Visoka škola strukovnih studija za informacione i komunikacione tehnologije



# Dokumentacija

AllMyCoins

Nikola Nedeljković 2/17

# Sadržaj

[Dokumentacija 1](#_Toc41602138)

[Sadržaj 2](#_Toc41602139)

[1. UVOD 4](#_Toc41602140)

[1.1 Korišćeni programski / markap jezici 4](#_Toc41602141)

[1.2 Opis funkcionalnosti 4](#_Toc41602142)

[1.3 Izgled stranica 6](#_Toc41602143)

[2. ORGANIZACIJA 7](#_Toc41602144)

[2.1 Organizacija sajta 7](#_Toc41602145)

[2.2 Organizacija baze 8](#_Toc41602149)

[2.3 Slike stranica sa funkcionalnostima 9](#_Toc41602154)

[3. KODOVI 12](#_Toc41602155)

[3.1 styles.scss 12](#_Toc41602156)

[3.2. app.component.html 12](#_Toc41602161)

[3.3. app.module.ts 12](#_Toc41602162)

[3.4 app.component.ts 13](#_Toc41602169)

[3.5. app-routing.module.ts 14](#_Toc41602171)

[3.6 coin.service.ts 14](#_Toc41602180)

[3.7 customvalue.pipe.ts 16](#_Toc41602190)

[3.8. customyear.pipe.ts 17](#_Toc41602201)

[3.9 navigation.component.html 17](#_Toc41602213)

[3.10 navigation.component.scss 18](#_Toc41602226)

[3.11. navigation.component.ts 19](#_Toc41602240)

[3.12 not-found.component.scss 20](#_Toc41602255)

[3.13. not-found.component.ts 21](#_Toc41602271)

[3.14. not-found.component.html 21](#_Toc41602288)

[3.15. search-coins.component.scss 22](#_Toc41602306)

[3.16. search-coins.component.ts 22](#_Toc41602325)

[3.17. search-coins.component.html 23](#_Toc41602328)

[3.18. title-area.component.scss 23](#_Toc41602329)

[3.19. title-area.component.ts 24](#_Toc41602351)

[3.20. title-area.component.html 24](#_Toc41602355)

[3.21. coin.component.scss 24](#_Toc41602379)

[3.22. coin.component.ts 25](#_Toc41602404)

[3.23. coin.component.html 25](#_Toc41602430)

[3.24. logout.php 🡪 brisanje sesije korisnika 26](#_Toc41602457)

[3.25. add-coin.component.scss 26](#_Toc41602485)

[3.26. add-coin.component.ts 27](#_Toc41602514)

[3.27. add-coin.component.html 30](#_Toc41602544)

[3.28. author-coin.component.scss 33](#_Toc41602575)

[3.29. author-coin.component.ts 33](#_Toc41602607)

[3.30. author-coin.component.html 33](#_Toc41602640)

[3.31. coin-fallin.component.html 33](#_Toc41602674)

[3.32. coin-fallin.component.scss 34](#_Toc41602709)

[3.33. coin-fallin.component.ts 34](#_Toc41602745)

[3.34. edit-coin.component.html 34](#_Toc41602782)

[3.35. edit-coin.component.scss 37](#_Toc41602820)

[3.36. edit-coin.component.ts 38](#_Toc41602859)

[3.37. no-coin-show.component.scss 42](#_Toc41602899)

[3.38. no-coin-show.component.ts 42](#_Toc41602940)

[3.39. no-coin-show.component.html 43](#_Toc41602982)

[3.40. preview-coin.component.html 43](#_Toc41603025)

[3.41. preview-coin.component.scss 43](#_Toc41603069)

[3.42. preview-coin.component.ts 44](#_Toc41603114)

[3.43. single-coin.component.scss 45](#_Toc41603160)

[3.44. single-coin.component.ts 46](#_Toc41603207)

[3.45. single-coin.component.html 47](#_Toc41603255)

[4. HIJERARHIJSKI PREGLED 49](#_Toc41603304)

# UVOD

## Korišćeni programski / markap jezici

* Za izradu ovog sajta korišćene su sledeće tehnologije:
* Mark-up jezici: HTML
* SCSS
* Skripting-programski jezici: TypeScript
* Bootstrap
* Angular – JavaScript framework
* JSON
* Sajt je napisan u Visual Studio Code-u

## Opis funkcionalnosti

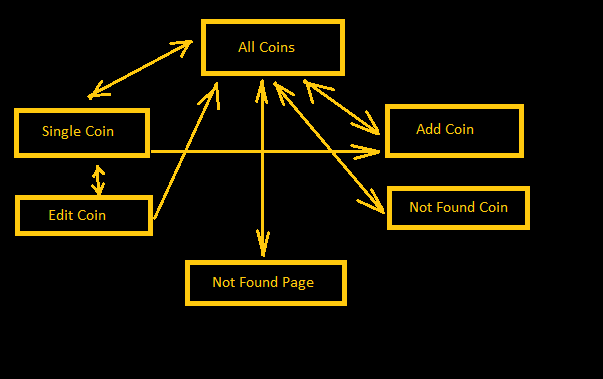
* **src\app\app-routing.module.ts**
* Module za routing
* **src\app\coin.service.ts**
* Coin service za komunikaciju sa serverom
* **src\app\customvalue.pipe.ts**
* Pipe za konverziju vrednosti iz mm u m i iz g u kg
* **src\app\customyear.pipe.ts**
* Pipe za konverziju vrednosti godine u BC i AD
* **src\app\component\fixed\navigation\navigation.component.html**
* Komponenta za navigaciju
* **src\app\component\fixed\not-found\not-found.component.html**
* Komponenta za not found rutu
* **src\app\component\fixed\search-coins\search-coins.component.html**
* Komponenta za pretragu kovanica
* **src\app\component\fixed\title-area\title-area.component.html**
* Komponenta za naslov stranice kad se ne učitava pretraga kovanica
* **src\app\component\coin\coin.component.html**
* Komponenta za prikaz početni prikaz sa virtual list-om za kovanice
* **src\app\component\coin\add-coin\add-coin.component.html**
* Komponenta za dodavanje jedne kovanice
* **src\app\component\coin\author-coin\author-coin.component.html**
* Komponenta za deo o autoru sa animacijom
* **src\app\component\coin\coin-fallin\coin-fallin.component.html**
* Komponenta za padajuću paru kad nije nađena ni jedna kovanica u pretrazi
* **src\app\component\coin\edit-coin\edit-coin.component.html**
* Komponenta za izmenu jedne kovanice
* **src\app\component\coin\no-coin-show\no-coin-show.component.html**
* Komponenta za prikaz kad nije ni jedna kovanice pronađena
* **src\app\component\coin\preview-coin\preview-coin.component.html**
* Komonenta za preview prikaz jedne komponente
* **src\app\component\coin\single-coin\single-coin.component.html**
* Komponenta za prikaz jedne kovanice

