

Akademija tehničko-umetničkih strukovnih Studija

Odsek Visoka ICT škola

# Collēctiō

<https://nikola-project-418a3.web.app/>

# Sadržaj

Uvod.....	6
Korišćene tehnologije .....	6
Okruženja .....	6
Opis projekta.....	6
Organizacija.....	7
Organizaciona šema baze .....	7
Organizaciona šema .....	8
Struktura projekta.....	9
App .....	9
Components.....	9
Interfaces .....	9
Services .....	10
Opis i slike stranica.....	10
Home.....	10
Collection .....	11
Collection item .....	11
Contact.....	12
Sign in, Sign up, Forgot Password & Verify Email .....	13
Dashboard .....	14
My Collections.....	14
Create Collection.....	15
Edit Collection .....	15
Create Item .....	16
Update Collection .....	16
Kodovi .....	17
Banner component .....	17
Html.....	17
Scss.....	17
TS.....	18
Collections Component.....	19
Html.....	19

Scss.....	19
TS.....	19
Single Collection Component.....	20
Html.....	20
Scss.....	20
TS.....	21
Header Component.....	24
Html.....	24
Scss.....	26
TS.....	27
Footer Component.....	29
Html.....	29
Scss.....	29
Author Component .....	31
Html.....	31
Scss.....	32
TS.....	33
About (Contact) Component.....	34
Html.....	34
Scss.....	36
TS.....	37
Dashboard Component.....	38
Html.....	38
TS.....	39
Forgot Password Component .....	39
Html.....	39
Scss.....	40
Sign in Component.....	40
Html.....	40
Scss.....	41
Sign up Component.....	41
Html.....	41
Scss.....	42

TS.....	42
Verify Email Component .....	44
Html.....	44
Collection Component .....	44
Html.....	44
Scss.....	45
TS.....	47
Collection Item Component.....	48
Html.....	48
Scss.....	48
TS.....	50
Item Component.....	51
Html.....	51
Scss.....	52
TS.....	53
Item Info Component.....	54
Html.....	54
Scss.....	54
Create Collection Component.....	54
Html.....	54
Scss.....	56
TS.....	56
Create Item Component .....	59
Html.....	59
Scss.....	60
TS.....	60
Edit Collection Component .....	62
Html.....	62
Scss.....	63
TS.....	65
My Collections Component.....	67
Html.....	67
Scss.....	67

TS.....	68
Update Collection Component .....	69
Html.....	69
Scss.....	70
TS.....	70
Interfaces .....	73
Categories .....	73
Collections.....	73
Errors.....	73
Infos.....	73
Items .....	73
Paginate .....	74
User .....	74
Services .....	74
Auth.....	74
Categories .....	78
Collections.....	79
Items .....	80
Files .....	82
Messages.....	82
Guard/Auth.guard.....	83
Animations.ts .....	83
App-routing.module.....	84
App-component.ts .....	87
App-module.ts .....	88

# Uvod

## Korišćene tehnologije

- HTML
- CSS
- TypeScript
- Angular
- Bootstrap
- Firebase

## Okruženja

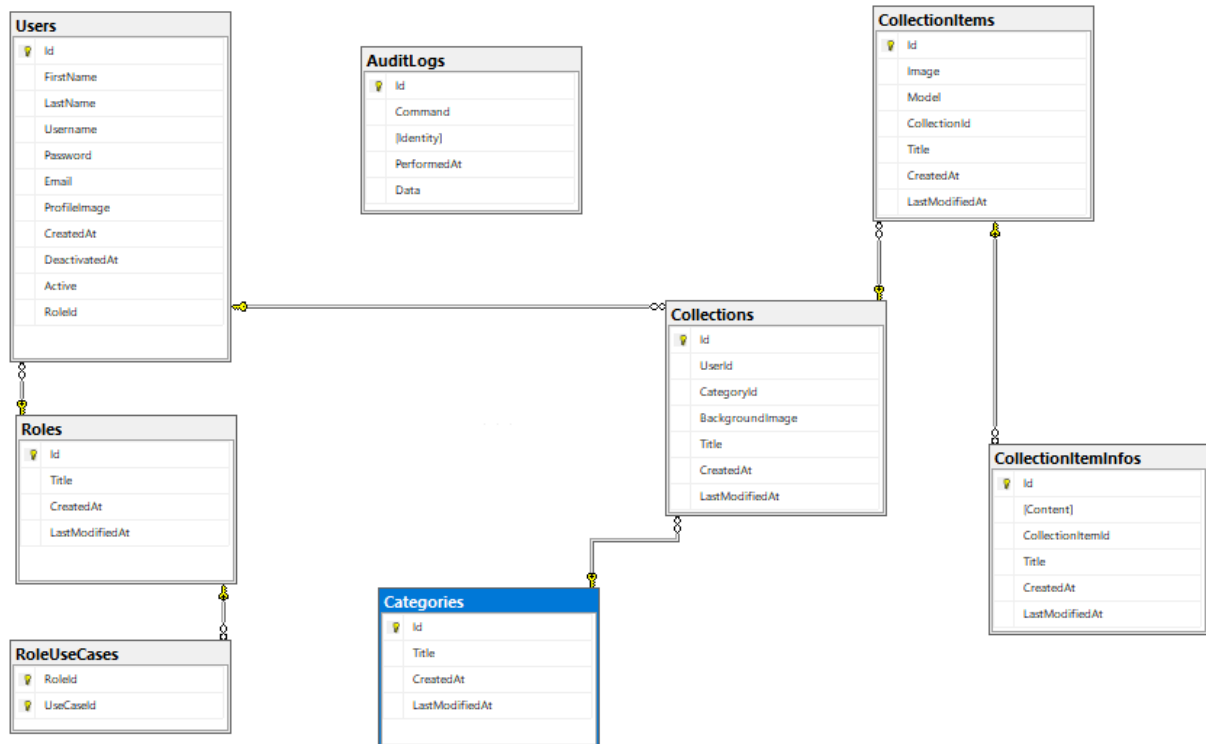
Za izradu projekta korišten je Visual Studio Code.

## Opis projekta

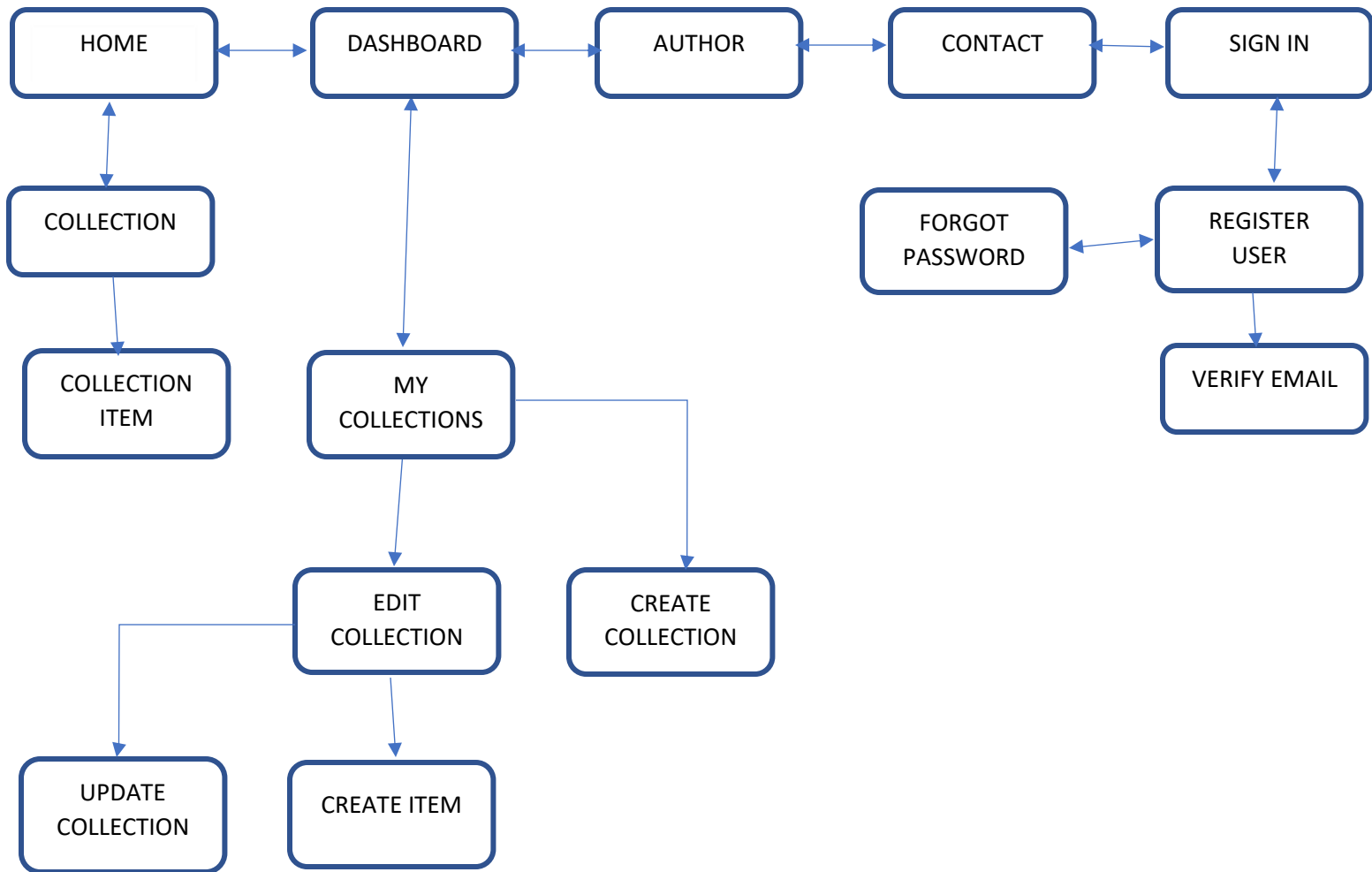
Projekat predstavlja online enciklopediju ili raznovrsne online kolekcije koje korisnici mogu da posjećuju i o kojim mogu da čitaju i nauče nešto novo. Autorizovani korisnici mogu kreirati i editovati svoje kolekcije.

# Organizacija

Organizaciona šema baze



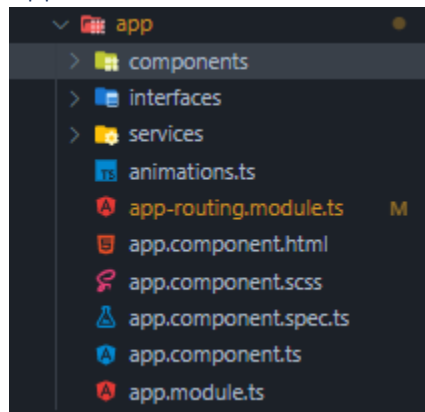
## Organizaciona šema



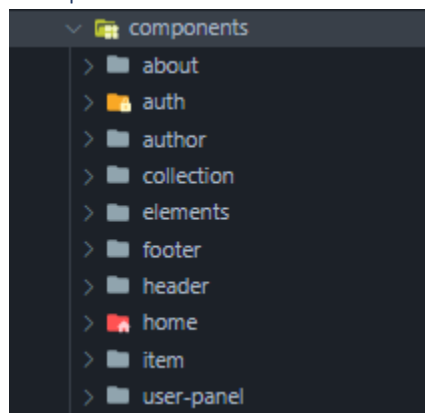


## Struktura projekta

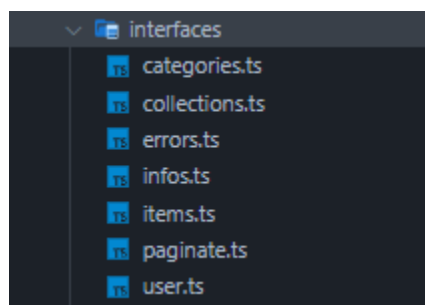
### App



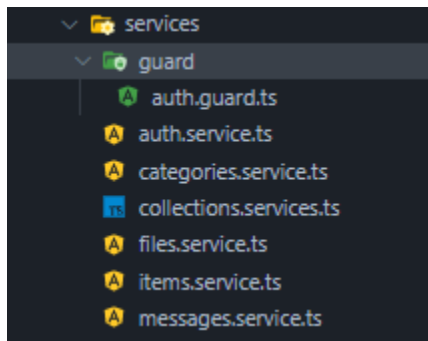
### Components



### Interfaces



## Services



## Opis i slike stranica

### Home



Na Home stranici prikazane su sve kolekcije, klikom na neku od kolekcija otvara se nova stranica sa svim njenim elementima. Pretraga kolekcija po nazivu.

## Collection



Na stranici kolekcije prikazani su autor, naziv kolekcije i svi njeni elementi, klikom na neki od elemenata prikazuje se detaljniji prikaz.

