

Visoka škola strukovnih studija za informacione i komunikacione tehnologije



Dokumentacija

<https://anotez.000webhostapp.com>

Nikola Nedeljković 2/17

Sadržaj

Dokumentacija	1
Sadržaj	2
1. UVOD	5
1.1 Korišćeni programski / markap jezici	5
1.2 Opis funkcionalnosti.....	5
1.3 Izgled stranica	10
2. ORGANIZACIJA.....	11
2.1 Organizacija sajta	11
2.2 Organizacija baze	12
2.3 Slike stranica sa funkcionalnostima	13
3. KODOVI	20
3.1 web.php.....	20
3.2. admin.blade.php	21
3.3. index.blade.php	22
3.4 footer.blade.php	22
3.5. header.blade.php.....	22
3.6 contact.blade.php	23
3.7 login.blade.php	23
3.8. main.blade.php	24
3.9 note.blade.php.....	24
3.10 register.blade.php	25
3.11. author.blade.php	26
3.12 notes_filter.blade.php.....	26
3.13. addnote.blade.php.....	27
3.14. admin_edit_category.blade.php.....	27
3.15. admin_edit_user.blade.php.....	27
3.16. admin_navigation.blade.php	28

3.17.	admin_show_author.blade.php.....	28
3.18.	admin_show_categories.blade.php.....	29
3.19.	admin_show_logs.blade.php.....	29
3.20.	admin_show_users.blade.php.....	30
3.21.	admin_add_author.blade.php.....	30
3.22.	admin_add_category.blade.php.....	31
3.23.	admin_add_user.blade.php.....	31
3.24.	admin_edit_author.blade.php.....	32
3.25.	Kernel.php.....	32
3.26.	AddCategoryRequest.php.....	34
3.27.	AuthorEditImageRequest.php.....	35
3.28.	AuthorImageRequest.php.....	35
3.29.	ContactRequest.php.....	36
3.30.	DeleteUserRequest.php.....	36
3.31.	EditUserRequest.php.....	37
3.32.	GetNotesRequest.php.....	38
3.33.	LoginRequest.php.....	38
3.34.	NoteRequest.php.....	39
3.35.	AuthorImage.php.....	39
3.36.	Category.php.....	40
3.37.	LogAdmin.php.....	41
3.38.	Note.php.....	41
3.39.	User.php.....	43
3.40.	IsAdminMiddleware.php.....	45
3.41.	LogAdminMiddleware.php.....	45
3.42.	LoggedInMiddleware.php.....	46
3.43.	LogVisit.php.....	46
3.44.	ContactController.php.....	47
3.45.	LoginController.php.....	48
3.46.	MainController.php.....	49
3.47.	NoteController.php.....	50
3.48.	RegisterController.php.....	51

3.49.	AdminAuthorController.php.....	52
3.50.	AdminCategoryController.php.....	54
3.51.	AdminLogController.php	55
3.52.	AdminUserController.php	55
3.53.	AuthorController.php.....	57
3.54.	main.js	57
3.55.	search.js.....	59
3.56.	author_game.js	61
3.57.	style.css.....	62
4.	HIJERARHIJSKI PREGLED.....	72

1. UVOD

1.1 Korišćeni programski / markap jezici

- Za izradu ovog sajta korišćene su sledeće tehnologije:
 - Mark-up jezici: HTML 5
 - CSS3
 - Skripting-programski jezici: JavaScript, PHP
 - JQuery – JavaScript framework
 - Laravel – PHP framework
- Sajt je napisan u PHP Storm-u

1.2 Opis funkcionalnosti

- **web.php**
 - ✓ Rad sa rutama
- **views/layouts/admin.blade.php**
 - ✓ Layout za admin panel
- **views/layouts/index.blade.php**
 - ✓ Layout za sve ostale stranice sajta
- **views/fixed/footer.blade.php**
 - ✓ Footer stranice
- **views/fixed/header.blade.php**
 - ✓ Header stranice
 - ✓ Dinamički prikaz naslova
 - ✓ Dinamički prikaz login i logout linka
 - ✓ Generisanje csrf tokena za ajax
- **views/pages/contact.blade.php**
 - ✓ Kontakt stranica sajta
 - ✓ Forma za slanje kontakt poruke

- **views/pages/login.blade.php**
 - ✓ Login stranica sajta
 - ✓ Forma za logovanje korisnika
- **views/pages/main.blade.php**
 - ✓ Prikaz svih note-a korisnika
 - ✓ Pretraživanje po naslovu putem ajax-a
 - ✓ Pretraživanje po kategoriji putem ajax-a
 - ✓ Straničenje note-a
- **views/pages/note.blade.php**
 - ✓ Prikaz jedne note ili dodavanje nove
 - ✓ Forma za insert i edit note
- **views/pages/register.blade.php**
 - ✓ Stranica za registraciju korisnika
 - ✓ Forma za registraciju
- **views/pages/author.blade.php**
 - ✓ Autor stranica
 - ✓ Dohvatanje slika putem ajax-a
 - ✓ Igra sa slikama putem ajax-a i javascript-a
- **views/partials/notes_filter.blade.php**
 - ✓ Filtriranje note-a parcijalni view
- **views/partials/addnote.blade.php**
 - ✓ Parcijalni view dodavanja nove note
- **views/admin/admin_edit_category.blade.php**
 - ✓ Admin stranica za edit kategorije
 - ✓ Forma za edit kategorije
- **views/admin/admin_edit_user.blade.php**
 - ✓ Admin stranica za edit korisnika
 - ✓ Forma za editovanje korisnika
- **views/admin/admin_navigation.blade.php**
 - ✓ Admin deo za navigaciju

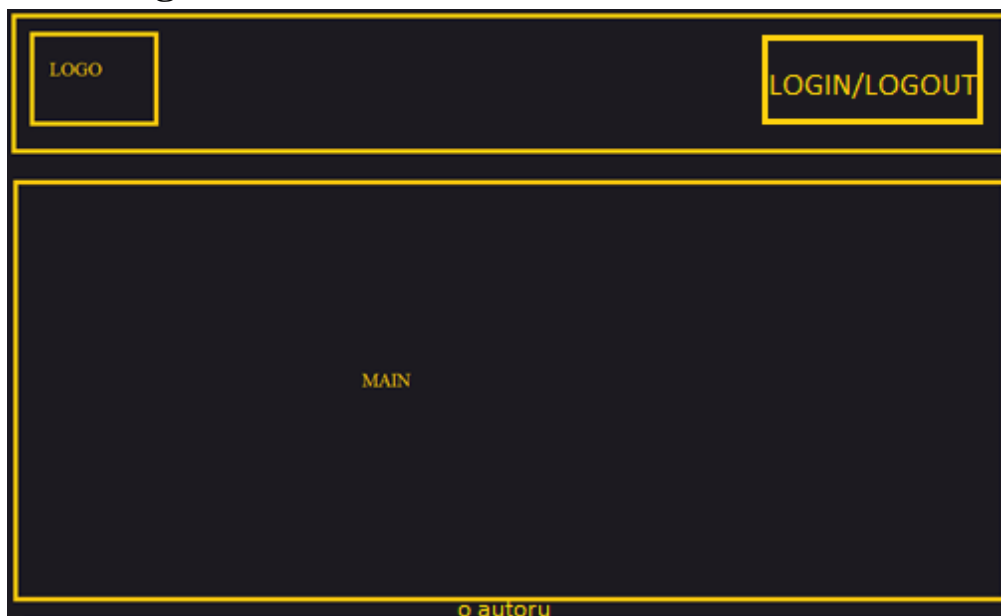
- **views/admin/admin_show_author.blade.php**
 - ✓ Tabelarni prikaz svih slika za igru na stranica autora
- **views/admin/admin_show_categories.blade.php**
 - ✓ Tabelarni prikaz svih kategorija
- **views/admin/admin_show_logs.blade.php**
 - ✓ Tabelarni prikaz svih logova u tabeli log
- **views/admin/admin_show_users.blade.php**
 - ✓ Tabelarni prikaz svih korisnika
- **views/admin/admin_add_author.blade.php**
 - ✓ Forma za dodavanje nove slike autora za igru na autor stranici
- **views/admin/admin_add_category.blade.php**
 - ✓ Forma za dodavanje nove kategorije
- **views/admin/admin_add_user.blade.php**
 - ✓ Forma za dodavanje novog korisnika
- **views/admin/admin_edit_author.blade.php**
 - ✓ Forma za izmenu već postojećeg korisnika
- **Kernel.php**
 - ✓ Rutiranje middleware-a
- **Requests/AddCategoryRequest.php**
 - ✓ Request za dodavanje nove kategorije
- **Requests/AuthorEditImageRequest.php**
 - ✓ Request za izmenu slike
- **Requests/AuthorImageRequest.php**
 - ✓ Request za dodavanje nove slike
- **Requests/ContactRequest.php**
 - ✓ Request za kontakt formu

- **Requests/DeleteUserRequest.php**
 - ✓ Request za brisanje korisnika
- **Requests/EditUserRequest.php**
 - ✓ Request za izmenu korisnika
- **Requests/GetNotesRequest.php**
 - ✓ Request za dohvaćanje svih note-a
- **Requests/LoginRequest.php**
 - ✓ Request za logovanje korisnika
- **Requests/NoteRequest.php**
 - ✓ Request za dodavanje nove note
- **Models/AuthorImage.php**
 - ✓ Model za tabelu slika autora
- **Models/Category.php**
 - ✓ Model za tabelu kategorije
- **Models/LogAdmin.php**
 - ✓ Model za tabelu log
- **Models/Note.php**
 - ✓ Model za tabelu note
- **Models/User.php**
 - ✓ Model za tabelu korisnika
- **Middleware/IsAdminMiddleware.php**
 - ✓ Middleware za proveru da li je korisnik admin
- **Middleware/LogAdminMiddleware.php**
 - ✓ Middleware za logovanje desavanja na admin panelu

- **Middleware/LoggedInMiddleware.php**
 - ✓ Middleware za proveru da li je korisnik logovan
- **Middleware/LogVisit.php**
 - ✓ Middleware za logovanje posete stranice
- **Controllers/ContactController.php**
 - ✓ Kontroler za slanje kontakt mail-a putem PHPMailer-a
- **Controllers/LoginController.php**
 - ✓ Kontroler za login i logout
- **Controllers/MainController.php**
 - ✓ Kontroler za stranicu prikaza svih note-a
- **Controllers/NoteController.php**
 - ✓ Kontroler za dohvatanje, dodavanje, izmenu i brisanje note-a
- **Controllers/RegisterController.php**
 - ✓ Kontroler za registraciju korisnika
- **Controllers/AdminAuthorController.php**
 - ✓ Kontroler za prikaz, dodavanje, izmenu i brisanje autor slika sa admin strane
- **Controllers/AdminCategoryController.php**
 - ✓ Kontroler za prikaz, dodavanje, izmenu i brisanje kategorija sa admin strane
- **Controllers/AdminLogController.php**
 - ✓ Kontroler za prikaz logova na admin strani
- **Controllers/AdminUserController.php**
 - ✓ Kontroler za prikaz, dodavanje, izmenu i brisanje korisnika sa admin strane
- **Controllers/AuthorController.php**
 - ✓ Kontroler za prikaz slika o autoru na autor stranici
- **public/main.js**
 - ✓ Javascript za front-end proveru svih formi

