Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektronsko poslovanje

Laravel

Domaći zadatak 2

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
|  |  |  |
| Link ka Github-u |  | |

Sadržaj

[1 Korisnički zahtev 3](#_Toc154095131)

[2 Implementacija 4](#_Toc154095132)

[2.1 Model 4](#_Toc154095133)

[2.2 Migracije 4](#_Toc154095134)

[2.3 Kontroleri 6](#_Toc154095135)

[2.4 Api rute 7](#_Toc154095136)

[3 REST API 8](#_Toc154095137)

[4 Korisničko uputstvo 17](#_Toc154095138)

# Korisnički zahtev

Laravel aplikacija je dizajnirana kao sistem za upravljanje naučnim sadržajem i događajima. U suštini, aplikacija omogućava korisnicima da kreiraju, ažuriraju, pretražuju i brišu naučne članke. Ovi članci su povezani sa korisnicima, a svaki članak može sadržavati komentare i tagove za bolju organizaciju i pretragu. Dodatno, aplikacija poseduje funkcionalnost za autentifikaciju korisnika, uključujući registraciju, prijavu i odjavu. Takođe, implementirana je integracija sa Ticketmaster API-jem za pretragu i prikaz naučnih događaja, omogućavajući korisnicima da pronađu događaje od interesa. Ova funkcionalnost podržava paginaciju za efikasnije pregledanje sadržaja.

# Implementacija

## Model

Aplikacija koristi niz modela koji čine osnovu sistema za upravljanje naučnim sadržajem. Ključni modeli uključuju:

1. Model Article: Predstavlja srce aplikacije. Svaki članak je povezan sa korisnikom (autorom) i može sadržavati višestruke komentare i tagove. Model čuva osnovne informacije o članku, kao što su naslov, sadržaj i datum objavljivanja.
2. Model User: Ovaj model upravlja registracijom korisnika, njihovim profilima i autentifikacijom. Korisnici mogu biti autori članaka i ostavljati komentare.
3. Model Comment: Omogućava korisnicima da komentarišu članke. Svaki komentar je povezan sa specifičnim člankom i korisnikom
4. Model Tag: Tagovi služe za kategorizaciju članaka. Jedan članak može imati više tagova, što olakšava pretragu i organizaciju sadržaja.
5. Model Role: Ovaj model definiše različite uloge korisnika u sistemu, kao što su administrator, autor ili običan korisnik, pružajući različite nivoe pristupa i dozvola.

Ovi modeli su međusobno povezani i formiraju osnovu za interakciju korisnika sa aplikacijom, omogućavajući efikasno upravljanje, pretragu i interakciju sa naučnim sadržajem.

## Migracije

Migracije u ovoj Laravel aplikaciji su ključne za definisanje i održavanje strukture baze podataka. Svaka migracija služi specifičnoj svrsi, kreirajući ili modifikujući strukturu tabela u skladu sa zahtevima aplikacije.

* Migracija za Kreiranje Tabela: U aplikaciji, postoje migracije za kreiranje osnovnih tabela kao što su users, articles, comments, roles, i tags. Svaka od ovih tabela ima definisane kolone koje odražavaju atribute odgovarajućih modela u aplikaciji.

Slika na kojoj se nalazi tekst, snimak ekrana, Font

Opis je automatski generisan

Ilustracija 1primer migracije za kreiranje tabele

* Migracija za Dodavanje Stranih Ključeva: Posebna migracija je zadužena za dodavanje stranih ključeva, posebno u tabeli articles, čime se uspostavlja veza između članaka i korisnika, kao i drugih relevantnih entiteta.

Slika na kojoj se nalazi snimak ekrana, tekst, softver, Multimedijski softver

Opis je automatski generisan

Ilustracija 2 migracije za dodavanje spoljnih kljuceva

* Migracija za Modifikaciju Tabela: Postoje migracije koje uvode promene u već postojeće tabele. Na primer, migracija za dodavanje novih kolona u tabelu users, što omogućava proširenje funkcionalnosti korisničkih profila.
* Migracija za Brisanje Kolona: Takođe, postoji migracija namenjena uklanjanju kolona, kao što je uklanjanje remember\_token iz tabele users, što je deo optimizacije i prilagođavanja strukture baze.