## Collection item



Element kolekcije može sadržati dodatne informacije kao i 3D model.

## Contact

*author*[Home](#)[Contact](#)[Sign In](#)

---

# Contact Us

Feel free to contact us and we will get back to you as soon as we can.

Your Name \*

Email \*

Message \*

SEND

Kontakt stranica sadrži formu sa proverom elemenata formulara i obradom grešaka. Ukoliko je validacija prošla dugme za slanje se enable-uje. Slanje mail-a je realizovano pomoću Formspree-a.

Sign in, Sign up, Forgot Password & Verify Email

### Sign In

Email

Password

Sign In

[Forgot Password](#)

[Create Account](#)

### Sign Up

First Name

Last Name

Username

Email Address

Password

Sign Up

[Sign In](#)

### Reset Password

Please enter your email address to request a password reset.

Email Address

Reset Password

[Sign In](#)

### Thank You for Registering

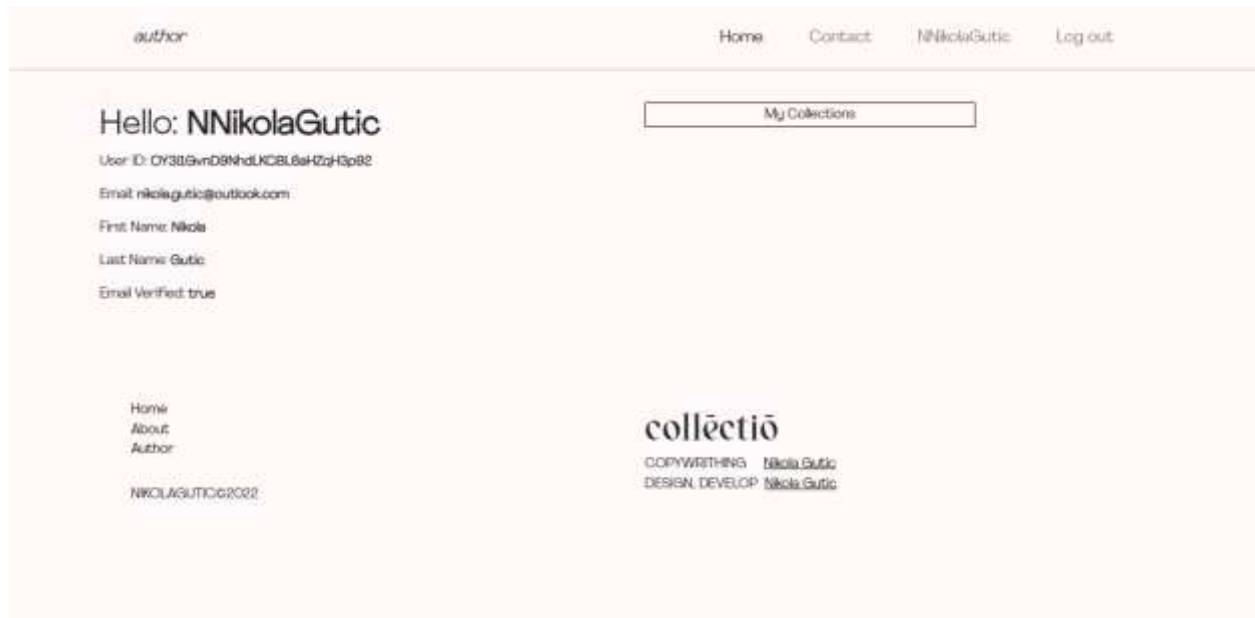
We have sent a confirmation email to [gutic.radosaaman@outlook.com](mailto:gutic.radosaaman@outlook.com).

Please check your email and click on the link to verify your email address.

Resend Verification Email

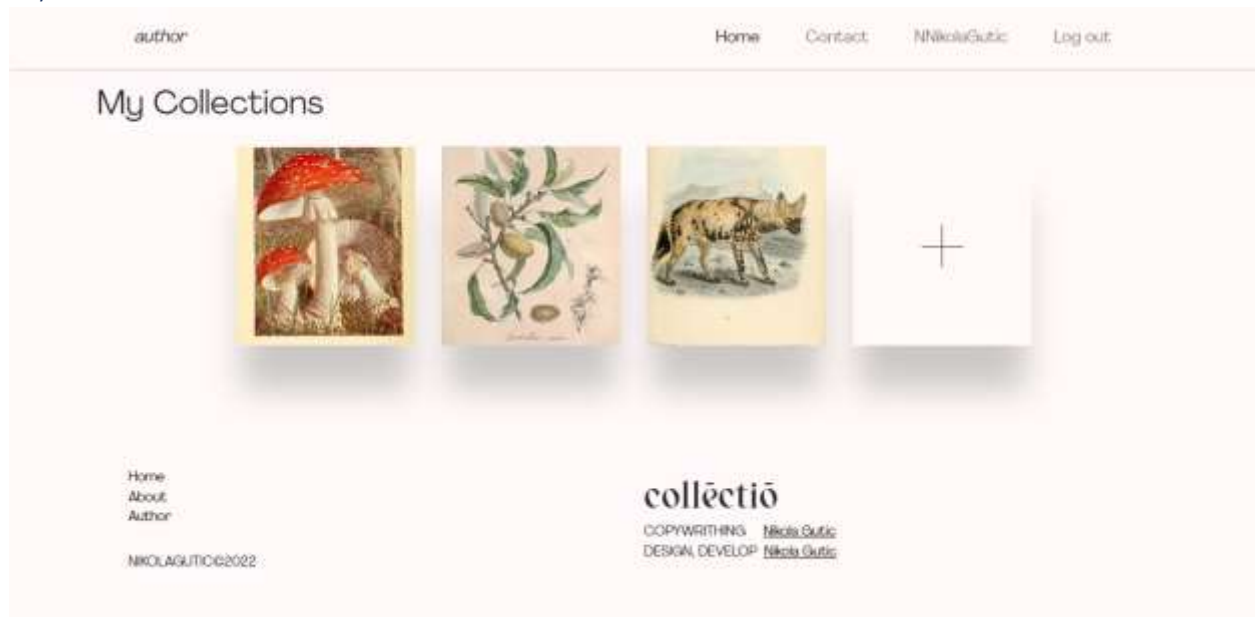
Sign In

## Dashboard



Nakon uspešnog logovanja i verifikacije emaila u navigaciji se prikazuje korisničko ime, klikom na isto korisnik se void da Dashboard stranicu. Dashboard stranica prikazuje osnovne podatke korisnika i link ka svim njegovim kolekcijama.

## My Collections



My Collections stranica prikazuje sve korisnikove kolekcije i opciju za kreiranje nove.

## Create Collection

author

Home Contact NNikolaGutic Log out

### Create new Collection

Collection Title \*

Category \*

BackgroundImage  No file chosen

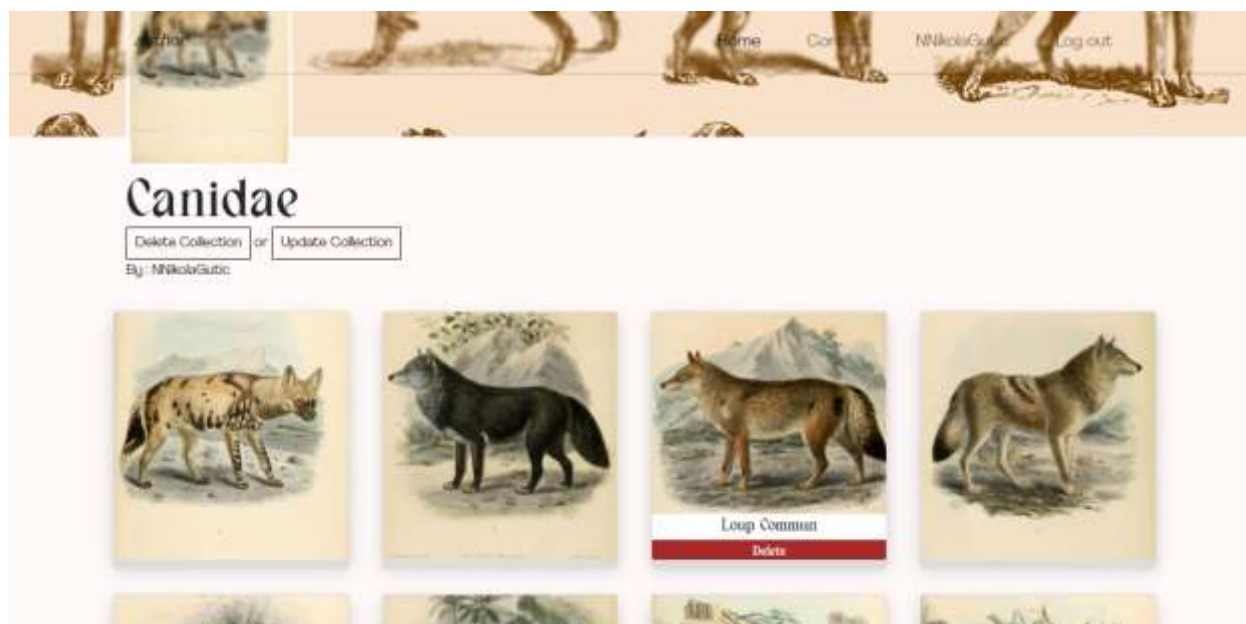
Home  
About  
Author

© NNikolaGutic 2022

collētiō  
COPYWRITING: NNikolaGutic  
DESIGN, DEVELOP: NNikolaGutic

Stranica za kreiranje nove kolekcije, korisnik unosi naslov kolekcije, kategoriju i pozadinsku sliku.

## Edit Collection



Klikom na neku od svojih kolekcija korisnik može da edituje svoju kolekciju. Briše i dodaje elemente kolekcije, obriše kompletnu kolekciju ili joj mijenja naslov i kategoriju.

## Create Item

[author](#)

[Home](#) [Contact](#) [NNikolaGutic](#) [Log out](#)

---

Create new Item

Item Title \*

Item Image \*  No file chosen

[Home](#)  
[About](#)  
[Author](#)  
  
NNIKLAGUTIC©2022

**collēctiō**  
COPYWRITING [Nikola Gutic](#)  
DESIGN, DEVELOP [Nikola Gutic](#)

Stranica za dodavanje novog elementa kolekciji.

## Update Collection

[author](#)

[Home](#) [Contact](#) [NNikolaGutic](#) [Log out](#)

---

Update Collection

Collection Title \*

Category \*

[Home](#)  
[About](#)  
[Author](#)  
  
NNIKLAGUTIC©2022

**collēctiō**  
COPYWRITING [Nikola Gutic](#)  
DESIGN, DEVELOP [Nikola Gutic](#)



# Kodovi

Banner component

Html

```
<div class="banner">
  <div class="heading" @fade>
    <h1>collectiō </h1>
    <h3>The encyclopedia of everything</h3>
    <span class="opis">discover, learn & share your work</span>
    <br>
    <input [ngModel]="value" (ngModelChange)="searchCollection($event)"
class="search" placeholder="  search for collections">
  </div>
</div>
```

Scss

```
.banner {
  height: 50vh;
  width: 100%;
  display: flex;
  justify-content: center;

  .heading {
    text-align: center;
    padding-top: 8%;
    font-family: 'Bagerich';
    z-index: 99;

    h1 {
      z-index: 99;
      font-size: 13vw;
    }

    .search {
      margin: 10px 0px;
      border: none;
      border-bottom: 1px solid black;
      background-color: rgba(0, 0, 0, 0);

      &:focus {
        outline: none;
      }
    }
  }
}
```

```

        .opis {
            font-family: "Agrandir";
            z-index: 99;
            font-stretch: extra-condensed;
        }
    }

    @media only screen and (max-width: 992px) {
        .heading {
            padding-top: 20vh;
        }
    }
}

```

TS

```

import { $animations } from 'src/app/animations';
import { Component, OnInit } from '@angular/core';
import { CollectionsServices } from 'src/app/services/collections.services';

@Component({
    selector: 'app-banner',
    templateUrl: './banner.component.html',
    styleUrls: ['./banner.component.scss'],
    animations: [$animations]
})
export class BannerComponent implements OnInit {

    constructor(private collectionServices: CollectionsServices) { }

    ngOnInit(): void {
    }

    value!: string;

    searchCollection(keyword: string) {
        this.collectionServices.searchCollection(keyword);
    }
}

```

## Collections Component

### Html

```
<div class="table-fungi">
  <div class="row">
    <div class="col-12 d-flex flex-wrap justify-content-center">
      <app-single-collection class="col-lg-2 col-md-3 col-12 ms-3 me-3 mb-5 mt-3" *ngFor="let collection of collections" [data]="collection" [link]="'collection'"></app-single-collection>
    </div>
  </div>
</div>
```

