns: displayedColumns;"></tr>
24 </table>

```

f1-result.components.css

```

.selection-container {
  margin: 24px;
}

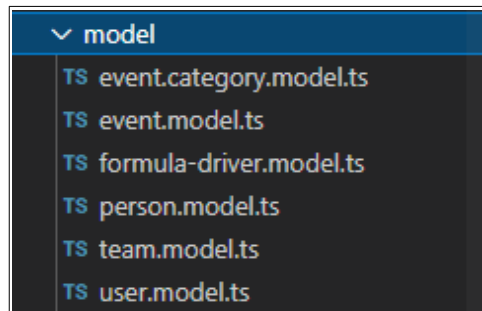
table {
  width: 100%;
}

```

sport.module.ts

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { F1ResultsComponent } from '../f1-results/f1-results.component';
import { AppMaterialModule } from '../app-material/app-material.module';

@NgModule({
  declarations: [F1ResultsComponent],
  imports: [
    CommonModule,
    AppMaterialModule
  ],
  exports: [
    F1ResultsComponent
  ]
})
export class SportModule { }
```



Модели за објекте и класе који се користе за баратање доменким подацима у апликацији

event.category.model.ts

```
export class EventCategory {
  id: number | null;
  name: string | null;

  constructor(id?: number, name?: string) {
    this.id = id || null;
    this.name = name || null;
  }
}
```

event.model.ts

```
export class Event {
  id: number | null;
  name: string | null;
  description: string | null;
  image: string | null;
  date: Date | null;
  time: string | null;
  category: EventCategory | null;

  constructor(id?: number, name?: string, description?: string, image?: string,
    date?: Date, time?: string, category?: EventCategory) {
    this.id = id || null;
    this.name = name || null;
    this.description = description || null;
    this.image = image || null;
    this.date = date || null;
    this.time = time || null;
    this.category = category || null;
  }
}
```

formula-driver.model.ts

```
1  import { Person } from './person.model';
2  import { Team } from './team.model';
3
4  export class FormulaDriver extends Person {
5      points: number;
6      team: Team;
7
8      constructor(firstName: string, lastName: string, points: number, team: Team) {
9          super(firstName, lastName);
10         this.points = points;
11         this.team = team;
12     }
13 }
```

person.model.ts

```
1  export class Person {
2      firstName: string;
3      lastName: string;
4
5      constructor(firstName: string,
6          lastName: string) {
7          this.firstName = firstName;
8          this.lastName = lastName;
9      }
10
11     getFullName(): string {
12         return this.firstName + ' ' + this.lastName;
13     }
14 }
```

team.model.ts

```
1  export class Team {
2      name: string;
3      nationality: string;
4
5      constructor(name: string, nationality: string) {
6          this.name = name;
7          this.nationality = nationality;
8      }
9  }
```

user.model.ts

```
1  export class User {  
2      firstName: string;  
3      lastName: string;  
4      username: string;  
5      password: string;  
6  
7      constructor(firstName: string, lastName: string, username: string, password: string) {  
8          this.firstName = firstName;  
9          this.lastName = lastName;  
10         this.username = username;  
11         this.password = password;  
12     }  
13 }
```

- ▼ auth
 - ▼ auth
 - # auth.component.css
 - <> auth.component.html
 - TS auth.component.spec.ts
 - TS auth.component.ts
 - ▼ login
 - # login.component.css
 - <> login.component.html
 - TS login.component.spec.ts
 - TS login.component.ts
 - ▼ services
 - TS guard.service.spec.ts
 - TS guard.service.ts
 - TS login.service.spec.ts
 - TS login.service.ts
 - TS auth-routing.module.ts
 - TS auth.module.ts

Модел за аутентикацију

auth.module.ts

```
1  import { NgModule } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3
4  import { AuthRoutingModule } from '../auth-routing.module';
5  import { LoginComponent } from '../login/login.component';
6  import { AuthComponent } from '../auth/auth.component';
7  import { FormsModule, ReactiveFormsModule } from '@angular/forms';
8  import { AppMaterialModule } from '../app-material/app-material.module';
9
10
11  @NgModule({
12    declarations: [LoginComponent, AuthComponent],
13    imports: [
14      CommonModule,
15      AuthRoutingModule,
16      FormsModule,
17      ReactiveFormsModule,
18      AppMaterialModule,
19    ]
20  })
21  export class AuthModule { }
22
```

auth-routing.module.ts

```
1  import { NgModule } from '@angular/core';
2  import { Routes, RouterModule } from '@angular/router';
3  import { AuthComponent } from '../auth/auth.component';
4  import { LoginComponent } from '../login/login.component';
5
6
7  const routes: Routes = [{
8    path: '',
9    component: AuthComponent,
10   children: [
11     {
12       path: '',
13       component: LoginComponent,
14     }
15   ]
16 }];
17
18
19  @NgModule({
20    imports: [RouterModule.forChild(routes)],
21    exports: [RouterModule]
22  })
23  export class AuthRoutingModule { }
24
```

guard.service.ts

```
1  import { Injectable } from '@angular/core';
2  import { CanActivate, Router } from '@angular/router';
3  import { LoginService } from '../login.service';
4
5  @Injectable({
6    providedIn: 'root'
7  })
8  export class GuardService implements CanActivate {
9
10     constructor(private loginService: LoginService,
11                 private router: Router) { }
12
13     canActivate(): boolean {
14         if (this.loginService.signIn) {
15             return true;
16         } else {
17             this.router.navigate(['']);
18             return false;
19         }
20     }
21 }
```



```
1 import { Injectable } from '@angular/core';
2 import { User } from '../../model/user.model';
3 import { Router } from '@angular/router';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class LoginService {
9
10   private users: Array<User> = [];
11   public signedIn: boolean = false;
12   public activeUser: User|null = null;
13
14   constructor(private router: Router) {
15     //testni podaci
16     this.users.push(new User("Ime 1", "Prezime 1", "user1", "pass1"));
17     this.users.push(new User("Ime 2", "Prezime 2", "user2", "pass2"));
18     this.users.push(new User("Ime 3", "Prezime 3", "user3", "pass3"));
19   }
20
21   public login(username: string, password: string): boolean {
22     let result = false;
23     this.activeUser = null;
24     this.users.forEach(user => {
25       if (user.username == username && user.password == password) {
26         result = true;
27         this.activeUser = user;
28       }
29     });
30     this.signedIn = result;
31     return result;
32   }
33
34   public logout(){
35     this.activeUser = null;
36     this.signedIn = false;
37     this.router.navigate(['/']);
38   }
39 }
40
```