Slika na kojoj se nalazi tekst, snimak ekrana, ekran, softver

Opis je automatski generisan

Ilustracija 3 primer migracije za modifikaciju tabele

## Kontroleri

Kontroleri u Laravel aplikaciji su ključni za upravljanje interakcijom između korisničkog interfejsa i modela aplikacije. Svaki kontroler je specijalizovan za određeni aspekt funkcionalnosti aplikacije.

* ArticleController: Ovaj kontroler upravlja svim operacijama vezanim za članke. Uključuje metode za prikazivanje svih članaka, prikazivanje detalja jednog članka, kao i kreiranje, ažuriranje i brisanje članaka. Pored toga, implementirana je paginacija za efikasno pregledanje članaka. Na slican nacin je implemnetiran i CommentControler.
* AuthController: Ovaj kontroler je zadužen za autentifikaciju korisnika, uključujući registraciju novih korisnika, prijavu i odjavu. Obezbeđuje sigurnost i privatnost korisničkih podataka.

<?php

namespace App\Http\Controllers;

use App\Models\Article;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Http;

use Illuminate\Support\Facades\Validator;

class ArticleController extends Controller

{

    public function index()

    {

        $articles = Article::paginate(3);   //https:127.0.0.8000/api/articles?page=3

        return response()->json($articles);

    }

    public function show($id)

    {

        $article = Article::find($id);

        if (!$article) {

            return response()->json(['message' => 'Article not found'], 404);

        }

        return response()->json($article);

    }

    public function store(Request $request)

    {

        $validator = Validator::make($request->all(), [

            'title' => 'required|max:255',

            'content' => 'required',

            'user\_id' => 'required|exists:users,id',

            'published\_at' => 'nullable|date',

            'image\_path' => 'nullable|string'

        ]);

        if ($validator->fails()) {

            return response()->json($validator->errors(), 400);

        }

        $article = Article::create($validator->validated());

        return response()->json($article, 201);

    }

    public function update(Request $request, $id)

    {

        $article = Article::find($id);

        if (!$article) {

            return response()->json(['message' => 'Article not found'], 404);

        }

        $validator = Validator::make($request->all(), [

            'title' => 'sometimes|required|max:255',

            'content' => 'sometimes|required',

            'user\_id' => 'sometimes|required|exists:users,id',

            'published\_at' => 'nullable|date',

            'image\_path' => 'nullable|string'

        ]);

        if ($validator->fails()) {

            return response()->json($validator->errors(), 400);

        }

        $article->update($validator->validated());

        return response()->json($article);

    }

    public function destroy($id)

    {

        $article = Article::find($id);

        if (!$article) {

            return response()->json(['message' => 'Article not found'], 404);

        }

        $article->delete();

        return response()->json(['message' => 'Article deleted']);

    }

    public function fetchScienceEvents(Request $request)  //potrebno je kroz request poslati parametar page za paginaciju  http://127.0.0.1:8000/api/fetchScienceEvents?page=2

    {

        $apiKey = 'rQaLInSZHng8AiA3h8qSt41RdHFKBmd3';

        $page = $request->input('page', 0);

        $response = Http::get("https://app.ticketmaster.com/discovery/v2/events.json", [

            'apikey' => $apiKey,

            'keyword' => 'science',

            'size' => 10,

            'page' => $page

        ]);

        if ($response->successful()) {

            $events = $response->json();

            return response()->json($events);

        } else {

            return response()->json(['message' => 'Nije moguće dobiti podatke'], $response->status());

        }

    }

    public function search(Request $request)

    {

        $query = Article::query();

        if ($request->has('title')) {

            $query->where('title', 'like', '%' . $request->input('title') . '%');

        }

        if ($request->has('content')) {

            $query->where('content', 'like', '%' . $request->input('content') . '%');

        }

        if ($request->has('user\_id')) {

            $query->where('user\_id', $request->input('user\_id'));

        }

        if ($request->has('published\_from') && $request->has('published\_to')) {

            $query->whereBetween('published\_at', [$request->input('published\_from'), $request->input('published\_to')]);

        }

        $articles = $query->paginate(2);

        return response()->json($articles);

    }

}

<?php

namespace App\Http\Controllers;

use App\Models\User;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Hash;

use Illuminate\Support\Facades\Validator;

class AuthController extends Controller

{

    public function register(Request $request)

    {

        $validator = Validator::make($request->all(), [

            'name' => 'required|string|max:255',

            'email' => 'required|string|email|max:255|unique:users',

            'password' => 'required|string|min:6',

        ]);

        if ($validator->fails()) {

            return response()->json($validator->errors(), 400);

