



## SEMINARSKI RAD – INTERNET PROGRAMIRANJE

### TEMA: ANSWER FINDER

Student: Milan Nikolić

Profesor: docent dr Željko Jovanović

Br. Indexa: 68/2020

Čačak 2024.

## Sadržaj

1. Opis sistema .....	2
2. Specifikacija zahteva .....	4
3. MVC projektovanje .....	6
3.1. Model (M) .....	6
4. VIEV (V) .....	8
4.1. Controller (C).....	15
5. Realizovane funkcionalnosti. ....	16

## 1. Opis sistema

Projekat „Answer Finder“ predstavlja izradu web sajta za postavljanje pitanja na koje drugi korisnici mogu dati odgovor, koji se može glasati (upvotovati). Cilj projekta je pružanje korisnicima funkcionalnu platformu za istraživanje, edukovanje ali i zabavu, nezavisno od njihovog obrazovanja ili godina. U daljem tekstu je dat pregled funkcionalnosti po ulogama na sajtu.

Neophodno je realizovati projekat „Answer Finder“ sa sledećim funkcionalnostima:

Uloge:

1. Gost - Osoba koja nije prijavljena na sajt, ima osnovne mogućnosti pregledanja pitanja, pretrage, sortiranja, filtriranja po različitim kriterijumima. Gost takođe može gledati komentare ispod svakog pitanja.
2. Korisnik – Registrovani korisnik sajta koji može pristupiti dodatnim funkcionalnostima. Korisnik može objavljivati pitanja na sajtu, komentarisati na svakom pitanju ali i glasati, odnosno upvotovati komentare po želji.
3. Administrator – Osoba sa privilegijama za upravljanje i održavanje sajta. Administrator ima mogućnost da briše i edituje pitanja, ali i komentare koji su na bilo koji način neprikladni ili neodgovarajući. Administrator takođe ima pristup svim funkcionalnostima koje su dostupne korisnicima.

Funkcionalnosti za svaku ulogu:

Za gosta:

- Registracija na web sajt
- Prijava na web sajt
- Pregled svih pitanja
- Pregled svih komentara i glasova
- Sortiranje i filtriranje

Za korisnika:

- Sve funkcionalnosti dostupne gostima
- Postavljanje novih pitanja
- Komentarisanje svih pitanja
- Upravljanje svojim pitanjima/komentarima
- Glasanje komentara

Za administratora:

- Sve funkcionalnosti dostupne korisnicima
- Upravljanje svim pitanjima i komentarima

Napredne funkcionalnosti:

- Pamćene poslednje posećenog pitanja koristeći kolačiće (Cookies)
- Dinamičko pretraživanje
- Dinamičko filtriranje
- Dinamička navigacija

Projekat „Answer Finder“ će biti hostovan na besplatnom hosting servisu „“ sa njihovim besplatnim domenom „“. Ova infrastruktura će omogućiti pouzdanost, skalabilnost i brzinu pristupa korisnicima koji će pristupati sajtu.

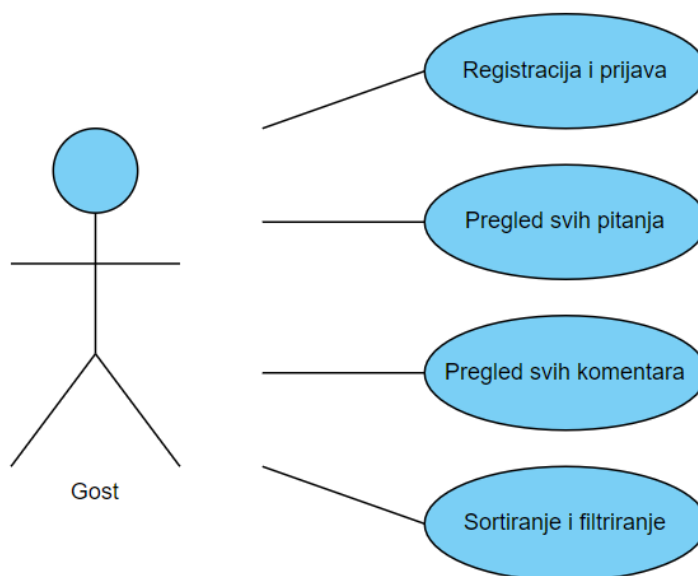
Kroz različite uloge (gost, korisnik, administrator), svaki korisnik će imati prilagođen pristup funkcionalnostima sistema, što će omogućiti interakciju, zajedničko učenje i unapređenje kulinarskih veština.

## 2. Specifikacija zahteva

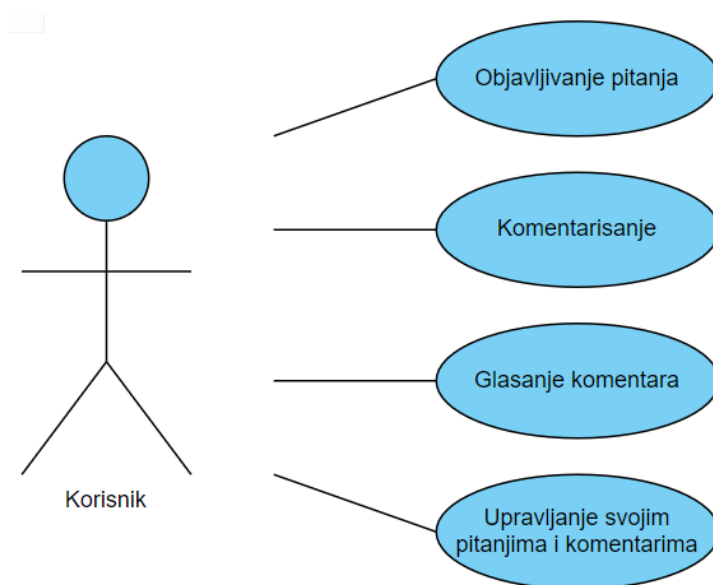
Za potrebe projekta „Answer Finder“ kreirani su Use Case dijagrami i tabela specifikacija, koji predstavljaju ključne komponente sistema.

Use Case dijagrami pružaju vizuelni prikaz interakcije između korisnika i sistema. Oni prikazuju različite uloge (gosta, korisnika, administratora) i akcije koje svaka uloga može izvršiti na web sajtu. Ovi dijagrami pomažu u razumevanju funkcionalnosti sistema i odnosa između korisnika i sistema.