## Izgled stranica



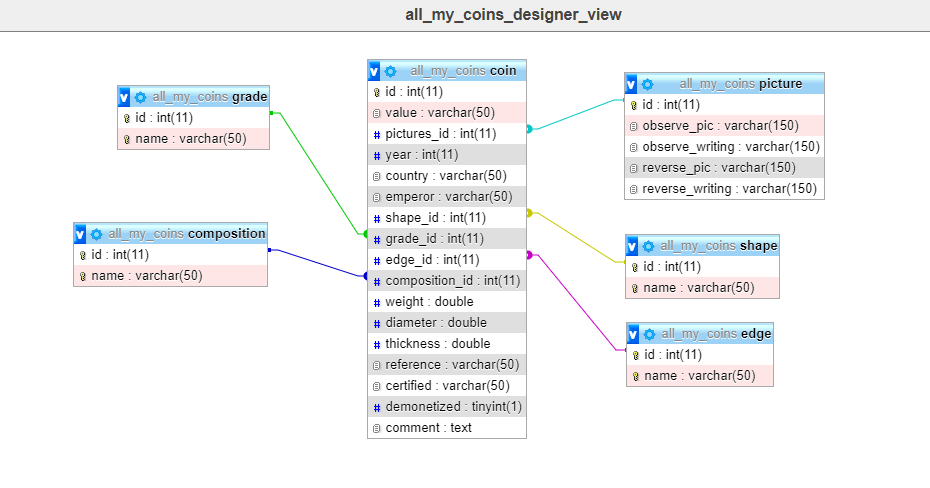
# ORGANIZACIJA

## Organizacija sajta



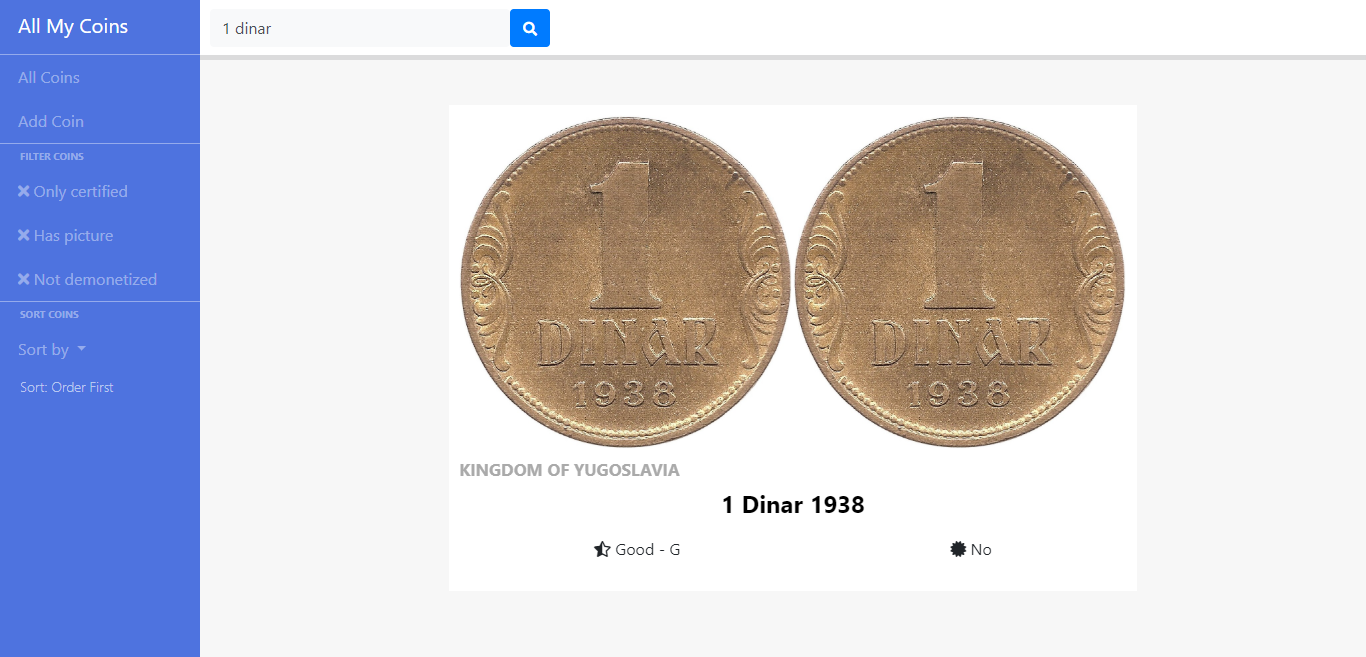


## Organizacija baze

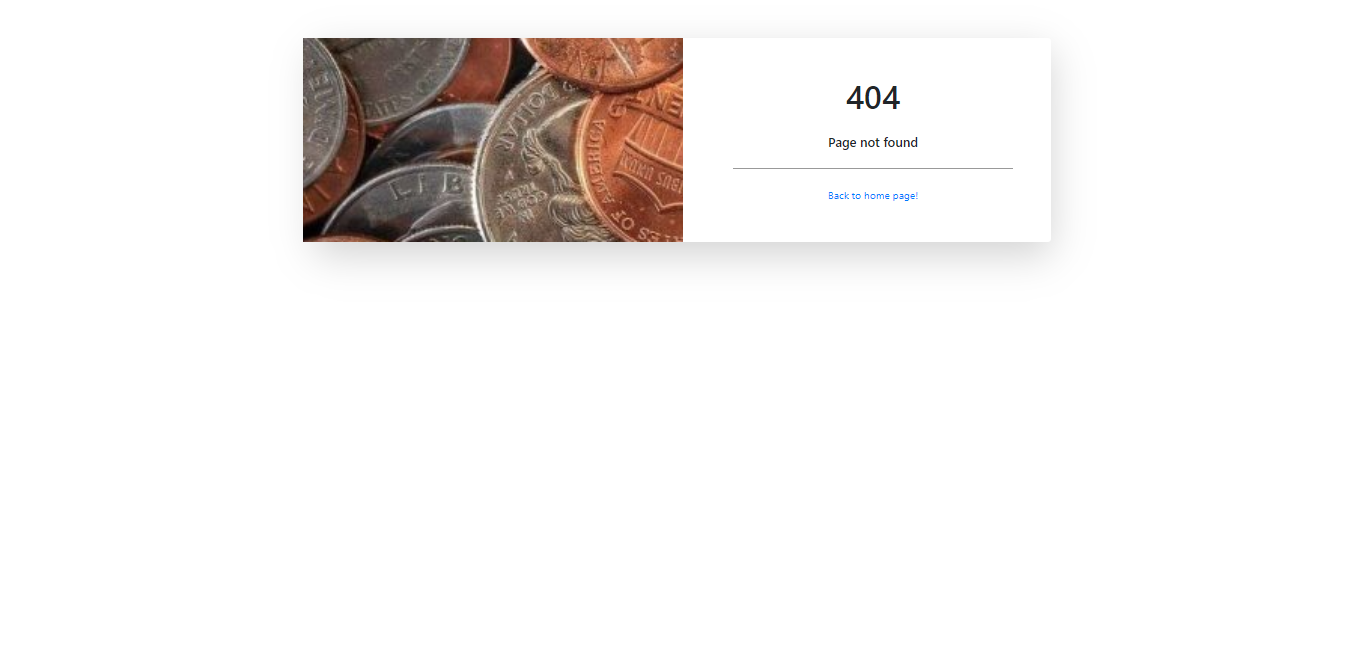




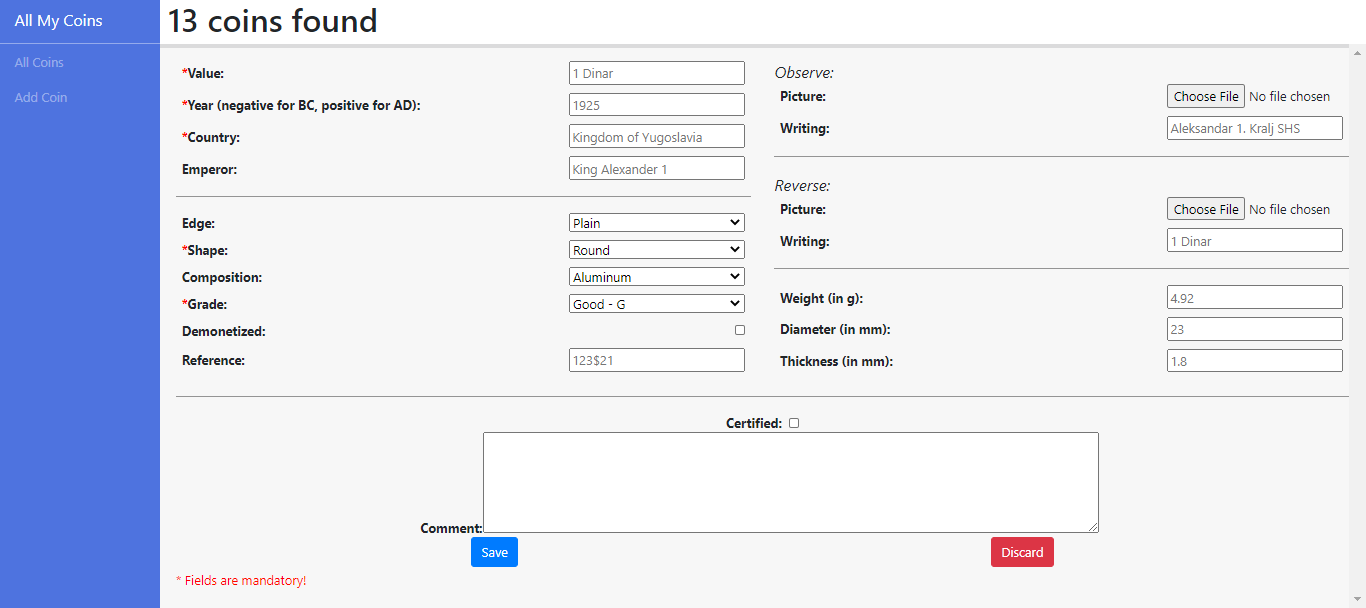
## Slike stranica sa funkcionalnostima



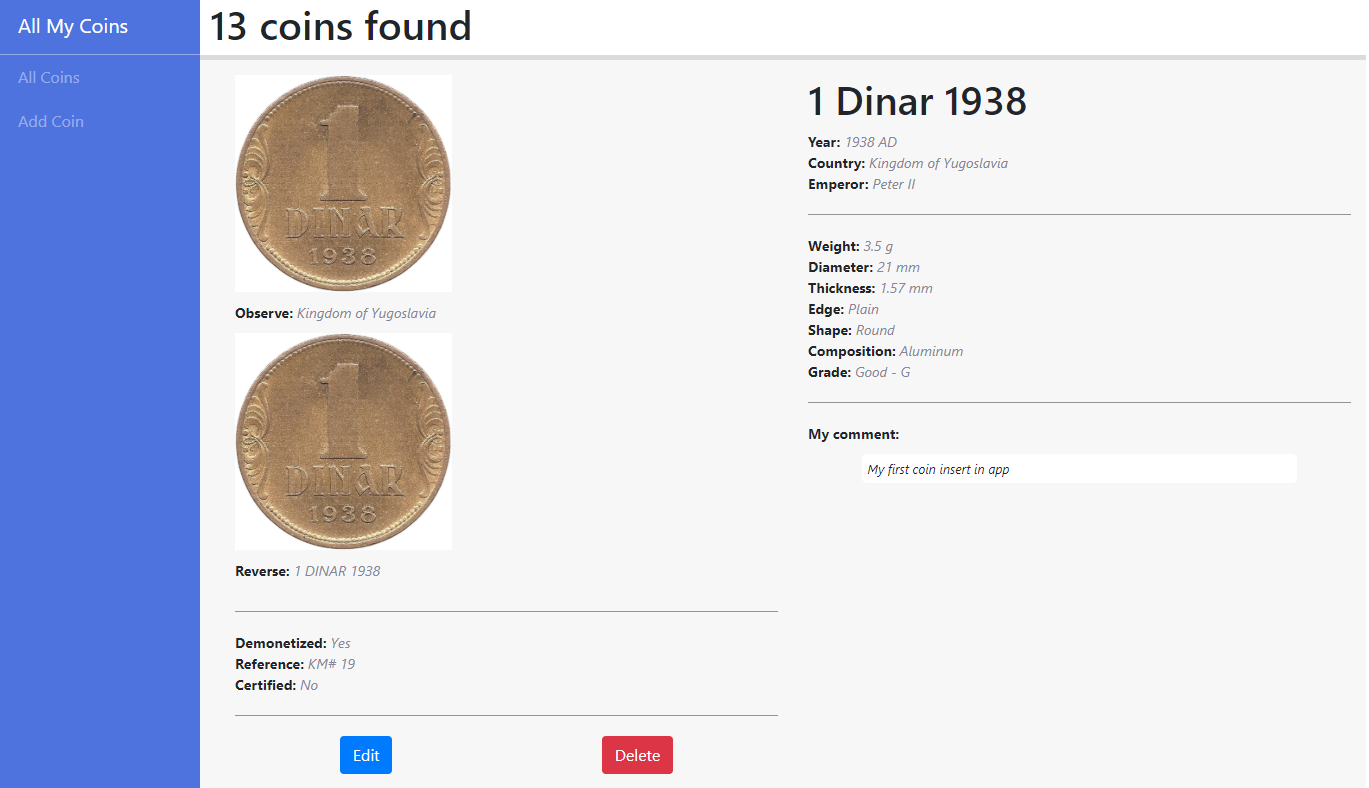
* Dinamičko ispisivanje kovanica, pretraga, sortiranje i filtriranje.



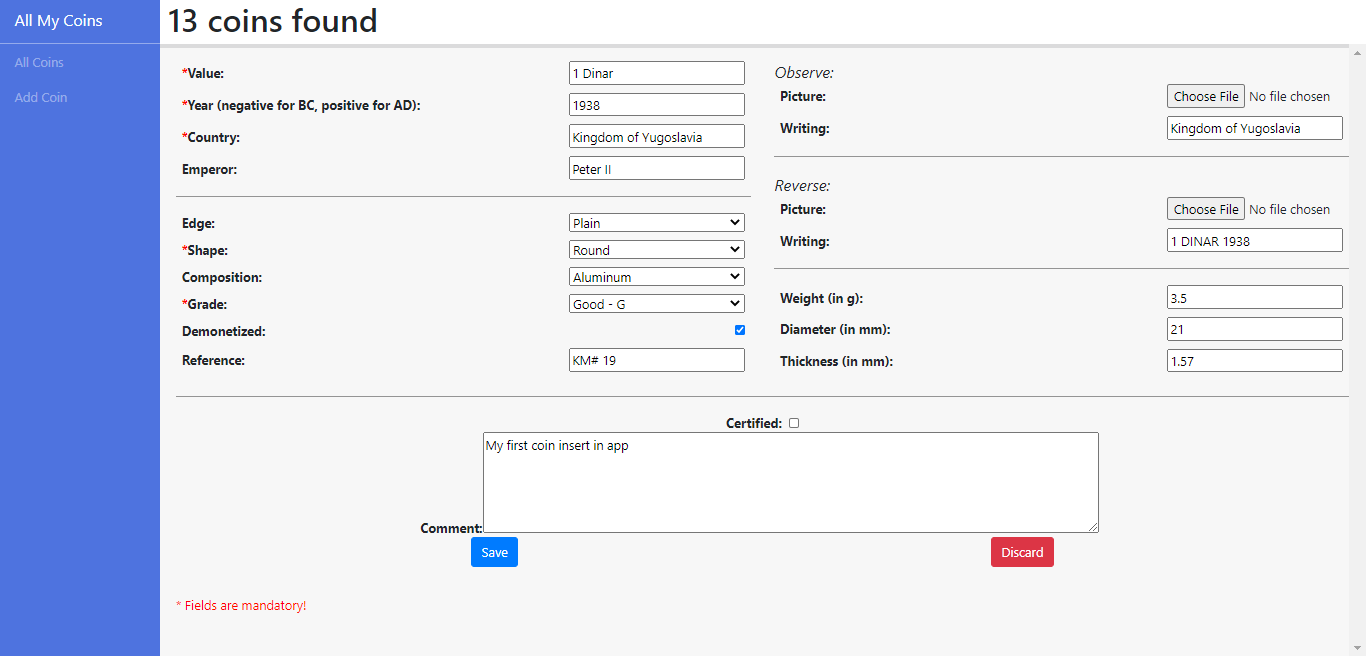
* 404 stranica kada se unese pogrešna adresa u url.