login.component.ts

```
1 import { Component, OnInit } from '@angular/core';
2 import { FormGroup, FormBuilder, Validators } from '@angular/forms';
3 import { LoginService } from '../services/login.service';
4 import { Router } from '@angular/router';
5 import { MatSnackBar } from '@angular/material/snack-bar';
6 import { User } from 'src/app/model/user.model';
7
8 @Component({
9   selector: 'app-login',
10  templateUrl: './login.component.html',
11  styleUrls: ['./login.component.css']
12 })
13 export class LoginComponent implements OnInit {
14
15   public form: FormGroup = new FormGroup({});
16
17   constructor(private formBuilder: FormBuilder,
18     private loginService: LoginService,
19     private snackBar: MatSnackBar,
20     private router: Router
21   ) { }
22
23   ngOnInit() {
24     this.form = this.formBuilder.group({
25       username: [null, Validators.required],
26       password: [null, Validators.required]
27     });
28   }
29
30   public login(form: any) {
31     if (this.loginService.login(form.value.username, form.value.password)) {
32       this.router.navigate(['events']);
33     } else {
34       this.snackBar.open("Podaci nisu dobri", undefined, {
35         duration: 2000
36       })
37     }
38   }
39
40 }
```

login.component.html


```
1 <div class="login-container">
2   <form [formGroup]="form" class="login-form">
3     <h3>Prijava na sistem</h3>
4     <mat-form-field>
5       <input matInput type="text" formControlName="username" required autocomplete="off"
6         placeholder="Korisnicko ime" />
7     </mat-form-field>
8     <mat-form-field>
9       <input matInput type="password" formControlName="password" required autocomplete="off"
10        placeholder="Lozinka" />
11     </mat-form-field>
12     <button mat-button [disabled]="form.invalid" (click)="login(form)">Prijava se</button>
13   </form>
14 </div>
```

auth.component.ts

```
1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-auth',
5   templateUrl: './auth.component.html',
6   styleUrls: ['./auth.component.css']
7 })
8 export class AuthComponent implements OnInit {
9
10   constructor() { }
11
12   ngOnInit() {
13   }
14
15 }
16
```

auth.component.html


```
<router-outlet></router-outlet>
```

[Događaji](#) [Kategorije događaja](#)

Pregled događaja

+

Naziv	Opis	Slika	Kategorija	Datum	Vrijeme

[Događaji](#) [Kategorije događaja](#)

Pregled kategorija

+

Naziv

Prijava na sistem

Korisnicko ime *

Lozinka *

Prijavi se

Podaci nisu dobri

Novi događaj

Naziv *

Opis *

Slika

Datum



Vrijeme *

Kategorija *



Odustani

Sacuvaj

Nova kategorija

Naziv *

Odustani

Sacuvaj



Pregled kategorija



Naziv

Спорт



Култура



Маинфестација



Novi događaj

Naziv *

Мали фудбал 2021

Opis *

Одржавање турнира у малом фудбалу.

Slika

Datum

8/28/2021



Vrijeme *

11:00

Kategorija *

Спорт



Odustani

Sacuvaj

Спорт

Култура

Маинфестација

<div> Događaji Kategorije događaja </div>						
Pregled događaja						+
Naziv ↓	Opis	Slika	Kategorija	Datum	Vrijeme	
Мали фудбал 2021	Одржавање турнира у малом фудбалу.		Спорт	28.08.2021	11:00	
Стони тенис 2021	Одржавање турнира у стоном тенису.		Спорт	11.08.2021	18:00	
Рибља Чорба ВЕКОВНА	Концерт рок групе у Бањој Луци 2021		Маинфестација	01.09.2021	20:00	
Дани Петра Кочића	Културна манифестација поводом обиљежавања дана сјећања на рад и живот нашег познатог писца.		Култура	12.09.2021	9:00	

Potvrda

Da li ste sigurni da želite obrisati podatak?

Ne

Da

Današnji događaji

Стони тенис 2021

Спорт

Одржавање турнира у стоном тенису.

11.08.2021 18:00

Najave

Kategorija ▾

Мали фудбал 2021

Спорт

Одржавање турнира у малом фудбалу.

28.08.2021 11:00

Рибља Чорба ВЕКОВНА

Маинфестација

Концерт рок групе у Бањој Луци 2021

01.09.2021 20:00

Дани Петра Кочића

Култура

Културна манифестација поводом обиљежавања дана сјећања на рад и живот нашег познатог писца.