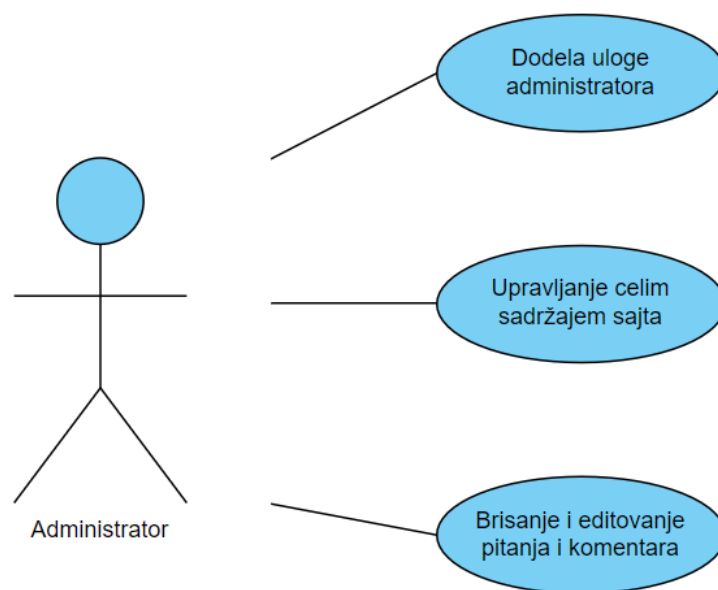
### USE CASE DIJAGRAMI



Slika 1 Use Case Dijagram Gost



Slika 2 Use Case Dijagram Korisnik



Slika 3 Use Case Dijagram Administrator

**TABELA SPECIFIKACIJE ZAHTEVA**

Rbz	Zahtev	Segment	Prioritet	Akter
1.	Pregled objavljenih pitanja		O	Svi
2.	Filtriranje i sortiranje pitanja		O	Svi
3.	Pretraga pitanja po nazivu		O	Svi
4.	Prijava i registracija		O	Gost
5.	Objavljivanje pitanja		O	Korisnik
6.	Postavljanje komentara		O	Korisnik
7.	Glasanje komentara		O	Korisnik
8.	Upravljanje svojim objavama		O	Korisnik
9.	Brisanje svih komentara i pitanja		O	Admin
10.	Davanje uloge administratora		B	Admin
11.	Mogućnost revizije pitanja pre objave		M	Admin

Tabela 1. Specifikacija funkcionalnih zahteva

**Zahtev** je tekstualni opis FZ-a.

**Prioritet** je jedno od sledećih slova (O – obavezan, V – važan, M – moguć, B - budući).

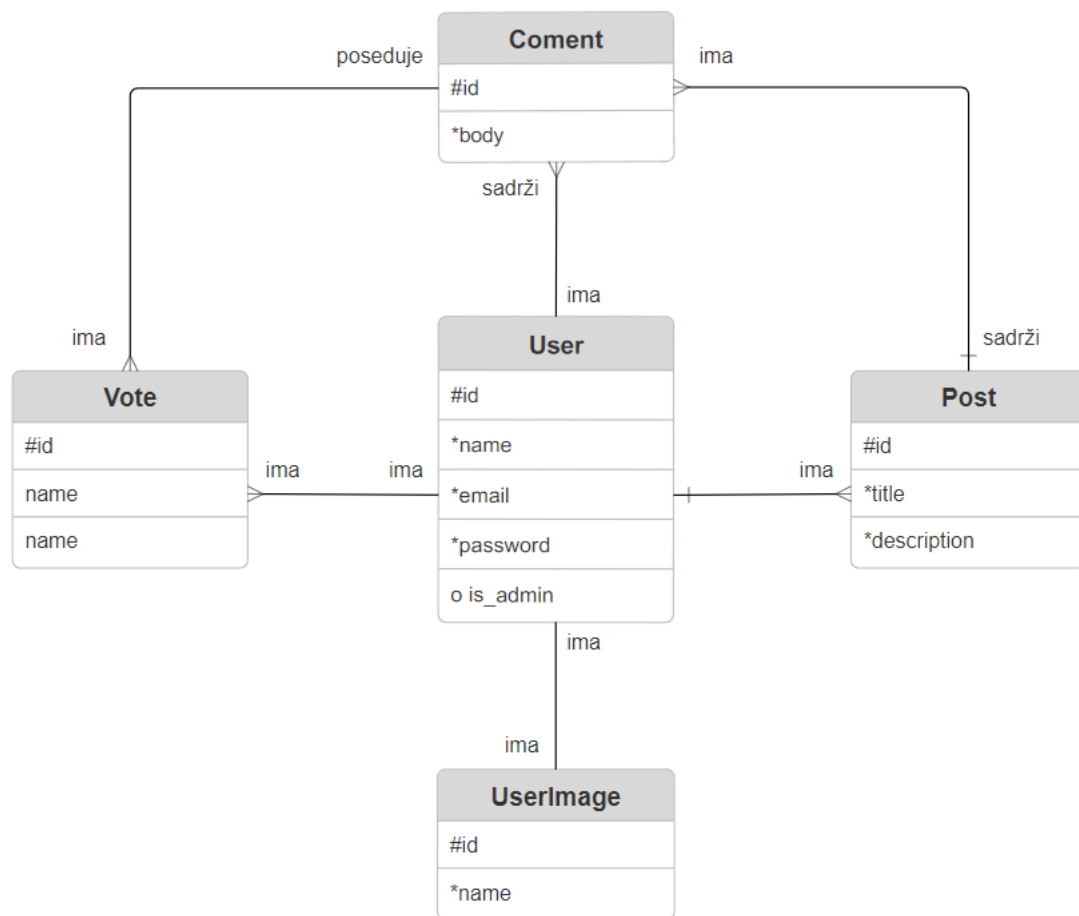
**Akter** je tip korisnika koji će izvršavati opisanu funkcionalnost (gost, admin, klijent...)

### 3. MVC projektovanje

#### 3.1. Model (M)

Model podataka (slika ispod) sadrži sledeće tabele:

- User – komentari recepta
- Post - korisnici
- Comment - recepti
- Vote– glasovi korisnika (da li im se dopada web sajt)
- UserImage – veza korisnika sa slikom u skladištu



Za svaku od tabela u bazi dati opis neophodnih DAO metoda .