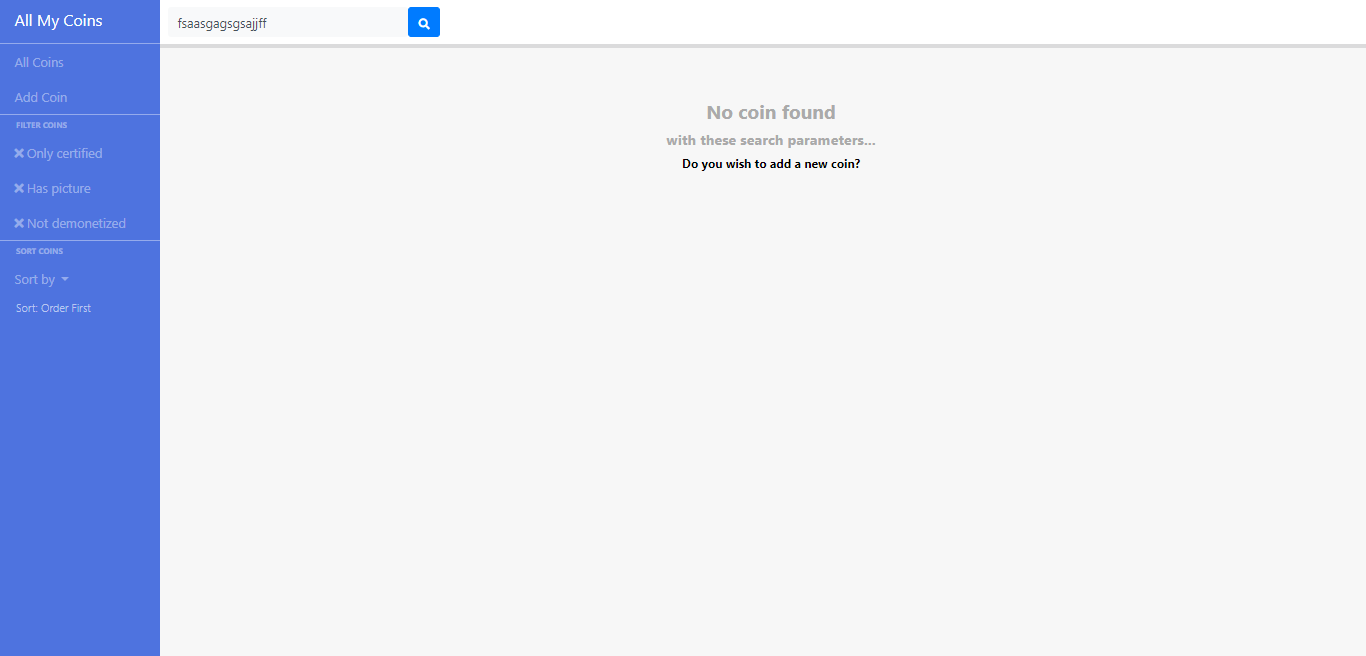
* Dodavanje nove kovanice.



* Prikaz jedne kovanice, sa korišćenjem pipe-ova za prikaz podataka.



* Stranica za edit kovanice.



* Prikaz kada ni jedna kovanica nije pronađena na stranici.

# KODOVI

## styles.scss

/\* You can add global styles to this file, and also import other style files \*/

body {

overflow: hidden;

}

.my-horizontal-line {

margin-bottom: 20px;

margin-top: 20px;

width: 100%;

height: 1px;

border-top: 1px solid rgba(0, 0, 0, 0.4);

}

.my-red-indicator {

color: #ff0000;

}



## app.component.html

<router-outlet></router-outlet>

## app.module.ts

**import** **{** BrowserModule **}** from '@angular/platform-browser'**;**

**import** **{** NgModule **}** from '@angular/core'**;**

**import** **{** ScrollingModule**}** from '@angular/cdk/scrolling'**;**

**import** **{** HttpClientModule **}** from '@angular/common/http'**;**

**import** **{** ReactiveFormsModule **}** from '@angular/forms'**;**

**import** **{** FormsModule **}** from '@angular/forms'**;**

**import** **{** BrowserAnimationsModule **}** from '@angular/platform-browser/animations'**;**

**import** **{** ToastrModule **}** from 'ngx-toastr'**;**

**import** **{** AppRoutingModule **}** from './app-routing.module'**;**

**import** **{** AppComponent **}** from './app.component'**;**

**import** **{** NavigationComponent **}** from './component/fixed/navigation/navigation.component'**;**

**import** **{** SearchCoinsComponent **}** from './component/fixed/search-coins/search-coins.component'**;**

**import** **{** NotFoundComponent **}** from './component/fixed/not-found/not-found.component'**;**

**import** **{** CoinComponent **}** from './component/coin/coin.component'**;**

**import** **{** PreviewCoinComponent **}** from './component/coin/preview-coin/preview-coin.component'**;**

**import** **{** SingleCoinComponent **}** from './component/coin/single-coin/single-coin.component'**;**

**import** **{** EditCoinComponent **}** from './component/coin/edit-coin/edit-coin.component'**;**

**import** **{** AddCoinComponent **}** from './component/coin/add-coin/add-coin.component'**;**

**import** **{** TitleAreaComponent **}** from './component/fixed/title-area/title-area.component'**;**

**import** **{** CustomyearPipe **}** from './customyear.pipe'**;**

**import** **{** CustomvaluePipe **}** from './customvalue.pipe'**;**

**import** **{** NoCoinShowComponent **}** from './component/coin/no-coin-show/no-coin-show.component'**;**

**import** **{** CoinFallinComponent **}** from './component/coin/coin-fallin/coin-fallin.component'**;**

**import** **{** AuthorCoinComponent **}** from './component/coin/author-coin/author-coin.component'**;**

@NgModule**({**

declarations**:** **[**

AppComponent**,**

NavigationComponent**,**

SearchCoinsComponent**,**

NotFoundComponent**,**

CoinComponent**,**

PreviewCoinComponent**,**

SingleCoinComponent**,**

EditCoinComponent**,**

AddCoinComponent**,**

TitleAreaComponent**,**

CustomyearPipe**,**

CustomvaluePipe**,**

NoCoinShowComponent**,**

CoinFallinComponent**,**

AuthorCoinComponent

**],**

imports**:** **[**

BrowserModule**,**

AppRoutingModule**,**

ScrollingModule**,**

HttpClientModule**,**

ReactiveFormsModule**,**

FormsModule**,**

BrowserAnimationsModule**,**

ToastrModule**.**forRoot**()**

**],**

providers**:** **[],**

bootstrap**:** **[**AppComponent**]**

**})**

**export** **class** AppModule **{** **}**



## app.component.ts

**import** **{** Component **}** from '@angular/core'**;**

@Component**({**

selector**:** 'app-root'**,**

templateUrl**:** './app.component.html'**,**

styleUrls**:** **[**'./app.component.scss'**]**

**})**

**export** **class** AppComponent **{**

title **=** 'all-my-coins'**;** **}**



## app-routing.module.ts

**import** **{** SingleCoinComponent **}** from './component/coin/single-coin/single-coin.component'**;**

**import** **{** EditCoinComponent **}** from './component/coin/edit-coin/edit-coin.component'**;**

**import** **{** AddCoinComponent **}** from './component/coin/add-coin/add-coin.component'**;**

**import** **{** NotFoundComponent **}** from './component/fixed/not-found/not-found.component'**;**

**import** **{** CoinComponent **}** from './component/coin/coin.component'**;**

**import** **{** NgModule **}** from '@angular/core'**;**

**import** **{** Routes**,** RouterModule **}** from '@angular/router'**;**

**const** routes**:** Routes **=** **[**

**{** path**:** ''**,** component**:** CoinComponent **},**

**{** path**:** 'coin'**,** children**:** **[**

**{** path**:** 'add'**,** component**:** AddCoinComponent **},**

**{** path**:** 'edit/:id'**,** component**:** EditCoinComponent **},**

**{** path**:** ':id'**,** component**:** SingleCoinComponent **},**

**{** path**:** ''**,** component**:** CoinComponent**,** pathMatch**:** 'full' **}**

**]**

**},**

**{** path**:** 'not-found'**,** component**:** NotFoundComponent **},**

**{** path**:** '\*\*'**,** redirectTo**:** '/not-found' **}**

**];**

@NgModule**({**

imports**:** **[**RouterModule**.**forRoot**(**routes**)],**

exports**:** **[**RouterModule**]**

**})**

**export** **class** AppRoutingModule **{** **}**



## coin.service.ts

**import** **{** Injectable **}** from '@angular/core'**;**

**import** **{** HttpClient**,** HttpHeaders**}** from '@angular/common/http'**;**

**import** **{** BehaviorSubject **}** from 'rxjs'**;**

@Injectable**({**

providedIn**:** 'root'

**})**

**export** **class** CoinService **{**

serviceFilterOrder **=** **{**

certified**:** **false,**

picture**:** **false,**

demonetized**:** **false,**

order**:** 'order-up'**,**

search**:** ''

**};**

**private** messageSource **=** **new** BehaviorSubject**(this.**serviceFilterOrder**);**

currentMessage **=** **this.**messageSource**.**asObservable**();**

header **=** **{**

headers**:** **new** HttpHeaders**({**

'Access-Control-Allow-Origin'**:**'\*'

**})**

**};**

**constructor(private** service**:** HttpClient**)** **{** **}**

getEdges**()** **{**

**var** edges **=** **this.**service**.**get**(**"http://localhost/all-my-coins-php/edges.php"**,** **this.**header**);**

**return** edges**;**

**}**

getShapes**()** **{**

**var** shapes **=** **this.**service**.**get**(**"http://localhost/all-my-coins-php/shapes.php"**,** **this.**header**);**

**return** shapes**;**

**}**

getCompositions**()** **{**

**var** compositions **=** **this.**service**.**get**(**"http://localhost/all-my-coins-php/compositions.php"**,** **this.**header**);**

**return** compositions**;**

**}**

getGrades() {

var grades = this.service.get("http://localhost/all-my-coins-php/grades.php", this.header);

return grades;

}

addCoin(data) {

var response = this.service.post("http://localhost/all-my-coins-php/add\_coin.php", data);

return response;

}

getCoins(filterOrder) {

var location = "http://localhost/all-my-coins-php/get\_coins.php?cert=" + filterOrder.certified + "&pic=" + filterOrder.picture + "&dem=" + filterOrder.demonetized + "&srtx=" + filterOrder.search + "&order=" + filterOrder.order;

var coins = this.service.get(location, this.header);

return coins;

}

getCoin(id) {

var coin = this.service.get("http://localhost/all-my-coins-php/get\_coin.php?id="+id, this.header);

return coin;

}

getEditCoin(id) {

var coin = this.service.get("http://localhost/all-my-coins-php/get\_edit\_coin.php?id="+id, this.header);

return coin;

}

editCoin(data, id) {

var response = this.service.post("http://localhost/all-my-coins-php/edit\_coin.php?id="+id, data);

return response;

}

deleteCoin(id) {

var response = this.service.get("http://localhost/all-my-coins-php/delete\_coin.php?id="+id, this.header);

return response;

}

coinCount() {

var num = this.service.get("http://localhost/all-my-coins-php/coin\_number.php", this.header);

return num;

}

setFilterOrder(filterOrder) {

this.serviceFilterOrder.certified = filterOrder.certified;

this.serviceFilterOrder.demonetized = filterOrder.demonetized;

this.serviceFilterOrder.order = filterOrder.order;

this.serviceFilterOrder.picture = filterOrder.picture;

this.messageSource.next(this.serviceFilterOrder);

}

setSearchText(searchText) {

this.serviceFilterOrder.search = searchText;

this.messageSource.next(this.serviceFilterOrder);