12.09.2021 9:00

Sport			
Ime	Prezime	Bodovi	Konstruktor
Lewis	Hamilton	347	Mercedes
Valtteri	Bottas	223	Mercedes
Max	Verstappen	214	Red Bull
Sergio	Pérez	125	Racing Point
Daniel	Ricciardo	119	Renault

```

v confirm-modal
# confirm-modal.component.css
<> confirm-modal.component.html
TS confirm-modal.component.spec.ts
TS confirm-modal.component.ts

```

confirm-modal.component.html

```

1 <h1 mat-dialog-title>Potvrda</h1>
2 <div mat-dialog-content>
3   Da li ste sigurni da želite obrisati podatak?
4 </div>
5 <div mat-dialog-actions>
6   <button mat-button (click)="close()">Ne</button>
7   <button mat-button (click)="confirm()">Da</button>
8 </div>

```

confirm-modal.component.ts

```
1  import { async, ComponentFixture, TestBed } from '@angular/core/testing';
2
3  import { ConfirmModalComponent } from './confirm-modal.component';
4
5  describe('ConfirmModalComponent', () => {
6    let component: ConfirmModalComponent;
7    let fixture: ComponentFixture<ConfirmModalComponent>;
8
9    beforeEach(async(() => {
10      TestBed.configureTestingModule({
11        declarations: [ ConfirmModalComponent ]
12      })
13      .compileComponents();
14    }));
15
16    beforeEach(() => {
17      fixture = TestBed.createComponent(ConfirmModalComponent);
18      component = fixture.componentInstance;
19      fixture.detectChanges();
20    });
21
22    it('should create', () => {
23      expect(component).toBeTruthy();
24    });
25  });
```

- ▼ event-category
 - > event-category-container
 - > event-category-edit
 - > event-category-list
 - > services
 - TS event-category-routing.module.ts
 - TS event-category.module.ts

event-category.module.ts

```
1  import { NgModule } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3
4  import { EventCategoryRoutingModule } from './event-category-routing.module';
5  import { EventCategoryListComponent } from './event-category-list/event-category-list.component';
6  import { EventCategoryEditComponent } from './event-category-edit/event-category-edit.component';
7  import { EventCategoryContainerComponent } from './event-category-container/event-category-container.component';
8  import { AppMaterialModule } from '../app-material/app-material.module';
9  import { FormsModule, ReactiveFormsModule } from '@angular/forms';
10
11
12  @NgModule({
13    declarations: [EventCategoryListComponent, EventCategoryEditComponent, EventCategoryContainerComponent],
14    imports: [
15      CommonModule,
16      EventCategoryRoutingModule,
17      AppMaterialModule,
18      FormsModule,
19      ReactiveFormsModule
20    ],
21    entryComponents: [
22      EventCategoryEditComponent
23    ]
24  })
25  export class EventCategoryModule { }
26
```

event-category-routing.model.ts

```
1  import { NgModule } from '@angular/core';
2  import { Routes, RouterModule } from '@angular/router';
3  import { EventCategoryListComponent } from './event-category-list/event-category-list.component';
4  import { EventCategoryContainerComponent } from './event-category-container/event-category-container.component';
5
6
7  const routes: Routes = [
8    {
9      path: '',
10     component: EventCategoryContainerComponent,
11     children: [
12       {
13         path: '',
14         component: EventCategoryListComponent
15       }
16     ]
17   }
18 ];
19
20 @NgModule({
21   imports: [RouterModule.forChild(routes)],
22   exports: [RouterModule]
23 })
24 export class EventCategoryRoutingModule { }
```

▼ event-category

▼ event-category-container

- # event-category-container.component.css
- ◀ event-category-container.component.html
- TS event-category-container.component.spec.ts
- TS event-category-container.component.ts

▼ event-category-edit

- # event-category-edit.component.css
- ◀ event-category-edit.component.html
- TS event-category-edit.component.spec.ts
- TS event-category-edit.component.ts

▼ event-category-list

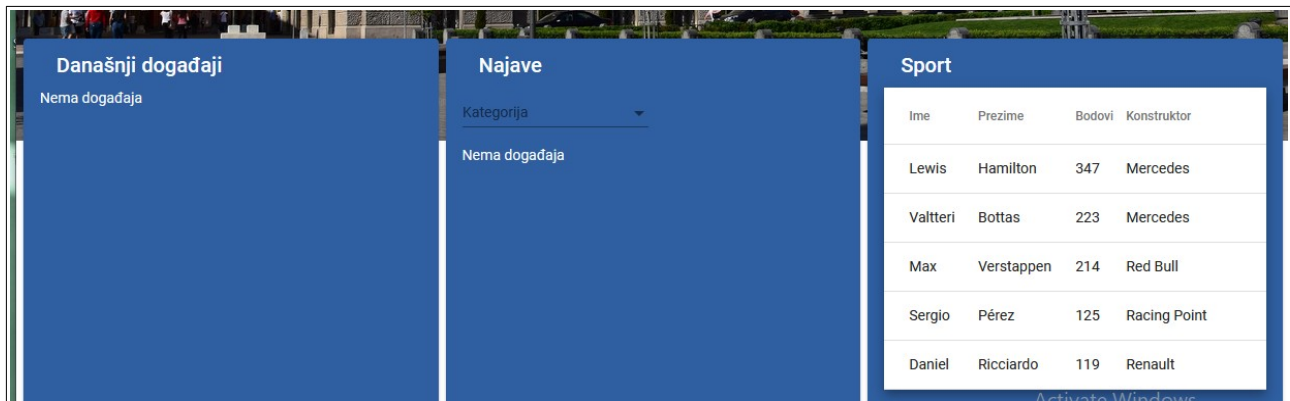
- # event-category-list.component.css
- ◀ event-category-list.component.html
- TS event-category-list.component.spec.ts
- TS event-category-list.component.ts

▼ services

- TS event-category.service.spec.ts
- TS event-category.service.ts
- TS event-category-routing.module.ts
- TS event-category.module.ts

event-category.service.ts

```
1  import { Injectable } from '@angular/core';
2  import { EventCategory } from 'src/app/model/event.category.model';
3  import { EventService } from 'src/app/event/services/event.service';
4
5  @Injectable({
6    providedIn: 'root'
7  })
8  export class EventCategoryService {
9
10     private STORAGE_KEY: string = "event-category";
11     public data: Array<EventCategory> = [];
12
13     constructor(private eventService: EventService) { }
14
15     public saveToLocalStorage() {
16         localStorage.setItem(this.STORAGE_KEY, JSON.stringify(this.data));
17     }
18
19     public loadFromLocalStorage() {
20         let data = localStorage.getItem(this.STORAGE_KEY) || "[]";
21         this.data = JSON.parse(data) || [];
22     }
23
24     public add(eventCategory: EventCategory) {
25         eventCategory.id = new Date().getTime();
26         this.data.push(eventCategory);
27         this.saveToLocalStorage();
28     }
29
30     public edit(id: number|null, eventCategory: EventCategory) {
31         this.data.forEach(item => {
32             if (item.id == id) {
33                 Object.assign(item, eventCategory);
34                 this.saveToLocalStorage();
35                 this.updateEvents(item);
36             }
37         })
38     }
39 }
```