### Scss

```
.table-fungi{
  margin-top: 150px;
  width: 100%;
  padding: 0px 5%;
  z-index: 60;
}
```

### TS

```
import { CollectionsServices } from './../../../../services/collections.services';
import { Component, OnInit } from '@angular/core';
import { Collections } from 'src/app/interfaces/collections';

@Component({
  selector: 'app-collections',
  templateUrl: './collections.component.html',
  styleUrls: ['./collections.component.scss']
})
export class CollectionsComponent implements OnInit {

  public page!: number;
  public perPage!: number;
  public pagesCount!: number;
  public totalCount!: number;

  collections!: Collections[];
  data!: Collections[];
  link!: string;

  constructor(private collectionServices: CollectionsServices) { }
```

```

ngOnInit(): void {
  this.getCollections();
}

getCollections(): void {
  this.collectionServices.getCollections();

  this.collectionServices.colx$.subscribe(data => {

    this.collections = data.data;
    this.page = data.page;
    this.perPage = data.perPage;
    this.pagesCount = data.pagesCount;
    this.totalCount = data.totalCount;

  })
}
}

```

## Single Collection Component

### Html

```

<div class="fungi-content" *ngIf="data.items[0]" [ngStyle]="animateBlock"
style="background-image:
url('../assets/images/{{data.items[0].image}}'); background-position:
center;
        background-repeat: no-repeat;
        background-size: cover;">
  <div class="layer" [ngStyle]="hoverLayer">
  </div>
  <span class="content">
    <a aria-current="page" [routerLink]="[getLink(),
transformUrl(data.title)]" routerLinkActive="active">
      <p class="info" [ngStyle]="animateText">{{data.title}}</p>
    </a>
  </span>
</div>

```

### Scss

```

.fungi-content {
  position: relative;
  background-position: center;

```

```

background-repeat: no-repeat;
background-size: cover;
height: 240px;
display: flex;
align-items: center;
justify-content: center;
z-index: 60;

box-shadow: rgba(0, 0, 0, 0.2) 0px 60px 40px -7px;

.layer {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
}

.content {
  display: flex;
  align-items: center;
  justify-content: center;
  width: 100%;
  .info{
    p{
      width: 100%;
    }
  }
  a {
    width: 100%;
    p {
      font-weight: 300;
      width: 100%;
      z-index: 99;
    }
  }
}
}
}

```

TS

```

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

@Component({

```

```

    selector: 'app-single-collection',
    templateUrl: './single-collection.component.html',
    styleUrls: ['./single-collection.component.scss']
  })
}
export class SingleCollectionComponent implements OnInit {

  @Input() data!: any;
  @Input() link!: string;

  isVisible = false;
  brojac!: number;
  interval!: any;

  constructor(public el: ElementRef<HTMLElement>) {
  }

  public transformUrl(data: string): string {
    let newString = data.replace(/\s/g, '-');
    return newString;
  }

  @HostListener('mouseenter')
  onMouseEnter() {
    this.isVisible = true;
    this.brojac = 1;
    clearInterval(this.interval);
    this.interval = setInterval(() => {
      if (this.brojac < 1.3) {
        this.brojac += 0.01
      }
      else {
        clearInterval(this.interval);
      }
    }, 5);
  }

  @HostListener('mouseleave')
  onMouseLeave() {
    this.isVisible = false;
    clearInterval(this.interval);
    this.interval = setInterval(() => {
      if (this.brojac > 1) {
        this.brojac -= 0.01
      }
    }, 5);
  }
}

```

```

    }
    else {
        clearInterval(this.interval);
    }
}, 5);
}

ngOnInit(): void {
}

getLink(): string{
    return "/" + this.link;
}

get animateText() {
    if (this.isVisible) {
        return {
            "text-align": "center",
            "animation-name": "fontZoom",
            "animation-duration": "0.5s",
            "animation-fill-mode": "forwards"
        }
    }
    return {
        "text-align": "center",
        "animation-name": "fontOut",
        "animation-duration": "0.5s",
        "animation-fill-mode": "forwards"
    }
}

get animateBlock() {
    const scaler = this.brojac;
    if (this.isVisible) {
        return {
            "transform": "scale(" + scaler + "," + scaler + ")",
            "animation-name": "borderCurver",
            "animation-duration": "0.5s",
            "animation-fill-mode": "forwards",
            "z-index": "10"
        }
    }
    return {
        "transform": "scale(" + scaler + "," + scaler + ")",
        "animation-name": "borderCurverOut",
    }
}

```

```

        "animation-duration": "0.5s",
        "animation-fill-mode": "forwards",
        "z-index": "5"
    }
}

get hoverLayer() {
    if (this.isVisible) {
        return {
            "animation-name": "hoverLayer",
            "animation-duration": "0.5s",
            "animation-fill-mode": "forwards"
        }
    }
    return {
        "animation-name": "hoverLayerExit",
        "animation-duration": "0.10s",
        "animation-fill-mode": "forwards"
    }
}

get borderCurver() {
    if (this.isVisible) {
        return {
            "animation-name": "borderCurver",
            "animation-duration": "0.5s",
            "animation-fill-mode": "forwards"
        }
    }
    return {
        "animation-name": "borderCurverOut",
        "animation-duration": "0.5s",
        "animation-fill-mode": "forwards"
    }
}
}

```

## Header Component

Html

```

<nav class="navbar fixed-top navbar-expand-lg navbar-light bg-light">
  <div class="contain" [ngStyle]="headerOverlay">

```



```

    <a class="navbar-brand" routerLink="/author"
routerLinkActive="active">author</a>
    <button class="navbar-toggler navButton" type="button" data-bs-
toggle="collapse"
        data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false"
        aria-label="Toggle navigation" (click)="animateButton()"
[ngStyle]="buttonAnimation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav me-auto mb-2 mb-lg-0 navi">
            <li class="nav-item">
                <a class="nav-link" routerLink="/" routerLinkActive="active">Home</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" routerLink="/about"
routerLinkActive="active">Contact</a>
            </li>
            <span *ngIf="authService.isLoggedIn == true" class="d-flex">
                <li class="nav-item" *ngIf="authService.userData as user">
                    <a class="nav-link" routerLink="/dashboard">
                        <i class="fas fa-user"></i>{{authService.moreData.username}}
                    </a>
                </li>
                <li class="nav-item" *ngIf="authService.userData as user">
                    <a class="nav-link" (click)="authService.SignOut()">
                        <i class="fas fa-sign-out-alt"></i>Log out
                    </a>
                </li>
            </span>
            <div *ngIf="authService.isLoggedIn == false">
                <li class="nav-item">
                    <a class="nav-link" routerLink="/sign-in" routerLinkActive="active">
                        <i class="fas fa-sign-out-alt"></i>Sign In
                    </a>
                </li>
            </div>
        </ul>
    </div>
</div>
</nav>

```

## Scss

```
.navbar-light .navbar-toggler {
  color: rgba(0, 0, 0, 0.55);
  border-color: rgba(0, 0, 0, 0) !important;

  &:focus {
    border: none !important;
    box-shadow: none !important;
  }
}

.navbar {
  padding: 0px 150px;
  font-family: Agrandir;
  color: black;
  width: 100%;

  .contain {
    box-shadow: rgba(0, 0, 0, 0.15) 1.95px 1.95px 2.6px;
    backdrop-filter: blur(2px);

    position: absolute;
    height: 10vh;
    top: 0px;
    left: 0px;
    width: 100%;
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 0px 10%;
  }

  .navi {
    display: flex;
    align-items: center;
    width: 100%;
    justify-content: end;
    flex-direction: row;

    li {
      padding: 15px 20px;
      font-size: 1.2rem;
    }
  }
}
```

```

}

.navbar-brand {
  color: #212529;
  font-style: italic;
}

```

TS

```

import { AuthService } from 'src/app/services/auth.service';
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-header',
  templateUrl: './header.component.html',
  styleUrls: ['./header.component.scss']
})
export class HeaderComponent implements OnInit {

  public clicked: boolean = false;
  public triggerAnimation!: boolean;
  public scrollValue!: number;

  constructor(public authService: AuthService) { }

  ngOnInit(): void {
    this.authService.getMoreUserInfo();
    this.animateHeader();
  }

  //OnScroll trigger za promenu stilova header-a
  animateHeader() {
    window.addEventListener('scroll', () => {
      this.scrollValue = window.pageYOffset;
      if (this.scrollValue > 100) {
        this.triggerAnimation = true;
      }
      else {
        this.triggerAnimation = false;
      }
    })
  }

  animateButton() {
    this.clicked = !this.clicked;
  }
}

```

```

}

//Dodavanje stilova i animacija menu-button-u
get buttonAnimation() {
    if (this.clicked) {
        return {
            "animation-name": "animateMenuIn",
            "animation-duration": "3s",
            "animation-fill-mode": "forwards"
        }
    }
    return {
        "animation-name": "animateMenuOut",
        "animation-duration": "3s",
        "animation-fill-mode": "forwards"
    }
}

//stilovi
get headerOverlay() {
    if (this.triggerAnimation) {
        return {
            "animation-name": "headerIn",
            "animation-duration": "0.5s",
            "animation-fill-mode": "forwards"
        }
    }
    return {
        "animation-name": "headerOut",
        "animation-duration": "0.5s",
        "animation-fill-mode": "forwards"
    }
}

get aTagColor() {
    if (this.triggerAnimation) {
        return {
            "color": "#ffedc5"
        }
    }
    return {
        "color": "#212529"
    }
}
}

```

## Footer Component

### Html

```
<div class="container foot">
  <div class="row">
    <div class="col-lg-6 col-md-6 col-12 copy ">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0 navi">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" routerLink="/"
routerLinkActive="active">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" routerLink="/main"
routerLinkActive="active">About</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" routerLink="/author"
routerLinkActive="active">Author</a>
        </li>
      </ul>
      <p>NIKOLAGUTIC&#169;2022</p>
    </div>
    <div class="col-lg-6 col-md-6 col-12">
      <h1>collēctiō</h1>
      <ul>
        <li>COPYWRITHING&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</li><a
href="">Nikola Gutic</a><br>
        <li>DESIGN, DEVELOP&nbsp;&nbsp;&nbsp;</li><a href="">Nikola
Gutic</a><br>
      </ul>
    </div>
  </div>
</div>
```

### Scss

```
.container {
  padding: 50px;
  margin-top: 50px;
  height: 40vh;
  font-family: Agrandir;
```

```
h1 {
  font-family: 'Bagerich';
  font-size: 3rem;
}

ul {
  padding: 0px;

  li {
    display: inline;
  }

  justify-content: flex-end;
}

a {
  display: inline;
  color: black;
  text-decoration: underline;
}

.copy {
  display: flex;
  flex-direction: column;
  justify-content: flex-end;

  a {
    text-decoration: none;
  }

  p {
    margin-top: 30px;
  }

  .navi {
    width: 100%;
  }
}

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

  h1,
  ul,
  p {
    text-align: center !important;
  }
}
```

```

    }
  }
}

```

## Author Component

Html

```

<div class="main">
  <div class="row">
    <div class="col-2">
      <div class="side side-left">
        <p>less.less.less.less.</p>
      </div>
    </div>
    <div class="col-8">
      <div class="info">
        <div class="row d-flex justify-content-center">
          <div class="col-md-6 col-10 image">
            
          </div>
          <div class="col-md-6 col-10 author">
            <h1>Hello I'm Nikola</h1>
            <p>I am studying at the College of Information and
Communication Technologies, majoring in
Internet
Technology and I enjoy everything this college has to
offer, web3 enthusiast and I work at A
Tavola, a
great startup company that develops technology based
on computer vision, designed for
collection,
analysis and data use for improving restaurant front
and back of the house operations, as
well
as to optimize overall operations and efficiency of
the restaurants.</p>
            <p><a href="../../assets/dokumentacija.pdf"
target="_blank">---Documentation</a></p>
            <p><a href="https://github.com/NNikolaG/TheFungus"
target="_blank" >---Source Code</a></p>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

```

        <div class="col-2">
            <div class="side side-right">
                <p>less.less.less.less.</p>
            </div>
        </div>
    </div>
</div>

```

Scss

```

.main {
    .side {
        font-size: 10rem;
        font-family: Bagerich;
    }

    .side-left {
        p {
            position: relative;
            top: -50px;
            transform: rotate(90deg) scale(1, -1);
        }
    }

    .side-right {
        p {
            position: relative;
            top: -50px;
            transform: rotate(90deg);
        }
    }

    .info {
        padding-top: 150px;
        width: 100%;
        margin-bottom: 150px;
    }

    .image {
        img {
            width: 80%;
            float: right;
        }
    }
}

```