##### •PostDAO

- getAllPosts(\$search, \$hasComments, \$sortDirection); - vraća sva pitanja koja zadovoljavaju određene upite za dinamičko renderovanje
- createPost(\$title, \$description, \$user\_id); - kreira pitanje
- findPost(Post \$post); - vraća pitanje sa datim id-em
- deletePost(Post \$post); - briše pitanje
- updatePost(Post \$post, \$title, \$description); - ažurira pitanje
- getPostComments(Post \$post); - vraća sve komentare ispod tog pitanja

##### •CommentDAO

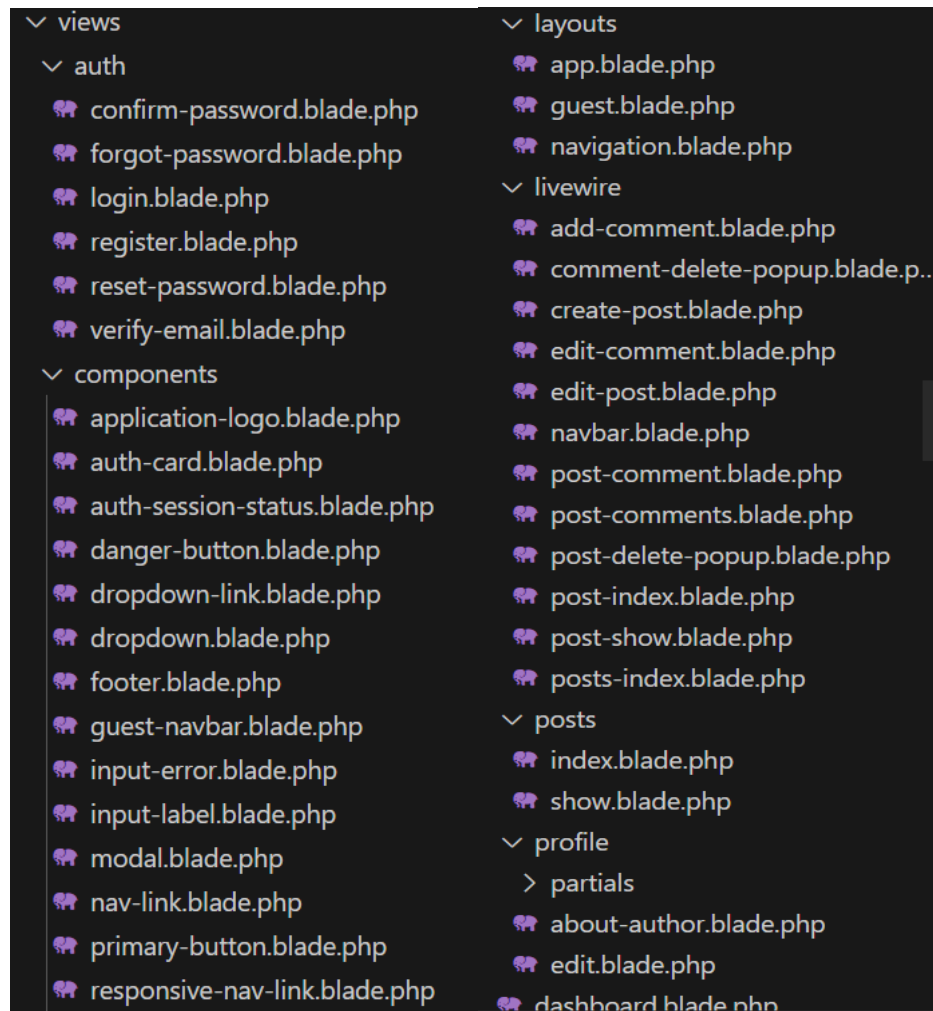
- getAllComments(); - vraća sve komentare
- createComment(\$user\_id, \$post\_id, \$body); - kreira komentar

- deleteComment(Comment \$comment); - briše komentar
- updateComment(Comment \$comment, \$body); - ažurira komentar
- VoteDAO
  - isVotedByUser(?User \$user, \$comment\_id); - proverava da li je korisnik već glasao za komentar i izvršava logiku na osnovu toga
  - vote(User \$user, \$comment\_id); - glasa za komentar
  - removeVote(User \$user, \$comment\_id); - uklanja glas koliko je korisnik već glasao
- UserImageDAO
  - createUserImage(\$user\_id, \$name); - kreira UserImage unutar tabele sa datim podacima



## 4. VIEV (V)

Pregledi predstavljaju jedan od najvažnijih delova svakog websajta. Oni predstavljaju ono što korisnik vidi i ono što ostavlja najveći utisak.



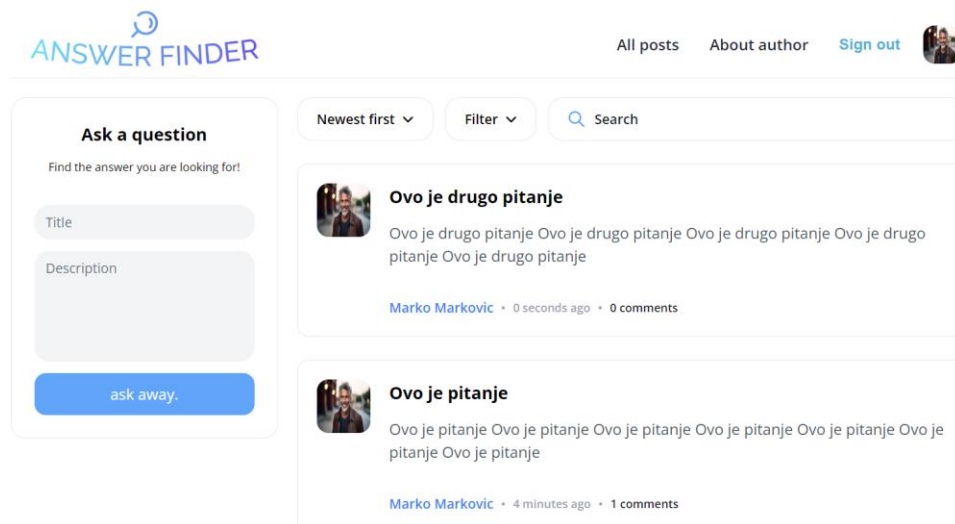
Slika 4 Struktura pregleda (View-a)

Ovi fajlovi predstavljaju komponente svakog view-a. Postoje obične komponente i Livewire komponente (unutar Livewire direktorijuma) koje za razliku od običnih služe za dinamičko renderovanje i vrlo su korisne.



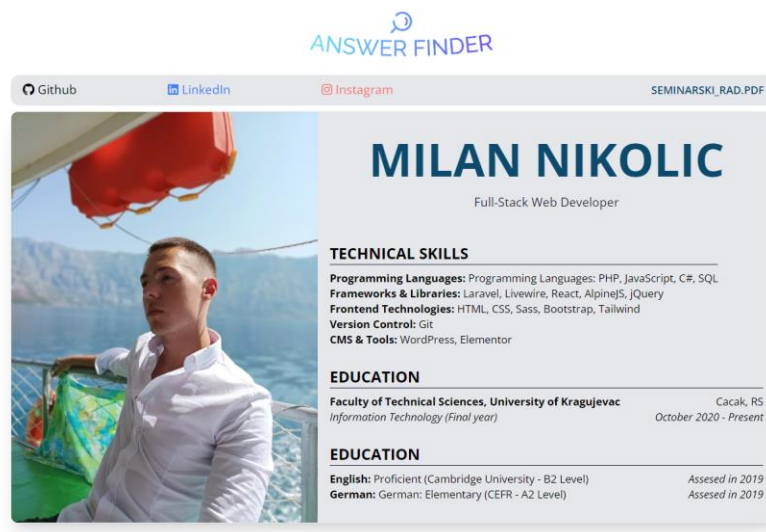
Slika 5 Navbar

Slika u navbaru predstavlja sliku ulogovanog korisnika. Slika se, u zavisnosti od toga da li je korisnik ima, vuče iz baze na osnovu njenog imena ili se ipak koristi „no-image.jpg“ slika za korisnike koji je nemaju.