event-category-list.component.html

```

1  <mat-toolbar class="options-menu">
2    <h3>Pregled kategorija</h3>
3    <button mat-icon-button (click)="add()" class="right-button">
4      <mat-icon>add</mat-icon>
5    </button>
6  </mat-toolbar>
7  <table mat-table [dataSource]="dataSource" matSort class="mat-elevation-z8">
8    <ng-container matColumnDef="name">
9      <th mat-header-cell *matHeaderCellDef mat-sort-header> Naziv </th>
10     <td mat-cell *matCellDef="let element"> {{element.name}} </td>
11   </ng-container>
12
13   <ng-container matColumnDef="edit">
14     <th mat-header-cell *matHeaderCellDef> </th>
15     <td mat-cell *matCellDef="let element">
16       <button mat-icon-button (click)="edit(element)">
17         <mat-icon>edit</mat-icon>
18       </button>
19     </td>
20   </ng-container>
21   <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
22   <tr mat-row *matRowDef="let row; columns: displayedColumns;"></tr>
23 </table>

```

event-category-list.component.css

```

1  table {
2    width: 100%;
3    position: absolute;
4    top: 128px;
5    bottom: 0;
6    left: 0;
7    right: 0;
8  }
9
10 .right-button {
11   position: absolute;
12   right: 8px;
13 }

```

event-category-list.component.ts

```
1 import { Component, OnInit, ViewChild } from '@angular/core';
2 import { EventCategory } from 'src/app/model/event.category.model';
3 import { EventCategoryService } from '../services/event-category.service';
4 import { EventCategoryEditComponent } from '../event-category-edit/event-category-edit.comp
5 import { MatTableDataSource } from '@angular/material/table';
6 import { MatSort } from '@angular/material/sort';
7 import { MatDialog } from '@angular/material/dialog';
8
9 @Component({
10   selector: 'app-event-category-list',
11   templateUrl: './event-category-list.component.html',
12   styleUrls: ['./event-category-list.component.css']
13 })
14 export class EventCategoryListComponent implements OnInit {
15
16   displayedColumns: string[] = ['name', 'edit'];
17   dataSource = new MatTableDataSource<EventCategory>();
18   // @ViewChild(MatSort, { static: true }) sort: MatSort;
19
20   constructor(private service: EventCategoryService,
21     private dialog: MatDialog) {
22   }
23
24   ngOnInit() {
25     this.dataSource.data = this.service.getAll();
26     // this.dataSource.sort = this.sort;
27   }
28
29   add() {
30     this.dialog.open(EventCategoryEditComponent, {
31       width: '600px',
32       data: {
33         eventCategory: new EventCategory(),
34         isEdit: false
35       }
36     })
37     .afterClosed()
38     .subscribe(result => {
39       this.dataSource.data = this.service.getAll();
40     });
41   }
42 }
```



```
20     constructor(private service: EventCategoryService,  
21     | private dialog: MatDialog) {  
22     }  
23  
24     ngOnInit() {  
25         this.dataSource.data = this.service.getAll();  
26         //this.dataSource.sort = this.sort;  
27     }  
28  
29     add() {  
30         this.dialog.open(EventCategoryEditComponent, {  
31             width: '600px',  
32             data: {  
33                 eventCategory: new EventCategory(),  
34                 isEdit: false  
35             }  
36         })  
37         .afterClosed()  
38         .subscribe(result => {  
39             this.dataSource.data = this.service.getAll();  
40         });  
41     }  
42  
43     edit(element: any) {  
44         this.dialog.open(EventCategoryEditComponent, {  
45             width: '600px',  
46             data: {  
47                 eventCategory: element,  
48                 isEdit: true  
49             }  
50         })  
51         .afterClosed()  
52         .subscribe(result => {  
53             this.dataSource.data = this.service.getAll();  
54         });  
55     }  
56  
57 }  
58
```

login.component.css

```
1  .login-container {
2    height: 100%;
3    display: flex;
4    align-items: center;
5    justify-content: center;
6  }
7
8  .login-form {
9    display: flex;
10   flex-direction: column;
11 }
```

event-category-edit.component.html

```
1  <h1 mat-dialog-title>Nova kategorija</h1>
2  <form [formGroup]="form">
3    <div mat-dialog-content class="create-form">
4      <mat-form-field>
5        <input matInput placeholder="Naziv" required formControlName="name" autocomplete="off" />
6      </mat-form-field>
7    </div>
8    <div mat-dialog-actions>
9      <button mat-button (click)="close()">Odustani</button>
10     <button mat-button [disabled]="form.invalid" type="submit" (click)="save(form)">Sacuvaj</button>
11   </div>
12 </form>
```

event-category-edit.component.css

```
1  .create-form {
2    display: flex;
3    flex-direction: column;
4    height: 250px;
5  }
```