        }

        $user = User::create(array\_merge(

                    $validator->validated(),

                    ['password' => Hash::make($request->password)]

                ));

        $token = $user->createToken('authToken')->plainTextToken;

        return response()->json(['user' => $user, 'token' => $token], 201);

    }

    public function login(Request $request)

    {

        $validator = Validator::make($request->all(), [

            'email' => 'required|string|email',

            'password' => 'required|string',

        ]);

        if ($validator->fails()) {

            return response()->json($validator->errors(), 400);

        }

        if (!Auth::attempt($validator->validated())) {

            return response()->json(['message' => 'Invalid credentials'], 401);

        }

        $user = User::where('email', $request->email)->firstOrFail();

        $token = $user->createToken('authToken')->plainTextToken;

        return response()->json(['user' => $user, 'token' => $token]);

    }

    public function logout(Request $request)

    {

        $request->user()->currentAccessToken()->delete();

        return response()->json(['message' => 'Successfully logged out']);

    }

}

<?php

namespace App\Http\Controllers;

use App\Models\Comment;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Validator;

class CommentController extends Controller

{

    public function index()

    {

        $comments = Comment::paginate(2);

        return response()->json($comments);

    }

    public function show($id)

    {

        $comment = Comment::find($id);

        if (!$comment) {

            return response()->json(['message' => 'Comment not found'], 404);

        }

        return response()->json($comment);

    }

    public function store(Request $request)

    {

        $validator = Validator::make($request->all(), [

            'content' => 'required',

            'user\_id' => 'required|exists:users,id',

            'article\_id' => 'required|exists:articles,id',

            'status' => 'required'

        ]);

        if ($validator->fails()) {

            return response()->json($validator->errors(), 400);

        }

        $comment = Comment::create($validator->validated());

        return response()->json($comment, 201);

    }

    public function update(Request $request, $id)

    {

        $comment = Comment::find($id);

        if (!$comment) {

            return response()->json(['message' => 'Comment not found'], 404);

        }

        $validator = Validator::make($request->all(), [

            'content' => 'sometimes|required',

            'user\_id' => 'sometimes|required|exists:users,id',

            'article\_id' => 'sometimes|required|exists:articles,id',

            'status' => 'sometimes|required'

        ]);

        if ($validator->fails()) {

            return response()->json($validator->errors(), 400);

        }

        $comment->update($validator->validated());

        return response()->json($comment);

    }

    public function destroy($id)

    {

        $comment = Comment::find($id);

        if (!$comment) {

            return response()->json(['message' => 'Comment not found'], 404);

        }

        $comment->delete();

        return response()->json(['message' => 'Comment deleted']);

    }

}

## Api rute

Autentifikacija: Aplikacija ima rute za registraciju, prijavu i odjavu korisnika, omogućavajući bezbedan pristup i upravljanje korisničkim nalogom.

Članci: Postoje rute za prikaz svih članaka, pojedinačnih članaka, pretragu članaka po ključnim rečima, i za CRUD (Create, Read, Update, Delete) operacije nad člancima.

Komentari: Implementirane su rute za pregled, dodavanje, ažuriranje i brisanje komentara na članke, neke od kojih zahtevaju autentifikaciju korisnika.

Naučni Događaji: Posebna ruta omogućava pristup spoljnim naučnim događajima preko Ticketmaster API-ja, pružajući korisnicima mogućnost da istraže relevantne događaje.Slika na kojoj se nalazi tekst, snimak ekrana

Opis je automatski generisan

Ilustracija 4 prikaz api ruta

# REST API

U ovom poglavlju ce biti prikazani detalji o api rutama za kontrolere.