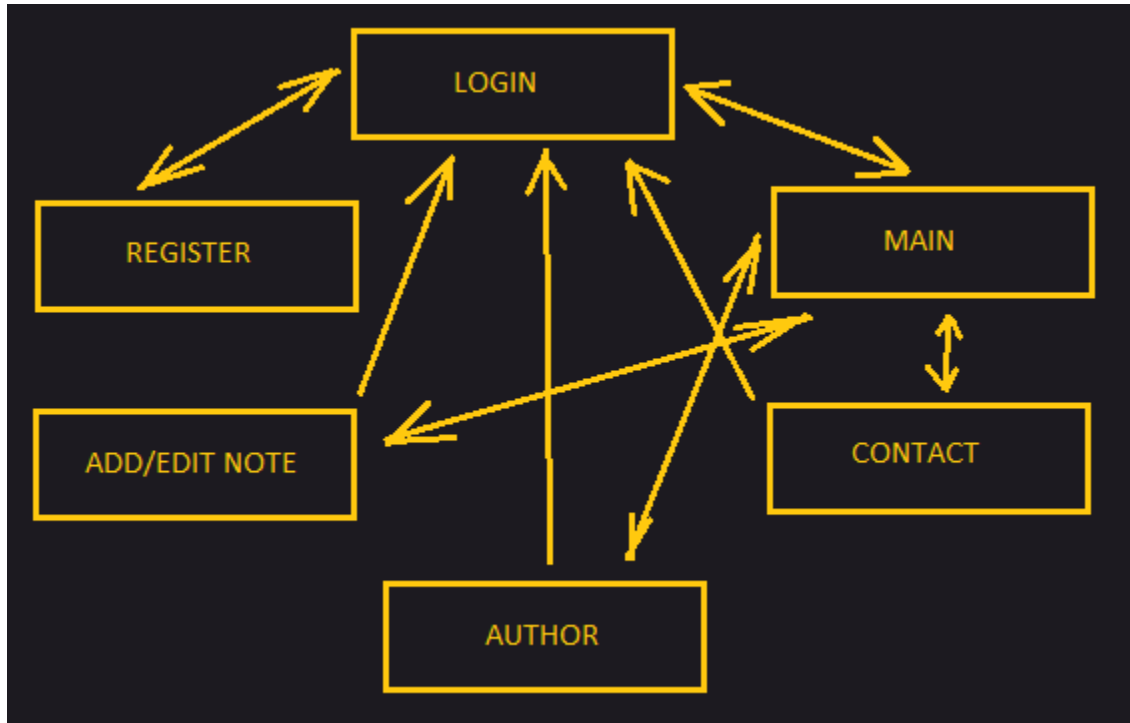
- **public/search.js**
 - ✓ Javascript za pretragu i prikazivanje svih note-a putem javascripta sa stranicenjem
- **public/author_game.js**
 - ✓ Javascript za prikaz slika i igru na stranici o autoru putem ajax-a
- **public/style.css**
 - ✓ CSS za dizajn celog sajta

1.3 Izgled stranica

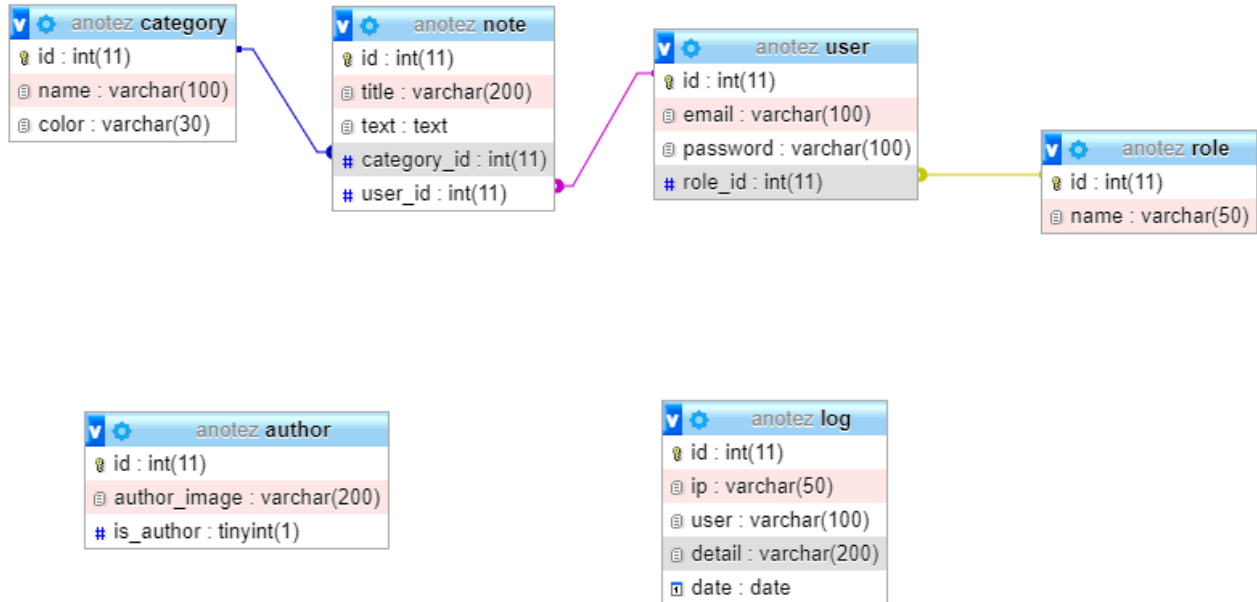


2. ORGANIZACIJA

2.1 Organizacija sajta



2.2 Organizacija baze



2.3 Slike stranica sa funkcionalnostima

AnoteZ Login

Email: someone@something.com

Password: *****

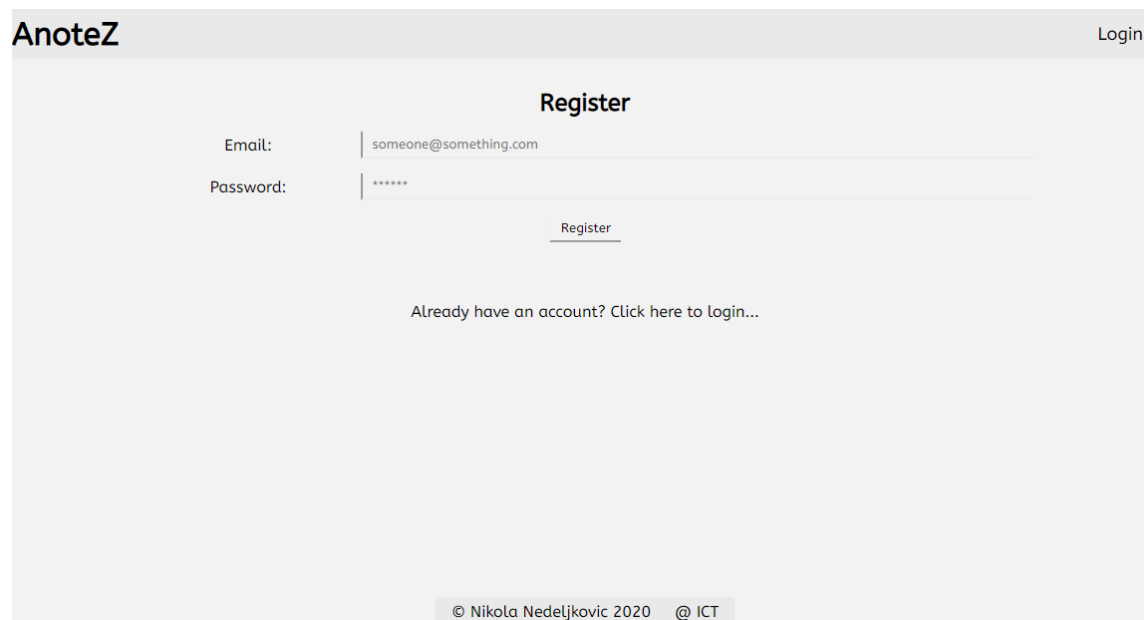
Login

Don't have an account? Click here to register...

© Nikola Nedeljkovic 2020 @ ICT

Login

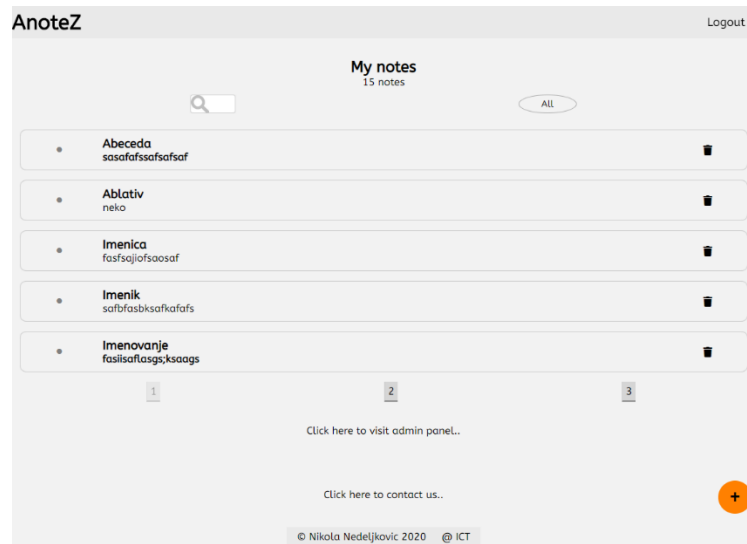
- Login forma, obrada forme na korisničkoj strani, logovanje



The screenshot shows the 'AnoteZ' registration page. At the top left is the 'AnoteZ' logo, and at the top right is a 'Login' link. The main heading is 'Register'. Below it are two input fields: 'Email:' with the placeholder 'someone@something.com' and 'Password:' with a masked password '*****'. A 'Register' button is centered below the fields. Below the button is a link: 'Already have an account? Click here to login...'. At the bottom, a footer contains the copyright notice '© Nikola Nedeljkovic 2020' and a social media icon '@ ICT'.

Login

- Registraciona forma, obrada forme na korisničkoj strani, registracija, provera duplog maila



Main

- Dinamičko dohvaćanje svih note-a preko ajax-a, straničenje, brisanje, formatiranje preko teksta i kategorije

AnoteZ Logout

Edit Note

Unknown

Title: Ablativ

File Edit View Format

Formats **B** *I* [Text Color] [Background Color] [Bulleted List] [Numbered List] [Link] [Unlink]

1. neko
2. drugo
3. navodjenje
4. blyat

ol

Submit Discard

[Click here to see all your notes...](#)

© Nikola Nedeljkovic 2020 @ ICT

Add i Edit note

- Forma za izmenu i dodavanje note-e u zavisnosti od prosledjenih parametar, korisnička provera.

AnoteZLogout

Navigation:

- Show Users
- Add User
- Show Category
- Add Category
- Show Author Images
- Add Author Image
- Show Logs
- Return to website

Logs

IP	User	Detail	Date
127.0.0.1	nikola.nini@gmail.com	/admin/add_category	2020-03-18
127.0.0.1	nikola.nini@gmail.com	/admin/add_category	2020-03-18
127.0.0.1	nikola.nini@gmail.com	/admin/add_category	2020-03-18
127.0.0.1	nikola.nini@gmail.com	/admin/add_category	2020-03-18
127.0.0.1	nikola.nini@gmail.com	/admin/add_category	2020-03-18
127.0.0.1	nikola.nini@gmail.com	/admin/edit_category	2020-03-18
127.0.0.1	nikola.nini@gmail.com	/admin/add_author	2020-03-18
127.0.0.1	nikola.nini@gmail.com	/admin/add_author	2020-03-18
127.0.0.1	nikola.nini@gmail.com	/admin/add_author	2020-03-18
127.0.0.1	nikola.nini@gmail.com	/admin/add_author	2020-03-18
127.0.0.1	nikola.nini@gmail.com	/admin/add_author	2020-03-18

© Nikola Nedeljkovic 2020 @ ICT

Admin

- Admin panel za sve funkcije administratora, tabelarni prikaz, izmenu i brisanje, kao i prikaz svih logova

AnoteZLogout

Contact:

Title:

Content:

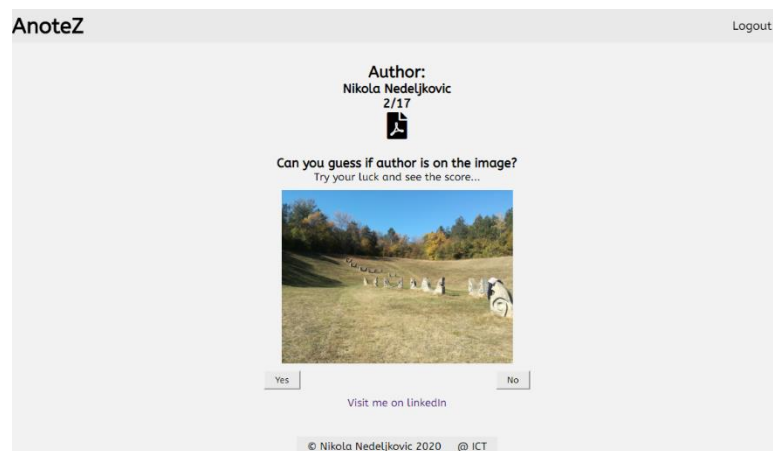
Submit

Back

© Nikola Nedeljkovic 2020 @ ICT

Contact

- Forma za slanje poruke administratoru sajta.