event-category-edit.component.ts

```
1  import { Component, OnInit, Inject } from '@angular/core';
2  import { FormGroup, FormBuilder, Validators } from '@angular/forms';
3  import { MatDialogRef, MAT_DIALOG_DATA } from '@angular/material/dialog';
4  import { MatSnackBar } from '@angular/material/snack-bar';
5  import { EventCategory } from 'src/app/model/event.category.model';
6  import { EventCategoryService } from '../services/event-category.service';
7
8  @Component({
9    selector: 'app-event-category-edit',
10   templateUrl: './event-category-edit.component.html',
11   styleUrls: ['./event-category-edit.component.css']
12 })
13 export class EventCategoryEditComponent implements OnInit {
14
15   public form: FormGroup = new FormGroup({}); //forma
16   public eventCategory: EventCategory = new EventCategory();
17   private isEdit = false; //true ako je izmjena, false ako je kreiranje
18
19   constructor(public formBuilder: FormBuilder, //formBuilder služi za kreiranje forme
20     private service: EventCategoryService, //service koristimo za čuvanje podataka
21     private snackBar: MatSnackBar,
22     private dialogRef: MatDialogRef<EventCategoryEditComponent>,
23     @Inject(MAT_DIALOG_DATA) data: any) {
24     this.isEdit = data.isEdit;
25     this.eventCategory = data.eventCategory;
26
27   } //snackBar za prikaz obavještenja
28
29   ngOnInit() {
30     //prilikom učitavanja stranice pravimo formu
31     this.form = this.formBuilder.group({
32       name: [this.eventCategory.name, Validators.required]
33     });
34   }
35 }
```

```

36 save({ value, valid }: { value: EventCategory, valid: boolean }) {
37     if (valid) { //ako su OK
38
39         if(this.isEdit) {
40             this.service.edit(this.eventCategory.id, value);
41         } else {
42             this.service.add(value); //koristimo nas servis da ih sacuvamo
43         }
44
45
46         this.form.reset(); //ponistimo prethodno unesene podatke
47         this.snackBar.open("Podaci su sacuvani", undefined, { //i prikazemo poruku koja nestaje nakon 2
48             duration: 2000,
49         });
50         this.close();
51     }
52 }
53
54 close() {
55     this.dialogRef.close();
56 }
57
58
59 }

```

Activate Windows

events-category-container.html

```

1 <mat-toolbar>
2     <mat-toolbar-row>
3         <button mat-icon-button routerLink="/">
4             <mat-icon>home</mat-icon>
5         </button>
6         <a routerLink="/events" class="link" routerLinkActive="primary">Događaji </a>
7         <a routerLink="/event-categories" class="link" routerLinkActive="primary"> Kategorije događaj
8     </mat-toolbar-row>
9 </mat-toolbar>
10 <router-outlet></router-outlet>

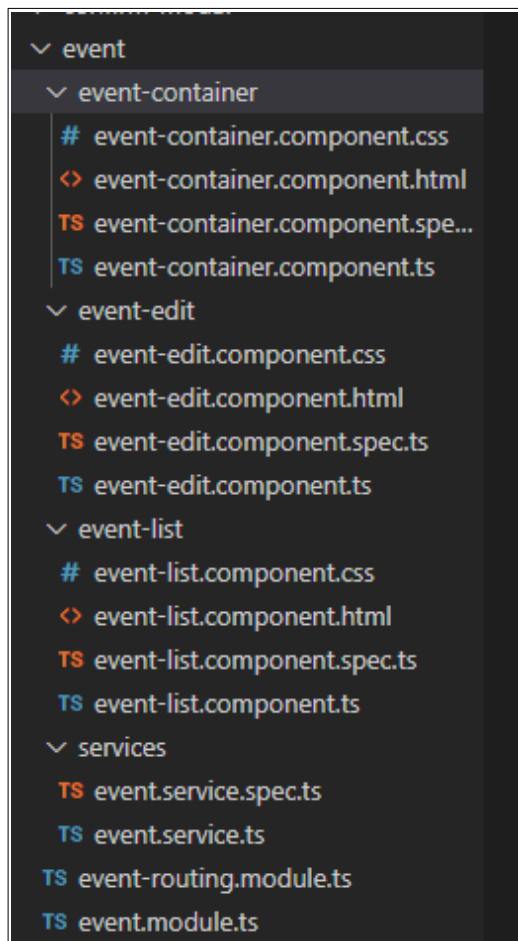
```

events-category-container.ts

```

1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4     selector: 'app-event-category-container',
5     templateUrl: './event-category-container.component.html',
6     styleUrls: ['./event-category-container.component.css']
7 })
8 export class EventCategoryContainerComponent implements OnInit {
9
10     constructor() { }
11
12     ngOnInit() {
13     }
14
15 }
16

```



event.module.ts

```
1 import { NgModule } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3
4 import { EventRoutingModule } from './event-routing.module';
5 import { EventListComponent } from './event-list/event-list.component';
6 import { EventContainerComponent } from './event-container/event-container.component';
7 import { EventEditComponent } from './event-edit/event-edit.component';
8 import { AppMaterialModule } from '../app-material/app-material.module';
9 import { FormsModule, ReactiveFormsModule } from '@angular/forms';
10
11
12 @NgModule({
13   declarations: [EventListComponent, EventContainerComponent, EventEditComponent],
14   imports: [
15     CommonModule,
16     EventRoutingModule,
17     AppMaterialModule,
18     FormsModule,
19     ReactiveFormsModule
20   ],
21   entryComponents: [
22     EventEditComponent
23   ]
24 })
25 export class EventModule { }
26
```

event-routing-module.ts

```
1 import { NgModule } from '@angular/core';
2 import { Routes, RouterModule } from '@angular/router';
3 import { EventListComponent } from './event-list/event-list.component';
4 import { EventContainerComponent } from './event-container/event-container.component';
5
6
7 const routes: Routes = [{
8   path: '',
9   component: EventContainerComponent,
10   children: [
11     {
12       path: '',
13       component: EventListComponent
14     }
15   ]
16 }];
17
18 @NgModule({
19   imports: [RouterModule.forChild(routes)],
20   exports: [RouterModule]
21 })
22 export class EventRoutingModule { }
23
```

```
1 import { Injectable } from '@angular/core';
2 import { Event } from 'src/app/model/event.model';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class EventService {
8
9   private readonly STORAGE_KEY: string = "events";
10   public data: Array<Event> = [];
11
12   constructor() { }
13
14   public saveToLocalStorage() {
15     localStorage.setItem(this.STORAGE_KEY, JSON.stringify(this.data));
16   }
17
18   public loadFromLocalStorage() {
19     let data = localStorage.getItem(this.STORAGE_KEY) || "[]";
20     this.data = JSON.parse(data) || [];
21     this.data.forEach((e: any) => {
22       e.date = new Date(e.date);
23     });
24   }
25
26   public add(event: Event) {
27     event.id = new Date().getTime();
28     this.data.push(event);
29     this.saveToLocalStorage();
30   }
31
32   public delete(id: number) {
33     let position = -1;
34     this.data.forEach((item, index) => {
35       if (item.id == id) {
36         position = index;
37       }
38     });
39     if (position > -1) {
40       this.data.splice(position, 1);
41       this.saveToLocalStorage();
42     }
43   }
44 }
```