|  |  |
| --- | --- |
| Opis funkcije | Prikaz svih artikala |
| HTTP metoda | GET |
| URL | /api/ articles |
| URL parametri | ?page=2 |
| HTTP body parametri | nema |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "current\_page": 2,      "data": [          {              "id": 4,              "user\_id": 14,              "title": "Modi ea eos et ut magnam placeat dolore.",              "content": "Et voluptatem aut deleniti. Officia nostrum dolores expedita non. Iusto tempora explicabo aperiam ipsa.",              "image\_path": "example.jpg",              "published\_at": "2023-12-29T07:52:45.000000Z",              "deleted\_at": **null**,              "created\_at": "2024-01-11T23:08:39.000000Z",              "updated\_at": "2024-01-11T23:08:39.000000Z"          },          {              "id": 5,              "user\_id": 15,              "title": "Non illum facilis at.",              "content": "Aut aliquid consequatur aliquam ex quaerat sunt. Qui aut reprehenderit laborum consequuntur voluptas. Ducimus voluptatum facilis et magni.",              "image\_path": "example.jpg",              "published\_at": "2023-12-14T08:17:12.000000Z",              "deleted\_at": **null**,              "created\_at": "2024-01-11T23:08:39.000000Z",              "updated\_at": "2024-01-11T23:08:39.000000Z"          },          {              "id": 6,              "user\_id": 16,              "title": "Nulla velit sed voluptatum corporis consequatur eaque.",              "content": "Reiciendis sed dignissimos placeat provident molestiae tempora qui. Laboriosam harum architecto aut ut sit alias. Atque ab ut dolore blanditiis maxime cupiditate.",              "image\_path": "example.jpg",              "published\_at": "2024-01-03T17:36:23.000000Z",              "deleted\_at": **null**,              "created\_at": "2024-01-11T23:08:39.000000Z",              "updated\_at": "2024-01-11T23:08:39.000000Z"          }      ],      "first\_page\_url": "http://127.0.0.1:8000/api/articles?page=1",      "from": 4,      "last\_page": 7,      "last\_page\_url": "http://127.0.0.1:8000/api/articles?page=7",      "links": [          {              "url": "http://127.0.0.1:8000/api/articles?page=1",              "label": "&laquo; Previous",              "active": **false**          },          {              "url": "http://127.0.0.1:8000/api/articles?page=1",              "label": "1",              "active": **false**          },          {              "url": "http://127.0.0.1:8000/api/articles?page=2",              "label": "2",              "active": **true**          },          {              "url": "http://127.0.0.1:8000/api/articles?page=3",              "label": "3",              "active": **false**          },          {              "url": "http://127.0.0.1:8000/api/articles?page=4",              "label": "4",              "active": **false**          },          {              "url": "http://127.0.0.1:8000/api/articles?page=5",              "label": "5",              "active": **false**          },          {              "url": "http://127.0.0.1:8000/api/articles?page=6",              "label": "6",              "active": **false**          },          {              "url": "http://127.0.0.1:8000/api/articles?page=7",              "label": "7",              "active": **false**          },          {              "url": "http://127.0.0.1:8000/api/articles?page=3",              "label": "Next &raquo;",              "active": **false**          }      ],      "next\_page\_url": "http://127.0.0.1:8000/api/articles?page=3",      "path": "http://127.0.0.1:8000/api/articles",      "per\_page": 3,      "prev\_page\_url": "http://127.0.0.1:8000/api/articles?page=1",      "to": 6,      "total": 20  } |
| Format izlaznih parametara | Application/json |
|  |  |
| Opis funkcije | Kreiranje artikla |
| HTTP metoda | POST |
| URL | /api/ articles |
| URL parametri | ?title=artikal 1&content=sadryaj artikla 1&user\_id=1 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "title": "artikal 1",      "content": "sadryaj artikla 1",      "user\_id": "1",      "updated\_at": "2024-01-12T02:56:26.000000Z",      "created\_at": "2024-01-12T02:56:26.000000Z",      "id": 21  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Ažuriranje artikla |
| HTTP metoda | PUT |
| URL | /api/ articles |
| URL parametri | 21?title=artikal 1 novi naslov&content=sadryaj artikla 1&user\_id=1 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "id": 21,      "user\_id": "1",      "title": "artikal 1 novi naslov",      "content": "sadryaj artikla 1",      "image\_path": **null**,      "published\_at": **null**,      "deleted\_at": **null**,      "created\_at": "2024-01-12T02:56:26.000000Z",      "updated\_at": "2024-01-12T02:57:44.000000Z"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje artikla |
| HTTP metoda | DELETE |
| URL | /api/ articles |
| URL parametri | 21 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "message": "Article deleted"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz svih naucnih dogadjaja |
| HTTP metoda | GET |
| URL | /api/fetchScienceEvents |
| URL parametri | nema |
| HTTP body parametri | nema |
| Format HTTP body parametara | JSON |
| Izlazni parametri | "\_embedded": {          "events": [              {                  "name": "Rude Science",                  "type": "event",                  "id": "G5dbZ9VJN6wgD",                  "test": **false**,                  "url": "https://www.ticketmaster.co.uk/rude-science-scunthorpe-17-02-2024/event/36005F49A31F2D7B",                  "locale": "en-us",                  "images": [                      {                          "ratio": "3\_2",                          "url": "https://s1.ticketm.net/dam/c/2b5/5a0a3487-492b-4072-9d8d-1fcda02bd2b5\_781101\_ARTIST\_PAGE\_3\_2.jpg",                          "width": 305,                          "height": 203,                          "fallback": **true**                      },                      {                          "ratio": "4\_3",                          "url": "https://s1.ticketm.net/dam/c/2b5/5a0a3487-492b-4072-9d8d-1fcda02bd2b5\_781101\_CUSTOM.jpg",                          "width": 305,                          "height": 225,                          "fallback": **true**                      },                      {                          "ratio": "16\_9",                          "url": "https://s1.ticketm.net/dam/c/2b5/5a0a3487-492b-4072-9d8d-1fcda02bd2b5\_781101\_RETINA\_PORTRAIT\_16\_9.jpg",                          "width": 640,                          "height": 360,                          "fallback": **true**                      },                      {                          "ratio": "16\_9",                          "url": "https://s1.ticketm.net/dam/c/2b5/5a0a3487-492b-4072-9d8d-1fcda02bd2b5\_781101\_TABLET\_LANDSCAPE\_LARGE\_16\_9.jpg",                          "width": 2048,                          "height": 1152,                          "fallback": **true**                      },                      {                          "ratio": "3\_2",                          "url": "https://s1.ticketm.net/dam/c/2b5/5a0a3487-492b-4072-9d8d-1fcda02bd2b5\_781101\_RETINA\_PORTRAIT\_3\_2.jpg",                          "width": 640,                          "height": 427,                          "fallback": **true**                      },                      {                          "ratio": "16\_9",                          "url": "https://s1.ticketm.net/dam/c/2b5/5a0a3487-492b-4072-9d8d-1fcda02bd2b5\_781101\_RECOMENDATION\_16\_9.jpg",                          "width": 100,                          "height": 56,                          "fallback": **true**                      },                      {  … |
| Format izlaznih parametara | Application/json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz svih komentara |
| HTTP metoda | GET |
| URL | /api/comments |
| URL parametri | nema |
| HTTP body parametri | nema |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "current\_page": 1,      "data": [          {              "id": 1,              "user\_id": 21,              "article\_id": 11,              "content": "Corporis aut animi et officiis ipsa esse. Quibusdam facere placeat magni illum animi tempore. Vel impedit voluptas at voluptatum ab.",              "status": "published",              "created\_at": "2024-01-11T23:08:40.000000Z",              "updated\_at": "2024-01-11T23:08:40.000000Z"          },          {              "id": 2,              "user\_id": 23,              "article\_id": 12,              "content": "Molestias magni eos cum atque aspernatur libero consequuntur. Iusto est dignissimos perferendis ad quae maiores officiis. Maxime ab magnam voluptas sint earum laudantium tempore. Eum in qui maxime fugiat laborum.",              "status": "published",              "created\_at": "2024-01-11T23:08:40.000000Z",              "updated\_at": "2024-01-11T23:08:40.000000Z"          }      ],      "first\_page\_url": "http://127.0.0.1:8000/api/comments?page=1",      "from": 1,      "last\_page": 5,      "last\_page\_url": "http://127.0.0.1:8000/api/comments?page=5",      "links": [          {              "url": **null**,              "label": "&laquo; Previous",              "active": **false**          },          {              "url": "http://127.0.0.1:8000/api/comments?page=1",              "label": "1",              "active": **true**          },          {              "url": "http://127.0.0.1:8000/api/comments?page=2",              "label": "2",              "active": **false**          },          {              "url": "http://127.0.0.1:8000/api/comments?page=3",              "label": "3",              "active": **false**          },          {              "url": "http://127.0.0.1:8000/api/comments?page=4",              "label": "4",              "active": **false**          },          {              "url": "http://127.0.0.1:8000/api/comments?page=5",              "label": "5",              "active": **false**          },          {              "url": "http://127.0.0.1:8000/api/comments?page=2",              "label": "Next &raquo;",              "active": **false**          }      ],      "next\_page\_url": "http://127.0.0.1:8000/api/comments?page=2",      "path": "http://127.0.0.1:8000/api/comments",      "per\_page": 2,      "prev\_page\_url": **null**,      "to": 2,      "total": 10  } |
| Format izlaznih parametara | Application/json |
|  |  |
| Opis funkcije | Kreiranje komentara |
| HTTP metoda | POST |
| URL | /api/comments |
| URL parametri | ? user\_id=1&content=sadrzaj prvog komentara&article\_id=1&status=status |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "content": "sadrzaj prvog komentara",      "user\_id": "1",      "article\_id": "1",      "status": "status",      "updated\_at": "2024-01-12T03:03:48.000000Z",      "created\_at": "2024-01-12T03:03:48.000000Z",      "id": 11  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Ažuriranje komentara |
| HTTP metoda | PUT |
| URL | /api/comments |
| URL parametri | 11?user\_id=1&content=sadrzaj prvog komentara&article\_id=1&status=status |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "id": 11,      "user\_id": "1",      "article\_id": "1",      "content": "sadrzaj prvog komentara",      "status": "status",      "created\_at": "2024-01-12T03:03:48.000000Z",      "updated\_at": "2024-01-12T03:03:48.000000Z"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje komentara |
| HTTP metoda | DELETE |
| URL | /api/comments |
| URL parametri | 11 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "message": "Comment deleted"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Registracija korisnika |
| HTTP metoda | POST |
| URL | /api/register |
| URL parametri | name=Milica&email=milica@gmail.com&password=milica123 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "user": {          "name": "Milica",          "email": "milica@gmail.com",          "role\_id": 1,          "updated\_at": "2024-01-12T02:53:16.000000Z",          "created\_at": "2024-01-12T02:53:16.000000Z",          "id": 41      },      "token": "1|XUJ9I5StVzuAkvhWS7rbF52ao3QnwDnXTVybv3tzf2619251"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Prijava korisnika |
| HTTP metoda | POST |
| URL | /api/login |
| URL parametri | email=milica@gmail.com&password=milica123 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "user": {          "id": 41,          "name": "Milica",          "email": "milica@gmail.com",          "email\_verified\_at": **null**,          "profile\_photo": **null**,          "created\_at": "2024-01-12T02:53:16.000000Z",          "updated\_at": "2024-01-12T02:53:16.000000Z",          "role\_id": 1,          "bio": **null**,          "last\_login\_at": **null**      },      "token": "2|7sxTE4myg54HbGZarWSAToYO4oGu2Q9AOnM2rqpq938b8ed1"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Odjava korisnika |
| HTTP metoda | POST |
| URL | /api/logout |
| URL parametri | nema |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "message": "Successfully logged out"  } |
| Format izlaznih parametara | application/json |