```

    .author {
      h1 {
        font-size: 5rem;
        font-family: Bagerich;
      }

      p {
        text-align: justify;
      }
      a{
        color:#222222;
      }
    }
  }
}

```

TS

```

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

@Component({
  selector: 'app-author',
  templateUrl: './author.component.html',
  styleUrls: ['./author.component.scss']
})
export class AuthorComponent implements OnInit {

  public infoBlock!: any;
  constructor() { }

  ngOnInit(): void {
  }

  click(event: { currentTarget: any; }) {
    var idAttr = event.currentTarget.id
    this.change(idAttr);
  }

  change(id: string) {
    var selector = "#info" + id;
    this.infoBlock = document.querySelector(selector);
    if (this.infoBlock.classList.contains('none')) {
      this.infoBlock.classList.remove('none');
      this.infoBlock.classList.add('block');
    }
  }
}

```

```

    else {
      this.infoBlock.classList.add('none');
      this.infoBlock.classList.remove('block');
    }
  }
}

```

## About (Contact) Component

Html

```

<div class="container">
  <div class="about-main">
    <div class="col-6">
      <h1>Contact Us</h1>
      <p>Feel free to contact us and we will get back to you as soon as we
can.</p>
    </div>
  </div>
  <div class="contact-area">
    <div class="container">
      <form [formGroup]="contactForm" (ngSubmit)="onSubmit()">
        <div class="row">
          <div class="col-sm-6 col-12">
            <mat-form-field appearance="outline">
              <mat-label>Your Name <span class="req">*</span></mat-label>
              <input matInput formControlName="name" />
              <div *ngIf="name?.invalid && (name?.dirty ||
name?.touched)" class="alert alert-danger">
                <mat-error>
                  <ng-container
*ngIf="name?.errors?.['required']">
                    Your full name is required.
                  </ng-container>
                </mat-error>
              </div>
            </mat-form-field>
          </div>
          <div class="col-sm-6 col-12">
            <mat-form-field appearance="outline">
              <mat-label>Email <span class="req">*</span></mat-label>
              <input matInput formControlName="email" />

```

```

        <div *ngIf="email?.invalid && (email?.dirty ||
email?.touched)" class="alert alert-danger">
            <mat-error>
                <ng-container
*ngIf="email?.errors?.['required']">
                    Email is required.
                </ng-container>
                <ng-container
*ngIf="email?.errors?.['minlength']">
                    The field is under min Len.
                </ng-container>
            </mat-error>
        </div>
    </mat-form-field>
</div>
</div>
<div class="row">
    <div class="col-12">
        <mat-form-field appearance="outline">
            <mat-label>Message <span class="req">*</span></mat-
label>
            <textarea name="msg" id="msg" matInput
formControlName="msg">
            </textarea>
            <div *ngIf="msg?.invalid && (msg?.dirty ||
msg?.touched)" class="alert alert-danger">
                <mat-error>
                    <ng-container
*ngIf="msg?.errors?.['required']">
                        Message is required.
                    </ng-container>
                </mat-error>
            </div>
        </mat-form-field>
    </div>
</div>
<div class="row">
    <div class="col-12">
        <button class='basic-button' mat-flat-button
color="primary" (ngSubmit)="onSubmit()"
[disabled]="contactForm.invalid">send</button>
    </div>
</div>
</form>
</div>

```

```
    </div>
  </div>
```

Scss

```
input {
  background-color: rgba(0, 0, 0, 0);
  outline: none;
  border: none;
  border-bottom: 1.3px solid black;
  width: 100%;
}

mat-label {
  margin-top: 20px;
  display: block;
}

.alert-danger {
  background-color: rgba(0, 0, 0, 0);
  border: none;
  height: 30px;
  padding: 0px;
  margin: 0px;
}

.req {
  color: brown;
}

#msg {
  width: 100%;
  background-color: rgba(0, 0, 0, 0);
  padding: 20px;

  &:focus {
    outline: none;
  }
}

.about-main {
  padding: 0px 12px;
  margin: 20vh 0px 10vh 0px;
  h1{
```

```

        font-family: Bagerich;
        font-size: 5rem;
    }
}

.basic-button{
    width: 200px;
    height: 50px;
}

```

TS

```

import { MessagesService } from './../../services/messages.service';
import { Component, OnInit } from '@angular/core';
import { FormGroup, FormControl, Validators } from '@angular/forms';

@Component({
  selector: 'app-about',
  templateUrl: './about.component.html',
  styleUrls: ['./about.component.scss']
})
export class AboutComponent implements OnInit {

  public contactForm = new FormGroup({
    name: new FormControl('', [Validators.required]),
    email: new FormControl('', [Validators.required, Validators.minLength(5)]),
    msg: new FormControl('', [Validators.required])
  });

  constructor(private messagesServices: MessagesService) { }

  ngOnInit(): void {
  }

  onSubmit(): void {
    if (this.contactForm.valid) {
      this.messagesServices.sendEmail(this.contactForm.value);
    }
    this.contactForm.reset();
  }

  get name() { return this.contactForm.get('name'); }
  get email() { return this.contactForm.get('email'); }
}

```

```

    get msg() { return this.contactForm.get('msg'); }
}

```

## Dashboard Component

Html

```

<div class="container">
  <div class="row pt-5 mt-5">
    <div role="main" class="col-lg-6 col-12">
      <div class="inner-adjust">
        <!-- Show user data when logged in -->
        <div class="row" *ngIf="authService.userData as user">
          <div class="col-md-12">
            <div class="media">
              
              <div class="media-body">
                <h1>Hello: <strong>{{(authService.moreData.username) ?
authService.moreData.username : 'User'}}</strong>
                </h1>
                <p>User ID: <strong>{{user.uid}}</strong></p>
                <p>Email: <strong>{{authService.moreData.email}}</strong></p>
                <p>First Name:
<strong>{{authService.moreData.firstName}}</strong></p>
                <p>Last Name:
<strong>{{authService.moreData.lastName}}</strong></p>
                <p>Email Verified: <strong>{{user.emailVerified}}</strong></p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div class="col-12 col-lg-6">
      <button class="basic-button" routerLink="/my-collections">
        <a style="color:#222222;">My Collections</a>
      </button>
    </div>
  </div>
</div>

```

TS

```
import { Component, OnInit } from '@angular/core';
import { AuthService } from 'src/app/services/auth.service';
@Component({
  selector: 'app-dashboard',
  templateUrl: './dashboard.component.html',
  styleUrls: ['./dashboard.component.scss'],
})
export class DashboardComponent implements OnInit {

  constructor(public authService: AuthService) { }
  ngOnInit(): void {
    this.authService.getMoreUserInfo();
  }
}
```

Forgot Password Component

Html

```
<div class="displayTable">
  <div class="displayTableCell">
    <div class="authBlock">
      <h3 style="margin-left: 10px;">Reset Password</h3>
      <p style="margin-left: 10px; max-width: 400px;">Please enter your email
address to request a password reset.</p>
      <div class="formGroup">
        <input type="email" class="input-style" placeholder="Email Address"
#passwordResetEmail required>
      </div>
      <!-- Calling ForgotPassword from AuthService Api -->
      <div class="formGroup" style="margin-left: 10px;">
        <input type="submit" class="basic-button" value="Reset Password"
(click)="authService.ForgotPassword(passwordResetEmail.value)">
      </div>
    </div>
    <div class="redirectToLogin" style="margin-left: 10px;">
      <span class="redirect small-link" routerLink="/sign-in">Sign In</span>
    </div>
    <div class="alert-msg" *ngIf="authService.alertMsgFirebase">
      <p>{{authService.alertMsgFirebase}}</p>
    </div>
  </div>
</div>
```

Scss

```
.alert-msg {  
  p {  
    color: coral;  
    max-width: 400px;  
    margin-left: 10px;  
  }  
}
```

Sign in Component

Html

```
<div class="displayTable d-flex">  
  <div class="displayTableCell">  
    <div class="authBlock">  
      <h3 style="margin-left: 10px;">Sign In</h3>  
      <div class="formGroup">  
        <input type="text" class="input-style" placeholder="Email" #userName  
required>  
      </div>  
      <div class="formGroup">  
        <input type="password" class="input-style" placeholder="Password"  
#userPassword required>  
      </div>  
      <!-- Calling SignIn Api from AuthService -->  
      <div class="formGroup" style="margin-left: 10px;">  
        <input type="button" class="basic-button" value="Sign in"  
        (click)="authService.SignIn(userName.value, userPassword.value)">  
      </div>  
      <div class="forgotPassword">  
        <span routerLink="/forgot-password" class="small-link" style="margin-  
left: 10px;">Forgot Password</span>  
      </div>  
      <div class="redirectToLogin" style="margin-left: 10px;">  
        <span class="redirect" routerLink="/register-user" class="small-  
link">Create Account</span>  
      </div>  
      <div *ngIf="authService.errorMsgFirebase" class="error-msg mt-3">  
        <p>{{authService.errorMsgFirebase}}</p>  
      </div>  
    </div>  
  </div>
```



Scss

```
.model-back {
  model-viewer {
    width: 100%;
    height: 500px;
  }
}

.error-msg {
  p {
    color: brown;
    max-width: 400px;
    margin-left: 10px;
  }
}
```

Sign up Component

Html

```
<div class="displayTable">
  <div class="displayTableCell">
    <div class="authBlock">
      <h3 style="margin-left: 10px;">Sign Up</h3>
      <div class="formGroup">
        <input type="text" class="input-style" placeholder="First Name"
#userFirstName required />
        <span class="error" *ngIf="errors"> {{errors.firstName}}</span>
      </div>
      <div class="formGroup">
        <input type="text" class="input-style" placeholder="Last Name"
#userLastName required />
        <span class="error" *ngIf="errors"> {{errors.lastName}}</span>
      </div>
      <div class="formGroup">
        <input type="text" class="input-style" placeholder="Username" #username
required />
        <span class="error" *ngIf="errors"> {{errors.username}}</span>
      </div>
      <div class="formGroup">
        <input type="email" class="input-style" placeholder="Email Address"
#userEmail required />
        <span class="error" *ngIf="errors"> {{errors.email}}</span>
      </div>
    </div>
  </div>
</div>
```

```

    <div class="formGroup">
      <input type="password" class="input-style" placeholder="Password"
#userPwd required />
      <span class="error" *ngIf="errors"> {{errors.password}}</span>
    </div>
    <div class="formGroup" style="margin-left: 10px;">
      <input type="button" class="basic-button" value="Sign Up"
      (click)="authService.registerUser(userEmail.value,
userPwd.value, userFirstName.value, userLastName.value, username.value)" />
    </div>
  </div>
  <div class="redirectToLogin" style="margin-left: 10px;">
    <span class="redirect small-link" routerLink="/sign-in">Sign In</span>
  </div>
  <div *ngIf="authService.errorMsgFirebase" class="error-msg">
    <p>{{authService.errorMsgFirebase}}</p>
  </div>
</div>
</div>

```

Scss

```

.error{
  color:brown;
  font-size: 12px;
  display: block;
  margin-left: 10px;
  max-width: 400px;
}

.error-msg{
  p{
    color: brown;
    max-width: 400px;
    margin-left: 10px;
  }
}

```

TS

```

import { Errors } from '../../../interfaces/errors';
import { Component, OnInit } from '@angular/core';
import { AuthService } from 'src/app/services/auth.service';

@Component({
  selector: 'app-sign-up',

```

```

        templateUrl: './sign-up.component.html',
        styleUrls: ['./sign-up.component.scss']
    })
    export class SignUpComponent implements OnInit {

        public errors: Errors = {
            email: '',
            username: '',
            password: '',
            firstName: '',
            lastName: ''
        }

        constructor(public authService: AuthService) { }

        ngOnInit(): void {
            this.getErrors();
        }

        public getErrors() {
            this.authService.errors$.subscribe(data => {
                this.errorPacker(data.errors);
            })
        }

        errorPacker(data: any) {
            //Resetting errors
            this.errors.email = '';
            this.errors.username = '';
            this.errors.firstName = '';
            this.errors.lastName = '';
            this.errors.password = '';

            //Packing errors
            data.forEach((e: any) => {
                switch(e.PropertyName){
                    case "Username" : this.errors.username = e.ErrorMessage;
                    break;
                    case "Email" : this.errors.email = e.ErrorMessage;
                    break;
                    case "FirstName" : this.errors.firstName = e.ErrorMessage;
                    break;
                    case "LastName" : this.errors.lastName = e.ErrorMessage;
                    break;
                    case "Password" : this.errors.password += " " + e.ErrorMessage;
                    break;
                }
            })
        }
    }

```