events-list.component.ts

```
1  import { Component, OnInit, ViewChild } from '@angular/core';
2  import { Event } from 'src/app/model/event.model';
3  import { EventEditComponent } from '../event-edit/event-edit.component';
4  import { EventService } from '../services/event.service';
5  import { ConfirmModalComponent } from 'src/app/confirm-modal/confirm-modal.component';
6  import { MatTableDataSource } from '@angular/material/table';
7  import { MatSort } from '@angular/material/sort';
8  import { MatDialog } from '@angular/material/dialog';
9
10 @Component({
11   selector: 'app-event-list',
12   templateUrl: './event-list.component.html',
13   styleUrls: ['./event-list.component.css']
14 })
15 export class EventListComponent implements OnInit {
16
17   displayedColumns: string[] = ['name', 'description', 'image', 'category', 'date', 'time', 'delete']
18   dataSource = new MatTableDataSource<Event>();
19   // @ViewChild(MatSort, { static: true }) sort: MatSort|undefined;
20
21   constructor(private service: EventService,
22     private dialog: MatDialog) {
23   }
24
25   ngOnInit() {
26     this.dataSource.data = this.service.getAll();
27     // this.dataSource.sort = this.sort;
28   }
29 }
```



```
30     add() {
31         this.dialog.open(EventEditComponent, {
32             width: '600px'
33         })
34         .afterClosed()
35         .subscribe(result => {
36             this.dataSource.data = this.service.getAll();
37         });
38     }
39
40     delete(element: any) {
41         this.dialog.open(ConfirmModalComponent, {
42             width: '300px'
43         })
44         .afterClosed()
45         .subscribe(result => {
46             if (result) {
47                 this.service.delete(element.id);
48                 this.dataSource.data = this.service.getAll();
49             }
50         });
51     }
52
53 }
54
```

event-list.component.html

```

1  <mat-toolbar class="options-menu">
2    <h3>Pregled događaja</h3>
3    <button mat-icon-button (click)="add()" class="right-button">
4      <mat-icon>add</mat-icon>
5    </button>
6  </mat-toolbar>
7  <table mat-table [dataSource]="dataSource" matSort class="mat-elevation-z8">
8    <ng-container matColumnDef="name">
9      <th mat-header-cell *matHeaderCellDef mat-sort-header> Naziv </th>
10     <td mat-cell *matCellDef="let element"> {{element.name}} </td>
11   </ng-container>
12
13   <ng-container matColumnDef="description">
14     <th mat-header-cell *matHeaderCellDef mat-sort-header> Opis </th>
15     <td mat-cell *matCellDef="let element"> {{element.description}} </td>
16   </ng-container>
17
18   <ng-container matColumnDef="image">
19     <th mat-header-cell *matHeaderCellDef mat-sort-header> Slika </th>
20     <td mat-cell *matCellDef="let element">
21       <img height="50" width="100" [src]="element.image" alt="-" *ngIf="element.image"/>
22     </td>
23   </ng-container>
24

```

```

25   <ng-container matColumnDef="date">
26     <th mat-header-cell *matHeaderCellDef mat-sort-header> Datum </th>
27     <td mat-cell *matCellDef="let element"> {{element.date | date: "dd.MM.yyyy"}} </td>
28   </ng-container>
29
30   <ng-container matColumnDef="time">
31     <th mat-header-cell *matHeaderCellDef mat-sort-header> Vrijeme </th>
32     <td mat-cell *matCellDef="let element"> {{element.time}} </td>
33   </ng-container>
34
35   <ng-container matColumnDef="category">
36     <th mat-header-cell *matHeaderCellDef mat-sort-header> Kategorija </th>
37     <td mat-cell *matCellDef="let element"> {{element.category.name}} </td>
38   </ng-container>
39
40   <ng-container matColumnDef="delete">
41     <th mat-header-cell *matHeaderCellDef> </th>
42     <td mat-cell *matCellDef="let element">
43       <button mat-icon-button (click)="delete(element)">
44         <mat-icon>delete</mat-icon>
45       </button>
46     </td>
47   </ng-container>
48   <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
49   <tr mat-row *matRowDef="let row; columns: displayedColumns;"></tr>
50 </table>

```

event-list.component.css

```
1  table {
2    width: 100%;
3    position: absolute;
4    top: 128px;
5    bottom: 0;
6    left: 0;
7    right: 0;
8  }
9
10  .right-button {
11    position: absolute;
12    right: 8px;
13  }
```

event-edit.component.css

```
1  .create-form {
2    display: flex;
3    flex-direction: column;
4  }
5
6  .row {
7    display: flex;
8    flex-direction: row;
9    justify-content: space-between;
10 }
```

event-edit.component.css

```
1  .link {
2    text-decoration: none;
3    font-size: 12px;
4    padding: 8px;
5  }
```

event-container.component.css

```
1  .link {
2    text-decoration: none;
3    font-size: 12px;
4    padding: 8px;
5  }
```