Author

- Autor stranica sa igrom za rad sa slikama, putem ajax-a.

3. KODOVI

3.1 web.php

```
<?php
```

```
/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/
Route::get('/', 'MainController@main')->middleware('loggedIn');
Route::get('/get_notes', 'MainController@getNotes')->middleware('loggedIn');

Route::post('/login', 'LoginController@loginDo');
Route::get('/login', 'LoginController@login')->middleware('loggedIn');

Route::get('/logout', 'LoginController@logout');

Route::get('/register', 'RegisterController@register')->
    middleware('loggedIn');
Route::post('/register', 'RegisterController@registerDo');

Route::get('/note/{id?}', 'NoteController@index')->middleware('loggedIn')->
    where(["id"=> "\d+"]);
Route::post('/editnote', 'NoteController@editNote')->middleware('loggedIn');
Route::post('/addnote', 'NoteController@addNote')->middleware('loggedIn');
Route::get('/deletenote/{id?}', 'NoteController@deleteNote')->
    middleware('loggedIn')->where(["id"=> "\d+"]);

Route::get('/author', 'AuthorController@index')->middleware('loggedIn');
Route::get('/author_get_images', 'AuthorController@getImages')->
    middleware('loggedIn');

Route::get('/contact', 'ContactController@index')->middleware('loggedIn');
Route::post('/contact_us', 'ContactController@contactUs')->
    middleware('loggedIn');

Route::prefix("/admin")->middleware(["loggedIn", "isAdmin"])->
    group(function() {
        Route::get('/', 'AdminUserController@showUsers');

        Route::get("/show_users", 'AdminUserController@showUsers');
        Route::get("/add_user", 'AdminUserController@addUser');
        Route::post("/add_user", 'AdminUserController@adminAddUser')->
            middleware('logAdminVisit');
        Route::get('/edit_user/{id?}', 'AdminUserController@editUser')->
            where(["id"=> "\d+"]);
```

```

Route::post('/edit_user', 'AdminController@adminEditUser')->
>middleware('logAdminVisit');
Route::get('/delete_user/{id?}', 'AdminController@deleteUser')->
>where(["id"=> "\d+"])>middleware('logAdminVisit');

Route::get("/show_category", 'AdminCategoryController@showCategory');
Route::get("/add_category", 'AdminCategoryController@addCategory');
Route::post("/add_category", 'AdminCategoryController@adminAddCategory')->
>middleware('logAdminVisit');
Route::get('/edit_category/{id?}',
'AdminCategoryController@editCategory')->where(["id"=> "\d+"]);
Route::post('/edit_category',
'AdminCategoryController@adminEditCategory')->middleware('logAdminVisit');
Route::get('/delete_category/{id?}',
'AdminCategoryController@deleteCategory')->where(["id"=> "\d+"])>
>middleware('logAdminVisit');

Route::get("/show_author", 'AdminAuthorController@showAuthor');
Route::get("/add_author", 'AdminAuthorController@addAuthor');
Route::post("/add_author", 'AdminAuthorController@adminAddAuthor')->
>middleware('logAdminVisit');
Route::get('/edit_author/{id?}', 'AdminAuthorController@editAuthor')->
>where(["id"=> "\d+"]);
Route::post('/edit_author', 'AdminAuthorController@adminEditAuthor')->
>middleware('logAdminVisit');
Route::get('/delete_author/{id?}', 'AdminAuthorController@deleteAuthor')->
>where(["id"=> "\d+"])>middleware('logAdminVisit');

Route::get("/show_logs", 'AdminLogController@showLogs');
});

```

3.2. admin.blade.php

```

@include('fixed.header')
<div class="admin_part">
    @include('admin.admin_navigation')
    <div class="admin_layout_content">
        @yield('admin_part')
        <p class="error_text" id="error_text">
            @if(session()->has('message'))
                {{ session('message') }}
            @endif
            @isset($errors)
                @foreach($errors->all() as $er)
                    {{$er}}
                @endforeach
            @endisset
        </p>
    </div>
</div>
@include('fixed.footer')

```

3.3. index.blade.php

```
@include('fixed.header')
<div class="middle_part">
    @yield('middle_part')
</div>
@include('fixed.footer')
```

3.4 footer.blade.php

```
<div class="footer_data"><a href="{{url('/author')}}"><p  
class="copyright">&copy; Nikola Nedeljkovic 2020 &nbsp;&nbsp;&nbsp;&nbsp; @  
ICT</p></a></div>  
<script type="text/javascript" src="{{ asset('main.js') }}"></script>  
</body>  
</html>
```

3.5. header.blade.php

```
<!DOCTYPE html>
<html>
<head>
  <title>Anotez - @yield('title_part')</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <meta name="author" content="Nikola Nedeljkovic"/>
  <meta name="copyright" content="RarOks 1008 Entertainment @ 2020"/>
  <meta name="robots" content="index,follow"/>
  <meta name="keywords" content="notes, anotez"/>
  <meta name="description" content="From any device access your notes."/>
  <meta name="abstract" content="Best notes app around."/>
  <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.8.1/css/all.css"
integrity="sha384-
50oBUHEmvpQ+1lW4y57PTFmhCaXp0ML5d60M1M7uH2+nqUivzIebhndOJK28anvf"
crossorigin="anonymous"/>
  <link href="https://fonts.googleapis.com/css?family=ABeeZee&display=swap"
rel="stylesheet"/>
  <link rel="stylesheet" href="{{ asset('style.css') }}" />

  <script src="http://tinymce.cachefly.net/4.0/tinymce.min.js"></script>
  <script type="text/javascript" src="{{ asset('jquery.min.js')
}}"></script>
  <script type="text/javascript">
    var csrf = '{{csrf_token()}}';
  </script>
</head>
<body>
<div class="header_holder">
  <a href="{{url('/') }}" >
    <h1>Anotez</h1>
  </a>
  @if(session()->has('user'))
    <a href="{{url('/logout')}}" class="login_logout_button">Logout</a>
  @else
    <a href="{{url('/login')}}" class="login_logout_button">Login</a>
  @endif
</div>
```

3.6 contact.blade.php

```
@extends('layouts.index')
@section('title_part')
    Contact Us
@endsection
@section('middle_part')
    <h2>Contact:</h2>
    <form class="contact_form" action="{{url('contact_us')}}" method="POST"
onsubmit="return contact_check()">
        @csrf
        <label>Title: </label>
        <input type="text" id="title" name="title"/>
        <label>Content: </label>
        <textarea name="content" id="content" rows="15"
columns="50"></textarea>
        <div>
            <input type="submit" value="Submit" name="note_submit"/>
            <a href="{{url('/') }}">Back</a>
        </div>
        <p class="error_text">
            @if(session()->has('message'))
                {{ session('message') }}
            @endif
            @isset($errors)
                @foreach($errors->all() as $er)
                    {{$er}}
                @endforeach
            @endisset
        </p>
    </form>
@endsection
```

3.7 login.blade.php

```
@extends('layouts.index')
@section('title_part')
    Login
@endsection
@section('middle_part')
    <h2>Login</h2>
    <form action="{{ url('/login') }}" method="POST" onsubmit="return
login_check()">
        @csrf
        <div class="divform">
            <label>Email:</label>
            <input type="email" name="email" id="email"
placeholder="someone@something.com"/>
        </div>
        <div class="divform">
            <label>Password:</label>
            <input type="password" name="password" id="password"
placeholder="*****"/>
        </div>
        <div class="divform">
            <input type="submit" name="btnLogin" value="Login"/>
        </div>
    </form>
```

```

        <p class="error_text" id="error_text_place">
            @if(session()->has('message'))
                {{ session('message') }}
            @endif
            @isset($errors)
                @foreach($errors->all() as $er)
                    {{$er}}
                @endforeach
            @endisset
        </p>
    </form>
    <a class="login_register_link" href="{{url('/register')}}">Don't have an
account? Click here to register...</a>
@endsection

```

3.8. main.blade.php

```

@extends('layouts.index')
@section('title_part')
    My notes
@endsection
@section('middle_part')
    <h2>My notes</h2>
    <p class="note_count">{{$note_count}} notes</p>
    @include("partials.notes_filter")
    <div id="my_notes"></div>
    <p class="error_text">
        @if(session()->has('message'))
            {{ session('message') }}
        @endif
    </p>
    @if(strtolower(session('user')->name) == 'admin')
        <a class="admin_panel_link" href="{{url('/admin')}}"><p>Click here to
visit admin panel...</p></a>
    @endif
    <a class="admin_panel_link" href="{{url('/contact')}}"><p>Click here to
contact us...</p></a>
    <script type="text/javascript" src="{{ asset('search.js') }}"></script>
@endsection
@include('partials.addnote')

```

3.9 note.blade.php

```

@extends('layouts.index')
@section('title_part')
    @isset($note)
        {{$note->title}}
    @else
        Add note
    @endisset
@endsection
@section('middle_part')
    @isset($note)
        <h2>Edit Note</h2>
    @else
        <h2>Add note</h2>
    @endisset

```



```

<form class="note_form" action="@isset($note) {{url("/editnote")}} @else
{{url("/addnote")}} @endisset" method="POST" onsubmit="return note_check()">
    @csrf
    <select name="note_category" id="note_category">
        @foreach( $categories as $cat)
            <option value="{{ $cat->id }}" @isset($note) @if($note-
>name==$cat->name) selected @endif @endisset>{{ $cat->name }}</option>
        @endforeach
    </select>
    <div class="divform addedittitle">
        <label>Title: </label>
        <input type="text" @isset($note) value="{{ $note->title }}"
@endisset id="note_title" name="note_title"/>
    </div>
    <textarea name="note_text" id="note_text" rows="15"
columns="50">@isset($note) {{ $note->text }}@endisset</textarea>
    @isset($note)
        <input type="hidden" value="{{ $note->id }}" name="note_id"/>
    @endisset
    <div class="divform">
        <input type="submit" value="Submit" name="note_submit"/>
        <a href="{{url('/') }}">Discard</a>
    </div>
</form>
<p class="error_text" id="error_text">
    @if(session()->has('message'))
        {{ session('message') }}
    @endif
    @isset($errors)
        @foreach($errors->all() as $er)
            {{ $er }}
        @endforeach
    @endisset
</p>
<a class="login_register_link" href="{{url('/') }}">Click here to see all
your notes...</a>
<script type="application/x-javascript">
    tinymce.init({selector: '#note_text'});
</script>
@endsection

```

3.10 register.blade.php

```

@extends('layouts.index')
@section('title_part')
    Register
@endsection
@section('middle_part')
    <h2>Register</h2>
    <form action="{{ url("/register") }}" method="POST" onsubmit="return
register_check()">
        @csrf
        <div class="divform">
            <label>Email:</label>
            <input type="email" name="email" id="email"
placeholder="someone@something.com"/>
        </div>
    </form>