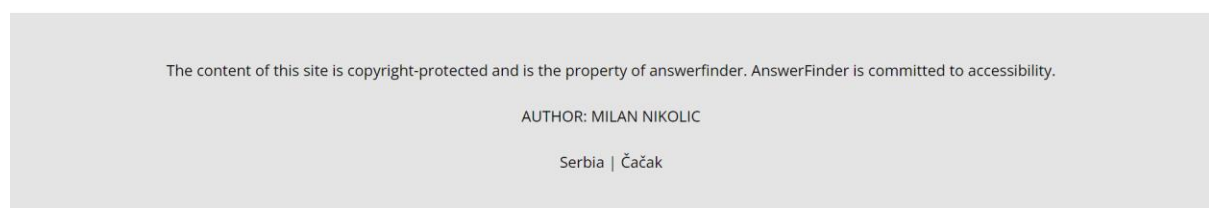
Slika 6 Početna strana

Početna strana prikazuje sva pitanja i mesto za kreiranje novog pitanja ali i mesto za filtriranje i pretragu. Svaka od ovih delova stranice je Livewire komponenta.



Slika 7 Stranica About Author

Stranica „About author“ je jedna od rektih stranica na ovom websajtu koja ne sadrži ni jednu komponentu. Ona je rađena u jednoj celini, i predstavlja jedan „mini CV“.



Slika 8 Footer

**Provident quod hic id voluptas voluptatem in.**

Commodi sint ea similique provident maxime. Officia voluptatem et sunt aspernatur. Omnis non nulla ullam aperiam fugit saepe. Consequatur quam alias tempore sit aut voluptatem enim qui. Optio quis non nihil quo quis in...

[Milan](#) • 4 days ago • 2 comments

**Et omnis aut sequi amet.**

Eum qui repellendus necessitatibus iure. Animi quos suscipit autem nesciunt voluptatibus maxime. Inventore saepe quia omnis suscipit cumque et et. Minus quod consequuntur iste perferendis eum. Ullam ducimus autem iusto...

[Donna Jerde](#) • 4 days ago • 2 comments

Slika 9 Pregled pitanja

Newest first ▾

Filter ▾

Search

Slika 10 Sortiranje, filteri i pretraga

< Back

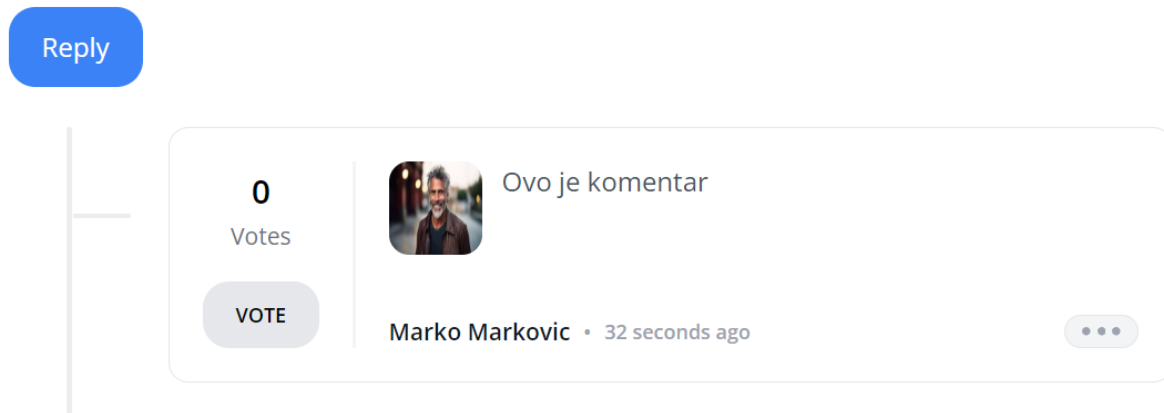
**Test123123**

TEadlkjf akjsldf lkj;dsalkjf ;dsa;lk fkjdsa;lkf aslkdf jlkdsaf ;lkdsalkj fsd

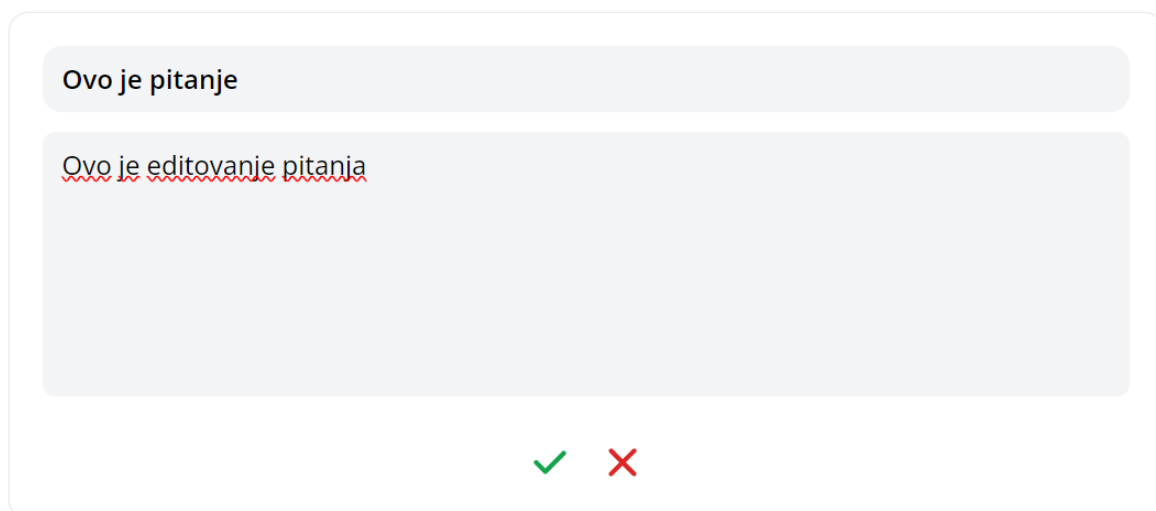
[Milan Nikolic](#) • 1 hour ago • 2 comments

Slika 11 Singularno pitanje

Prethodne komponente su sastavni deo jednog view-a (posts-index.blade.php). Oni se nalaze na istom view-u ali su komponente potpuno odvojene što se tiče funkcionalnosti i lokacije. Sve komponente ovog tipa se nalaze u views/livewire direktorijumu ali u odvojenim fajlovima.