}

}



## customvalue.pipe.ts

**import** **{** Pipe**,** PipeTransform **}** from '@angular/core'**;**

@Pipe**({**

name**:** 'customvalue'

**})**

**export** **class** CustomvaluePipe **implements** PipeTransform **{**

transform**(**value**:** number**,** type**:** string**)** **{**

**if** **(**value **<=** 0**)** **{**

**return** 'Wrong on no value inserted'**;**

**}** **else** **{**

**}**

**switch(**type**)** **{**

**case** 'g'**:**

**if** **(**value**/**1000 **>=** 1**)** **{**

**return** **(**value**/**1000**)** **+** ' kg'**;**

**}** **else** **{**

**return** value **+** ' g'**;**

**}**

**case** 'm'**:**

**if** **(**value**/**1000 **>=** 1**)** **{**

**return** **(**value**/**1000**)** **+** ' m'**;**

**}** **else** **{**

**return** value **+** ' mm'**;**

**}**

**}**

**}**

**}**



## customyear.pipe.ts

**import** **{** Pipe**,** PipeTransform **}** from '@angular/core'**;**

@Pipe**({**

name**:** 'customyear'

**})**

**export** **class** CustomyearPipe **implements** PipeTransform **{**

transform**(**value**:** number**)** **{**

**if** **(**value **>=** 0**)** **{**

**return** value **+** ' AD'**;**

**}** **else** **{**

**return** value**\*(-**1**)** **+** ' BC'**;**

**}**

**}**

**}**



## navigation.component.html

<nav class=**"navbar-primary"**>

<ul class=**"navbar-primary-menu"**>

<a class=**"navbar-brand"** routerLink=**"/"**>**All My Coins**</a>

<li class=**"nav-item"**><a routerLink=**"/coin"**><span class=**"glyphicon glyphicon-list-alt"**></span><span class=**"nav-label"**>**All Coins**</span></a></li>

<li class=**"nav-item"**><a routerLink=**"/coin/add"**><span class=**"glyphicon glyphicon-envelope"**></span><span class=**"nav-label"**>**Add Coin**</span></a></li>

<ng-container \*ngIf="shouldShowFilterSort">

<div class=**"area\_heading"**>**Filter Coins**</div>

<li class=**"nav-item"**><a (click)=**"certClick()"**><span class=**"fas"** [ngClass]=**"onlyCertified ? 'fa-check' : 'fa-times'"**></span>*&nbsp;*<span class=**"nav-label"**>**Only certified**</span></a></li>

<li class=**"nav-item"**><a (click)=**"pictClick()"**><span class=**"fas"** [ngClass]=**"hasPicture ? 'fa-check' : 'fa-times'"**></span>*&nbsp;*<span class=**"nav-label"**>**Has picture**</span></a></li>

<li class=**"nav-item"**><a (click)=**"demoClick()"**><span class=**"fas"** [ngClass]=**"notDemonetized ? 'fa-check' : 'fa-times'"**></span>*&nbsp;*<span class=**"nav-label"**>**Not demonetized**</span></a></li>

<div class=**"area\_heading"**>**Sort Coins**</div>

<li class=**"nav-item dropdown"**>

<a class=**"nav-link dropdown-toggle"** id=**"navbarDropdownMenuLink"** data-toggle=**"dropdown"** aria-haspopup=**"true"** aria-expanded=**"false"**>

**Sort by**

</a>

<div class=**"my-sort-show"**>**Sort: {{orderValue == "order-up" ? "Order First" : (orderValue == "order-down" ? "Order Last" : (orderValue == "year-up" ? "Year First": "Year Last"))}}**</div>

<div class=**"dropdown-menu"** aria-labelledby=**"navbarDropdownMenuLink"**>

<a class=**"dropdown-item"** (click)=**"orderValueChange('order-up')"**>**Order** <span class=**"fas fa-arrow-up"**></span></a>

<a class=**"dropdown-item"** (click)=**"orderValueChange('order-down')"**>**Order** <span class=**"fas fa-arrow-down"**></span></a>

<a class=**"dropdown-item"** (click)=**"orderValueChange('year-up')"**>**Year** <span class=**"fas fa-arrow-up"**></span></a>

<a class=**"dropdown-item"** (click)=**"orderValueChange('year-down')"**>**Year** <span class=**"fas fa-arrow-down"**></span></a>

</div>

</li>

</ng-container>

</ul>

</nav>



## navigation.component.scss

.navbar-primary {

background-color: #4e73df;

bottom: 0;

left: 0;

width: 200px;

min-width: 200px;

position: absolute;

top: 0;

.navbar-primary-menu {

margin:0;

padding:0;

list-style: none;

.navbar-brand {

padding: 10px 18px;

color: #fff;

border-bottom: 1px solid rgba(255,255,255,.4);

min-height: 55px;

width: 100%;

}

.area\_heading {

color: rgba(255,255,255,.4);

text-transform: uppercase;

cursor: default;

font-weight: bold;

font-size: 10px;

padding: 5px 20px;

border-top: 1px solid rgba(255,255,255,.4);

}

.my-sort-show {

color: rgba(255,255,255,.6);

cursor: default;

font-weight: lighter;

font-size: 14px;

padding: 5px 20px;

}

li {

margin:0;

padding:0;

a {

display: block;

color: rgba(255,255,255,.4);

padding: 10px 18px;

cursor: pointer;

user-select: none;

&:hover {

color: #fff;

text-decoration: none;

}

}

}

.dropdown-menu a {

color: #000;

&:hover {

color: #4e73df;

}

&:active {

color: #000;

}

}

}

}

@media screen and (max-width: 500px) {

.navbar-primary {

width: 100px;

min-width: 100px;

.navbar-primary-menu {

.area\_heading, li a, .my-sort-show {

padding: 2px;

}

.navbar-brand {

display: none;

}

}

}

}



## navigation.component.ts

**import** **{** CoinService **}** from './../../../coin.service'**;**

**import** **{** Component**,** OnInit**,** Input **}** from '@angular/core'**;**

@Component**({**

selector**:** 'app-navigation'**,**

templateUrl**:** './navigation.component.html'**,**

styleUrls**:** **[**'./navigation.component.scss'**]**

**})**

**export** **class** NavigationComponent **implements** OnInit **{**

onlyCertified **=** **false;**

hasPicture **=** **false;**

notDemonetized **=** **false;**

orderValue **=** 'order-up'**;**

@Input**()**

shouldShowFilterSort**:** string**;**

**constructor(private** coinService**:** CoinService**)** **{** **}**

ngOnInit**()** **{**

**this.**sendValues**();**

**}**

certClick**()** **{**

**this.**onlyCertified **=** **!this.**onlyCertified**;**

**this.**sendValues**();**

**}**

pictClick**()** **{**

**this.**hasPicture **=** **!this.**hasPicture**;**

**this.**sendValues**();**

**}**

demoClick**()** **{**

**this.**notDemonetized **=** **!this.**notDemonetized**;**

**this.**sendValues**();**

**}**

orderValueChange**(**value**)** **{**

**this.**orderValue **=** value**;**

**this.**sendValues**();**

**}**

sendValues() {

var values = {

certified: this.onlyCertified,

picture: this.hasPicture,

demonetized: this.notDemonetized,

order: this.orderValue

};

this.coinService.setFilterOrder(values);

}

}



## not-found.component.scss

.my-lost-image {

background-image: url('lost-coins.jpg');

background-attachment: fixed ;

background-repeat: no-repeat;

background-size: cover;

background-position: center center;

}

.my-hr {

padding-top: 19px;

border-top: 1px solid rgba(0, 0, 0, 0.4);

}

.my-flex-image {

display: flex;

align-items: center;

justify-content: center;

}



## not-found.component.ts

**import** **{** Component**,** OnInit **}** from '@angular/core'**;**

@Component**({**

selector**:** 'app-not-found'**,**

templateUrl**:** './not-found.component.html'**,**

styleUrls**:** **[**'./not-found.component.scss'**]**

**})**

**export** **class** NotFoundComponent **implements** OnInit **{**

secondPassed **=** **false;**

waitTime **=** 30 **\*** 1000**;**

**constructor()** **{** **}**

ngOnInit**()** **{**

**setTimeout(()=>{**

**this.**secondPassed **=** **true;**

**},** **this.**waitTime**);**

**}**

**}**



## not-found.component.html

<div class=**"container"**>

<div class=**"row justify-content-center"**>

<div class=**"col-xl-10 col-lg-12 col-md-9"**>

<div class=**"card o-hidden border-0 shadow-lg my-5"**>

<div class=**"card-body p-0"**>

<div class=**"row"**>

<div class=**"col-lg-6 d-none d-lg-block my-lost-image"**></div>

<div class=**"col-lg-6"**>

<div class=**"p-5"**>

<div class=**"text-center"**>

<h1 class=**"h1 text-gray-900 mb-4"**>**404**</h1>

</div>

<div class=**"text-center"**>

<h1 class=**"h6 text-gray-900 mb-4"**>**Page not found**</h1>

</div>

<div class=**"text-center my-hr"**>

<a class=**"small"** routerLink=**"/coin"**>**Back to home page!**</a>

</div>

<div class=**"my-flex-image"** \*ngIf=**"secondPassed"**>

<app-author-coin></app-author-coin>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>



## search-coins.component.scss

.my-container {

right: 0;

left: 200px;

position: absolute;

top: 0;

height: 55px;

min-height: 55px;

box-shadow: 0 5px 0 rgba(58, 59, 69, 0.15);

z-index: -1;

display: flex;

justify-content: space-between;

.my-container-item {

margin-left: 10px;

margin-right: 10px;

height: 55px;

min-height: 55px;

display: flex;

align-items: center;

input {

width: 300px;

font-size: 16px;

}

}

}

@media screen and (max-width: 670px) {

.my-container .my-container-item input {

width: 80%;

}

}

@media screen and (max-width: 500px) {

.my-container {

left: 100px;

}

}



## search-coins.component.ts

**import** **{** CoinService **}** from './../../../coin.service'**;**

**import** **{** Component**,** OnInit **}** from '@angular/core'**;**

**import** **{**FormsModule**}** from '@angular/forms'**;**

@Component**({**

selector**:** 'app-search-coins'**,**

templateUrl**:** './search-coins.component.html'**,**

styleUrls**:** **[**'./search-coins.component.scss'**]**

**})**

**export** **class** SearchCoinsComponent **implements** OnInit **{**

searchText **=** ''**;**

**constructor(private** coinService**:** CoinService**)** **{** **}**

ngOnInit**()** **{**

**}**

searchSubmit**()** **{**

**this.**coinService**.**setSearchText**(this.**searchText**);**

**}**

searchTextChange**(event)** **{**

**if** **(event.**key **!==** "Enter"**)** **{** **return;** **}**

**this.**searchSubmit**();**

**}**

**}**



## search-coins.component.html

<div class=**"my-container"**>

<div class=**"my-container-item"**>

<input type=**"text"** class=**"form-control bg-light border-0 small"** placeholder=**"Search coins..."** aria-label=**"Search"** [(ngModel)]=**"searchText"** (keyup)=**"searchTextChange($event)"** />

<button class=**"btn btn-primary"** type=**"button"** (click)=**"searchSubmit();"**><i class=**"fas fa-search fa-sm"**></i></button>

</div>

</div>

## title-area.component.scss

.my-container {

right: 0;

left: 200px;

position: absolute;

top: 0;

height: 55px;

min-height: 55px;

box-shadow: 0 5px 0 rgba(58, 59, 69, 0.15);

z-index: -1;

display: flex;

justify-content: space-between;

.my-container-item {

margin-left: 10px;

margin-right: 10px;

height: 55px;

min-height: 55px;

display: flex;

align-items: center;