# Korisničko uputstvo

Prilikom pokretanja aplikacije je potrebno registrovati se

Slika na kojoj se nalazi tekst, snimak ekrana, ekran, softver

Opis je automatski generisan

Nakon uspešne registracije je potrebno prijaviti se

Slika na kojoj se nalazi tekst, snimak ekrana, softver, Ikona na računaru

Opis je automatski generisan

Token koji smo dobili treba da kopiramo u Authorization karticu kako bismo imali pristup zasticenim rutama

Slika na kojoj se nalazi tekst, snimak ekrana, softver, ekran

Opis je automatski generisan

Prikaz svih artikala

Slika na kojoj se nalazi tekst, snimak ekrana, softver, Ikona na računaru

Opis je automatski generisan

Kreiranje artikla

Slika na kojoj se nalazi tekst, snimak ekrana, softver, Ikona na računaru

Opis je automatski generisan

Ažuriranje artikla

Slika na kojoj se nalazi tekst, snimak ekrana, softver, Ikona na računaru

Opis je automatski generisan

Brisanje artikla

Slika na kojoj se nalazi tekst, snimak ekrana, softver, Ikona na računaru

Opis je automatski generisan

Prikaz naučnih događaja

Slika na kojoj se nalazi tekst, snimak ekrana, softver, ekran

Opis je automatski generisan

Prikaz svih komentara

Slika na kojoj se nalazi tekst, snimak ekrana, softver, ekran

Opis je automatski generisan

Kreiranje komentara

Slika na kojoj se nalazi tekst, snimak ekrana, softver, Ikona na računaru

Opis je automatski generisan

Izmena komentara

Slika na kojoj se nalazi tekst, snimak ekrana, softver, ekran

Opis je automatski generisan

Brisanje komentara

Slika na kojoj se nalazi tekst, snimak ekrana, softver, ekran

Opis je automatski generisan

Odjava korisnika

Slika na kojoj se nalazi tekst, snimak ekrana, softver, ekran

Opis je automatski generisan

# dsada