```

    }
  });
}
}

```

## Verify Email Component

Html

```

<div class="displayTable d-flex justify-content-center mt-5 pt-5">
  <div class="displayTableCell">
    <div class="authBlock">
      <h3 style="text-align: center;">Thank You for Registering</h3>
      <div class="formGroup" *ngIf="authService.userData as user">
        <p class="text-center">We have sent a confirmation email to
<strong>{{user.email}}</strong>.</p>
        <p class="text-center">Please check your email and click on the link to
verfiy your email address.</p>
      </div>

      <!-- Calling SendVerificationMail() method using authService Api -->
      <div class="formGroup d-flex flex-column justify-content-center">
        <div class="col-12 d-flex justify-content-center">
          <button type="button" class="basic-button"
(click)="authService.SendVerificationMail()">
            <i class="fas fa-redo-alt"></i>
            Resend Verification Email
          </button>
        </div>
        <div class="col-12 d-flex justify-content-center">
          <button type="button" class="basic-button mt-3" routerLink="/sign-in">
            <i class="fas fa-redo-alt"></i>
            Sign In
          </button>
        </div>
      </div>
    </div>
  </div>
</div>

```

## Collection Component

Html

```

<div class="row" *ngIf="collection">
  <div class="background-image col-12">

```

```

        style="background-
image:url('../.../assets/images/{{collection.backgroundImage}}')">
    </div>
    <div class="image-info-container">
        <div class="collection-image">
            
        </div>
    </div>
</div>
<div class="row" *ngIf="collection">
    <div class="image-info-container">
        <div class="collection-info">
            <div class="author">
                <h1 class="title">{{collection.title}}</h1>
                <p>By : {{collection.username}}</p>
            </div>
        </div>
    </div>
</div>
<div class="container" *ngIf="collection">
    <div class="row">
        <div class="col-12 d-flex flex-wrap justify-content-around">
            <app-collection-item class="col-lg-3 col-md-6 col-sm-6 col-12"
*ngFor="let item of items" [itemData]="item"
                [title]="collection.title" [mode]='none'
[deleteColl]="deleteColl">
            </app-collection-item>
        </div>
    </div>
</div>
</div>

```

Scss

```

.background-image {
    height: 50vh;
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;

    img {
        height: 100%;
        width: 100%;
    }
}

```

```

    }
}

.row {
    position: relative;
}

.collection-info {
    font-family: "Agrandir";
    padding-top: 2rem;

    .title {
        font-family: "Bagerich";
        font-size: 4rem;
    }
}

.image-info-container {
    width: 100%;
    padding: 0px 10%;
}

.collection-image {
    position: absolute;
    bottom: -10%;
    width: 200px;
    border: 5px solid white;
    overflow: hidden;
    display: flex;
    align-items: center;
    justify-content: center;

    img {
        width: 100%;
    }
}

@media only screen and (max-width: 992px) {
    .image-info-container {
        display: flex;
        justify-content: center;

        .collection-info {
            text-align: center;
        }
    }
}

```

```
}  
}
```

TS

```
import { Component, OnInit } from '@angular/core';  
import { ActivatedRoute } from '@angular/router';  
import { Collections } from 'src/app/interfaces/collections';  
import { CollectionsServices } from 'src/app/services/collections.services';  
  
@Component({  
  selector: 'app-collection',  
  templateUrl: './collection.component.html',  
  styleUrls: ['./collection.component.scss']  
})  
export class CollectionComponent implements OnInit {  
  
  private sub!: any;  
  private collectionTitle!: string;  
  public deleteColl!: number;  
  
  public collection!: Collections;  
  public item!: any;  
  public items!: any[];  
  
  constructor(private route: ActivatedRoute, private collectionServices:  
CollectionsServices) { }  
  
  ngOnInit(): void {  
    this.getCollectionName();  
    this.getCollection();  
  }  
  
  getCollectionName() {  
    this.sub = this.route.params.subscribe(params => {  
      this.collectionTitle = params['title'].replaceAll('-', ' ');  
    });  
  }  
  
  getCollection(){  
    this.collectionServices.getCollection(this.collectionTitle).subscribe(data  
=>{  
      this.collection = data.data[0];  
      this.item = data.data[0].items[0];  
      this.items = data.data[0].items;
```

```

        if(this.items.length == 1){
            this.deleteColl = this.collection.id;
        }
    })
}
}

```

## Collection Item Component

Html

```

<div class="card" *ngIf="itemData">
    <div class="content">
        <div class="imgBx">
            <div *ngIf="itemData.image">
                <img [src]="imageSrc">
            </div>
        </div>
    </div>
    <div class="item-info">
        <a [routerLink]="['/collection',transformUrl(title),
transformUrl(itemData.title)]">
            <h5>{{itemData.title}}</h5>
        </a>
        <a *ngIf="mode == 'editor' && deleteColl == undefined"
(click)="deleteMe(itemData.id)">
            <p class="delete">Delete</p>
        </a>
        <a *ngIf="mode == 'editor' && deleteColl"
(click)="deleteCollection(deleteColl)">
            <p class="delete">Delete</p>
        </a>
    </div>
</div>

```

Scss

```

.card {
    position: relative;
    height: 300px;
    background: rgba(255, 255, 255, 0.05);
    margin: 20px;
    display: flex;

```



```
justify-content: center;
align-items: center;
overflow: hidden;

box-shadow: 0px 10px 1px rgba(221, 221, 221, 1), 0 10px 20px rgba(204, 204,
204, 1);

.content {
  position: relative;
  display: flex;
  width: 100%;
  justify-content: center;
  align-items: center;
  flex-direction: column;
  transition: 0.5s;

  .imgBx {
    position: relative;
    width: 100%;
    height: 300px;
    overflow: hidden;

    img {
      position: absolute;
      inset: 0px;
      box-sizing: border-box;
      padding: 0px;
      border: none;
      margin: auto;
      display: block;
      width: 100%;
    }
  }
}

.item-info {
  position: absolute;
  bottom: -70px;
  width: 100%;
  height: 70px;
  transition: 0.5s;
  background-color: #fff;

  display: flex;
```

```

        flex-direction: column;
        justify-content: center;
        align-items: center;

        font-family: "Bagerich";
        h5{
            text-align: center;
        }
        a{
            color: #212529;
            width: 100%;
            position: relative;
        }

        .delete{
            width: 100%;
            background-color: brown;
            color:white;
            text-align: center;
            cursor: pointer;
        }
    }

    &:hover .item-info {
        transform: translateY(-55px);
    }
}

```

TS

```

import { FileService } from './../../../services/files.service';
import { CollectionsServices } from 'src/app/services/collections.services';
import { ItemsService } from './../../../services/items.service';
import { Component, Input, OnInit } from '@angular/core';

@Component({
  selector: 'app-collection-item',
  templateUrl: './collection-item.component.html',
  styleUrls: ['./collection-item.component.scss']
})
export class CollectionItemComponent implements OnInit {

  @Input() itemData!: any;
  @Input() title!: string;

```

```

@Input() model!: string;
@Input() deleteColl!: number;

public imageSrc!: string;

constructor(
  private itemsService: ItemsService,
  private collectionsService: CollectionsServices,
  private filesService: FilesService
) { }

ngOnInit(): void {
  this.filesService.getImage(this.itemData.image).subscribe((data: any) => {
    this.imageSrc = data;
  })
}

public transformUrl(data: string): string {
  return data.replace(/\\s/g, '-');
}

public deleteMe(id: number) {
  this.itemsService.deleteItem(id);
}

public deleteCollection(id: number) {
  this.collectionsService.deleteCollection(id);
}
}

```

## Item Component

### Html

```

<div class="container" *ngIf="item">
  <div class="" *ngIf="item.model != 'nomodel'; else noModel">
    <div class="row">
      <div class="image col-lg-6 col-12 mt-5 mb-5 d-flex justify-content-center"></div>
      <div class="model col-lg-6 col-12 mt-5 mb-5">
        <div class="model-back">
          <model-viewer src="assets/3Dmodels/{{item.model}}/scene.gltf"
alt="fungus model" camera-controls ar
ios-
src="assets/3Dmodels/AmanitaMuscaria/scene.gltf"></model-viewer>

```

```

        </div>
      </div>
    </div>
  </div>
  <ng-template #noModel>
    <div class="row d-flex justify-content-center">
      <div class="image col-6 mt-5 mb-5 d-flex justify-content-center"></div>
      </div>
    </ng-template>
    <div class="title">
      {{item.title}}
    </div>
    <div class="row">
      <app-item-infos class="col-6 mt-5" *ngFor="let info of infos"
[info]="info">
      </app-item-infos>
    </div>
  </div>
</div>

```

Scss

```

.image{
  padding: 5%;
  img{
    width: 100%;
  }
}
.title{
  width: 100%;
  font-family: Bagerich;
  font-size: 3rem;
  text-align: center;
}
.model-back{
  model-viewer{
    width: 100%;
    height: 500px;
  }
}

```

TS

```
import { Infos } from './../../interfaces/infos';
import { ItemsService } from './../../services/items.service';
import { Observable } from 'rxjs';
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { Items } from 'src/app/interfaces/items';

@Component({
  selector: 'app-item',
  templateUrl: './item.component.html',
  styleUrls: ['./item.component.scss']
})
export class ItemComponent implements OnInit {

  private sub!: any;
  private itemName!: string;

  public item!: Items;
  public infos!: [Infos]

  constructor(private route: ActivatedRoute, private itemServices: ItemsService)
{ }

  ngOnInit(): void {
    this.getItemName();
    this.getItems();
  }

  getItemName() {
    this.sub = this.route.params.subscribe((params: { [x: string]: any; }) => {
      this.itemName = params['name'].replaceAll('-', ' ');
    });
  }

  public getItems() {
    this.itemServices.getItems(this.itemName).subscribe(data => {
      this.item = data.data[0];
      this.infos = data.data[0].collectionItemInfos;
    })
  }
}
```

## Item Info Component

Html

```
<h2 *ngIf="info.title != ''; else invisHeading">{{info.title}}</h2>
<ng-template #invisHeading><h2 class="invisible">NO HEADER</h2></ng-template>
<p>{{info.content}}</p>
```

Scss

```
h2{
  font-family: "Bagerich";
}
.invisible{
  opacity: 100%;
}
p{
  text-align: justify;
}
```

## Create Collection Component

Html

```
<div class="container">
  <div class="row mt-5 pt-5 d-flex justify-content-center">
    <div class="col-6">
      <h1>Create new Collection</h1>
      <form [formGroup]="createCollection">
        <div class="row">
          <input hidden [value]="userId" />
          <mat-form-field appearance="outline">
            <mat-label>Collection Title <span class="req">*</span>
          </mat-label><br>
          <input matInput formControlName="title" class="input-style" />
          <div *ngIf="title?.invalid && (title?.dirty || title?.touched)" class="alert alert-danger">
            <mat-error>
              <ng-container
                *ngIf="title?.errors?.['required']">
                  Collection Title is required.
                </ng-container>
            </mat-error>
          </div>
        </mat-form-field>
        <mat-form-field appearance="outline">
```

```

<mat-label>Category <span class="req">*</span></mat-
label><br>
<select formControlName="categoryId" id="category"
class="input-style">
    <option *ngFor="let category of categories; let i =
index"
        [value]="categories[i].categoryId">
            {{categories[i].title}}
        </option>
</select>
<div *ngIf="categoryId?.invalid && (categoryId?.dirty ||
categoryId?.touched)"
    class="alert alert-danger">
        <mat-error>
            <ng-container
*ngIf="categoryId?.errors?.['required']">
                Category is required.
            </ng-container>
            <ng-container
*ngIf="categoryId?.errors?.['minlength']">
                The field is under min Len.
            </ng-container>
        </mat-error>
    </div>
</mat-form-field>
</div>
<mat-form-field appearance="outline">
    <mat-label>BackgroundImage <span class="req">*</span></mat-
label>
    <input (change)="getFile($event)" placeholder="Picture"
formControlName="backgroundImage"
    type="file" multiple="multiple" required>
    <div *ngIf="backgroundImage?.invalid &&
(backgroundImage?.dirty || backgroundImage?.touched)"
    class="alert alert-danger">
        <mat-error>
            <ng-container
*ngIf="backgroundImage?.errors?.['required']">
                Image is required.
            </ng-container>
        </mat-error>
    </div>
</mat-form-field>
<div class="row">
    <div class="col-12">

```

```

        <button class='basic-button' mat-flat-button
color="primary" (click)="onSubmit()"
        [disabled]="createCollection.invalid">Create
Collection</button>
    </div>
</div>
</form>
<div *ngIf="collectionsService.message" class="succ-msg">
    <p>{{collectionsService.message}}</p>
</div>
<div *ngIf="errors" class="error-msg">
    <p>{{errors}}</p>
</div>
</div>
</div>
</div>

```

Scss