event-edit.component.ts

```
1  import { Component, OnInit } from '@angular/core';
2  import { FormGroup, FormBuilder, Validators } from '@angular/forms';
3  import { Event } from 'src/app/model/event.model';
4  import { EventService } from '../services/event.service';
5  import { EventCategoryService } from 'src/app/event-category/services/event-category.service';
6  import { EventCategory } from 'src/app/model/event.category.model';
7  import { MatSnackBar } from '@angular/material/snack-bar';
8  import { MatDialogRef } from '@angular/material/dialog';
9
10 @Component({
11   selector: 'app-event-edit',
12   templateUrl: './event-edit.component.html',
13   styleUrls: ['./event-edit.component.css']
14 })
15 export class EventEditComponent implements OnInit {
16
17   public form: FormGroup = new FormGroup({}); //forma
18   public event: Event = new Event();
19   public categories: Array<EventCategory> = [];
20
21   constructor(public formBuilder: FormBuilder, //formBuilder služi za kreiranje forme
22     private service: EventService, //service koristimo za cuvanje podataka
23     private snackBar: MatSnackBar,
24     private dialogRef: MatDialogRef<EventEditComponent>,
25     private categoryService: EventCategoryService) {
26     this.categories = this.categoryService.getAll();
27   }
28
29   ngOnInit() {
30     //prilikom učitavanja stranice pravimo formu
31     this.form = this.formBuilder.group({
32       name: [this.event.name, Validators.required],
33       description: [this.event.description, Validators.required],
34       image: [this.event.image],
35       date: [this.event.date, Validators.required],
36       time: [this.event.time, Validators.required],
37       category: [this.event.category, Validators.required]
38     });
39   }
40
41   save({ value, valid }: { value: Event, valid: boolean }) {
42     if (valid) { //ako su OK
43
44       this.service.add(value); //koristimo naš servis da ih sačuvamo
45       this.form.reset(); //ponistimo prethodno unesene podatke
46       this.snackBar.open("Podaci su sačuvani", undefined, { //i prikazemo poruku koja nestaje nakon 2
47         duration: 2000,
48       });
49       this.close();
50     }
51   }
52
53   close() {
54     this.dialogRef.close();
55   }
56
57 }
58 }
```

event-edit.component.html

```

1  <h1 mat-dialog-title>Novi događaj</h1>
2  <form [formGroup]="form">
3    <div mat-dialog-content class="create-form">
4      <mat-form-field>
5        <input matInput placeholder="Naziv" required formControlName="name" autocomplete="off" />
6      </mat-form-field>
7
8      <mat-form-field>
9        <input matInput placeholder="Opis" required formControlName="description" autocomplete="off" />
10     </mat-form-field>
11
12     <mat-form-field>
13       <input matInput placeholder="Slika" formControlName="image" autocomplete="off" />
14     </mat-form-field>
15
16     <div class="row">
17       <mat-form-field>
18         <input matInput [matDatepicker]="picker" placeholder="Datum" formControlName="date">
19         <mat-datepicker-toggle matSuffix [for]="picker"></mat-datepicker-toggle>
20         <mat-datepicker #picker></mat-datepicker>
21       </mat-form-field>
22
23       <mat-form-field>
24         <input matInput placeholder="Vrijeme" required formControlName="time" autocomplete="off" />
25       </mat-form-field>
26     </div>
27
28     <mat-form-field>
29       <mat-label>Kategorija</mat-label>
30       <mat-select formControlName="category" required>
31         <mat-option *ngFor="let category of categories" [value]="category">
32           {{category.name}}
33         </mat-option>
34       </mat-select>
35     </mat-form-field>
36   </div>
37   <div mat-dialog-actions>
38     <button mat-button (click)="close()">Odustani</button>
39     <button mat-button [disabled]="form.invalid" type="submit" (click)="save(form)">Sacuvaj</button>
40   </div>
41 </form>

```

event-container.component.ts

```
1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-event-container',
5   templateUrl: './event-container.component.html',
6   styleUrls: ['./event-container.component.css']
7 })
8 export class EventContainerComponent implements OnInit {
9
10   constructor() { }
11
12   ngOnInit() {
13   }
14
15 }
```

event-container.component.html

```
1 <mat-toolbar>
2   <mat-toolbar-row>
3     <button mat-icon-button routerLink="/">
4       <mat-icon>home</mat-icon>
5     </button>
6     <a routerLink="/events" class="link" routerLinkActive="primary">Dogadaji </a>
7     <a routerLink="/event-categories" class="link" routerLinkActive="primary"> Kategorije događaja</a>
8   </mat-toolbar-row>
9 </mat-toolbar>
10 <router-outlet></router-outlet>
```

```
▼ home
  ▼ home
    # home.component.css
    <> home.component.html
    TS home.component.spec.ts
    TS home.component.ts
    TS home-routing.module.ts
    TS home.module.ts
```

home.module.ts

```
1  import { NgModule } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3
4  import { HomeRoutingModule } from '../home-routing.module';
5  import { HomeComponent } from '../home/home.component';
6  import { AppModule } from '../app-material/app-material.module';
7  import { FormsModule, ReactiveFormsModule } from '@angular/forms';
8  import { SportModule } from '../sport/sport.module';
9
10
11  @NgModule({
12    declarations: [HomeComponent],
13    imports: [
14      CommonModule,
15      HomeRoutingModule,
16      AppModule,
17      FormsModule,
18      ReactiveFormsModule,
19      SportModule
20    ]
21  })
22  export class HomeModule { }
```

home-routing.model.ts

```
src > app > home > TS home-routing.module.ts > ...
1  import { NgModule } from '@angular/core';
2  import { Routes, RouterModule } from '@angular/router';
3  import { HomeComponent } from '../home/home.component';
4
5
6  const routes: Routes = [
7    {
8      path: '',
9      component: HomeComponent
10   }
11 ];
12
13  @NgModule({
14    imports: [RouterModule.forChild(routes)],
15    exports: [RouterModule]
16  })
17  export class HomeRoutingModule { }
18  |
```