```

```

        <div class="divform">
            <label>Password:</label>
            <input type="password" name="password" id="password"
placeholder="*****"/>
        </div>
        <div class="divform">
            <input type="submit" name="btnRegister" value="Register"/>
        </div>
        <p class="error_text" id="error_text_place">
            @if(session()->has('message'))
                {{ session('message') }}
            @endif
            @isset($errors)
                @foreach(($errors->all()) as $er)
                    {{$er}}
                @endforeach
            @endisset
        </p>
    </form>
    <a class="login_register_link" href="{{url('/login')}}">Already have an
account? Click here to login...</a>
</endsection>

```

3.11. author.blade.php

```

@extends('layouts.index')
@section('title_part')
    Author
</endsection>
@section('middle_part')
    <h2>Author:</h2>
    <h3 class="author_center">Nikola Nedeljkovic</h3>
    <h3 class="author_center">2/17</h3>
    <a href="{{ asset('documentation.pdf') }}"
class="documentation_holder"><span class="fas fa-file-pdf"></span></a>
    <div class="author_game" id="author_game">
    </div>
    <p class="error_text author_error_text" id="error_text">
        <a href="https://www.linkedin.com/in/nikola-nedeljkovic-
8110b5150/">Visit me on linkedIn</a>
    </p>
    <script type="text/javascript" src="{{ asset('author_game.js')
}}"></script>
</endsection>

```

3.12 notes_filter.blade.php

```

<div class="notes_filter">
    <input type="text" class="search_notes" id="search_notes"
name="searchtext"/>
    <select name="category_select" id="category_select">
        <option value="0">All</option>
        @foreach( $categories as $cat)
            <option value="{{ $cat->id }}">{{ $cat->name }}</option>
        @endforeach
    </select>

```

```
</div>
```

3.13. addnote.blade.php

```
<div class="button_add_note"><a
href="{{url('/note')}}">&nbsp;&#43;&nbsp;</a></div>
```

3.14. admin_edit_category.blade.php

```
@extends('layouts.admin')
@section('title_part')
    Admin Edit Category
@endsection
@section('admin_part')
    <h2>Edit Category</h2>
    <form action="{{ url('/admin/edit_category') }}" method="POST"
onsubmit="return addcategory_check()" >
        @csrf
        <div>
            <label>Name:</label>
            <input type="text" name="categoryname" id="categoryname"
value="{{ $category->name }}" />
        </div>
        <div>
            <label>Color:</label>
            <input type="color" name="categorycolor" id="categorycolor"
value="{{ $category->color }}" />
        </div>
        <input type="hidden" name="id" value="{{ $category->id }}" />
        <div>
            <input type="submit" name="btnSubmit" value="Submit" />
            <a href="{{ url('/admin/show_category') }}"
class="admin_back_link">Back</a>
        </div>
    </form>
@endsection
```

3.15. admin_edit_user.blade.php

```
@extends('layouts.admin')
@section('title_part')
    Admin Edit User
@endsection
@section('admin_part')
    <h2>Edit User</h2>
    <form action="{{ url('/admin/edit_user') }}" method="POST"
onsubmit="return admin_edit_user()" >
        @csrf
        <div>
            <label>Email:</label>
            <input type="email" name="email" id="email" value="{{ $user-
>email }}" placeholder="someone@something.com" />
        </div>
        <div>
            <label>Password:</label>
            <input type="password" name="password" id="password"
placeholder="*****" />
        </div>
    </form>
@endsection
```

```

    </div>
    <input type="hidden" name="id" value="{{ $user->id }}" />
  </div>
    <input type="submit" name="btnSubmit" value="Submit" />
    <a href="{{ url('/admin/show_users') }}"
class="admin_back_link">Back</a>
  </div>
</form>
@endsection

```

3.16. admin_navigation.blade.php

```

<div class="admin_navigation">
  <h2>Navigation:</h2>
  <ul>
    <li><a href="{{ url('admin/show_users') }}">Show Users</a></li>
    <li><a href="{{ url('admin/add_user') }}">Add User</a></li>
    <li><a href="{{ url('admin/show_category') }}">Show Category</a></li>
    <li><a href="{{ url('admin/add_category') }}">Add Category</a></li>
    <li><a href="{{ url('admin/show_author') }}">Show Author
Images</a></li>
    <li><a href="{{ url('admin/add_author') }}">Add Author Image</a></li>
    <li><a href="{{ url('admin/show_logs') }}">Show Logs</a></li>
    <li><a href="{{ url('/') }}">Return to website</a></li>
  </ul>
</div>

```

3.17. admin_show_author.blade.php

```

@extends('layouts.admin')
@section('title_part')
  Admin Show Author Images
@endsection
@section('admin_part')
  <h2>Author Images</h2>
  <table>
    <tr class="fourcolumn">
      <th>Image</th>
      <th>Is Author?</th>
      <th>Edit</th>
      <th>Delete</th>
    </tr>
    @foreach($author_images as $img)
      <tr class="fourcolumn">
        <td>id }}" /></td>
        <td>{{ $img->is_author?'true':'false' }}</td>
        <td><a href="{{ url('admin/edit_author/' . $img->id) }}"><span
class="fas fa-edit"></span></a></td>
        <td><a href="{{ url('admin/delete_author/' . $img->id) }}"><span
class="fa fa-trash" aria-hidden="true"></span></a></td>
      </tr>
    @endforeach
  </table>
@endsection

```

3.18. admin_show_categories.blade.php

```
@extends('layouts.admin')
@section('title_part')
    Admin Show Categories
@endsection
@section('admin_part')
    <h2>Categories</h2>
    <table>
        <tr class="fourcolumn">
            <th>Category</th>
            <th>Color</th>
            <th>Edit</th>
            <th>Delete</th>
        </tr>
        @foreach($categories as $category)
            <tr class="fourcolumn">
                <td>{{ $category->name }}</td>
                <td style="color: {{ $category->
>color}}</td>
                    <td><a href="{{ url('admin/edit_category/' . $category-
>id) }}"><span class="fas fa-edit"></span></a></td>
                    <td><a href="{{ url('admin/delete_category/' . $category-
>id) }}"><span class="fa fa-trash" aria-hidden="true"></span></a></td>
                </tr>
            @endforeach
        </table>
    @endsection
```

3.19. admin_show_logs.blade.php

```
@extends('layouts.admin')
@section('title_part')
    Admin Logs
@endsection
@section('admin_part')
    <h2>Logs</h2>
    <table>
        <tr class="fourcolumn">
            <th>IP</th>
            <th>User</th>
            <th>Detail</th>
            <th>Date</th>
        </tr>
        @foreach($logs as $log)
            <tr class="fourcolumn">
                <td>{{ $log->ip }}</td>
                <td>{{ $log->user }}</td>
                <td>{{ $log->detail }}</td>
                <td>{{ $log->date }}</td>
            </tr>
        @endforeach
    </table>
    @endsection
```

3.20. admin_show_users.blade.php

```
@extends('layouts.admin')
@section('title_part')
    Admin Show Users
@endsection
@section('admin_part')
    <h2>Users</h2>
    <table>
        <tr class="fourcolumn">
            <th>Email</th>
            <th>Role</th>
            <th>Edit</th>
            <th>Delete</th>
        </tr>
        @foreach($users as $user)
            <tr class="fourcolumn">
                <td>{{ $user->email }}</td>
                <td>{{ $user->name }}</td>
                <td><a href="{{ url('admin/edit_user/'. $user->id) }}"><span
class="fas fa-edit"></span></a></td>
                <td><a href="{{ url('admin/delete_user/'. $user->id) }}"><span
class="fa fa-trash" aria-hidden="true"></span></a></td>
            </tr>
        @endforeach
    </table>
@endsection
```

3.21. admin_add_author.blade.php

```
@extends('layouts.admin')
@section('title_part')
    Admin Add Author Image
@endsection
@section('admin_part')
    <h2>Add Author Image</h2>
    <form action="{{ url('/admin/add_author') }}" method="POST"
onsubmit="return admin_add_author();" enctype="multipart/form-data">
        @csrf
        <div>
            <label>Image:</label>
            <input type="file" name="authorimage" id="authorimage"
accept=".jpeg, .png, .jpg"/>
        </div>
        <div>
            <label>Is Author?</label>
            <span>True <input type="radio" id="isauthor" name="isauthor"
value="true" checked="checked"/></span>
            <span>False <input type="radio" id="isnotauthor" name="isauthor"
value="false"/></span>
        </div>
        <div>
            <input type="submit" name="btnSubmit" value="Submit"/>
        </div>
    </form>
@endsection
```

3.22. admin_add_category.blade.php

```
@extends('layouts.admin')
@section('title_part')
    Admin Add Category
@endsection
@section('admin_part')
    <h2>Add Category</h2>
    <form action="{{ url('/admin/add_category') }}" method="POST"
onsubmit="return addcategory_check()">
        @csrf
        <div>
            <label>Name:</label>
            <input type="text" name="categoryname" id="categoryname"
placeholder="Name"/>
        </div>
        <div>
            <label>Color:</label>
            <input type="color" name="categorycolor" id="categorycolor"/>
        </div>
        <div>
            <input type="submit" name="btnSubmit" value="Submit"/>
        </div>
    </form>
@endsection
```

3.23. admin_add_user.blade.php

```
@extends('layouts.admin')
@section('title_part')
    Admin Add User
@endsection
@section('admin_part')
    <h2>Add User</h2>
    <form action="{{ url('/admin/add_user') }}" method="POST"
onsubmit="return admin_add_user()">
        @csrf
        <div>
            <label>Email:</label>
            <input type="email" name="email" id="email"
placeholder="someone@something.com"/>
        </div>
        <div>
            <label>Password:</label>
            <input type="password" name="password" id="password"
placeholder="*****"/>
        </div>
        <div>
            <input type="submit" name="btnSubmit" value="Submit"/>
        </div>
    </form>
@endsection
```

3.24. admin_edit_author.blade.php

```
@extends('layouts.admin')
@section('title_part')
    Admin Edit Author
@endsection
@section('admin_part')
    <h2>Edit Author Image</h2>
    <form action="{{ url('/admin/edit_author') }}" method="POST"
    enctype="multipart/form-data">
        @csrf
        <div>
            id}}"/>
        </div>
        <div>
            <label>Image:</label>
            <input type="file" name="authorimage" id="authorimage"
            accept=".jpeg, .png, .jpg"/>
        </div>
        <div>
            <label>Is Author?</label>
            <span>True <input type="radio" id="isauthor" name="isauthor"
            value="true" {{ $image->is_author ? 'checked' : '' }}/></span>
            <span>False <input type="radio" id="isnotauthor" name="isauthor"
            value="false" {{ $image->is_author ? '' : 'checked' }}/></span>
        </div>
        <input type="hidden" value="{{ $image->id }}" name="id"/>
        <input type="hidden" value="{{ $image->author_image }}"
        name="lastImageUrl"/>
        <div>
            <input type="submit" name="btnSubmit" value="Submit"/>
            <a href="{{ url('/admin/show_author') }}"
            class="admin_back_link">Back</a>
        </div>
    </form>
@endsection
```

3.25. Kernel.php

```
<?php

namespace App\Http;

use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel
{
    /**
     * The application's global HTTP middleware stack.
     *
     * These middleware are run during every request to your application.
     *
     * @var array
     */
    protected $middleware = [
        \App\Http\Middleware\TrustProxies::class,
```



```

        \App\Http\Middleware\CheckForMaintenanceMode::class,
        \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
        \App\Http\Middleware\TrimStrings::class,

\Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
    ];

/**
 * The application's route middleware groups.
 *
 * @var array
 */
protected $middlewareGroups = [
    'web' => [
        \App\Http\Middleware\EncryptCookies::class,
        \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
        \Illuminate\Session\Middleware\StartSession::class,
        // \Illuminate\Session\Middleware\AuthenticateSession::class,
        \Illuminate\View\Middleware\ShareErrorsFromSession::class,
        \App\Http\Middleware\VerifyCsrfToken::class,
        \Illuminate\Routing\Middleware\SubstituteBindings::class,

        \App\Http\Middleware\LogVisit::class,
    ],

    'api' => [
        'throttle:60,1',
        \Illuminate\Routing\Middleware\SubstituteBindings::class,
    ],
];

/**
 * The application's route middleware.
 *
 * These middleware may be assigned to groups or used individually.
 *
 * @var array
 */
protected $routeMiddleware = [
    'auth' => \App\Http\Middleware\Authenticate::class,
    'auth.basic' =>
\Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'bindings' =>
\Illuminate\Routing\Middleware\SubstituteBindings::class,
    'cache.headers' =>
\Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'password.confirm' =>
\Illuminate\Auth\Middleware\RequirePassword::class,
    'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    'verified' =>
\Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,

    'loggedIn' => \App\Http\Middleware\LoggedInMiddleware::class,
    'isAdmin' => \App\Http\Middleware\IsAdminMiddleware::class,