}

}

@media screen and (max-width: 500px) {

.my-container {

left: 100px;

}

}



## title-area.component.ts

**import** **{** CoinService **}** from './../../../coin.service'**;**

**import** **{** Subscription **}** from 'rxjs'**;**

**import** **{** Component**,** OnInit **}** from '@angular/core'**;**

@Component**({**

selector**:** 'app-title-area'**,**

templateUrl**:** './title-area.component.html'**,**

styleUrls**:** **[**'./title-area.component.scss'**]**

**})**

**export** **class** TitleAreaComponent **implements** OnInit **{**

coinNumber**;**

subscription**:** Subscription **=** **new** Subscription**();**

**constructor(private** coinService**:** CoinService**)** **{** **}**

ngOnInit**()** **{**

**this.**subscription**.**add**(this.**coinService**.**coinCount**().**subscribe**(**

response **=>** **{** **this.**coinNumber **=** response**;** **},**

error **=>** **{** **this.**coinNumber **=** **{**number**:** 0**};** **}**

**))**

**}**

**}**



## title-area.component.html

<div class=**"my-container"**>

<div class=**"my-container-item"**>

<h1>**{{coinNumber ? coinNumber.number : '0'}} coins found**</h1>

</div>

</div>



## coin.component.scss

.my-row {

position: absolute;

top: 55px;

left: 200px;

right: 0;

bottom: 0;

margin: 0;

background-color: rgba(0, 0, 0, 0.03);

padding: 20px;

padding-right: 0;

cdk-virtual-scroll-viewport, app-no-coin-show {

width: 100%;

height: 100%;

}

}

@media screen and (max-width: 500px) {

.my-row {

left: 100px;

}

}



## coin.component.ts

**import** **{** ToastrService **}** from 'ngx-toastr'**;**

**import** **{** Subscription **}** from 'rxjs'**;**

**import** **{** CoinService **}** from './../../coin.service'**;**

**import** **{** Component**,** OnInit **}** from '@angular/core'**;**

@Component**({**

selector**:** 'app-coin'**,**

templateUrl**:** './coin.component.html'**,**

styleUrls**:** **[**'./coin.component.scss'**]**

**})**

**export** **class** CoinComponent **implements** OnInit **{**

subscription**:** Subscription **=** **new** Subscription**();**

coins**;**

filterOrder**;**

**private** waitTime **=** 3000**;**

**constructor(private** coinService**:** CoinService**,** **private** toastr**:** ToastrService**)** **{** **}**

ngOnInit**()** **{**

**this.**coinService**.**currentMessage**.**subscribe**(**data **=>** **{** **this.**filterOrder **=** data**;** **this.**getCoins**();** **});**

**}**

errorFunc**(**title**,** message**)** **{**

**this.**toastr**.**error**(**message**,** title **,** **{**

timeOut **:** **this.**waitTime

**});**

**}**

getCoins**()** **{**

**this.**subscription**.**add**(this.**coinService**.**getCoins**(this.**filterOrder**).**subscribe**(**

response **=>** **{**

**this.**coins **=** response**;**

**},**

error **=>** **{** **this.**errorFunc**(**"Error"**,** "Unexpected error occured. Please try again later."**);** **}**

**));**

**}**

**}**



## coin.component.html

<app-navigation [shouldShowFilterSort]="true"></app-navigation>

<app-search-coins></app-search-coins>

<div class=**"row my-row"**>

<cdk-virtual-scroll-viewport itemSize="5" \*ngIf="coins && coins.length > **0">**

<app-preview-coin \*cdkVirtualFor="let coinEl of coins" [coin]="coinEl">

</app-preview-coin>

</cdk-virtual-scroll-viewport>

<app-no-coin-show \*ngIf="coins && coins.length == 0"></app-no-coin-show>

</div>



## logout.php 🡪 brisanje sesije korisnika

<?php

**session\_start**();

**if**(**isset**($\_SESSION['user'])) {

**unset**($\_SESSION['user']);

**unset**($\_SESSION['user\_rights\_id']);

**session\_destroy**();

}

**echo** **true**;

?>



## add-coin.component.scss

.my-row {

position: absolute;

top: 55px;

left: 200px;

right: 0;

bottom: 0;

margin: 0;

background-color: rgba(0, 0, 0, 0.03);

padding: 20px;

padding-right: 0;

overflow-x: hidden;

overflow-y: scroll;

form {

width: 100%;

display: flex;

flex-direction: column;

.my-100-div {

width: 100%;

display: flex;

align-items: center;

justify-content: space-between;

&.my-100-div-column {

flex-direction: column;

label {

margin: 0;

font-weight: bold;

}

input[type="text"], input[type="number"], input[type="file"], select {

width: 220px;

min-width: 220px;

}

textarea {

width: 90%;

}

.my-button-holder {

width: 90%;

display: flex;

justify-content: space-around;

}

}

.my-part-holder {

width: 49%;

.my-sub-title {

font-size: 20px;

font-style: italic;

}

.my-item-holder {

width: 98%;

margin-left: 1%;

margin-bottom: 10px;

display: flex;

justify-content: space-between;

align-items: center;

label {

margin: 0;

font-weight: bold;

}

input[type="text"], input[type="number"], input[type="file"], select {

width: 220px;

min-width: 220px;

}

}

}

}

}

}

@media screen and (max-width: 930px) {

.my-row form .my-100-div {

flex-direction: column;

.my-part-holder {

width: 98%;

}

}

}

@media screen and (max-width: 500px) {

.my-row {

left: 100px;

}

}



## add-coin.component.ts

**import** **{** Router **}** from '@angular/router'**;**

**import** **{** CoinService **}** from './../../../coin.service'**;**

**import** **{** Component**,** OnInit**,** ChangeDetectorRef **}** from '@angular/core'**;**

**import** **{** Subscription **}** from 'rxjs'**;**

**import** **{** FormBuilder**,** FormControl**,** Validators **}** from '@angular/forms'**;**

**import** **{** ToastrService **}** from 'ngx-toastr'**;**

@Component**({**

selector**:** 'app-add-coin'**,**

templateUrl**:** './add-coin.component.html'**,**

styleUrls**:** **[**'./add-coin.component.scss'**]**

**})**

**export** **class** AddCoinComponent **implements** OnInit **{**

subscription**:** Subscription **=** **new** Subscription**();**

edges**;**

shapes**;**

compositions**;**

grades**;**

valueReg **=** **/^[0-9]+\s[A-z]+/;**

addCoinForm **=** **this.**formBuilder**.**group**({**

coinValue**:** **[**''**,** Validators**.**pattern**(this.**valueReg**)],**

coinYear**:** **[**''**],**

coinCountry**:** **[**''**],**

coinEmperor**:** **[**''**],**

coinEdge**:** **[**'1'**],**

coinShape**:** **[**'1'**],**

coinComposition**:** **[**'1'**],**

coinGrade**:** **[**'1'**],**

coinDemonetized**:** **[**''**],**

coinReference**:** **[**''**],**

coinObservePic**:** **new** FormControl**(null),**

coinObserveWriting**:** **[**''**],**

coinReversePic**:** **new** FormControl**(null),**

coinReverseWriting**:** **[**''**],**

coinWeight**:** **[**''**,** Validators**.**min**(**0.01**)],**

coinDiameter**:** **[**''**,** Validators**.**min**(**0.01**)],**

coinThickness**:** **[**''**,** Validators**.**min**(**0.01**)],**

coinCertified**:** **[**''**],**

coinCertification**:** **[**''**],**

coinComment**:** **[**''**]**

**});**

**private** waitTime **=** 3000**;**

**constructor(private** coinService**:** CoinService**,** **private** formBuilder**:** FormBuilder**,** **private** cd**:** ChangeDetectorRef**,** **private** toastr**:** ToastrService**,** **private** router**:** Router**)** **{** **}**

ngOnInit**()** **{**

**this.**subscription**.**add**(this.**coinService**.**getEdges**().**subscribe**(**

response **=>** **{** **this.**edges **=** response**;** **},**

error **=>** **{** **this.**errorFunc**(**"Error"**,** "Unexpected error occured. Please try again later."**);** **}**

**));**

**this.**subscription**.**add**(this.**coinService**.**getShapes**().**subscribe**(**

response **=>** **{** **this.**shapes **=** response**;** **},**

error **=>** **{** **this.**errorFunc**(**"Error"**,** "Unexpected error occured. Please try again later."**);** **}**

**));**

**this.**subscription**.**add**(this.**coinService**.**getCompositions**().**subscribe**(**

response **=>** **{** **this.**compositions **=** response**;** **},**

error **=>** **{** **this.**errorFunc**(**"Error"**,** "Unexpected error occured. Please try again later."**);** **}**

**));**

**this.**subscription**.**add**(this.**coinService**.**getGrades**().**subscribe**(**

response **=>** **{** **this.**grades **=** response**;** **},**

error **=>** **{** **this.**errorFunc**(**"Error"**,** "Unexpected error occured. Please try again later."**);** **}**

**));**

**}**

coinFormDiscard**()** **{**

**this.**router**.navigate([**'/coin'**]);**

**}**

coinFormSubmit**()** **{**

**if** **(this.**addCoinForm**.status** **===** 'INVALID'**)** **{**

**this.**errorFunc**(**"Could not submit"**,** "Please check your input and try again."**);**

**return;**

**}**

**this.**subscription**.**add**(this.**coinService**.**addCoin**(this.**addCoinForm**.**value**).**subscribe**(**

response **=>** **{**

**if** **(**response **==** "OK"**)** **{** **this.**successFunc**(**"Success"**,** "Coin added successfully. Redirecting you to home page."**);** **}**

**else** **{** **this.**errorFunc**(**"Error"**,** "Unexpected error occured. Please try again later."**);** **}**

**},**

error **=>** **{** **this.**errorFunc**(**"Error"**,** "Unexpected error occured. Please try again later."**);** **}**

**));**

**}**

onObservePicChange**(event)** **{**

let reader = new FileReader();

if(event.target.files && event.target.files.length) {

const [file] = event.target.files;

reader.readAsDataURL(file);

reader.onload = () => {

this.addCoinForm.patchValue({

coinObservePic: reader.result

});

this.cd.markForCheck();

};

}

}

onReversePicChange(event) {

let reader = new FileReader();

if(event.target.files && event.target.files.length) {

const [file] = event.target.files;

reader.readAsDataURL(file);

reader.onload = () => {

this.addCoinForm.patchValue({

coinReversePic: reader.result

});

this.cd.markForCheck();

};

}

}

successFunc(title, message) {

this.toastr.success(message, title , {

timeOut : this.waitTime

});

setTimeout(()=>{

this.router.navigate(['/coin']);

}, this.waitTime);

}

errorFunc(title, message) {

this.toastr.error(message, title , {

timeOut : this.waitTime

});

}

}



## add-coin.component.html

<app-navigation [shouldShowFilterSort]="false"></app-navigation>

<app-title-area></app-title-area>

<div class=**"row my-row"**>

<form [formGroup]=**"addCoinForm"**>

<div class=**"my-100-div"**>

<div class=**"my-part-holder"**>

<div class=**"my-item-holder"**>

<label for=**"value"**><span class=**"my-red-indicator"**>**\***</span>**Value:**</label>

<input type=**"text"** name=**"value"** placeholder=**"1 Dinar"** formControlName=**"coinValue"** required/>

</div>

<div class=**"my-item-holder"**>

<label for=**"year"**><span class=**"my-red-indicator"**>**\***</span>**Year (negative for BC, positive for AD):**</label>

<input type=**"number"** name=**"year"** placeholder=**"1925"** formControlName=**"coinYear"** required/>

</div>

<div class=**"my-item-holder"**>

<label for=**"country"**><span class=**"my-red-indicator"**>**\***</span>**Country:**</label>