home.component.ts

```
1  import { Component, OnInit } from '@angular/core';
2  import { EventService } from 'src/app/event/services/event.service';
3  import { Event } from 'src/app/model/event.model';
4  import { EventCategory } from 'src/app/model/event.category.model';
5  import { EventCategoryService } from 'src/app/event-category/services/event-category.service';
6
7  @Component({
8    selector: 'app-home',
9    templateUrl: './home.component.html',
10   styleUrls: ['./home.component.css']
11 })
12 export class HomeComponent implements OnInit {
13
14   public eventsToday: Array<Event> = []; //danasnji
15   public events: Array<Event> = []; //svi za filter
16   public categories: Array<EventCategory> = [];
17
18   constructor(private service: EventService,
19     private categoryService: EventCategoryService) {
20     this.events = this.service.getFutureEvents();
21     this.eventsToday = this.service.getTodayEvents();
22     this.categories = this.categoryService.getAll();
23   }
24
25   ngOnInit() {
26   }
27
28   filterCategory(event: any) {
29     let category = event.value.id;
30     this.events = this.service.getFutureEvents().filter((e: any) => {
31       return e.category.id == category
32     });
33   }
34
35   onF1DataLoad(result: string) {
36     console.log("F1:", result);
37   }
38
39 }
40
```


home.component.html

```
1 <a routerLink="/login" class="link right-link" routerLinkActive="primary">Prijava </a>
2 
3 <div class="main-title">
4   <h1>Banja Luka događaji</h1>
5   <p style="width: 200px">Ovaj sajt je primjer i nije namijenjen komercijalnoj upotrebi.
6   Osnovna namjena sajta je da pokaže jedan način implementacije traženih funkcionalnosti.</p>
7 </div>
8 <div class="main-content">
9   <mat-card class="info-card">
10     <mat-card-header>
11       <mat-card-title>
12         Današnji događaji
13       </mat-card-title>
14     </mat-card-header>
15     <mat-card-content>
16       <p *ngIf="eventsToday.length == 0">Nema događaja</p>
17       <div class="event" *ngFor="let event of eventsToday">
18         <h3>{{event.name}}</h3>
19         <p>{{event.category?.name}}</p>
20         <p *ngIf="event.image">
21           <img [src]="event.image" width="200" />
22         </p>
23         <p>{{event.description}}</p>
24         <p>{{event.date | date:'dd.MM.yyyy'}} {{event.time}}</p>
25       </div>
26     </mat-card-content>
27   </mat-card>
```

```

28     <mat-card class="info-card">
29         <mat-card-header>
30             <mat-card-title>
31                 Najave
32             </mat-card-title>
33         </mat-card-header>
34         <mat-card-content>
35             <mat-form-field>
36                 <mat-label>Kategorija</mat-label>
37                 <mat-select (selectionChange)="filterCategory($event)">
38                     <mat-option *ngFor="let category of categories" [value]="category">
39                         {{category.name}}
40                     </mat-option>
41                 </mat-select>
42             </mat-form-field>
43             <p *ngIf="events.length == 0">Nema događaja</p>
44             <div class="event" *ngFor="let event of events">
45                 <h3>{{event.name}}</h3>
46                 <p>{{event.category?.name}}</p>
47                 <p *ngIf="event.image">
48                     <img [src]="event.image" width="200" />
49                 </p>
50                 <p>{{event.description}}</p>
51                 <p>{{event.date | date:'dd.MM.yyyy'}} {{event.time}}</p>
52             </div>
53         </mat-card-content>
54     </mat-card>
55     <mat-card class="info-card">
56         <mat-card-header>
57             <mat-card-title>
58                 Sport
59             </mat-card-title>
60         </mat-card-header>
61         <mat-card-content>
62             <app-f1-results season="2020" (onDataLoad) = "onF1DataLoad($event)"></app-f1-results>
63         </mat-card-content>
64     </mat-card>
65 </div>

```

```
1  .right-link {
2      right: 4px;
3      position: absolute;
4      top: 4px;
5      color: ■white;
6  }
7
8  .link {
9      text-decoration: none;
10     font-size: 12px;
11     padding: 8px;
12 }
13
14 .main-title {
15     top: 0px;
16     left: 8px;
17     position: absolute;
18     color: ■white;
19 }
20
21 .main-content {
22     position: absolute;
23     top: 500px;
24     left: 0;
25     right: 0;
26     display: flex;
27     flex-direction: row;
28     flex-wrap: wrap;
29     justify-content: space-evenly;
30     margin-bottom: 16px;
31 }
32
33 .info-card {
34     min-width: 30%;
35     background-color: ■rgb(47, 95, 161);
36     color: ■white;
37     min-height: 300px;
38 }
39
40 .event {
41     background-color: ■rgb(60, 106, 168);
42     padding: 8px;
43     margin: 4px;
44     text-align: center;
45 }
```

Neke karakteristike

```
<div class="event" *ngFor="let event of eventsToday">
  <h3>{{event.name}}</h3>
  <p>{{event.category?.name || "-"}}</p>
  <p *ngIf="event.image">
    <img [src]="event.image" width="200" />
  </p>
</div>
```

Operator rezervnog ispisa kada je upitanju nula pokazivac

```
private users: Array<User> = [];
public signedIn: boolean = false;
public activeUser: User|null = null;
```

Вишеструки типови

```
this.data.forEach((item, index) => {
  if (item.id == id) {
    position = index;
  }
});
```

```
//prilikom ucitavanja stranice pravimo formu
this.form = this.formBuilder.group({
  name: [this.event.name, Validators.required],
  description: [this.event.description, Validators.required],
  image: [this.event.image],
  date: [this.event.date, Validators.required],
  time: [this.event.time, Validators.required],
  category: [this.event.category, Validators.required]
});
```

```
delete(element: any) {
  this.dialog.open(ConfirmModalComponent, {
    width: '300px'
  })
  .afterClosed()
  .subscribe(result => {
    if (result) {
      this.service.delete(element.id);
      this.dataSource.data = this.service.getAll();
    }
  });
}
```