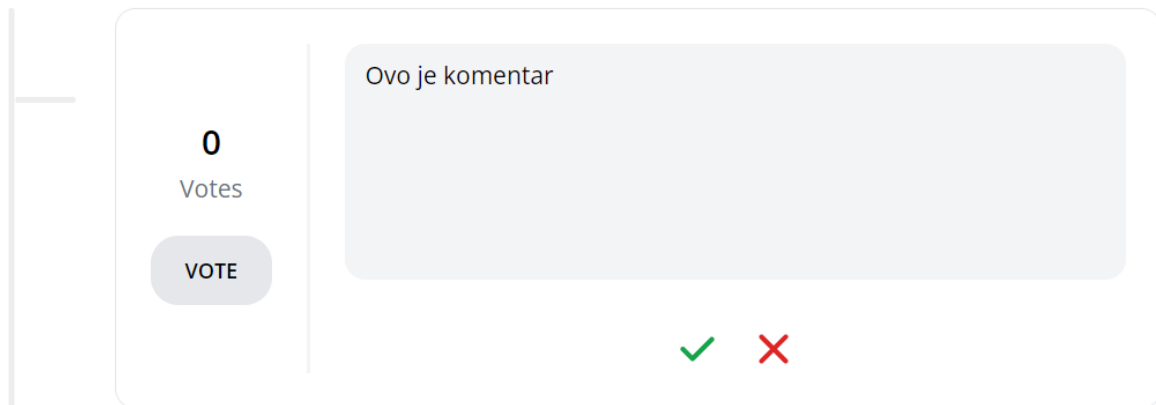


Slika 12 Komentar

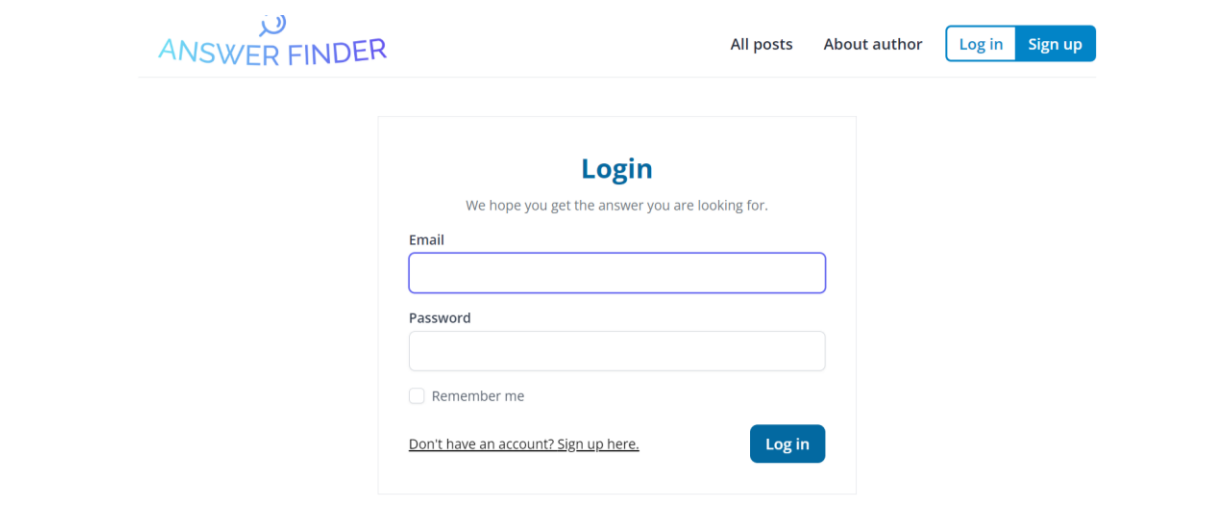


Slika 13 Editovanje pitanja

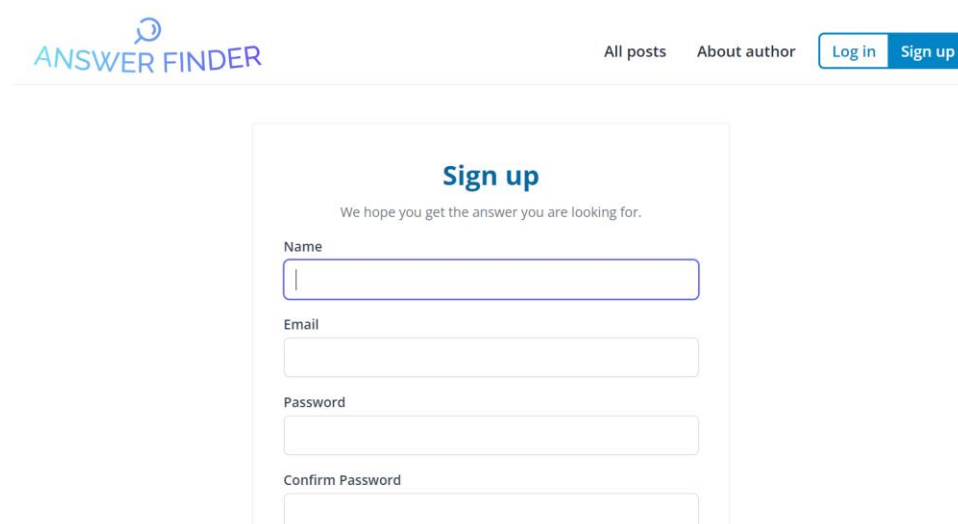
Editovanje pitanja se vrši dinamički koristeći PostEdit livewire komponentu u post-edit.blade.php pregledu. Logika za editovanje se nalazi u PostDAO klasi (repozitorijum).