```

.error-msg{
    p{
        color: brown;
        max-width: 400px;
        margin-left: 10px;
    }
}

.succ-msg{
    p{
        color: greenyellow;
        max-width: 400px;
        margin-left: 10px;
    }
}

.input-style{
    margin-left: 0px;
    margin-bottom: 20px;
}

```

TS

```

import { FileService } from '../services/files.service';
import { AuthService } from 'src/app/services/auth.service';
import { ICategories } from '../interfaces/categories';
import { CategoriesService } from '../services/categories.service';
import { CollectionsServices } from 'src/app/services/collections.services';

```



```

import { Component, OnInit } from '@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';

@Component({
  selector: 'app-create-collection',
  templateUrl: './create-collection.component.html',
  styleUrls: ['./create-collection.component.scss']
})
export class CreateCollectionComponent implements OnInit {

  public createCollection = new FormGroup({
    title: new FormControl('', [Validators.required]),
    categoryId: new FormControl('', [Validators.required]),
    backgroundImage: new FormControl('', [Validators.required])
  });

  public userId!: number;
  public image!: any;
  public errors: string = '';

  public categories!: ICategories[];
  constructor(
    public collectionsService: CollectionsServices,
    private categoriesService: CategoriesService,
    private filesService: FilesService,
    private authService: AuthService
  ) { }

  ngOnInit(): void {
    this.getUserId();
    this.getCategories();
  }

  onSubmit(): void {
    if (this.createCollection.valid && this.errors == '') {
      const image = this.backgroundImage?.value.split("\\\\")[2];
      const newCollection = JSON.stringify({
        'Title': this.title?.value,
        'BackgroundImage': image,
        'CategoryId': this.categoryId?.value,
        'UserId': this.userId
      });

      this.collectionsService.createCollection(newCollection);
    }
  }
}

```

```

    this.createCollection.reset();
  }

  getCategories() {
    this.categoriesService.getCategories().subscribe((data: any) => {
      this.categories = data;
    })
  }

  getUserId() {
    const email = JSON.parse(localStorage.getItem('user')).email;
    this.authService.getUserId(email).subscribe(data => {
      this.userId = data.id;
    })
  }

  getFile(file: any) {
    this.errors = '';
    let extensionAllowed = ['png', 'jpeg', 'jpg']

    if (file.target.files[0].size / 1024 / 1024 > 20) {
      this.errors = "File size should be less than 20MB";
      return;
    }
    if (extensionAllowed) {
      var nam = file.target.files[0].name.split('.').pop();
      if (!extensionAllowed.includes(nam)) {
        this.errors = "Please upload " + Object.values(extensionAllowed) + "
file.";
        return;
      }
    }
    this.filesService.imageUploadFirebase(file.target.files[0]);
  }

  get backgroundImage() { return this.createCollection.get('backgroundImage'); }
  get title() { return this.createCollection.get('title'); }
  get categoryId() { return this.createCollection.get('categoryId'); }
}

```

## Create Item Component

Html

```
<div class="container">
  <div class="row mt-5 pt-5 d-flex justify-content-center">
    <div class="col-6">
      <h1>Create new Item</h1>
      <form [formGroup]="createItem">
        <div class="row">
          <input hidden [value]="collectionId" />
          <mat-form-field appearance="outline">
            <mat-label>Item Title <span class="req">*</span> </mat-label><br>
            <input matInput formControlName="title" class="input-style" style="margin-left: 0px;"/>
            <div *ngIf="title?.invalid && (title?.dirty || title?.touched)" class="alert alert-danger">
              <mat-error>
                <ng-container
                  *ngIf="title?.errors?.['required']">
                    Item Title is required.
                  </ng-container>
                </mat-error>
              </div>
            </mat-form-field>
          </div>
          <mat-form-field appearance="outline">
            <mat-label>Item Image <span class="req">*</span></mat-label>
            <input (change)="getFile($event)" placeholder="Picture"
              formControlName="itemImage" type="file"
              multiple="multiple" required>
            <div *ngIf="itemImage?.invalid && (itemImage?.dirty || itemImage?.touched)"
              class="alert alert-danger">
              <mat-error>
                <ng-container
                  *ngIf="itemImage?.errors?.['required']">
                    Image is required.
                  </ng-container>
                </mat-error>
              </div>
            </mat-form-field>
          <div class="row">
            <div class="col-12">
              <button class='basic-button' mat-flat-button
                color="primary" (click)="onSubmit()">
```

```

        [disabled]="createItem.invalid">Create Item</button>
      </div>
    </div>
  </form>
  <div *ngIf="itemsService.message" class="succ-msg">
    <p>{{itemsService.message}}</p>
  </div>
  <div *ngIf="errors" class="error-msg">
    <p>{{errors}}</p>
  </div>
</div>
</div>
</div>

```

Scss

```

.succ-msg{
  p{
    color: greenyellow;
    max-width: 400px;
    margin-left: 10px;
  }
}

```

TS

```

import { FilesService } from './../../../../services/files.service';
import { ItemsService } from './../../../../services/items.service';
import { Component, OnInit } from '@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';
import { ActivatedRoute } from '@angular/router';

@Component({
  selector: 'app-create-item',
  templateUrl: './create-item.component.html',
  styleUrls: ['./create-item.component.scss']
})
export class CreateItemComponent implements OnInit {

  public createItem = new FormGroup({
    title: new FormControl('', [Validators.required]),
    itemImage: new FormControl('', [Validators.required])
  });

  public collectionId!: number;
  public image!: any;

```

```

public errors: string = '';

constructor(
  public itemsService: ItemsService,
  private route: ActivatedRoute,
  private filesServices: FilesService
) { }

ngOnInit(): void {
  this.getCollectionId();
}

onSubmit(): void {
  if (this.createItem.valid && this.errors == '') {
    const image = this.itemImage?.value.split("\\")[2];
    const newItem = JSON.stringify({
      'Title': this.title?.value,
      'Image': image,
      'Model': "nomodel",
      'CollectionId': this.collectionId
    });

    this.itemsService.createCollectionItem(newItem);
  }
  this.createItem.reset();
}

getCollectionId() {
  this.route.params.subscribe(params => {
    this.collectionId = params['id'];
  });
}

getFile(file: any) {
  this.errors = '';
  let extensionAllowed = ['png', 'jpeg', 'jpg']

  if (file.target.files[0].size / 1024 / 1024 > 20) {
    this.errors = "File size should be less than 20MB";
    return;
  }
  if (extensionAllowed) {
    var nam = file.target.files[0].name.split('.').pop();
    if (!extensionAllowed.includes(nam)) {

```

```

        this.errors = "Please upload " + Object.values(extensionAllowed) + "
file.";
        return;
    }
}
this.filesServices.imageUploadFirebase(file.target.files[0]);
}

get itemImage() { return this.createItem.get('itemImage'); }
get title() { return this.createItem.get('title'); }
get CollectionId() { return this.createItem.get('categoryId'); }
}

```

## Edit Collection Component

Html

```

<div *ngIf="collection && myCollection == true; else notAllowed">
    <div class="row">
        <div class="background-image col-12"
            style="background-
image:url('../.../assets/images/{{collection.backgroundImage}}')">
        </div>
        <div class="image-info-container">
            <div class="collection-image">
                
            </div>
        </div>
    </div>
    <div class="row" *ngIf="collection">
        <div class="image-info-container">
            <div class="collection-info">
                <div class="author">
                    <h1 class="title">{{collection.title}}</h1>
                    <a class='basic-button'
(click)="deleteCollection(collection.id)">Delete Collection</a> or
                    <a class='basic-button' [routerLink]="['/update-collection',
collection.title]">Update Collection</a>
                    <p style="margin-top: 10px;">By : {{collection.username}}</p>
                </div>
            </div>
        </div>
    </div>
    <div class="container" *ngIf="collection">

```

```

        <div class="row">
            <div class="col-12 d-flex flex-wrap justify-content-around">
                <app-collection-item class="col-lg-3 col-md-6 col-sm-6 col-12"
*ngFor="let item of items"
                    [itemData]="item" [title]="collection.title"
[mode]="'editor'" [deleteColl]="deleteColl">
                    </app-collection-item>
                    <div class="add-collection col-lg-3 col-md-6 col-sm-6 col-12 ms-3
me-3 mb-5 mt-3 d-flex justify-content-center align-items-center"
                        [routerLink]="['/create-item', collection.id]">
                            <div class="add d-flex justify-content-center align-items-
center">
                                <span class="up"></span>
                                <span class="down"></span>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
</div>
<ng-template #notAllowed>
    <div class="container mt-5 pt-5">
        <div class="row mt-5 pt-5">
            <div class="col-12 d-flex justify-content-center">
                <h1>You can't Edit others Collection.</h1>
            </div>
        </div>
    </div>
</ng-template>

```

Scss

```

.background-image {
    height: 50vh;
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;

    img {
        height: 100%;
        width: 100%;
    }
}

```

```
.row {
  position: relative;
}

.collection-info {
  font-family: "Agrandir";
  padding-top: 2rem;

  .title {
    font-family: "Bagerich";
    font-size: 4rem;
  }
}

.image-info-container {
  width: 100%;
  padding: 0px 10%;
}

.collection-image {
  position: absolute;
  bottom: -10%;
  width: 200px;
  border: 5px solid white;
  overflow: hidden;
  display: flex;
  align-items: center;
  justify-content: center;

  img {
    width: 100%;
  }
}

.add-collection {
  height: 300px;
  cursor: pointer;
  box-shadow: rgba(0, 0, 0, 0.2) 0px 60px 40px -7px;

  .add {
    position: relative;

    span {
      position: absolute;
      width: 50px;
    }
  }
}
```



```

        border-bottom: 2px solid #222222;
    }

    .up {
        transform: rotate(90deg);
    }
}

.basic-button {
    padding: 10px;
    color: #222222;
}

@media only screen and (max-width: 992px) {
    .image-info-container {
        display: flex;
        justify-content: center;

        .collection-info {
            text-align: center;
        }
    }
}

```

TS

```

import { Component, Input, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { Collections } from 'src/app/interfaces/collections';
import { CollectionsServices } from 'src/app/services/collections.services';

@Component({
    selector: 'app-edit-collection',
    templateUrl: './edit-collection.component.html',
    styleUrls: ['./edit-collection.component.scss']
})
export class EditCollectionComponent implements OnInit {

    private collectionTitle!: string;

    public collection!: Collections;
    public item!: any;
    public items!: any[];

```

```

public deleteColl!: number;

public myCollection: boolean = false;

constructor(private route: ActivatedRoute, private collectionServices:
CollectionsServices) { }

ngOnInit(): void {
  this.getUserCollections();
  this.getCollectionName();
  this.getCollection();
}

getCollectionName() {
  this.route.params.subscribe(params => {
    this.collectionTitle = params['collection'].replaceAll('-', ' ');
  });
}

getCollection() {
  this.collectionServices.getCollection(this.collectionTitle).subscribe(data =>
{
    this.collection = data.data[0];
    this.item = data.data[0].items[0];
    this.items = data.data[0].items;

    if (this.items.length == 1) {
      this.deleteColl = this.collection.id;
    }
  })
}

deleteCollection(id: number) {
  this.collectionServices.deleteCollection(id);
}

getUserCollections() {
  const email = JSON.parse(localStorage.getItem('user')!).email;
  this.collectionServices.getUserCollections(email).subscribe(data => {
    data.data.forEach((e) => {
      if (e.title == this.collectionTitle) {
        this.myCollection = true;
      }
    })
  });
}

```

```
}  
  
}
```

## My Collections Component

Html

```
<div class="container">  
  <div class="row mt-5 pt-5">  
    <div class="header">  
      <h1>My Collections</h1>  
    </div>  
    <div class="col-12 d-flex flex-wrap justify-content-center">  
      <app-single-collection class="col-lg-2 col-md-3 col-12 ms-3 me-3 mb-5  
mt-3"  
        *ngFor="let collection of collections" [data]="collection"  
[link]=" 'edit-collection' ">  
        </app-single-collection>  
        <div class="add-collection col-lg-2 col-md-3 col-12 ms-3 me-3 mb-5  
mt-3 d-flex justify-content-center align-items-center"  
          routerLink="/create-collection">  
            <div class="add d-flex justify-content-center align-items-  
center">  
              <span class="up"></span>  
              <span class="down"></span>  
            </div>  
          </div>  
        </div>  
      </div>  
    </div>  
  </div>
```

Scss

```
.add-collection {  
  cursor: pointer;  
  height: 240px ;  
  box-shadow: rgba(0, 0, 0, 0.2) 0px 60px 40px -7px;  
  .add {  
    position: relative;  
    span {  
      position: absolute;  
      width: 50px;  
      border-bottom: 2px solid #222222;  
    }  
  }  
}
```