<input type=**"text"** name=**"country"** placeholder=**"Kingdom of Yugoslavia"** formControlName=**"coinCountry"** required/>

</div>

<div class=**"my-item-holder"**>

<label for=**"emperor"**>**Emperor:**</label>

<input type=**"text"** name=**"emperor"** placeholder=**"King Alexander 1"** formControlName=**"coinEmperor"** />

</div>

<div class=**"my-horizontal-line"**></div>

<div class=**"my-item-holder"**>

<label for=**"edge"**>**Edge:**</label>

<select name=**"edge"** formControlName=**"coinEdge"**>

<option \*ngFor=**"let edge of edges"** [value]=**"edge.id"**>**{{edge.name}}**</option>

</select>

</div>

<div class=**"my-item-holder"**>

<label for=**"shape"**><span class=**"my-red-indicator"**>**\***</span>**Shape:**</label>

<select name=**"shape"** formControlName=**"coinShape"**>

<option \*ngFor=**"let shape of shapes"** [value]=**"shape.id"**>**{{shape.name}}**</option>

</select>

</div>

<div class=**"my-item-holder"**>

<label for=**"composition"**>**Composition:**</label>

**<select name="composition" formControlName="coinComposition">**

**<option \*ngFor="let composition of compositions" [value]="composition.id">{{composition.name}}</option>**

**</select>**

**</div>**

**<div class="my-item-holder">**

**<label for="grade"><span class="my-red-indicator">\*</span>Grade:</label>**

**<select name="grade" formControlName="coinGrade">**

**<option \*ngFor="let grade of grades" [value]="grade.id">{{grade.name}}</option>**

**</select>**

**</div>**

**<div class="my-item-holder">**

**<label for="demonetized">Demonetized:</label>**

**<input type="checkbox" name="demonetized" formControlName="coinDemonetized" />**

**</div>**

**<div class="my-item-holder">**

**<label for="reference">Reference:</label>**

**<input type="text" name="reference" placeholder="123$21" formControlName="coinReference" />**

**</div>**

**</div>**

**<div class="my-part-holder">**

**<div>**

**<div class="my-sub-title">Observe:</div>**

**<div class="my-item-holder">**

**<label for="observe\_pic">Picture:</label>**

**<input type="file" name="observe\_pic" (change)="onObservePicChange($event)" accept="image/\*" />**

**</div>**

**<div class="my-item-holder">**

**<label for="observe\_writing">Writing:</label>**

**<input type="text" name="observe\_writing" placeholder="Aleksandar 1. Kralj SHS" formControlName="coinObserveWriting" />**

**</div>**

**<div class="my-horizontal-line"></div>**

**<div class="my-sub-title">Reverse:</div>**

**<div class="my-item-holder">**

**<label for="reverse\_pic">Picture:</label>**

**<input type="file" name="reverse\_pic" (change)="onReversePicChange($event)" accept="image/\*" />**

**</div>**

**<div class="my-item-holder">**

**<label for="reverse\_writing">Writing:</label>**

**<input type="text" name="reverse\_writing" placeholder="1 Dinar" formControlName="coinReverseWriting" />**

**</div>**

**<div class="my-horizontal-line"></div>**

**</div>**

**<div class="my-item-holder">**

**<label for="weight">Weight (in g):</label>**

**<input type="number" name="weight" placeholder="4.92" formControlName="coinWeight" />**

**</div>**

**<div class="my-item-holder">**

**<label for="diameter">Diameter (in mm):</label>**

**<input type="number" name="diameter" placeholder="23" formControlName="coinDiameter" />**

**</div>**

**<div class="my-item-holder">**

**<label for="thickness">Thickness (in mm):</label>**

**<input type="number" name="thickness" placeholder="1.8" formControlName="coinThickness" />**

**</div>**

**</div>**

**</div>**

**<div class="my-horizontal-line my-100-div"></div>**

**<div class="my-100-div my-100-div-column">**

**<div>**

**<label for="certified">Certified: &nbsp;</label>**

**<input type="checkbox" name="certified" formControlName="coinCertified" />**

**</div>**

**<div \*ngIf="addCoinForm.value.coinCertified">**

**<label for="certification">Input certification: &nbsp;</label>**

**<input type="text" name="certification" placeholder="321%31" formControlName="coinCertification" />**

**</div>**

**<div>**

**<label for="comment">Comment:</label>**

**<textarea name="comment" cols="100" rows="5" formControlName="coinComment"></textarea>**

**</div>**

**<div class="my-button-holder">**

**<button type="button" class="btn btn-primary" (click)="coinFormSubmit()">Save</button>**

**<button type="button" class="btn btn-danger" (click)="coinFormDiscard()">Discard</button>**

**</div>**

**</div>**

**</form>**

**<span class="my-red-indicator">\* Fields are mandatory!</span>**

**</div>**



## author-coin.component.scss

img {

width: 150px;

margin-top: 30px;

cursor: pointer;

}

p {

margin-top: 10px;

font-size: 10px;

}



## author-coin.component.ts

**import** **{** Component**,** OnInit **}** from '@angular/core'**;**

**import** **{** trigger**,** transition**,** style**,** animate**,** state **}** from '@angular/animations'**;**

@Component**({**

selector**:** 'app-author-coin'**,**

templateUrl**:** './author-coin.component.html'**,**

styleUrls**:** **[**'./author-coin.component.scss'**],**

animations**:** **[**

trigger**(**'rotatedState'**,** **[**

state**(**'default'**,** style**({** transform**:** 'rotate(0)' **})),**

state**(**'rotated'**,** style**({** transform**:** 'rotate(720deg)' **})),**

transition**(**'rotated => default'**,** animate**(**'1000ms ease-in'**)),**

transition**(**'default => rotated'**,** animate**(**'1000ms ease-in'**))**

**])**

**]**

**})**

**export** **class** AuthorCoinComponent **implements** OnInit **{**

state**:** string **=** 'default'**;**

**constructor()** **{** **}**

ngOnInit**()** **{**

**}**

rotate**()** **{**

**this.**state **=** **(this.**state **===** 'default' **?** 'rotated' **:** 'default'**);**

**}**

**}**



## author-coin.component.html

<img src=**"assets/coins/author.png"** alt=**"Author"** (click)=**"rotate()"** [@rotatedState]=**'state'** />

<p class=**"text-center"**>**Click image to rotate!**</p>



## coin-fallin.component.html

<img class=**"coin"** src=**"assets/coins/bitcoin.png"** alt=**"Coin"** @slideInOut />



## coin-fallin.component.scss

.coin {

width: 70px;

position: absolute;

top: -300px;

left: 50%;

z-index: 50;

}



## coin-fallin.component.ts

**import** **{** Component**,** OnInit **}** from '@angular/core'**;**

**import** **{** trigger**,** transition**,** style**,** animate **}** from '@angular/animations'**;**

@Component**({**

selector**:** 'app-coin-fallin'**,**

templateUrl**:** './coin-fallin.component.html'**,**

styleUrls**:** **[**'./coin-fallin.component.scss'**],**

animations**:** **[**

trigger**(**'slideInOut'**,** **[**

transition**(**"void => \*"**,** **[**

style**({**'bottom'**:** '0px'**,** 'top'**:** '900px'**}),**

animate**(**3000**)**

**])**

**]),**

**]**

**})**

**export** **class** CoinFallinComponent **implements** OnInit **{**

**constructor()** **{** **}**

ngOnInit**():** **void** **{**

**}**

**}**



## edit-coin.component.html

<app-navigation [shouldShowFilterSort]="false"></app-navigation>

<app-title-area></app-title-area>

<div class=**"row my-row"**>

<form [formGroup]=**"editCoinForm"**>

<div class=**"my-100-div"**>

<div class=**"my-part-holder"**>

<div class=**"my-item-holder"**>

<label for=**"value"**><span class=**"my-red-indicator"**>**\***</span>**Value:**</label>

<input type=**"text"** name=**"value"** placeholder=**"1 Dinar"** formControlName=**"coinValue"** required/>

</div>

<div class=**"my-item-holder"**>

<label for=**"year"**><span class=**"my-red-indicator"**>**\***</span>**Year (negative for BC, positive for AD):**</label>

<input type=**"number"** name=**"year"** placeholder=**"1925"** formControlName=**"coinYear"** required/>

</div>

<div class=**"my-item-holder"**>

<label for=**"country"**><span class=**"my-red-indicator"**>**\***</span>**Country:**</label>

<input type=**"text"** name=**"country"** placeholder=**"Kingdom of Yugoslavia"** formControlName=**"coinCountry"** required/>

</div>

<div class=**"my-item-holder"**>

<label for=**"emperor"**>**Emperor:**</label>

<input type=**"text"** name=**"emperor"** placeholder=**"King Alexander 1"** formControlName=**"coinEmperor"** />

</div>

<div class=**"my-horizontal-line"**></div>

<div class=**"my-item-holder"**>

<label for=**"edge"**>**Edge:**</label>

<select name=**"edge"** formControlName=**"coinEdge"**>

<option \*ngFor=**"let edge of edges"** [value]=**"edge.id"**>**{{edge.name}}**</option>

</select>

</div>

<div class=**"my-item-holder"**>

<label for=**"shape"**><span class=**"my-red-indicator"**>**\***</span>**Shape:**</label>

<select name=**"shape"** formControlName=**"coinShape"**>

<option \*ngFor=**"let shape of shapes"** [value]=**"shape.id"**>**{{shape.name}}**</option>

</select>

</div>

<div class=**"my-item-holder"**>

<label for=**"composition"**>**Composition:**</label>

**<select name="composition" formControlName="coinComposition">**

**<option \*ngFor="let composition of compositions" [value]="composition.id">{{composition.name}}</option>**

**</select>**

**</div>**

**<div class="my-item-holder">**

**<label for="grade"><span class="my-red-indicator">\*</span>Grade:</label>**

**<select name="grade" formControlName="coinGrade">**

**<option \*ngFor="let grade of grades" [value]="grade.id">{{grade.name}}</option>**

**</select>**

**</div>**

**<div class="my-item-holder">**

**<label for="demonetized">Demonetized:</label>**

**<input type="checkbox" name="demonetized" formControlName="coinDemonetized" />**

**</div>**

**<div class="my-item-holder">**

**<label for="reference">Reference:</label>**

**<input type="text" name="reference" placeholder="123$21" formControlName="coinReference" />**

**</div>**

**</div>**

**<div class="my-part-holder">**

**<div>**

**<div class="my-sub-title">Observe:</div>**

**<div class="my-item-holder">**

**<label for="observe\_pic">Picture:</label>**

**<input type="file" name="observe\_pic" (change)="onObservePicChange($event)" accept="image/\*" />**

**</div>**

**<div class="my-item-holder">**

**<label for="observe\_writing">Writing:</label>**

**<input type="text" name="observe\_writing" placeholder="Aleksandar 1. Kralj SHS" formControlName="coinObserveWriting" />**