```

```

        'logAdminVisit' => \App\Http\Middleware\LogAdminMiddleware::class,
    ];

    /**
     * The priority-sorted list of middleware.
     *
     * This forces non-global middleware to always be in the given order.
     *
     * @var array
     */
    protected $middlewarePriority = [
        \Illuminate\Session\Middleware\StartSession::class,
        \Illuminate\View\Middleware\ShareErrorsFromSession::class,
        \App\Http\Middleware\Authenticate::class,
        \Illuminate\Routing\Middleware\ThrottleRequests::class,
        \Illuminate\Session\Middleware\AuthenticateSession::class,
        \Illuminate\Routing\Middleware\SubstituteBindings::class,
        \Illuminate\Auth\Middleware\Authorize::class,
    ];
}

```

3.26. AddCategoryRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class AddCategoryRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'categoryname' => 'required',
            'categorycolor' => 'required'
        ];
    }
}

```

3.27. AuthorEditImageRequest.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class AuthorEditImageRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "isauthor" => "required",
            "authorimage" => 'mimes:jpeg,png,jpg',
            "id" => "required",
            "lastImageUrl" => "required"
        ];
    }
}
```

3.28. AuthorImageRequest.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class AuthorImageRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
```

```
    * Get the validation rules that apply to the request.
    *
    * @return array
    */
    public function rules()
    {
        return [
            "isauthor" => "required",
            "authorimage" => 'required|mimes:jpeg,png,jpg'
        ];
    }
}
```

3.29. ContectRequest.php

```
<?php
```

```
namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class ContactRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "title" => "required|string|min:3|max:255",
            "content" => "required|string|min:3|max:255"
        ];
    }
}
```

3.30. DeleteUserRequest.php

```
<?php
```

```
namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class DeleteUserRequest extends FormRequest
{
```

```
/**
 * Determine if the user is authorized to make this request.
 *
 * @return bool
 */
public function authorize()
{
    return true;
}

/**
 * Get the validation rules that apply to the request.
 *
 * @return array
 */
public function rules()
{
    return [
        "userid" => "required"
    ];
}
```

3.31. EditUserRequest.php

<?php

```
namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class EditUserRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "email" => "required|email",
            "id" => "required"
        ];
    }
}
```

3.32. GetNotesRequest.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class GetNotesRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "category" => "required",
            "page" => "required"
        ];
    }
}
```

3.33. LoginRequest.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class LoginRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     */
}
```

```
* @return array
*/
public function rules()
{
    return [
        "email" => "required|email",
        "password" => "required"
    ];
}
```

3.34. NoteRequest.php

```
<?php
```

```
namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class NoteRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "note_title" => "required",
            "note_text" => "required",
            "note_category" => "required"
        ];
    }
}
```

3.35. AuthorImage.php

```
<?php
```

```
namespace App\Models;

class AuthorImage {
    private $table = 'author';

    public function getAll() {
```

```

        return \DB::table($this->table)
            ->select("*")
            ->get();
    }

    public function getImageWithId($id) {
        return \DB::table($this->table)
            ->select("*")
            ->where([
                ["id", "=", $id]
            ])
            ->first();
    }

    public function addAuthorImage($author_image, $is_author) {
        return \DB::table($this->table)
            ->insertGetId(['author_image' => $author_image, 'is_author' =>
$author_image]);
    }

    public function deleteAuthorImage($id) {
        \DB::table($this->table)
            ->where('id', $id)
            ->delete();
    }

    public function editWithAuthorImage($author_image, $is_author, $id) {
        \DB::table($this->table)
            ->where('id', $id)
            ->update(['author_image' => $author_image, 'is_author' =>
$is_author]);
    }

    public function editIsAuthor($is_author, $id) {
        \DB::table($this->table)
            ->where('id', $id)
            ->update(['is_author' => $is_author]);
    }
}

```

3.36. Category.php

```
<?php
```

```

namespace App\Models;

class Category {
    private $table = 'category';

    public function getCategories () {
        return \DB::table($this->table)
            ->select("id", "name", "color")
            ->get();
    }

    public function getCategoryWithId($id) {

```



```

        return \DB::table($this->table)
            ->select("id", "name", "color")
            ->where([
                ["id", "=", $id]
            ])
            ->first();
    }

    public function insertCategory($name, $color) {
        return \DB::table($this->table)
            ->insertGetId(['name' => $name, 'color' => $color]);
    }

    public function editCategory($name, $color, $id) {
        \DB::table($this->table)
            ->where('id', $id)
            ->update(['name' => $name, 'color' => $color]);
    }

    public function deleteCategory($id) {
        \DB::table($this->table)
            ->where('id', $id)
            ->delete();
    }
}

```

3.37. LogAdmin.php

```
<?php
```

```
namespace App\Models;
```

```

class LogAdmin {
    private $table = 'log';

    public function getLogs () {
        return \DB::table($this->table)
            ->select("ip", "user", "detail", "date")
            ->get();
    }

    public function insertLog($ip, $user, $detail, $date) {
        return \DB::table($this->table)
            ->insertGetId(['ip' => $ip, 'user' => $user, 'detail' => $detail,
            'date' => $date]);
    }
}

```

3.38. Note.php

```
<?php
```

```
namespace App\Models;
```

```
class Note {
```

```

private $table_long = 'note AS n';
private $table_short = 'note';

public function getNoteCount ($email) {
    return \DB::table($this->table_long)
        ->join("category AS c", "n.category_id", "=", "c.id")
        ->join("user AS u", "n.user_id", "=", "u.id")
        ->where([
            ["u.email", "=", $email]
        ])
        ->count();
}

public function getNoteWithId ($email, $id) {
    return \DB::table($this->table_long)
        ->join("category AS c", "n.category_id", "=", "c.id")
        ->join("user AS u", "n.user_id", "=", "u.id")
        ->select("n.title", "n.text", "n.id", "c.color", "c.name")
        ->where([
            ["u.email", "=", $email],
            ["n.id", "=", $id]
        ])
        ->first();
}

public function insertNote ($title, $text, $category, $user) {
    return \DB::table($this->table_short)
        ->insertGetId(['title' => $title, 'text' => $text, 'category_id' =>
=> $category, 'user_id' => $user]);
}

public function editNote ($id, $title, $text, $category) {
    \DB::table($this->table_short)
        ->where('id', $id)
        ->update(['title' => $title, 'text' => $text, 'category_id' =>
$category]);
}

public function deleteNote ($id) {
    \DB::table($this->table_short)
        ->where('id', $id)
        ->delete();
}

public function getNotes ($email) {
    return \DB::table($this->table_long)
        ->join("category AS c", "n.category_id", "=", "c.id")
        ->join("user AS u", "n.user_id", "=", "u.id")
        ->select("n.title", "n.text", "n.id", "c.color", "c.name")
        ->where([
            ["u.email", "=", $email]
        ])
        ->paginate(5);
}

public function getNotesWithCategory ($category, $email) {
    return \DB::table($this->table_long)
        ->join("category AS c", "n.category_id", "=", "c.id")
        ->join("user AS u", "n.user_id", "=", "u.id")
        ->select("n.title", "n.text", "n.id", "c.color", "c.name")
        ->where([

```

```

        ["u.email", "=", $email],
        ["c.id", "=", $category]
    ])
    ->paginate(5);
}

public function getNotesWithTitle ($title, $email) {
    return \DB::table($this->table_long)
        ->join("category AS c", "n.category_id", "=", "c.id")
        ->join("user AS u", "n.user_id", "=", "u.id")
        ->select("n.title", "n.text", "n.id", "c.color", "c.name")
        ->where([
            ["u.email", "=", $email],
            ["n.title", "like", '%'.$title.'%']
        ])
        ->paginate(5);
}

public function getNotesWithCategoryAndTitle ($category, $title, $email)
{
    return \DB::table($this->table_long)
        ->join("category AS c", "n.category_id", "=", "c.id")
        ->join("user AS u", "n.user_id", "=", "u.id")
        ->select("n.title", "n.text", "n.id", "c.color", "c.name")
        ->where([
            ["u.email", "=", $email],
            ["c.id", "=", $category],
            ["n.title", "like", '%'.$title.'%']
        ])
        ->paginate(5);
}
}

```

3.39. User.php

```
<?php
```

```
namespace App\Models;
```

```

class User {
    private $table_long = 'user AS u';
    private $table_short = 'user';

    public function getLogin($email, $password){
        return \DB::table($this->table_long)
            ->join("role AS r", "u.role_id", "=", "r.id")
            ->select("r.name", "u.email", "u.id")
            ->where([
                ["email", "=", $email],
                ["password", "=", md5($password)]
            ])
            ->first();
    }

    public function emailTaken($email) {
        return \DB::table($this->table_short)
            ->select("email")
    }
}

```

```
        ->where([
            ["email", "=", $email]
        ])
        ->first();
    }
    public function emailTakenWithId($email, $id) {
        return \DB::table($this->table_short)
            ->select("email")
            ->where([
                ["email", "=", $email],
                ["id", "!=" , $id]
            ])
            ->first();
    }
    public function addUser($email, $password) {
        \DB::table($this->table_short)
            ->insert(['email' => $email, 'password' => md5($password)]);
    }

    public function getUsers() {
        return \DB::table($this->table_long)
            ->join("role AS r", "u.role_id", "=", "r.id")
            ->select("r.name", "u.email", "u.id")
            ->get();
    }

    public function deleteUser($id) {
        \DB::table($this->table_short)
            ->where('id', $id)
            ->delete();
    }

    public function getUserWithId($id) {
        return \DB::table($this->table_long)
            ->join("role AS r", "u.role_id", "=", "r.id")
            ->select("r.name", "u.email", "u.id")
            ->where([
                ["u.id", "=", $id]
            ])
            ->first();
    }
    public function editUser($email, $id) {
        \DB::table($this->table_short)
            ->where('id', $id)
            ->update(['email' => $email]);
    }
    public function editUserWithPassword($email, $password, $id) {
        \DB::table($this->table_short)
            ->where('id', $id)
            ->update(['email' => $email, 'password' => md5($password)]);
    }
}
```

3.40. IsAdminMiddleware.php

```
<?php

namespace App\Http\Middleware;

use Closure;

class IsAdminMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if(strtolower($request->session()->get('user')->name) == "admin")
            return $next($request);
        else
            return redirect("/")->with("message", "You don't have the right
to visit this page!");
    }
}
```

3.41. LogAdminMiddleware.php

```
<?php

namespace App\Http\Middleware;

use App\Models\LogAdmin;
use Closure;

class LogAdminMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        $model = new LogAdmin();
        $date = date("Y-m-d H:i:s");
        $ip = $request->ip();
        $user = $request->session()->get('user')->email;
        $detail = $request->getPathInfo();
        $model->insertLog($ip, $user, $detail, $date);
        return $next($request);
    }
}
```

3.42. LoggedInMiddleware.php

```
<?php

namespace App\Http\Middleware;

use Closure;

class LoggedInMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if ($request->session()->has("user"))
            if ($request->getPathInfo() == "/login" || $request->
                >getPathInfo() == "/register")
                return redirect("/");
            else
                return $next($request);
        else
            if ($request->getPathInfo() == "/login" || $request->
                >getPathInfo() == "/register")
                return $next($request);
            else
                return redirect("/login")->with("message", "You must be
logged in to continue!");
    }
}
```

3.43. LogVisit.php