Slika 14 Editovanje komentara

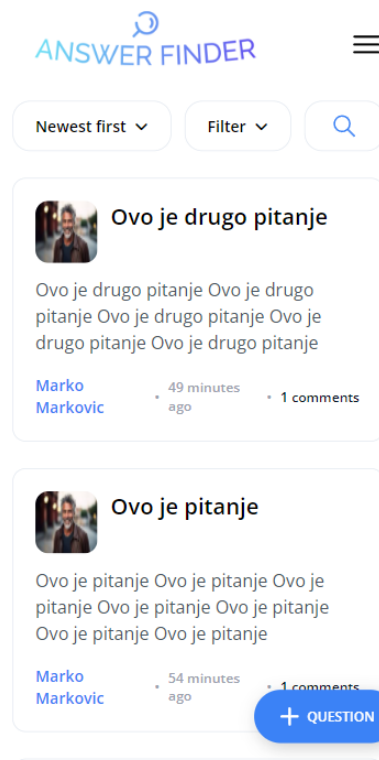


Slika 15 Login strana



Slika 16 Register strana

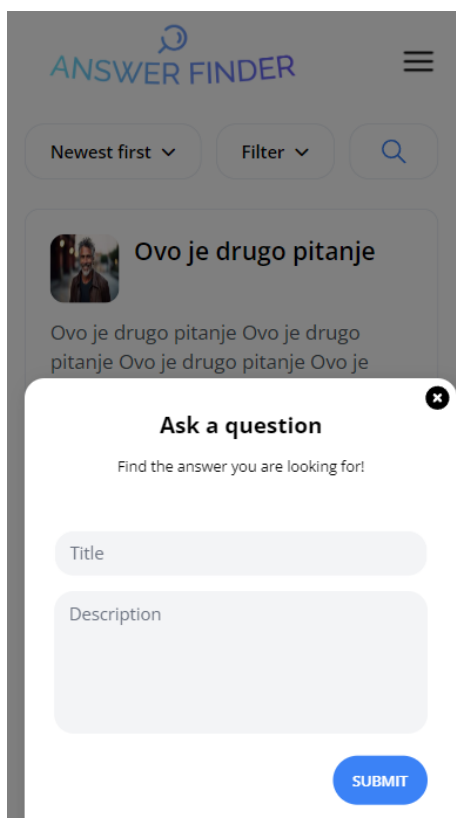
Login i register strane su sastavni deo Laravel/Breeze s' tim što su korigovane ovom projektu. One su automatski generisane.



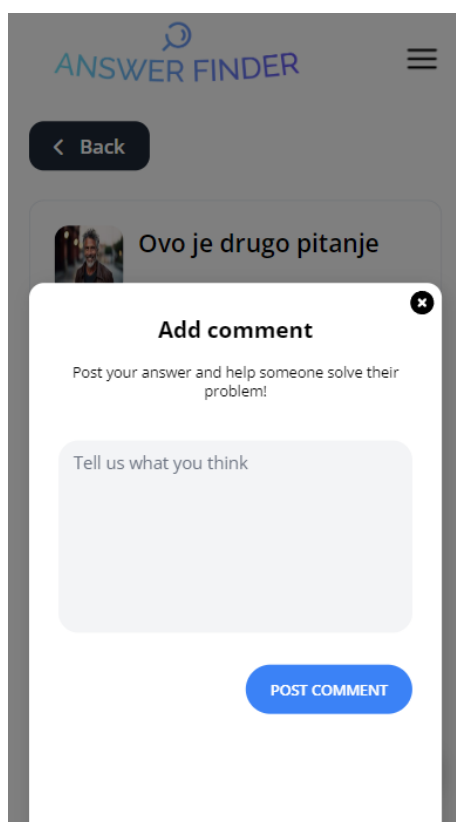
Slika 17 Početna strana na telefonu



Slika 18 Post show strana na telefonu



Slika 19 Kreiranje pitanja na telefonu



Slika 20 Kreiranje komentara na telefonu

### 4.1. Controller (C)

Kontroleri korišćeni u ovom projektu su: PagesController, PostsController, ProfileController, UserImageController. Kontroler ProfileController je automatski generisan kontroler od strane Laravel/Breeze dodatka za autentifikaciju.

TIP	RUTA	AKCIJA	PARAMETRI
GET	/dashboard	index	
GET	/posts	index	
GET	/about-author	index	
GET	/posts/{post}	show	Post
POST	/image-upload	store	
GET	/login	index	
POST	/login/{user}	login	User
GET	/register	register	
POST	/register/{user}	Store	User

Primeri celih ruta:

```
Route::get('/', [PostController::class, 'index'])->name('posts.index');

Route::get('/posts/{post}', [PostController::class, 'show'])->
>name('posts.show');

Route::get('/about-author', [PagesController::class, 'index'])->name('about-
author');

Route::get('/dashboard', function () {
    return view('dashboard');
})->middleware(['auth', 'verified'])->name('dashboard');

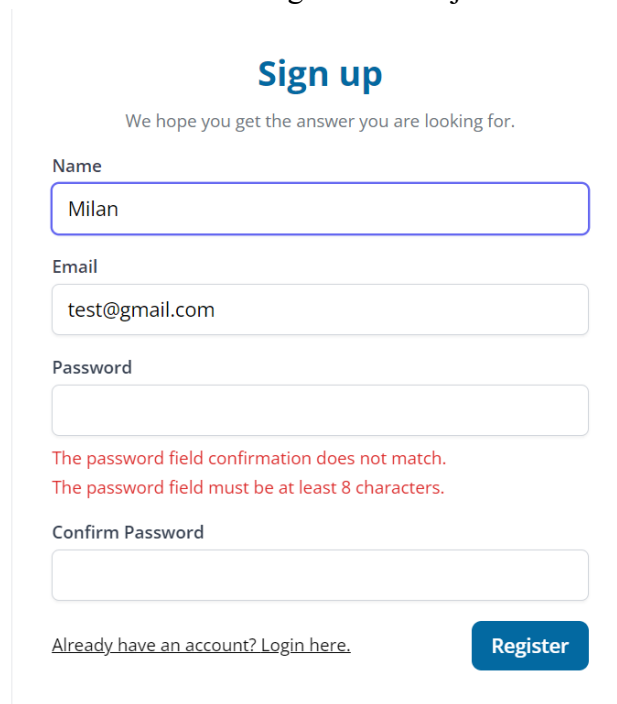
Route::post('/image-upload', [UserImageController::class, 'store'])->
>middleware(['auth', 'verified'])->name('upload.img');

Route::middleware('auth')->group(function () {
    Route::get('/profile', [ProfileController::class, 'edit'])->
>name('profile.edit');
    Route::patch('/profile', [ProfileController::class, 'update'])->
>name('profile.update');
    Route::delete('/profile', [ProfileController::class, 'destroy'])->
>name('profile.destroy');
});
```



## 5. Realizovane funkcionalnosti.

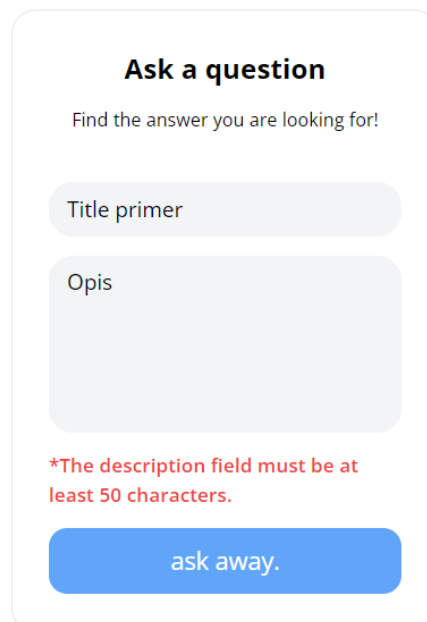
U nastavku ću vam prikazati sve realizovane funkcionalnosti u okviru projekta „Answer Finder“ uz dodatne slike radi lakšeg razumevanja.