**</div>**

**<div class="my-horizontal-line"></div>**

**<div class="my-sub-title">Reverse:</div>**

**<div class="my-item-holder">**

**<label for="reverse\_pic">Picture:</label>**

**<input type="file" name="reverse\_pic" (change)="onReversePicChange($event)" accept="image/\*" />**

**</div>**

**<div class="my-item-holder">**

**<label for="reverse\_writing">Writing:</label>**

**<input type="text" name="reverse\_writing" placeholder="1 Dinar" formControlName="coinReverseWriting" />**

**</div>**

**<div class="my-horizontal-line"></div>**

**</div>**

**<div class="my-item-holder">**

**<label for="weight">Weight (in g):</label>**

**<input type="number" name="weight" placeholder="4.92" formControlName="coinWeight" />**

**</div>**

**<div class="my-item-holder">**

**<label for="diameter">Diameter (in mm):</label>**

**<input type="number" name="diameter" placeholder="23" formControlName="coinDiameter" />**

**</div>**

**<div class="my-item-holder">**

**<label for="thickness">Thickness (in mm):</label>**

**<input type="number" name="thickness" placeholder="1.8" formControlName="coinThickness" />**

**</div>**

**</div>**

**</div>**

**<div class="my-horizontal-line my-100-div"></div>**

**<div class="my-100-div my-100-div-column">**

**<div>**

**<label for="certified">Certified: &nbsp;</label>**

**<input type="checkbox" name="certified" formControlName="coinCertified" />**

**</div>**

**<div \*ngIf="editCoinForm.value.coinCertified">**

**<label for="certification">Input certification: &nbsp;</label>**

**<input type="text" name="certification" placeholder="321%31" formControlName="coinCertification" />**

**</div>**

**<div>**

**<label for="comment">Comment:</label>**

**<textarea name="comment" cols="100" rows="5" formControlName="coinComment"></textarea>**

**</div>**

**<div class="my-button-holder">**

**<button type="button" class="btn btn-primary" (click)="coinFormSubmit()">Save</button>**

**<button type="button" class="btn btn-danger" (click)="coinFormDiscard()">Discard</button>**

**</div>**

**</div>**

**</form>**

**<span class="my-red-indicator">\* Fields are mandatory!</span>**

**</div>**



## edit-coin.component.scss

.my-row {

position: absolute;

top: 55px;

left: 200px;

right: 0;

bottom: 0;

margin: 0;

background-color: rgba(0, 0, 0, 0.03);

padding: 20px;

padding-right: 0;

overflow-x: hidden;

overflow-y: scroll;

form {

width: 100%;

display: flex;

flex-direction: column;

.my-100-div {

width: 100%;

display: flex;

align-items: center;

justify-content: space-between;

&.my-100-div-column {

flex-direction: column;

label {

margin: 0;

font-weight: bold;

}

input[type="text"], input[type="number"], input[type="file"], select {

width: 220px;

min-width: 220px;

}

textarea {

width: 90%;

}

.my-button-holder {

width: 90%;

display: flex;

justify-content: space-around;

}

}

.my-part-holder {

width: 49%;

.my-sub-title {

font-size: 20px;

font-style: italic;

}

.my-item-holder {

width: 98%;

margin-left: 1%;

margin-bottom: 10px;

display: flex;

justify-content: space-between;

align-items: center;

label {

margin: 0;

font-weight: bold;

}

input[type="text"], input[type="number"], input[type="file"], select {

width: 220px;

min-width: 220px;

}

}

}

}

}

}

@media screen and (max-width: 930px) {

.my-row form .my-100-div {

flex-direction: column;

.my-part-holder {

width: 98%;

}

}

}

@media screen and (max-width: 500px) {

.my-row {

left: 100px;

}

}



## edit-coin.component.ts

**import** **{** ToastrService **}** from 'ngx-toastr'**;**

**import** **{** CoinService **}** from './../../../coin.service'**;**

**import** **{** Component**,** OnInit**,** ChangeDetectorRef **}** from '@angular/core'**;**

**import** **{** ActivatedRoute**,** Router **}** from '@angular/router'**;**

**import** **{** FormBuilder**,** FormControl**,** Validators **}** from '@angular/forms'**;**

**import** **{** Subscription **}** from 'rxjs'**;**

@Component**({**

selector**:** 'app-edit-coin'**,**

templateUrl**:** './edit-coin.component.html'**,**

styleUrls**:** **[**'./edit-coin.component.scss'**]**

**})**

**export** **class** EditCoinComponent **implements** OnInit **{**

subscription**:** Subscription **=** **new** Subscription**();**

coinItem**;**

edges**;**

shapes**;**

compositions**;**

grades**;**

**private** coin\_id**:** number**;**

valueReg **=** **/^[0-9]+\s[A-z]+/;**

editCoinForm **=** **this.**formBuilder**.**group**({**

coinValue**:** **[**''**,** Validators**.**pattern**(this.**valueReg**)],**

coinYear**:** **[**''**],**

coinCountry**:** **[**''**],**

coinEmperor**:** **[**''**],**

coinEdge**:** **[**'1'**],**

coinShape**:** **[**'1'**],**

coinComposition**:** **[**'1'**],**

coinGrade**:** **[**'1'**],**

coinDemonetized**:** **[**''**],**

coinReference**:** **[**''**],**

coinObservePic**:** **new** FormControl**(null),**

coinObserveWriting**:** **[**''**],**

coinReversePic**:** **new** FormControl**(null),**

coinReverseWriting**:** **[**''**],**

coinWeight**:** **[**''**,** Validators**.**min**(**0.01**)],**

coinDiameter**:** **[**''**,** Validators**.**min**(**0.01**)],**

coinThickness**:** **[**''**,** Validators**.**min**(**0.01**)],**

coinCertified**:** **[**''**],**

coinCertification**:** **[**''**],**

coinComment**:** **[**''**]**

**});**

**private** waitTime **=** 3000**;**

**constructor(private** route**:** ActivatedRoute**,** **private** router**:** Router**,** **private** formBuilder**:** FormBuilder**,** **private** coinService**:** CoinService**,** **private** cd**:** ChangeDetectorRef**,** **private** toastr**:** ToastrService**)** **{** **}**

ngOnInit**()** **{**

**var** id\_param **=** **this.**route**.**snapshot**.**params**[**'id'**];**

**if** **(**isNaN**(**id\_param**))** **{**

**this.**router**.navigate([**'/coin'**]);**

**return;**

**}**

**this.**coin\_id **=** id\_param**;**

**this.**subscription**.**add**(this.**coinService**.**getEdges**().**subscribe**(**

response **=>** **{** **this.**edges **=** response**;** **},**

error **=>** **{** **this.**errorFunc**(**"Error"**,** "Unexpected error occured. Please try again later."**);** **}**

**));**

**this.**subscription**.**add**(this.**coinService**.**getShapes**().**subscribe**(**

response **=>** **{** **this.**shapes **=** response**;** **},**

error **=>** **{** **this.**errorFunc**(**"Error"**,** "Unexpected error occured. Please try again later."**);** **}**

**));**

**this.**subscription**.**add**(this.**coinService**.**getCompositions**().**subscribe**(**

response **=>** **{** **this.**compositions **=** response**;** **},**

error **=>** **{** **this.**errorFunc**(**"Error"**,** "Unexpected error occured. Please try again later."**);** **}**

**));**

this.subscription.add(this.coinService.getGrades().subscribe(

response => { this.grades = response; },

error => { this.errorFunc("Error", "Unexpected error occured. Please try again later."); }

));

this.subscription.add(this.coinService.getEditCoin(this.coin\_id).subscribe(

response => {

this.coinItem = response;

if (!this.coinItem) {

this.router.navigate(['/coin']);

}

this.patchValues();

},

error => { this.errorFunc("Error", "Unexpected error occured. Please try again later."); }

));

}

patchValues() {

this.editCoinForm.patchValue({

coinValue: this.coinItem.value,

coinYear: this.coinItem.year,

coinCountry: this.coinItem.country,

coinEmperor: this.coinItem.emperor,

coinEdge: this.coinItem.edge,

coinShape: this.coinItem.shape,

coinComposition: this.coinItem.composition,

coinGrade: this.coinItem.grade,

coinDemonetized: this.coinItem.demonetized,

coinReference: this.coinItem.reference,

coinObserveWriting: this.coinItem.observe\_writing,

coinReverseWriting: this.coinItem.reverse\_writing,

coinWeight: this.coinItem.weight,

coinDiameter: this.coinItem.diameter,

coinThickness: this.coinItem.thickness,

coinComment: this.coinItem.comment

});

if (this.coinItem.certified !== 'No') {

this.editCoinForm.patchValue({

coinCertified: true,

coinCertification: this.coinItem.certified,

});

}

}

coinFormDiscard() {

this.router.navigate(['/coin/' + this.coin\_id]);

}

coinFormSubmit() {

if (this.editCoinForm.status === 'INVALID') {

this.errorFunc("Could not submit", "Please check your input and try again.");

return;

}

this.subscription.add(this.coinService.editCoin(this.editCoinForm.value, this.coin\_id).subscribe(

response => {

if (response == "OK") { this.successFunc("Success", "Coin edited successfully. Redirecting you to coin page."); }

else { this.errorFunc("Error", "Unexpected error occured. Please try again later."); }

},

error => { this.errorFunc("Error", "Unexpected error occured. Please try again later."); }

));

}

onObservePicChange(event) {

let reader = new FileReader();

if(event.target.files && event.target.files.length) {

const [file] = event.target.files;

reader.readAsDataURL(file);

reader.onload = () => {

this.editCoinForm.patchValue({

coinObservePic: reader.result

});

this.cd.markForCheck();

};

}

}

onReversePicChange(event) {

let reader = new FileReader();

if(event.target.files && event.target.files.length) {

const [file] = event.target.files;

reader.readAsDataURL(file);

reader.onload = () => {

this.editCoinForm.patchValue({

coinReversePic: reader.result

});

this.cd.markForCheck();

};

}

}

successFunc(title, message) {

this.toastr.success(message, title , {

timeOut : this.waitTime

});

setTimeout(()=>{

this.router.navigate(['/coin/' + this.coin\_id]);

}, this.waitTime);

}

errorFunc(title, message) {

this.toastr.error(message, title , {

timeOut : this.waitTime

});

}

}



## no-coin-show.component.scss

.my-bg-preview {

width: 60%;

margin-left: 20%;

padding: 10px;

margin-top: 30px;

h4.my-no-color-ash {

font-size: 24px;

}

h5.my-no-color-ash {

font-size: 18px;

}

.my-no-color-ash {

color: #aaa;

font-weight: bold;

margin: 10px 0;

cursor: default;

}

.my-no-color-black {

font-weight: bold;

font-size: 15px;

a {

color: #000;

&:hover {

color: #F9B500;

text-decoration: none;

}

}

}

}



## no-coin-show.component.ts

**import** **{** Component**,** OnInit **}** from '@angular/core'**;**

@Component**({**

selector**:** 'app-no-coin-show'**,**

templateUrl**:** './no-coin-show.component.html'**,**

styleUrls**:** **[**'./no-coin-show.component.scss'**]**

**})**

**export** **class** NoCoinShowComponent **implements** OnInit **{**

**constructor()** **{** **}**

ngOnInit**():** **void** **{**

**}**

**}**



## no-coin-show.component.html

<div class=**"my-bg-preview"**>

<div>

<h4 class=**"my-no-color-ash text-center"**>**No coin found**</h4>

<h5 class=**"my-no-color-ash text-center"**>**with these search parameters...**</h5>

<h6 class=**"my-no-color-black text-center"**><a routerLink=**"/coin/add"**>**Do you wish to add a new coin?**</a></h6>

<app-coin-fallin></app-coin-fallin>

</div>

</div>



## preview-coin.component.html

<div class=**"bg-white my-bg-preview"**>

<div class=**"my-preview-images"**>

<img [src]=**"'**http://localhost/all-my-coins-php/coin-images/**' + (coin.observe\_pic || 'no\_image.png')"** [alt]=**"coin.observe\_writing || 'No text inserted'"** />

<img [src]=**"'**http://localhost/all-my-coins-php/coin-images/**' + (coin.reverse\_pic || 'no\_image.png')"** [alt]=**"coin.reverse\_writing || 'No text inserted'"** />

</div>

<div>

<h5 class=**"my-color-ash"**>**{{coin.country || 'No country inserted'}}**</h5>

<h4 class=**"my-color-black text-center"**><a routerLink=**"/coin/{{coin.id}}"**>**{{coin.value || 'Error'}} {{coin.year || 'Error'}}**</a></h4>