```
<?php

namespace App\Http\Middleware;

use Closure;

class LogVisit
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        // Storage -> Logs -> laravel.log
        \Log::info("Visit time: ". date("Y-m-d H:i:s"). ", visit ip: ".
            $request->ip(). ", location visited: ". $request->fullUrl());
    }
}
```

```
        return $next($request);
    }
}
```

3.44. ContactController.php

```
<?php
```

```
namespace App\Http\Controllers;

use App\Http\Requests\ContactRequest;
use Illuminate\Http\Request;

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\SMTP;
use PHPMailer\PHPMailer\Exception;

class ContactController extends Controller
{
    public function index() {
        return view('pages.contact');
    }

    public function contactUs(ContactRequest $request) {
        $subject = $request->input("title");
        $message = $request->input("content");
        $to = $request->session()->get('user')->email;

        $mail = new PHPMailer(true);
        try {
            $mail->SMTPDebug = 0;
            $mail->isSMTP();
            $mail->Host = 'smtp.gmail.com';
            $mail->SMTPAuth = true;
            $mail->Username = 'twentyfivekitchen@gmail.com';
            $mail->Password = PaziStaCitas;
            $mail->SMTPSecure = 'tls';
            $mail->Port = 587;

            $mail->SMTPOptions = array(
                'ssl' => array(
                    'verify_peer' => false,
                    'verify_peer_name' => false,
                    'allow_self_signed' => true
                )
            );

            $mail->setFrom('twentyfivekitchen@gmail.com', 'Anotez');
            $mail->addAddress('twentyfivekitchen@gmail.com');
            $mail->addReplyTo($to, 'Visitor');

            $mail->isHTML(true);
            $mail->Subject = $subject;
            $mail->Body = $message;

            $mail->send();
        }
    }
}
```

```

        return redirect("/contact")->with("message", "Message sent
successfully.");
    } catch (Exception $e) {
        dd($e);
        return redirect("/contact")->with("message", "Message sending
failed.");
    }
}
}

```

3.45. LoginController.php

<?php

```

namespace App\Http\Controllers;

use App\Http\Requests\LoginRequest;
use App\Models\User;
use Illuminate\Http\Request;

class LoginController extends Controller
{
    protected $model;

    public function __construct() {
        $this->model = new User();
    }

    public function login() {
        return view('pages.login');
    }

    public function loginDo(LoginRequest $request) {
        $email = $request->input("email");
        $password = $request->input("password");
        $user = $this->model->getLogin($email, $password);
        if ($user) {
            $request->session()->put("user", $user);
            return redirect("/");
        } else {
            return redirect("/login")->with("message", "Wrong params.");
        }
    }

    public function logout(Request $request) {
        $request->session()->forget("user");
        $request->session()->flush();
        return redirect("/login");
    }
}

```


3.46. MainController.php

```
<?php
```

```
namespace App\Http\Controllers;

use App\Http\Requests\GetNotesRequest;
use App\Models\Category;
use App\Models>Note;
use Illuminate\Http\Request;

class MainController extends Controller
{
    protected $model;
    protected $category_model;

    public function __construct() {
        $this->model = new Note();
        $this->category_model = new Category();
    }

    public function main(Request $request) {
        $db_data = $this->model->getNoteCount($request->session()->get('user')->email);
        $cat_data = $this->category_model->getCategories();
        $data = array();
        $data['note_count'] = $db_data;
        $data['categories'] = $cat_data;
        return view('pages.main', $data);
    }

    public function getNotes(GetNotesRequest $request) {
        $category = $request->input("category");
        $title = $request->input("title");
        $email = $request->session()->get('user')->email;
        if ($category == 0 && $title == null) {
            $notes = $this->model->getNotes($email);
        } else if ($title == null) {
            $notes = $this->model->getNotesWithCategory($category, $email);
        } else if ($category == 0) {
            $notes = $this->model->getNotesWithTitle($title, $email);
        } else {
            $notes = $this->model->getNotesWithCategoryAndTitle($category, $title, $email);
        }
        return $notes;
    }
}
```

3.47. NoteController.php

```
<?php
```

```
namespace App\Http\Controllers;

use App\Http\Requests>NoteRequest;
use App\Models\Category;
use App\Models>Note;
use Illuminate\Http\Request;

class NoteController extends Controller
{
    protected $model;

    public function __construct() {
        $this->model = new Note();
    }

    public function index(Request $request, $id = 0) {
        $category_model = new Category();
        $data = array();
        if ($id != 0) {
            $note = $this->model->getNoteWithId($request->session()-
>get('user')->email, $id);
            if ($note) {
                $data['note'] = $note;
            } else {
                return redirect("/")->with("message", "You don't have the
right to see this note or the note does not exist.");
            }
        }
        $category = $category_model->getCategories();
        $data['categories'] = $category;
        return view('pages.note', $data);
    }

    public function editNote(NoteRequest $request) {
        $category = $request->input("note_category");
        $title = $request->input("note_title");
        $text = $request->input("note_text");
        $id = $request->input("note_id");
        $note = $this->model->getNoteWithId($request->session()->get('user')-
>email, $id);
        if (!$note) {
            return redirect("/")->with("message", "You don't have the right
to edit this note or the note does not exist.");
        }
        $this->model->editNote($id, $title, $text, $category);
        return $this->index($request, $id);
    }

    public function addNote(NoteRequest $request) {
        $title = $request->input("note_title");
        $text = $request->input("note_text");
        $category = $request->input("note_category");
```

```

        $lastInsertId = $this->model->insertNote($title, $text, $category,
$request->session()->get('user')->id);
        return $this->index($request, $lastInsertId);
    }

    public function deleteNote(Request $request, $id = 0) {
        if ($id != 0) {
            $note = $this->model->getNoteWithId($request->session()-
>get('user')->email, $id);
            if ($note) {
                $this->model->deleteNote($id);
                return redirect("/");
            } else {
                return redirect("/")->with("message", "You don't have the
right to delete this note or the note does not exist.");
            }
        } else {
            return redirect("/")->with("message", "Note must exist.");
        }
    }
}

```

3.48. RegisterController.php

```
<?php
```

```

namespace App\Http\Controllers;

use App\Http\Requests\LoginRequest;
use App\Models\User;
use Illuminate\Http\Request;

class RegisterController extends Controller
{
    protected $model;

    public function __construct() {
        $this->model = new User();
    }

    public function register() {
        return view('pages.register');
    }

    public function registerDo(LoginRequest $request) {
        $email = $request->input("email");
        $password = $request->input("password");
        $user = $this->model->emailTaken($email);
        if ($user) {
            return redirect("/register")->with("message", "Email already
taken.");
        } else {
            $this->model->addUser($email, $password);
            return redirect("/login")->with("message", "User created
successfully.");
        }
    }
}

```

}

3.49. AdminAuthorController.php

<?php

```

namespace App\Http\Controllers;

use App\Http\Requests\AuthorEditImageRequest;
use App\Http\Requests\AuthorImageRequest;
use App\Models\AuthorImage;
use Illuminate\Http\Request;

class AdminAuthorController extends Controller
{
    private $model;

    public function __construct() {
        $this->model = new AuthorImage();
    }

    public function showAuthor(Request $request) {
        $data = [];
        $data['author_images'] = $this->model->getAll();
        return view('admin.admin_show_author', $data);
    }

    public function addAuthor(Request $request) {
        return view('admin.admin_add_author');
    }

    public function adminAddAuthor(AuthorImageRequest $request) {
        \DB::beginTransaction();
        $image = $request->file("authorimage");
        $name = time()."_" . $image->getClientOriginalName();
        $isUploaded = $image->move(public_path() . "/img/author", $name);
        if (!$isUploaded) {
            \DB::rollback();
            return redirect("admin/show_author");
        }
        $name_full = "/img/author/" . $name;
        $is_author = $request->input("isauthor");
        $author = $is_author == "true" ? 1 : 0;
        try {
            $lastId = $this->model->addAuthorImage($name_full, $author);
            if (!$lastId) {
                \DB::rollback();
                return redirect("admin/show_author");
            }
        } catch (\Exception $ex) {
            \DB::rollback();
            return redirect("admin/show_author");
        }
        \DB::commit();
        return redirect("admin/show_author");
    }
}

```

```

    public function editAuthor(Request $request, $id = 0) {
        $data = [];
        $data['image'] = $this->model->getImageWithId($id);
        if ($data['image']) {
            return view('admin.admin_edit_author', $data);
        }
        return redirect("admin/show_author")->with("message", "You don't have
the right to see this image or the image does not exist.");
    }

    public function adminEditAuthor(AuthorEditImageRequest $request) {
        $is_author = $request->input("isauthor");
        $author = $is_author == "true" ? 1 : 0;
        $lastImageUrl = $request->input("lastImageUrl");
        $id = $request->input("id");
        $image = $request->file("authorimage");
        if ($image) {
            \DB::beginTransaction();
            $name = time()."_".$image->getClientOriginalName();
            $isUploaded = $image->move(public_path()."/img/author", $name);
            if (!$isUploaded) {
                \DB::rollback();
                return redirect("admin/show_author");
            }
            $name_full = "/img/author/".$name;
            if (\File::exists(public_path().$lastImageUrl)) {
                \File::delete(public_path().$lastImageUrl);
            }
            try {
                $this->model->editWithAuthorImage($name_full, $author, $id);
                \DB::commit();
            } catch (\Exception $ex) {
                \DB::rollback();
                return redirect("admin/show_author");
            }
        } else {
            $this->model->editIsAuthor($author, $id);
        }
        return redirect("admin/show_author");
    }

    public function deleteAuthor(Request $request, $id = 0) {
        $image = $this->model->getImageWithId($id);
        if ($image) {
            if (\File::exists(public_path().$image->author_image)) {
                \File::delete(public_path().$image->author_image);
                $this->model->deleteAuthorImage($id);
            }
            return redirect("admin/show_author");
        }
        return redirect("admin/show_author")->with("message", "You don't have
the right to see this image or the image does not exist.");
    }
}

```

3.50. AdminCategoryController.php

```
<?php
```

```
namespace App\Http\Controllers;

use App\Http\Requests\AddCategoryRequest;
use App\Models\Category;
use Illuminate\Http\Request;

class AdminCategoryController extends Controller
{
    private $model;

    public function __construct() {
        $this->model = new Category();
    }

    public function showCategory(Request $request) {
        $data = [];
        $data['categories'] = $this->model->getCategories();
        return view('admin.admin_show_categories', $data);
    }

    public function addCategory(Request $request) {
        return view('admin.admin_add_category');
    }

    public function adminAddCategory(AddCategoryRequest $request) {
        $name = $request->input("categoryname");
        $color = $request->input("categorycolor");
        $this->model->insertCategory($name, $color);
        return redirect("/admin/show_category");
    }

    public function editCategory(Request $request, $id = 0) {
        $data = [];
        $data['category'] = $this->model->getCategoryWithId($id);
        if ($data['category']) {
            return view('admin.admin_edit_category', $data);
        }
        return redirect("admin/show_category")->with("message", "You don't
have the right to see this category or the category does not exist.");
    }

    public function adminEditCategory(AddCategoryRequest $request) {
        $name = $request->input("categoryname");
        $color = $request->input("categorycolor");
        $id = $request->input("id");
        $this->model->editCategory($name, $color, $id);
        return redirect("/admin/show_category");
    }

    public function deleteCategory(Request $request, $id = 0) {
        $category = $this->model->getCategoryWithId($id);
        if ($category) {
```

```
        $this->model->deleteCategory($id);
        return redirect("admin/show_category");
    }
    return redirect("admin/show_category")->with("message", "You don't
have the right to see this category or the category does not exist.");
}
}
```

3.51. AdminLogController.php