```

        .up {
            transform: rotate(90deg);
        }
    }
}

```

TS

```

import { CollectionsServices } from 'src/app/services/collections.services';
import { Observable } from 'rxjs';
import { Component, OnInit } from '@angular/core';
import { Collections } from 'src/app/interfaces/collections';
import { AuthService } from 'src/app/services/auth.service';

@Component({
  selector: 'app-my-collections',
  templateUrl: './my-collections.component.html',
  styleUrls: ['./my-collections.component.scss']
})
export class MyCollectionsComponent implements OnInit {

  collections!: Collections[];
  link!: string;

  constructor(private collectionsServices: CollectionsServices, private
  authService: AuthService) { }

  ngOnInit(): void {
    this.authService.getMoreUserInfo();
    this.getUserCollections();
  }

  getUserCollections() {
    const email = JSON.parse(localStorage.getItem('user')!).email;
    this.collectionsServices.getUserCollections(email).subscribe(data =>{
      this.collections = data.data;
    });
  }
}

```

## Update Collection Component

Html

```
<div class="container">
  <div class="row mt-5 pt-5 d-flex justify-content-center">
    <div class="col-6">
      <h1>Update Collection</h1>
      <form [formGroup]="updateCollection">
        <div class="row">
          <mat-form-field appearance="outline">
            <mat-label>Collection Title <span class="req">*</span>
          </mat-label><br>
            <input matInput formControlName="title" class="input-
style" />
            <div *ngIf="title?.invalid && (title?.dirty ||
title?.touched)" class="alert alert-danger">
              <mat-error>
                <ng-container
*ngIf="title?.errors?.['required']">
                  Collection Title is required.
                </ng-container>
              </mat-error>
            </div>
          </mat-form-field>
          <mat-form-field appearance="outline">
            <mat-label>Category <span class="req">*</span></mat-
label><br>
            <select formControlName="categoryId" id="category"
class="input-style">
              <option *ngFor="let category of categories; let i =
index"
                [value]="categories[i].categoryId">
                  {{categories[i].title}}
                </option>
            </select>
            <div *ngIf="categoryId?.invalid && (categoryId?.dirty ||
categoryId?.touched)"
              class="alert alert-danger">
                <mat-error>
                  <ng-container
*ngIf="categoryId?.errors?.['required']">
                    Category is required.
                  </ng-container>
                  <ng-container
*ngIf="categoryId?.errors?.['minlength']">
                    The field is under min Len.
```

```

        </ng-container>
        </mat-error>
      </div>
    </mat-form-field>
  </div>
  <div class="row">
    <div class="col-12">
      <button class='basic-button' mat-flat-button
color="primary" (click)="onSubmit()"
[disabled]="updateCollection.invalid">Update
Collection</button>
    </div>
  </div>
</form>
<div *ngIf="collectionsService.message" class="succ-msg">
  <p>{{collectionsService.message}}</p>
</div>
<div *ngIf="errors" class="error-msg">
  <p>{{errors}}</p>
</div>
</div>
</div>
</div>

```

Scss

```

.succ-msg{
  p{
    color: greenyellow;
    max-width: 400px;
    margin-left: 10px;
  }
}

```

TS

```

import { Collections } from './../../../../interfaces/collections';
import { Component, OnInit } from '@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';
import { ActivatedRoute } from '@angular/router';
import { ICategories } from 'src/app/interfaces/categories';
import { AuthService } from 'src/app/services/auth.service';
import { CategoriesService } from 'src/app/services/categories.service';
import { CollectionsServices } from 'src/app/services/collections.services';
import { FilesService } from 'src/app/services/files.service';

```

```

@Component({
  selector: 'app-update-collection',
  templateUrl: './update-collection.component.html',
  styleUrls: ['./update-collection.component.scss']
})
export class UpdateCollectionComponent implements OnInit {

  public updateCollection = new FormGroup({
    title: new FormControl('', [Validators.required]),
    categoryId: new FormControl('', [Validators.required]),
  });

  public userId!: number;
  public image!: any;
  public errors: string = '';
  private collectionTitle!: string;

  private collectionData!: any;

  public categories!: ICategories[];
  constructor(
    public collectionsService: CollectionsServices,
    private categoriesService: CategoriesService,
    private authService: AuthService,
    private route: ActivatedRoute
  ) { }

  ngOnInit(): void {
    this.getCategories();
    this.getCollectionName();
    this.getCollectionData();
  }

  onSubmit(): void {
    if (this.updateCollection.valid && this.errors == '') {
      const newCollection = JSON.stringify({
        'Id' : this.collectionData.id,
        'Title': this.title?.value,
        'CategoryId': this.categoryId?.value,
        'BackgroundImage': null
      });

      this.collectionsService.updateCollection(newCollection);
    }
  }
}

```

```

    this.updateCollection.reset();
  }

  setForm() {
    if (this.categories) {
      var catId = 0;

      this.categories.forEach(e => {
        if (e.title == this.collectionData.category) {
          catId = e.categoryId;
        }
      })

      this.updateCollection.patchValue({
        title: this.collectionData.title,
        categoryId: catId
      })
    }
  }

  getCollectionName() {
    this.route.params.subscribe(params => {
      this.collectionTitle = params['title'].replaceAll('-', ' ');
    });
  }

  getCollectionData() {
    this.collectionsService.getCollection(this.collectionTitle).subscribe(data =>
  {
    this.collectionData = data.data[0];
    this.setForm();

  });
  }

  getCategories() {
    this.categoriesService.getCategories().subscribe((data: any) => {
      this.categories = data;
    })
  }

  get title() { return this.updateCollection.get('title'); }
  get categoryId() { return this.updateCollection.get('categoryId'); }
}

```



## Interfaces

### Categories

```
export interface ICategories {  
  categoryId: number,  
  title: string  
}
```

### Collections

```
export interface Collections {  
  id : number;  
  title: string;  
  backgroundImage: string;  
  username: string;  
  category: string;  
  items: [number];  
}
```

### Errors

```
export interface Errors {  
  password: string,  
  email: string,  
  username: string,  
  firstName: string,  
  lastName: string  
}
```

### Infos

```
export interface Infos {  
  id : number;  
  title: string;  
  content: string;  
}
```

### Items

```
export interface Items {  
  id: number;  
  title: string;  
  image: string;
```

```
model: string;
infos: [string];
}
```

## Paginate

```
export interface Paginate {
  page: number;
  totalCount: number;
  perPage: number;
  pagesCount: number;
  data:[any];
}
```

## User

```
export interface User {
  uid: string;
  email: string;
  photoURL: string;
  emailVerified: boolean;
}
```

## Services

### Auth

```
import { HttpClient } from '@angular/common/http';
import { Injectable, NgZone } from '@angular/core';
import { AngularFireAuth } from '@angular/fire/compat/auth';
import {
  AngularFirestore,
  AngularFirestoreDocument,
} from '@angular/fire/compat/firestore';
import { Router } from '@angular/router';
import { Subject, Observable } from 'rxjs';
import { User } from '../interfaces/user';
@Injectable({
  providedIn: 'root',
})
export class AuthService {
  public userData: any; // Save logged in user data
  //User info from API
  public moreData: any = {
    username: '',
  }
```

```

    firstName: '',
    lastName: '',
    email: '',
  }
  constructor(
    public afs: AngularFirestore, // Inject Firestore service
    public afAuth: AngularFireAuth, // Inject Firebase auth service
    public router: Router,
    public ngZone: NgZone, // NgZone service to remove outside scope warning
    private http: HttpClient
  ) {
    /* Saving user data in localStorage when
    logged in and setting up null when logged out */

    this.afAuth.authState.subscribe((user) => {
      if (user) {
        this.userData = user;
        localStorage.setItem('user', JSON.stringify(this.userData));
        JSON.parse(localStorage.getItem('user')!);
      } else {
        localStorage.setItem('user', 'null');
        JSON.parse(localStorage.getItem('user')!);
      }
    });
  }

  private baseUrl: string = "https://collectio.azurewebsites.net/api/";
  public errors$: Subject<any> = new Subject<any>();
  private errorCount!: any;
  public errorMsgFirebase!: string;
  public alertMsgFirebase!: string;

  // Sign in with email/password
  SignIn(email: string, password: string) {
    return this.afAuth
      .signInWithEmailAndPassword(email, password)
      .then((result) => {
        if (result.user?.emailVerified === false) {
          throw new Error("Please verify your Email");
        }
        else {
          this.SetUserData(result.user);
          this.afAuth.authState.subscribe((user) => {
            if (user) {
              this.router.navigate(['dashboard']);
            }
          });
        }
      });
  }

```

```

        }
    });
}
})
.catch((error) => {
    this.errorMsgFirebase = error.message;
});
}

registerUser(email: string, password: string, firstName: string, lastName:
string, username: string) {

    const headers = { 'content-type': 'application/json' }
    const body = JSON.stringify({
        "Email": email,
        "Username": username,
        "FirstName": firstName,
        "LastName": lastName,
        "Password": password
    });

    this.http.post(this.baseUrl + "register", body, { 'headers': headers
}).subscribe({
    next: () => {
        this.SignUp(email, password);
    },
    error: (error) => {
        this.errors$.next(error.error)
        this.errorCount = error.error.errors;
    },
});
}

// Sign up with email/password
SignUp(email: string, password: string) {
    return this.afAuth
        .createUserWithEmailAndPassword(email, password)
        .then((result) => {
            /* Call the SendVerificationMail() function when new user sign
            up and returns promise */
            this.SendVerificationMail();
            this.SetUserData(result.user);
        })
        .catch((error) => {
            this.errorMsgFirebase = error.message;

```

```

    });
}
// Send email verification when new user sign up
SendVerificationMail() {
    return this.afAuth.currentUser
        .then((u: any) => u.sendEmailVerification())
        .then(() => {
            this.router.navigate(['verify-email-address']);
        });
}
// Reset Forggot password
ForgotPassword(passwordResetEmail: string) {
    return this.afAuth
        .sendPasswordResetEmail(passwordResetEmail)
        .then(() => {
            throw new Error("Password reseted. Check your email");
        })
        .catch((error) => {
            this.alertMsgFirebase = error.message;
        });
}
// Returns true when user is looged in and email is verified
get isLoggedIn(): boolean {
    const user = JSON.parse(localStorage.getItem('user')!);
    return user !== null && user.emailVerified !== false ? true : false;
}
// Auth logic to run auth providers
AuthLogin(provider: any) {
    return this.afAuth
        .signInWithPopup(provider)
        .then((result) => {
            this.router.navigate(['dashboard']);
            this.SetUserData(result.user);
        })
        .catch((error) => {
            window.alert(error);
        });
}
/* Setting up user data when sign in with username/password,
sign up with username/password and sign in with social auth
provider in Firestore database using AngularFirestore +
AngularFirestoreDocument service */
SetUserData(user: any) {
    const userRef: AngularFireStoreDocument<any> = this.afs.doc(
        `users/${user.uid}`
    );
}

```

```

    );
    const userData: User = {
      uid: user.uid,
      email: user.email,
      photoURL: user.photoURL,
      emailVerified: user.emailVerified,
    };
    return userRef.set(userData, {
      merge: true,
    });
  }

  getMoreUserInfo() {
    const email = JSON.parse(localStorage.getItem('user')).email;
    this.http.get(this.baseUrl + "register?keyword=" + email).subscribe((data:
any) => {
      this.moreData.username = data.username;
      this.moreData.firstName = data.firstName;
      this.moreData.lastName = data.lastName;
      this.moreData.email = data.email;
    });
  }
  // Sign out
  SignOut() {
    return this.afAuth.signOut().then(() => {
      localStorage.removeItem('user');
      this.router.navigate(['sign-in']);
      location.reload();
    });
  }
  getUserId(email: string): Observable<any>{
    return this.http.get(this.baseUrl + "/register?keyword=" + email);
  }
}

```

## Categories

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class CategoriesService {

```

```

private baseUrl: string = 'https://collectio.azurewebsites.net/api/categories'
constructor(private http: HttpClient) { }

getCategories(){
  return this.http.get(this.baseUrl);
}
}

```

## Collections

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { AngularFireStorage } from '@angular/fire/compat/storage';
import { finalize, Observable, Subject } from 'rxjs';
import { Paginate } from '../interfaces/paginate';