The screenshot shows a 'Sign up' form with the following elements:

- Sign up** (blue heading)
- We hope you get the answer you are looking for.*
- Name** field: Milan
- Email** field: test@gmail.com
- Password** field: (empty)
- Validation messages (in red):
  - The password field confirmation does not match.
  - The password field must be at least 8 characters.
- Confirm Password** field: (empty)
- [Already have an account? Login here.](#)
- Register** button

Slika 21 Validacija za registraciju

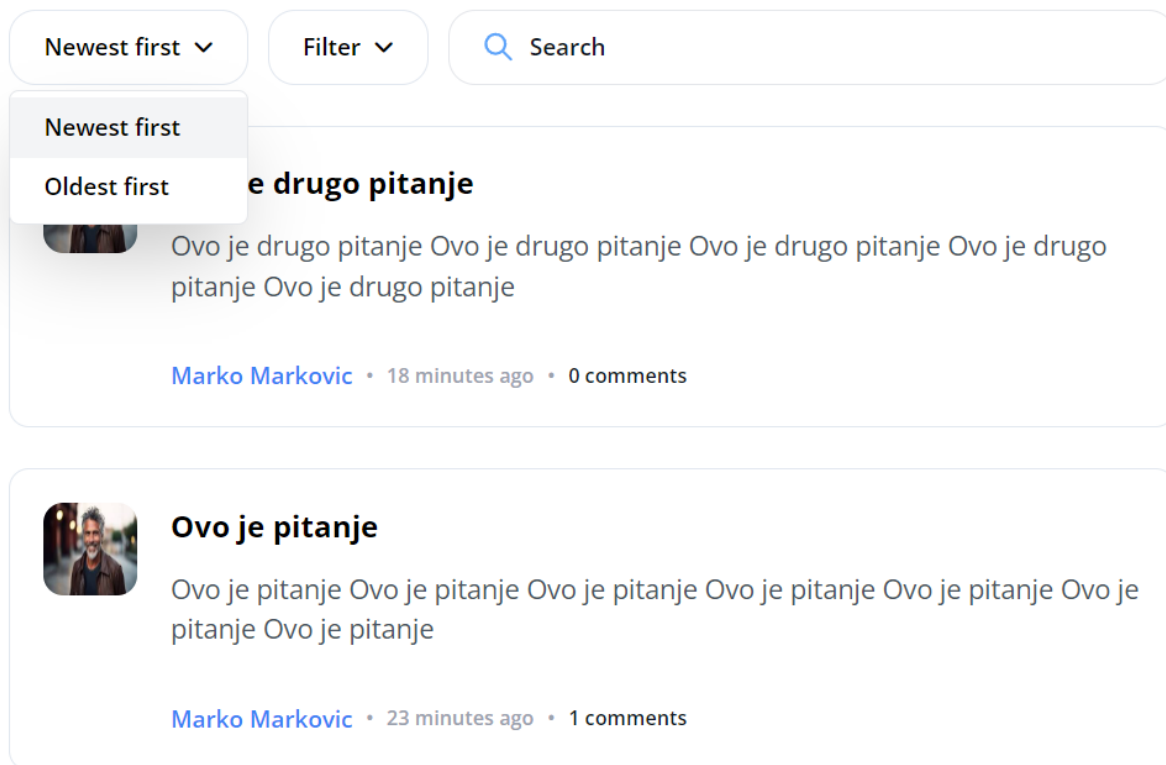


The screenshot shows an 'Ask a question' form with the following elements:

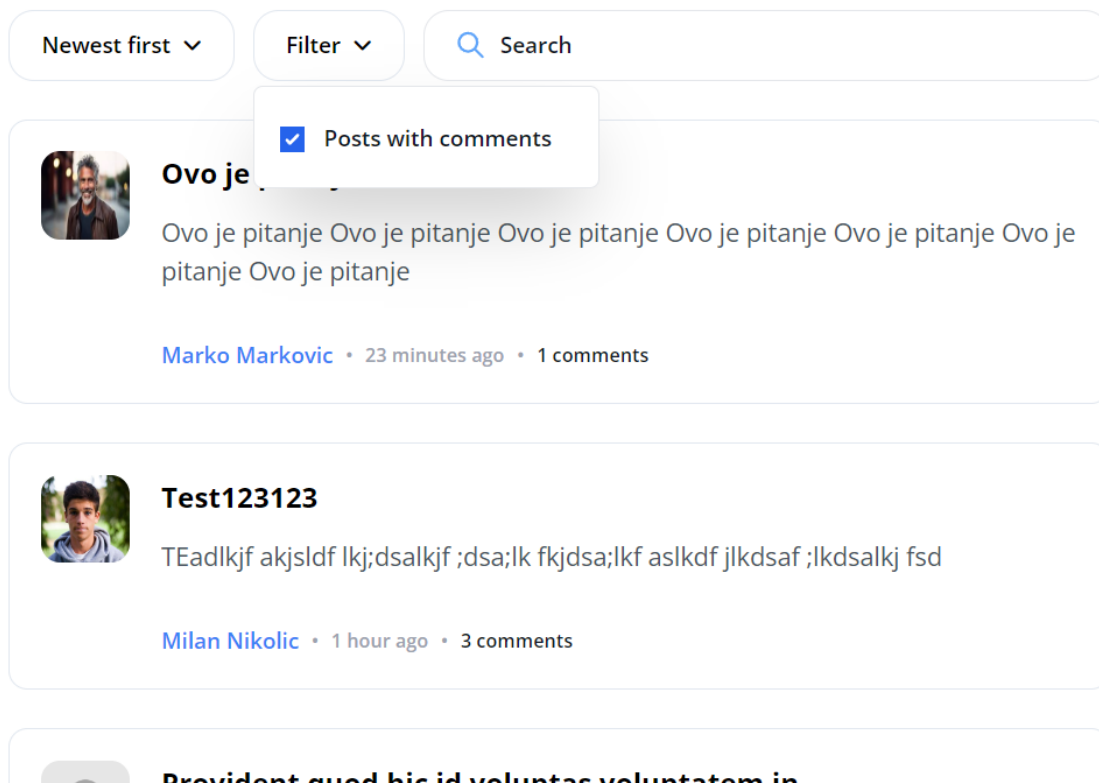
- Ask a question** (black heading)
- Find the answer you are looking for!*
- Title primer** field
- Opis** field
- Validation message (in red):
  - \*The description field must be at least 50 characters.
- ask away.** button

Slika 22 Validacija za kreiranje pitanja

Validacija je ključan deo svakog websajta. Ovde je prikazana validacija za minimalni broj karaktera za opis pitanja kao i uslovi za biranje šifre prilikom registracije novih korisnika.