```
<?php
```

```
namespace App\Http\Controllers;

use App\Models\LogAdmin;
use Illuminate\Http\Request;

class AdminLogController extends Controller
{
    private $model;

    public function __construct() {
        $this->model = new LogAdmin();
    }

    public function showLogs(Request $request) {
        $data = [];
        $data['logs'] = $this->model->getLogs();
        return view('admin.admin_show_logs', $data);
    }
}
```

3.52. AdminUserController.php

```
<?php
```

```
namespace App\Http\Controllers;

use App\Http\Requests\DeleteUserRequest;
use App\Http\Requests>EditUserRequest;
use App\Http\Requests>LoginRequest;
use App\Models\User;
use Illuminate\Http\Request;

class AdminUserController extends Controller
{
    private $model;

    public function __construct() {
        $this->model = new User();
    }

    public function showUsers(Request $request) {
        $data = [];
        $data['users'] = $this->model->getUsers();
        return view('admin.admin_show_users', $data);
    }
}
```

```
public function editUser(Request $request, $id = 0) {
    $data = [];
    $data['user'] = $this->model->getUserWithId($id);
    if ($data['user']) {
        return view('admin.admin_edit_user', $data);
    }
    return redirect("admin/show_users")->with("message", "You don't have
the right to see this user or the user does not exist.");
}

public function adminEditUser(EditUserRequest $request) {
    $email = $request->input("email");
    $password = $request->input("password");
    $id = $request->input("id");
    $user = $this->model->emailTakenWithId($email, $id);
    if ($user) {
        return redirect("/admin/edit_user/".$id->with("message", "Email
already taken.");
    } else {
        if (!$password) {
            $this->model->editUser($email, $id);
        } else {
            $this->model->editUserWithPassword($email, $password, $id);
        }
        return redirect("/admin/show_users");
    }
}

public function deleteUser(Request $request, $id = 0) {
    $user = $this->model->getUserWithId($id);
    if ($user) {
        $this->model->deleteUser($id);
        return redirect("admin/show_users");
    }
    return redirect("admin/show_users")->with("message", "You don't have
the right to see this user or the user does not exist.");
}

public function addUser(Request $request) {
    return view('admin.admin_add_user');
}

public function adminAddUser(LoginRequest $request) {
    $email = $request->input("email");
    $password = $request->input("password");
    $user = $this->model->emailTaken($email);
    if ($user) {
        return redirect("/admin/add_user")->with("message", "Email
already taken.");
    } else {
        $this->model->addUser($email, $password);
        return redirect("/admin/show_users");
    }
}
}
```


3.53. AuthorController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\AuthorImage;
use Illuminate\Http\Request;

class AuthorController extends Controller
{
    private $model;

    public function __construct() {
        $this->model = new AuthorImage();
    }

    public function index() {
        return view('pages.author');
    }

    public function getImages() {
        return $this->model->getAll();
    }
}
```

3.54. main.js

```
function check_params(email, password) {
    var email_pattern = /^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@[a-zA-Z0-9-
]+(?:\.[a-zA-Z0-9-]+)*$/;
    if (email == "" || password == "") {
        document.getElementById("error_text_place").innerHTML = "Email and
password must not be empty";
        return false;
    }
    if (!email_pattern.test(email)) {
        document.getElementById("error_text_place").innerHTML = "Email not
valid";
        return false;
    }
    document.getElementById("error_text_place").innerHTML = "";
    return true;
}

function login_check() {
    var email = document.getElementById("email").value,
        password = document.getElementById("password").value;
    return check_params(email, password);
}

function register_check() {
    var email = document.getElementById("email").value,
        password = document.getElementById("password").value;
    return check_params(email, password);
}

function note_check() {
    var title = document.getElementById("note_title").value,
        text = tinyMCE.get('note_text').getContent();
}
```

```

        if (title == '' || text == '') {
            document.getElementById("error_text").innerHTML = "None field should
be left empty.";
            return false;
        }
        document.getElementById("error_text").innerHTML = "";
        return true;
    }
    function addcategory_check() {
        var name = document.getElementById("categoryname").value,
            color = document.getElementById("categorycolor").value;
        if (name == '' || color == '') {
            document.getElementById("error_text").innerHTML = "None field should
be left empty.";
            return false;
        }
        document.getElementById("error_text").innerHTML = "";
        return true;
    }
    function admin_add_user() {
        var email = document.getElementById("email").value,
            password = document.getElementById("password").value,
            email_pattern = /^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~]+@[a-zA-Z0-9-
]+(?:\.[a-zA-Z0-9-]+)*$/;
        if (email == "" || password == "") {
            document.getElementById("error_text").innerHTML = "Email and password
must not be empty";
            return false;
        }
        if (!email_pattern.test(email)) {
            document.getElementById("error_text").innerHTML = "Email not valid";
            return false;
        }
        document.getElementById("error_text").innerHTML = "";
        return true;
    }
    function admin_edit_user() {
        var email = document.getElementById("email").value,
            email_pattern = /^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~]+@[a-zA-Z0-9-
]+(?:\.[a-zA-Z0-9-]+)*$/;
        if (email == "") {
            document.getElementById("error_text").innerHTML = "Email must not be
empty";
            return false;
        }
        if (!email_pattern.test(email)) {
            document.getElementById("error_text").innerHTML = "Email not valid";
            return false;
        }
        document.getElementById("error_text").innerHTML = "";
        return true;
    }
    function admin_add_author() {
        var authorimage = document.getElementById("authorimage").value;
        if (authorimage == "") {
            document.getElementById("error_text").innerHTML = "Must select an
image";

```

```

        return false;
    }
    document.getElementById("error_text").innerHTML = "";
    return true;
}
function contact_check() {
    var title = document.getElementById("title").value,
        content = document.getElementById("content").value;
    if (title == "" || content == "") {
        $(".error_text").html("None field must be left empty");
        return false;
    }
    $(".error_text").html("");
    return true;
}

```

3.55. search.js

```

$(document).ready(function () {
    get_notes("", 0, 1);
    $("#search_notes").keyup(function() {
        var search_text = $("#search_notes").val(),
            category_select = $("#category_select").val(),
            page = $("#current_page").html();
        if (!page) { page = 1; }
        get_notes(search_text, category_select, page);
    });
    $("#category_select").change(function() {
        var search_text = $("#search_notes").val(),
            category_select = $("#category_select").val(),
            page = $("#current_page").html();
        if (!page) { page = 1; }
        get_notes(search_text, category_select, page);
    });
});

function get_notes(title, category, page) {
    $.ajax({
        url: "/get_notes",
        method: "GET",
        dataType: "json",
        data: {
            title: title,
            category: category,
            page: page,
            _token: csrf
        },
        success: function (data) {
            if(!data.data.length && data.total > 0) {
                get_notes(title, category, page - 1);
            } else {
                fill_note_data(data);
            }
        },
        error: function (err) {
            $(".error_text").html(err.responseText);
        }
    });
}

```

```

    });
}

function fill_note_data(data) {
    var note_holder = document.getElementById("my_notes"),
        text = "";
    if (data.total == 0) {
        text = `<h2 class="no_notes_text">You don't have any notes so far.
You can add your first one by clicking + symbol.</h2>`;
    } else {
        text = `<ul class="my_notes">`;
        data.data.forEach(function (note) {
            var note_text = "";
            if (note.text.indexOf("<li>") >= 0) {
                note_text = note.text.substring(note.text.indexOf("<li>") +
4, note.text.indexOf("</li>"));
            } else if (note.text.indexOf("<p>") >= 0) {
                note_text = note.text.substring(note.text.indexOf("<p>") + 3,
note.text.indexOf("</p>"));
            } else if (note.text.indexOf("<h1>") >= 0) {
                note_text = note.text.substring(note.text.indexOf("<h1>") +
4, note.text.indexOf("</h1>"));
            } else if (note.text.indexOf("<h2>") >= 0) {
                note_text = note.text.substring(note.text.indexOf("<h2>") +
4, note.text.indexOf("</h2>"));
            } else if (note.text.indexOf("<h3>") >= 0) {
                note_text = note.text.substring(note.text.indexOf("<h3>") +
4, note.text.indexOf("</h3>"));
            } else if (note.text.indexOf("<h4>") >= 0) {
                note_text = note.text.substring(note.text.indexOf("<h4>") +
4, note.text.indexOf("</h4>"));
            } else if (note.text.indexOf("<h5>") >= 0) {
                note_text = note.text.substring(note.text.indexOf("<h5>") +
4, note.text.indexOf("</h5>"));
            } else if (note.text.indexOf("<h6>") >= 0) {
                note_text = note.text.substring(note.text.indexOf("<h6>") +
4, note.text.indexOf("</h6>"));
            } else {
                note_text = note.text.substring(note.text.indexOf("<div>") +
5, note.text.indexOf("</div>"));
            }
            text += `<li><div class="note_category_color" style="color:
${note.color};"><abbr title="${note.name}">#8226;</abbr></div><div
class="note_info_section"><a
href="/note/${note.id}"><h3>${note.title}</h3><p>${note_text}</p></a></div><d
iv class="note_delete_area"><a href="/deletenote/'${note.id}'"><span class="fa
fa-trash" aria-hidden="true"></span></a></div></li>`;
        });
        text += `</ul>`;
        text += `<ul id="note_paging">`;
        for (var i = 1; i <= data.last_page; i++) {
            if (i == data.current_page) {
                text += `<li id="current_page">${i}</li>`;
            } else {
                text += `<li>${i}</li>`;
            }
        }
    }
}

```

```

        text += `</ul>`;
    }
    note_holder.innerHTML = text;
    $("#note_paging li").on("click", function() {
        if ($(this).attr("id") == "current_page") { return; }
        var search_text = $("#search_notes").val(),
            category_select = $("#category_select").val(),
            page = $(this).html();
        get_notes(search_text, category_select, page);
    });
}

```

3.56. author_game.js

```

$(document).ready(function () {
    get_images();
});
function get_images() {
    $.ajax({
        url: "/author_get_images",
        method: "GET",
        dataType: "json",
        data: {
            _token: csrf
        },
        success: function (data) {
            var score = {
                correct: 0,
                done: 0
            };
            fill_author_image(score, data, 0);
        },
        error: function (err) {
            $("#error_text").html(err.responseText);
        }
    });
}
function fill_author_image(score, author_images, which_image) {
    var game_section = document.getElementById("author_game");
    if (score.done == author_images.length) {
        var percentage = (score.correct / score.done) * 100;
        game_section.innerHTML = `<h3>Congratulations!</h3>
        <p>You had ${percentage}% correct answers.</p>
        <button id="try_again">Try Again?</button>`;
        $("#try_again").on("click", function() {
            score.correct = 0;
            score.done = 0;
            fill_author_image(score, author_images, 0);
        });
    } else {
        game_section.innerHTML = `<h3>Can you guess if author is on the
        image?</h3>
        <p>Try your luck and see the score...</p>
        
        <div class="author game buttons"><button
        id="yes_button">Yes</button><button id="no_button">No</button></div>`;
    }
}

```

```

    $("#yes_button").on("click", function() {
        author_button_click(score, author_images, which_image, true);
    });
    $("#no_button").on("click", function() {
        author_button_click(score, author_images, which_image, false);
    });
}
}
function author_button_click(score, author_images, which_image, yes_no) {
    var round_result = (author_images[which_image].is_author == yes_no);
    score.done++;
    if(round_result) { score.correct++; }
    fill_author_image(score, author_images, which_image + 1);
}

```

3.57. style.css

```

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'ABeeZee';
}
html {
    background-color: rgb(242, 242, 242);
}
input, select {
    background-color: rgb(242, 242, 242);
}
.error_text {
    margin-top: 10px;
    color: #FF0000;
}
p.add_space_below {
    margin-bottom: 40px;
}
input:focus, textarea:focus, select:focus {
    border-color: #FF8000;
    outline: none;
}
.documentation_holder {
    font-size: 40px;
    color: #000;
}
.documentation_holder:hover {
    color: #FF8000;
}

/*Header*/
.header_holder {
    width: 100%;
    height: 50px;
    display: flex;
    justify-content: space-between;
    align-items: center;
    position: fixed;
}