<ul class=**"my-preview-list"**>

<li><span class=**"fas fa-star-half-alt"**></span> **{{coin.grade || 'No grade selected'}}**</li>

<li><span class=**"fas fa-certificate"**></span> **{{coin.certified || 'No'}}**</li>

</ul>

</div>

</div>



## preview-coin.component.scss

.my-bg-preview {

width: 60%;

margin-left: 20%;

padding: 10px;

margin-top: 30px;

.my-preview-images {

width: 100%;

img {

width: 50%;

}

}

.my-color-ash {

color: #aaa;

font-size: 17px;

font-weight: bold;

margin: 10px 0;

text-transform: uppercase;

}

.my-color-black {

font-weight: bold;

a {

color: #000;

&:hover {

color: #F9B500;

text-decoration: none;

}

}

}

.my-preview-list {

padding: 0;

list-style-type: none;

margin: 20px 0;

display: flex;

justify-content: space-around;

}

}



## preview-coin.component.ts

**import** **{** Component**,** OnInit**,** Input **}** from '@angular/core'**;**

@Component**({**

selector**:** 'app-preview-coin'**,**

templateUrl**:** './preview-coin.component.html'**,**

styleUrls**:** **[**'./preview-coin.component.scss'**]**

**})**

**export** **class** PreviewCoinComponent **implements** OnInit **{**

@Input**()** coin**:** **{**id**:** number**,** country**:** string**,** value**:** string**,** year**:** number**,** grade**:** string**,** certified**:** string**,** observe\_pic**:** string**,** reverse\_pic**:** string**,** observe\_writing**:** string**,** reverse\_writing**:** string**}** **;**

**constructor()** **{** **}**

ngOnInit**():** **void** **{**

**}**

**}**



## single-coin.component.scss

.my-row {

position: absolute;

top: 55px;

left: 200px;

right: 0;

bottom: 0;

margin: 0;

background-color: rgba(0, 0, 0, 0.03);

padding: 20px;

padding-right: 0;

overflow-x: hidden;

overflow-y: scroll;

display: flex;

.my-column {

display: flex;

flex-direction: column;

img {

cursor: pointer;

width: 40%;

}

.my-small-title {

font-weight: bold;

font-size: 14px;

&.small-title-margin {

margin: 10px 0;

}

.my-italic {

font-weight: normal;

font-style: italic;

color: #858796;

}

}

.my-comment-area {

font-size: 13px;

font-style: italic;

margin-top: 10px;

background-color: white;

width: 80%;

margin-left: 10%;

padding: 5px;

border-radius: 5px;

}

.my-button-holder {

display: flex;

justify-content: space-around;

}

}

}

.modal-content img {

width: 70%;

margin-left: 15%;

}

@media screen and (max-width: 500px) {

.my-row {

left: 100px;

}

}

@media screen and (max-width: 768px) {

.my-first-column {

order: 2;

}

.my-second-column {

order: 1;

margin-bottom: 30px;

}

}



## single-coin.component.ts

**import** **{** ToastrService **}** from 'ngx-toastr'**;**

**import** **{** CoinService **}** from './../../../coin.service'**;**

**import** **{** Component**,** OnInit **}** from '@angular/core'**;**

**import** **{** ActivatedRoute**,** Router **}** from '@angular/router'**;**

**import** **{** Subscription **}** from 'rxjs'**;**

@Component**({**

selector**:** 'app-single-coin'**,**

templateUrl**:** './single-coin.component.html'**,**

styleUrls**:** **[**'./single-coin.component.scss'**]**

**})**

**export** **class** SingleCoinComponent **implements** OnInit **{**

subscription**:** Subscription **=** **new** Subscription**();**

**private** coin\_id**:** number**;**

**public** selectedImage **=** **{**src**:** ''**,** alt**:** ''**};**

coinItem**;**

**private** waitTime **=** 3000**;**

**constructor(private** route**:** ActivatedRoute**,** **private** router**:** Router**,** **private** coinService**:** CoinService**,** **private** toastr**:** ToastrService**)** **{** **}**

ngOnInit**()** **{**

**var** id\_param **=** Number**(this.**route**.**snapshot**.**params**[**'id'**]);**

**if** **(**isNaN**(**id\_param**))** **{**

**this.**router**.navigate([**'/coin'**]);**

**return;**

**}**

**this.**coin\_id **=** id\_param**;**

**this.**subscription**.**add**(this.**coinService**.**getCoin**(this.**coin\_id**).**subscribe**(**

response **=>** **{**

**this.**coinItem **=** response**;**

**if** **(!this.**coinItem**)** **{**

**this.**router**.navigate([**'/coin'**]);**

**}**

**},**

error **=>** **{** **this.**errorFunc**(**"Error"**,** "Unexpected error occured. Please try again later."**);** **}**

**));**

**}**

**public** enlargeImage**(**imageSrc**,** imageAlt**):** **void** **{**

this.selectedImage.src = imageSrc;

this.selectedImage.alt = imageAlt;

}

editClick() {

this.router.navigate(['/coin/edit/' + this.coin\_id]);

}

deleteClick() {

this.subscription.add(this.coinService.deleteCoin(this.coin\_id).subscribe(

response => { this.successFunc("Success", "Coin deleted successfully. Redirecting you to home page."); },

error => { this.errorFunc("Error", "Unexpected error occured. Please try again later."); }

));

}

successFunc(title, message) {

this.toastr.success(message, title , {

timeOut : this.waitTime

});

setTimeout(()=>{

this.router.navigate(['/coin']);

}, this.waitTime);

}

errorFunc(title, message) {

this.toastr.error(message, title , {

timeOut : this.waitTime

});

}

}



## single-coin.component.html

<app-navigation [shouldShowFilterSort]="false"></app-navigation>

<app-title-area></app-title-area>

<div class=**"row my-row"**>

<ng-container \*ngIf="coinItem">

<div class=**"col-md-6 my-column my-first-column"**>

<img [src]=**"'**http://localhost/all-my-coins-php/coin-images/**' + (coinItem.observe\_pic || 'no\_image.png')"** [alt]=**"coinItem.observe\_writing || 'No text inserted'"** data-toggle=**"modal"** data-target=**".bd-example-modal-xl"** (click)=**"enlargeImage(coinItem.observe\_pic || 'no\_image.png', coinItem.observe\_writing || 'No text inserted')"**/>

<span class=**"my-small-title small-title-margin"**>**Observe:** <span class=**"my-italic"**>**{{coinItem.observe\_writing || 'No text inserted'}}**</span></span>

<img [src]=**"'**http://localhost/all-my-coins-php/coin-images/**' + (coinItem.reverse\_pic || 'no\_image.png')"** [alt]=**"coinItem.reverse\_writing || 'No text inserted'"** data-toggle=**"modal"** data-target=**".bd-example-modal-xl"** (click)=**"enlargeImage(coinItem.reverse\_pic || 'no\_image.png', coinItem.reverse\_writing || 'No text inserted')"**/>

<span class=**"my-small-title small-title-margin"**>**Reverse:** <span class=**"my-italic"**>**{{coinItem.reverse\_writing || 'No text inserted'}}**</span></span>

<div class=**"my-horizontal-line"**></div>

<span class=**"my-small-title"**>**Demonetized:** <span class=**"my-italic"**>**{{coinItem.demonetized ? 'Yes' : 'No'}}**</span></span>

<span class=**"my-small-title"**>**Reference:** <span class=**"my-italic"**>**{{coinItem.reference || 'No reference inserted'}}**</span></span>

<span class=**"my-small-title"**>**Certified:** <span class=**"my-italic"**>**{{coinItem.certified || 'No'}}**</span></span>

<div class=**"my-horizontal-line"**></div>

<div class=**"my-button-holder"**>

<button type=**"button"** class=**"btn btn-primary"** (click)=**"editClick();"**>**Edit**</button>

<button type=**"button"** class=**"btn btn-danger"** (click)=**"deleteClick();"**>**Delete**</button>

</div>

</div>

<div class=**"col-md-6 my-column my-second-column"**>

<h1>**{{coinItem.value || 'Error'}} {{coinItem.year || 'Error'}}**</h1>

<span class=**"my-small-title"**>**Year:** <span class=**"my-italic"**>**{{coinItem.year | customyear}}**</span></span>

<span class=**"my-small-title"**>**Country:** <span class=**"my-italic"**>**{{coinItem.country || 'No country inserted'}}**</span></span>

<span class=**"my-small-title"**>**Emperor:** <span class=**"my-italic"**>**{{coinItem.emperor || 'No emperor inserted'}}**</span></span>

<div class=**"my-horizontal-line"**></div>

<span class=**"my-small-title"**>**Weight:** <span class=**"my-italic"**>**{{coinItem.weight | customvalue:'g'}}**</span></span>

<span class=**"my-small-title"**>**Diameter:** <span class=**"my-italic"**>**{{coinItem.diameter | customvalue:'m'}}**</span></span>

<span class=**"my-small-title"**>**Thickness:** <span class=**"my-italic"**>**{{coinItem.thickness | customvalue:'m'}}**</span></span>

<span class=**"my-small-title"**>**Edge:** <span class=**"my-italic"**>**{{coinItem.edge || 'No edge selected'}}**</span></span>

<span class=**"my-small-title"**>**Shape:** <span class=**"my-italic"**>**{{coinItem.shape || 'No shape selected'}}**</span></span>

<span class=**"my-small-title"**>**Composition:** <span class=**"my-italic"**>**{{coinItem.composition || 'No composition selected'}}**</span></span>

<span class=**"my-small-title"**>**Grade:** <span class=**"my-italic"**>**{{coinItem.grade || 'No grade selected'}}**</span></span>

<div class=**"my-horizontal-line"** \*ngIf=**"coinItem.comment"**></div>

<span class=**"my-small-title"** \*ngIf=**"coinItem.comment"**>**My comment:**</span>

<div class=**"my-comment-area"** \*ngIf=**"coinItem.comment"**>**{{coinItem.comment}}**</div>

</div>

**<div class="modal fade bd-example-modal-xl" tabindex="-1" role="dialog" aria-labelledby="myExtraLargeModalLabel" aria-hidden="true">**

**<div class="modal-dialog modal-xl">**

**<div class="modal-content">**

**<img [src]="selectedImage.src ? 'http://localhost/all-my-coins-php/coin-images/' + selectedImage.src : ''" alt="{{selectedImage.alt}}" id="enlargedImageLoc"/>**

**</div>**

**</div>**

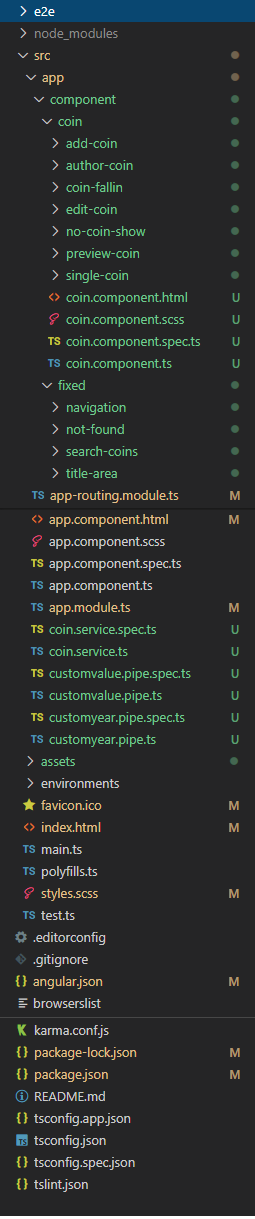
**</div>**

**</ng-container>**

**</div>**



# HIJERARHIJSKI PREGLED

****