@Injectable({
  providedIn: 'root'
})

export class CollectionsServices {

  private baseUrl: string =
"https://collectio.azurewebsites.net/api/collections";

  public message!: string;

  public colx$: Subject<Paginate> = new Subject<Paginate>();

  constructor(private http: HttpClient) { }

  private searchUrlConstructor(urlPart: string): string {
    return this.baseUrl + '?keyword=' + urlPart;
  }

  public getCollections(): void {
    this.http.get<Paginate>(this.baseUrl).subscribe(data => {
      this.colx$.next(data);
    });
  }

  public searchCollection(keyword: string) {
    this.http.get<Paginate>(this.searchUrlConstructor(keyword)).subscribe(data =>
{

```

```

        this.colx$.next(data);
    })
}

public getCollection(collectionName: string): Observable<Paginate> {
    return this.http.get<Paginate>(this.searchUrlConstructor(collectionName));
}

public getUserCollections(email: string): Observable<Paginate> {
    return this.http.get<Paginate>(this.baseUrl + "?email=" + email);
}

public createCollection(data: string) {
    const headers = { 'content-type': 'application/json' }
    this.http.post(this.baseUrl, data, { 'headers': headers }).subscribe({
        next: () => {
            this.message = "Collection Created";
        }
    });
}

public deleteCollection(id: number) {
    this.http.delete(this.baseUrl + '/' + id).subscribe({
        next: (result) => {
            window.location.replace("http://localhost:4200/my-collections");
        }
    });
}

public updateCollection(data: string) {
    const headers = { 'content-type': 'application/json' }
    this.http.put(this.baseUrl, data, { 'headers': headers }).subscribe({
        next: () => {
            this.message = "Collection Updated";
        }
    });
}
}

```

Items

```

import { Paginate } from '../interfaces/paginate';
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

```



```

import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class ItemsService {

  private baseUrl: string = "https://collectio.azurewebsites.net/api/items";
  public message!: string;

  constructor(private http: HttpClient) { }

  private searchUrlConstructor(urlPart: string): string {
    return this.baseUrl + '?keyword=' + urlPart;
  }

  public.getItems(itemCollection: string): Observable<Paginate> {
    return this.http.get<Paginate>(this.searchUrlConstructor(itemCollection));
  }

  public createCollectionItem(data: string) {
    console.log(data);
    const headers = { 'content-type': 'application/json' }

    this.http.post(this.baseUrl, data, { 'headers': headers }).subscribe({
      next: (result) => {
        console.log(result);
        this.message = "Item created";
      }
    })
  }

  public deleteItem(id: number) {
    this.http.delete(this.baseUrl + '/' + id).subscribe({
      next: (response) => {
        location.reload();
      }
    });
  }
}

```

## Files

```
import { Injectable } from '@angular/core';
import { AngularFireStorage } from '@angular/fire/compat/storage';

@Injectable({
  providedIn: 'root'
})
export class FilesService {

  private basePath: string = '/uploads';

  constructor(private storage: AngularFireStorage) { }

  public imageUploadFirebase(file: any) {
    const filePath = `${this.basePath}/${file.name}`;
    this.storage.ref(filePath);
    this.storage.upload(filePath, file);
  }

  public getImage(image: string) {
    var imageRef = this.storage.ref(`${this.basePath}/${image}`);
    return imageRef.getDownloadURL();
  }
}
```

## Messages

```
import { Injectable } from '@angular/core';
import { HttpClient, HttpHeaders } from '@angular/common/http';

@Injectable({
  providedIn: 'root'
})
export class MessagesService {

  constructor(private http: HttpClient) { }

  sendEmail(formData: any): void {
    const headers = new HttpHeaders({ 'Content-Type': 'application/json' });

    this.http.post('https://formspree.io/f/mbjbllkj',
      { name: formData.name, replyto: formData.email, message: formData.msg },
      { 'headers': headers }).subscribe(
        response => {
```

```

        console.log(response);
    }
    );
}
}

```

#### Guard/Auth.guard

```

import { Injectable } from '@angular/core';
import { CanActivate, ActivatedRouteSnapshot, RouterStateSnapshot, Router } from
 '@angular/router';

import { Observable } from 'rxjs';
import { AuthService } from '../auth.service';
@Injectable({
  providedIn: 'root'
})
export class AuthGuard implements CanActivate {

  constructor(
    public authService: AuthService,
    public router: Router
  ){ }

  canActivate(
    next: ActivatedRouteSnapshot,
    state: RouterStateSnapshot): Observable<boolean> | Promise<boolean> | boolean
  {
    if(this.authService.isLoggedIn !== true) {
      this.router.navigate(['sign-in'])
    }
    return true;
  }
}

```

#### Animations.ts

```

import { animate, query, stagger, style, transition, trigger } from
"@angular/animations";

export const $animations = [

  trigger('fade', [
    transition('void=>*', [

```

```

        style({
            opacity: 0
        }),
        animate('3s ease-in-out')
    ])
]),
trigger('slideSide', [
    transition('void=>*', [
        style({
            transform: "translateX(-300px)"
        }),
        animate("3s ease-in-out")
    ])
]),
trigger('slideUp', [
    transition('void=>*', [
        style({
            transform: "translateY(200px)"
        }),
        animate("3s ease-in-out")
    ])
])
]);

```

## App-routing.module

```

import { UpdateCollectionComponent } from './components/user-panel/update-collection/update-collection.component';
import { CreateItemComponent } from './components/user-panel/create-item/create-item.component';
import { EditCollectionComponent } from './components/user-panel/edit-collection/edit-collection.component';
import { CreateCollectionComponent } from './components/user-panel/create-collection/create-collection.component';
import { MyCollectionsComponent } from './components/user-panel/my-collections/my-collections.component';
import { VerifyEmailComponent } from './components/auth/verify-email/verify-email.component';
import { ForgotPasswordComponent } from './components/auth/forgot-password/forgot-password.component';
import { DashboardComponent } from './components/auth/dashboard/dashboard.component';
import { SignUpComponent } from './components/auth/sign-up/sign-up.component';
import { SignInComponent } from './components/auth/sign-in/sign-in.component';

```

```
import { CollectionComponent } from
'./components/collection/collection.component';
import { AuthorComponent } from './components/author/author.component';
import { AboutComponent } from './components/about/about.component';
import { HomeComponent } from './components/home/home.component';
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { ItemComponent } from './components/item/item.component';
import { AuthGuard } from './services/guard/auth.guard';

const routes: Routes = [
  {
    path: '',
    component: HomeComponent,
    data: { title: 'Collēctiō | Home' }
  },
  {
    path: 'sign-in',
    component: SignInComponent,
    data: { title: 'Collēctiō | Sign-in' }
  },
  {
    path: 'register-user',
    component: SignUpComponent,
    data: { title: 'Collēctiō | Sign-up' }
  },
  {
    path: 'dashboard',
    component: DashboardComponent,
    data: { title: 'Collēctiō | Dashboard' },
    canActivate: [AuthGuard]
  },
  {
    path: 'forgot-password',
    component: ForgotPasswordComponent,
    data: { title: 'Collēctiō | Password-Reset' }
  },
  {
    path: 'verify-email-address',
    component: VerifyEmailComponent,
    data: { title: 'Collēctiō | Verify-Email' }
  },
  {
    path: 'collection/:title',
    component: CollectionComponent,
```

```
data: { title: 'Collēctiō | Details' }
},
{
  path: 'collection/:title/:name',
  component: ItemComponent,
  data: { title: 'Collēctiō | Details' }
},
{
  path: 'about',
  component: AboutComponent,
  data: { title: 'Collēctiō | About' }
},
{
  path: 'author',
  component: AuthorComponent,
  data: { title: 'Nikola Gutic' }
},
{
  path: 'my-collections',
  component: MyCollectionsComponent,
  data: { title: 'Collēctiō | My Collections' },
  canActivate: [AuthGuard]
},
{
  path: 'create-collection',
  component: CreateCollectionComponent,
  data: { title: 'Collēctiō | Create Collection' },
  canActivate: [AuthGuard]
},
{
  path: 'edit-collection/:collection',
  component: EditCollectionComponent,
  data: { title: 'Collēctiō | Edit Collection' },
  canActivate: [AuthGuard]
},
{
  path: 'create-item/:id',
  component: CreateItemComponent,
  data: { title: 'Collēctiō | Create Item' },
  canActivate: [AuthGuard]
},
{
  path: 'update-collection/:title',
  component: UpdateCollectionComponent,
  data: { title: 'Collēctiō | Update Collection' },
```

```

        canActivate: [AuthGuard]
    }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule {
}

```

## App-component.ts

```

import { HeaderComponent } from './components/header/header.component';
import { Component, ViewChild } from '@angular/core';
import { Title } from '@angular/platform-browser';
import { Router, NavigationEnd, ActivatedRoute } from '@angular/router';
import { filter } from 'rxjs/operators';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})

export class AppComponent {

  title = 'Collectiō';

  constructor(
    private router: Router,
    private activatedRoute: ActivatedRoute,
    private titleService: Title
  ) { }

  ngOnInit() {
    this.router.events.pipe(
      filter(event => event instanceof NavigationEnd),
    )
      .subscribe(() => {

        var rt = this.getChild(this.activatedRoute)

        rt.data.subscribe((data: { title: string; }) => {

```

```

        this.titleService.setTitle(data.title)
    })
})

}

getChild(activatedRoute: ActivatedRoute): any {
    if (activatedRoute.firstChild) {
        return this.getChild(activatedRoute.firstChild);
    } else {
        return activatedRoute;
    }
}
}
}

```

## App-module.ts

```

import { CUSTOM_ELEMENTS_SCHEMA, NgModule } from '@angular/core';
import { BrowserModule, Title } from '@angular/platform-browser';
import { NgbModule } from '@ng-bootstrap/ng-bootstrap';
import { HttpClientModule } from '@angular/common/http';
import { FormsModule } from '@angular/forms';

import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { AppRoutingModuleModule } from './app-routing.module';
import { AngularFireModule } from "@angular/fire/compat";
import { AngularFireAuthModule } from '@angular/fire/compat/auth';
import { AngularFireDatabaseModule } from '@angular/fire/compat/database';
import { AngularFireStorageModule } from '@angular/fire/compat/storage';

import { AppComponent } from './app.component';
import { HeaderComponent } from './components/header/header.component';

import { BannerComponent } from './components/home/banner/banner.component';
import { HomeComponent } from './components/home/home.component';
import { environment } from 'src/environments/environment';
import { FooterComponent } from './components/footer/footer.component';
import { AboutComponent } from './components/about/about.component';
import { AuthorComponent } from './components/author/author.component';
import { ReactiveFormsModule } from '@angular/forms';
import { CollectionsComponent } from './components/home/collections/collections.component';

```



```
import { SingleCollectionComponent } from './components/home/collections/single-collection/single-collection.component';
import { CollectionComponent } from './components/collection/collection.component';
import { CollectionItemComponent } from './components/collection/collection-item/collection-item.component';
import { ItemComponent } from './components/item/item.component';
import { ItemInfosComponent } from './components/item/item-infos/item-infos.component';
import { DashboardComponent } from './components/auth/dashboard/dashboard.component';
import { SignInComponent } from './components/auth/sign-in/sign-in.component';
import { SignUpComponent } from './components/auth/sign-up/sign-up.component';
import { ForgotPasswordComponent } from './components/auth/forgot-password/forgot-password.component';
import { VerifyEmailComponent } from './components/auth/verify-email/verify-email.component';
import { AuthService } from './services/auth.service';
import { MyCollectionsComponent } from './components/user-panel/my-collections/my-collections.component';
import { CreateCollectionComponent } from './components/user-panel/create-collection/create-collection.component';
import { EditCollectionComponent } from './components/user-panel/edit-collection/edit-collection.component';
import { CreateItemComponent } from './components/user-panel/create-item/create-item.component';
import { UpdateCollectionComponent } from './components/user-panel/update-collection/update-collection.component';
```

```
@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    BannerComponent,
    HomeComponent,
    FooterComponent,
    AboutComponent,
    AuthorComponent,
    CollectionItemComponent,
    CollectionsComponent,
    SingleCollectionComponent,
    CollectionComponent,
    ItemComponent,
    ItemInfosComponent,
    DashboardComponent,
```

```
    SignInComponent,
    SignUpComponent,
    ForgotPasswordComponent,
    VerifyEmailComponent,
    MyCollectionsComponent,
    CreateCollectionComponent,
    EditCollectionComponent,
    CreateItemComponent,
    UpdateCollectionComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    NgbModule,
    BrowserModuleAnimationsModule,
    AngularFireModule.initializeApp(environment.firebaseConfig),
    AngularFireDatabaseModule,
    AngularFireAuthModule,
    ReactiveFormsModule,
    HttpClientModule,
    FormsModule,
    AngularFireStorageModule
  ],
  providers: [Title, AuthService],
  bootstrap: [AppComponent],
  schemas: [CUSTOM_ELEMENTS_SCHEMA]
})
export class AppModule { }
```