```

```
.header_holder, .footer_data {
    background-color: rgb(233, 233, 233);
}
.header_holder a {
    color: #000;
    text-decoration: none;
}
.header_holder .login_logout_button {
    font-size: 18px;
    padding: 5px 15px;
}
.header_holder .search_form input[type="submit"] {
    display: none;
}

/*Footer*/
.footer_data {
    position: fixed;
    bottom: 0;
    width: 300px;
    margin-left: calc(50% - 150px);
}
.footer_data a {
    text-decoration: none;
    color: #000;
}
.footer_data a p {
    text-align: center;
    padding: 5px 0;
}

/*Notes Filter*/
.notes_filter {
    width: 90%;
    margin: 10px 0;
    display: flex;
    justify-content: space-around;
}
.notes_filter .search_notes {
    width: 70px;
    border: 1px solid #ccc;
    border-radius: 4px;
    font-size: 14px;
    background-color: #FFFFFF;
    background-image: url('img/searchicon.png');
    background-repeat: no-repeat;
    padding: 4px 6px 4px 25px;
    -webkit-transition: width 0.4s ease-in-out;
    transition: width 0.4s ease-in-out;
}
.notes_filter .search_notes:focus {
    width: 130px;
}
.notes_filter select {
    align-self: flex-start;
    padding: 5px 10px;
    border-radius: 50%;
    -webkit-appearance: none;
```

```
-moz-appearance: none;
font-size: 14px;
text-align-last: center;
}

/*Middle Part*/
.middle_part {
  padding-top: 80px;
  display: flex;
  align-items: center;
  flex-direction: column;
}
.middle_part .my_notes {
  list-style-type: none;
  width: 98%;
}
.middle_part .admintitle {
  margin-top: 30px;
}
.middle_part .my_notes li {
  margin-top: 15px;
  border: 1px solid #ccc;
  padding: 10px 5px;
  border-radius: 10px;
  color: #000;
  text-decoration: none;
  display: flex;
  justify-content: space-between;
}
.middle_part .my_notes li:hover {
  border: 1px solid #FF8000;
}
.middle_part .my_notes li .note_category_color {
  width: 10%;
  display: flex;
  justify-content: center;
  align-items: center;
  font-size: 30px;
  user-select: none;
}
.middle_part .my_notes li .note_category_color abbr {
  text-decoration: none;
}
.middle_part .my_notes li .note_info_section {
  width: 78%;
}
.middle_part .my_notes li .note_info_section a {
  color: #000;
  text-decoration: none;
}
.middle_part .my_notes li .note_info_section h3, .middle_part .my_notes li
.note_info_section p {
  overflow: hidden;
  text-overflow: ellipsis;
  white-space: nowrap;
}
.middle_part .my_notes li .note_delete_area {
```



```
        width: 10%;
        display: flex;
        justify-content: center;
        align-items: center;
    }
    .middle_part .my_notes li .note_delete_area a {
        color: #000;
    }
    .middle_part form {
        width: 98%;
        display: flex;
        align-items: center;
        flex-direction: column;
    }
    .middle_part form div.divform.addeditttitle {
        margin-bottom: 15px;
    }
    .middle_part form div.divform {
        margin-top: 15px;
        width: 100%;
        display: flex;
        justify-content: center;
        align-items: center;
    }
    .middle_part form div input {
        padding: 5px 10px;
        border-top: none;
        border-right: none;
    }
    .middle_part form div input[type="submit"] {
        cursor: pointer;
    }
    .middle_part form div input[type="email"], .middle_part form div
input[type="password"] {
        width: 60%;
    }
    .middle_part form div label {
        width: 20%;
        text-align: center;
    }
    .middle_part .login_register_link {
        color: #000;
        text-align: center;
        text-decoration: none;
        margin-top: 50px;
        margin-bottom: 60px;
    }
    .middle_part .login_register_link:hover {
        text-decoration: underline;
    }
    .middle_part #my_notes {
        width: 100%;
        display: flex;
        flex-direction: column;
        align-items: center;
    }
    .middle_part #note_paging {
```

```
margin-top: 15px;
margin-bottom: 15px;
width: 96%;
margin-left: 2%;
display: flex;
flex-wrap: wrap;
justify-content: space-around;
list-style-type: none;
}
.middle_part #note_paging li {
background-color: rgb(213, 213, 213);
padding: 5px;
border-bottom: 2px solid #888888;
border-left: 1px solid rgba(88, 88, 88, 0.1);
cursor: pointer;
}
.middle_part #note_paging li:hover {
color: #FF0000;
}
.middle_part #note_paging li#current_page {
opacity: 0.4;
cursor: initial;
color: #000;
user-select: none;
}
.middle_part .no_notes_text {
margin-top: 20px;
text-align: center;
font-style: italic;
}
.middle_part .note_form {
margin-top: 10px;
width: 98%;
display: flex;
flex-direction: column;
}
.middle_part .note_form div.divform {
display: flex;
justify-content: space-between;
}
.middle_part .note_form div input[type="color"] {
align-self: flex-start;
}
.middle_part .note_form div label {
width: 30%;
}
.middle_part .note_form div input[type="text"] {
width: 70%;
}
.middle_part .note_form textarea {
width: 100%;
margin-top: 10px;
padding: 5px;
}
.middle_part .note_form div input[type="submit"], .middle_part .note_form a {
padding: 5px 10px;
border-bottom: 2px solid #888888;
```

```
border-left: 1px solid rgba(88, 88, 88, 0.1);
font-size: 16px;
color: #000;
}
.middle_part .note_form a {
text-decoration: none;
margin-left: 10px;
}
.middle_part .note_form select {
align-self: flex-start;
padding: 5px 10px;
border-radius: 50%;
-webkit-appearance: none;
-moz-appearance: none;
font-size: 14px;
text-align-last: center;
}
.middle_part .admin_panel_link {
margin-top: 10px;
text-align: center;
color: #000;
text-decoration: none;
margin-bottom: 70px;
}
.middle_part .author_center {
text-align: center;
}
.middle_part .author_game {
margin-top: 20px;
display: flex;
flex-direction: column;
align-items: center;
}
.middle_part .author_error_text {
margin-bottom: 70px;
}
.middle_part .author_error_text a {
text-decoration: none;
}
.middle_part .author_error_text a:hover {
color: #FF0000;
}
.middle_part .author_game img {
margin-top: 10px;
width: 30%;
}
.middle_part .author_game button {
margin-top: 10px;
padding: 5px 15px;
background-color: rgb(233, 233, 233);
cursor: pointer;
}
.middle_part .author_game .author_game_buttons {
width: 60%;
display: flex;
justify-content: space-around;
}
```

```
.middle_part .author_game h3, .middle_part .author_game p {
    text-align: center;
}
.middle_part .contact_form {
    margin-top: 15px;
}
.middle_part .contact_form div {
    margin-top: 10px;
    display: flex;
    align-items: center;
    justify-content: space-around;
    width: 80%;
}
.middle_part .contact_form input[type="text"], .middle_part .contact_form
textarea {
    margin: 10px 0;
    border-top: none;
    border-right: none;
    background-color: rgb(254, 254, 254);
    padding: 10px;
    width: 80%;
}
.middle_part .contact_form a {
    padding: 5px 10px;
    border-top: none;
    border-right: none;
    text-decoration: none;
    border-bottom: 2px solid #888888;
    border-left: 1px solid rgba(88, 88, 88, 0.1);
    color: #000;
}
.button_add_note {
    position: fixed;
    z-index: 10;
    bottom: 50px;
    right: 5px;
    padding: 5px;
    background-color: #FF8000;
    border-radius: 50%;
}
.button_add_note a {
    color: #000;
    background-color: #FF8000;
    text-decoration: none;
    font-size: 35px;
    border-radius: 50%;
}
/*Admin Panel*/
.admin_part {
    padding-top: 80px;
    display: flex;
    justify-content: space-around;
    flex-direction: row;
}
.admin_part .error_text {
    margin-bottom: 70px;
}
```

```
.admin_part .admin_navigation {
    width: 30%;
    background-color: rgb(233, 233, 233);
    padding: 5px;
}
.admin_part .admin_back_link {
    text-decoration: none;
    color: #000;
}
.admin_part .admin_navigation h2, .admin_part .admin_navigation ul,
.admin_part .admin_navigation ul li, .admin_part .admin_navigation ul li a {
    background-color: rgb(233, 233, 233);
}
.admin_part .admin_navigation h2 {
    margin-bottom: 20px;
}
.admin_part .admin_navigation ul {
    list-style-type: none;
}
.admin_part .admin_navigation ul li {
    margin: 5px 0;
}
.admin_part .admin_navigation ul li a {
    text-decoration: none;
    color: #000;
}
.admin_part .admin_navigation ul li a:hover {
    color: #FF8000;
}
.admin_part .admin_layout_content {
    width: 67%;
}
.admin_part .admin_layout_content h2 {
    text-align: center;
}
.admin_part .admin_layout_content table {
    width: 100%;
    border-spacing: 0;
    display: block;
    overflow-x: auto;
    margin-bottom: 70px;
}
.admin_part .admin_layout_content table tr {
    width: 100%;
}
.admin_part .admin_layout_content table tr:nth-child(2n) {
    background-color: rgb(225, 225, 225);
}
.admin_part .admin_layout_content table tr:hover {
    background-color: rgb(150, 150, 150);
}
.admin_part .admin_layout_content table th {
    padding: 5px;
}
.admin_part .admin_layout_content table td {
    padding: 2px;
}
```

```
.admin_part .admin_layout_content table th, .admin_part .admin_layout_content
table td {
    text-align: center;
}
.admin_part .admin_layout_content table td img {
    width: 50%;
}
.admin_part .admin_layout_content table .fourcolumn th, .admin_part
.admin_layout_content table .fourcolumn td {
    width: 25%;
}
.admin_part .admin_layout_content table .threecolumn th, .admin_part
.admin_layout_content table .threecolumn td {
    width: 25%;
}
.admin_part .admin_layout_content table a {
    color: #000;
}
.admin_part .admin_layout_content form {
    margin-top: 20px;
    display: flex;
    flex-direction: column;
    margin-bottom: 70px;
}
.admin_part .admin_layout_content form div {
    display: flex;
    justify-content: space-between;
}
.admin_part .admin_layout_content form input {
    margin-bottom: 10px;
    border-top: none;
    border-right: none;
}
.admin_part .admin_layout_content form input[type="email"], .admin_part
.admin_layout_content form input[type="password"], .admin_part
.admin_layout_content form input[type="text"], .admin_part
.admin_layout_content form input[type="color"] {
    width: 80%;
}
.admin_part .admin_layout_content form input[type="submit"] {
    cursor: pointer;
}
.admin_part .admin_layout_content form input, .admin_part
.admin_layout_content form label {
    padding: 5px;
}
.admin_part form div img {
    margin-left: 35%;
    width: 30%;
}

/*Media Queries*/
@media screen and (max-width: 500px) {
    .admin_part {
        flex-direction: column;
        align-items: center;
    }
}
```

```
.admin_part .admin_navigation, .admin_part .admin_layout_content {
    width: 99%;
}
}
@media screen and (max-width: 400px) {
    .middle_part form div {
        flex-direction: column;
        align-items: center;
    }
    .middle_part .note_form a {
        margin-left: 0;
        margin-top: 10px;
    }
    .middle_part form div input[type="email"], .middle_part form div
input[type="password"], .middle_part form div label {
        width: 90%;
    }
}
}
@media screen and (max-width: 320px) {
    .header_holder {
        flex-direction: column;
        height: 100px;
    }
    .header_holder .login_logout_button {
        margin-top: 5px;
        margin-bottom: 5px;
    }
    .footer_data {
        width: 200px;
        margin-left: calc(50% - 100px);
    }
    .middle_part, .admin_part {
        padding-top: 130px;
    }
}
}
```

4. HIJERARHIJSKI PREGLED