Slika 23 Sortiranje po vremenu objave



Slika 24 Prikazivanje pitanja koje imaju komentare



Slika 25 Pretraga u realnom vremenu

Sortiranje, filtriranje i pretraga su urađeni korišćenjem Livewire-a za dinamički prikaz. To znači da se sve dešava instantno. Nakon što se ukuca više od 2 karaktera u pretragu, automatski se aktivira query koji prikazuju pitanja čiji naslov sadrži ta dva karaktera.

To je urađeno koristeći sledeći kod:

```
public function getAllPosts($search, $hasComments, $sortDirection){
    return Post::when(strlen($search) >= 2, function($query) use
($search){
        return $query->where('title', 'like', '%'.$search.'%');
    })
    ->when($hasComments, function($query) {
        return $query->whereHas('comments', function($query) {
            $query->where('id', '>', 0);
        });
    })
    ->orderBy('created_at', $sortDirection)
    ->paginate(5);
}
```



Slika 26 Paginacija

Paginacija je urađena koristeći Livewire dodatak WithPagination i dodavanjem

```
->paginate(5);
```

na query, gde 5 označava broj postova po strani.

## RECENTLY SEEN



**Ovo je pitanje**


Ovo je pitanje Ovo je pitanje Ovo je pitanje Ovo je pitanje Ovo je pitanje Ovo je pitanje Ovo je pitanje

Marko Markovic • 24 minutes ago • 0 comments

Slika 27 Poslednje pogledano pitanje

**RECENTLY SEEN** je funkcionalnost koja je odrađena koristeći kolačiće. Kada korisnik uđe u neki post (pitanje) u **Cookies::queue** se ubacuju podaci o tom postu. Nakon vraćanja na početnu stranu, PostsIndex Livewire komponenta renderuje ovaj deo strane, koristeći podatke iz kolačića.

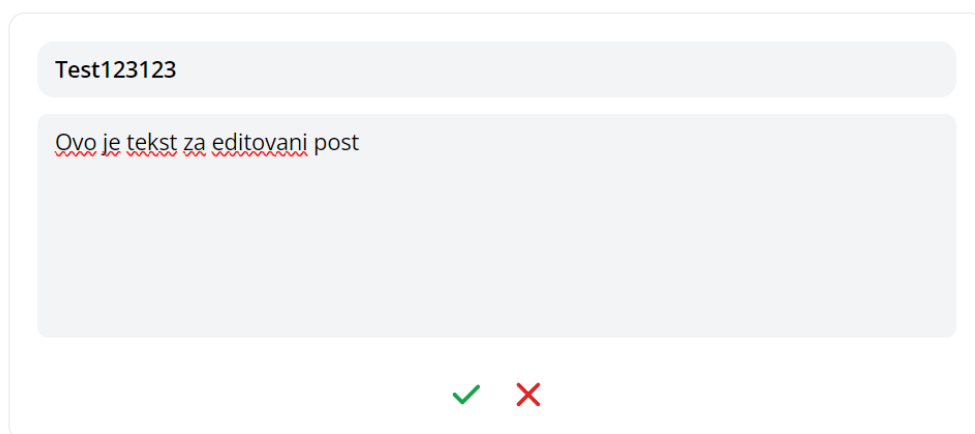
```
Cookie::queue('title', $this->post->title, 60);
Cookie::queue('description', $this->post->description, 60);
Cookie::queue('user_id', $this->post->user->id, 60);
Cookie::queue('post_id', $this->post->id, 60);
Cookie::queue('created_at', $this->post->created_at, 60);
Cookie::queue('user_name', $user_name, 60);
Cookie::queue('commentsCount', $this->post->comments()->count(), 60);
Cookie::queue('img', $img_name, 60);
```

All posts   About author   [Sign out](#) 

No file chosen

Slika 28 Uploadovanje slike

Uploadovanje slike se radi tako što se prvo validira slika, i nakon toga izabere ime za sliku na osnovu id-a korisnika i vremena kada je slika upload-ovana. Nakon toga se skladišti u app/storage. S' obzirom da sa frontend-a nemamo pristup tom folderu, potrebno je da se linkuje folder storage sa folderom po izboru koji se nalazi u answerfinder/public direktorijumu. Nakon što se linkuju, svaka slika koja je u storage folder, naći će se i u novom folderu po izboru unutar public direktorijuma.

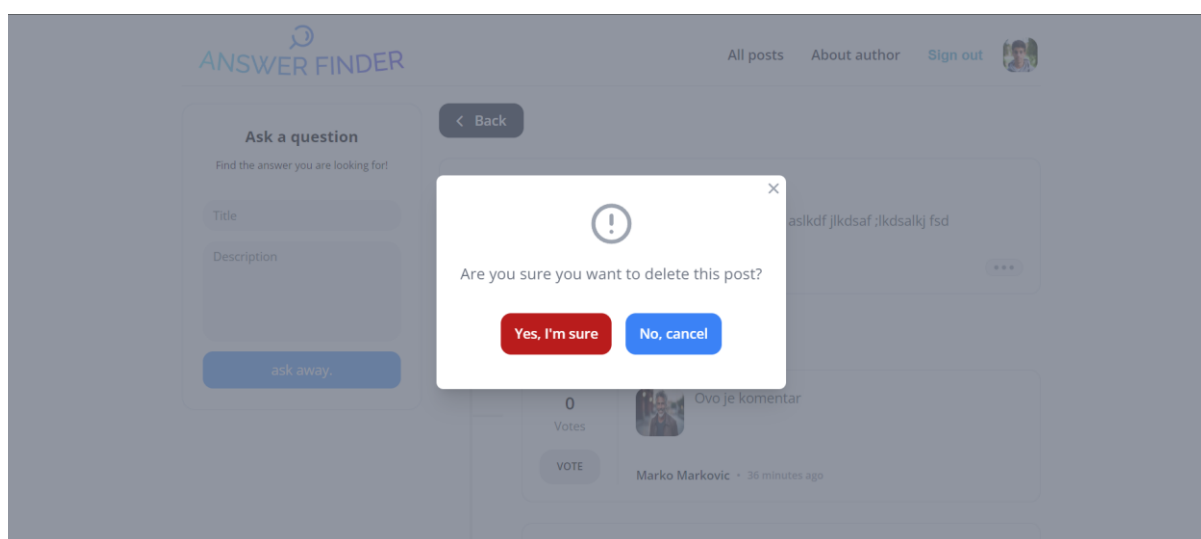


Test123123

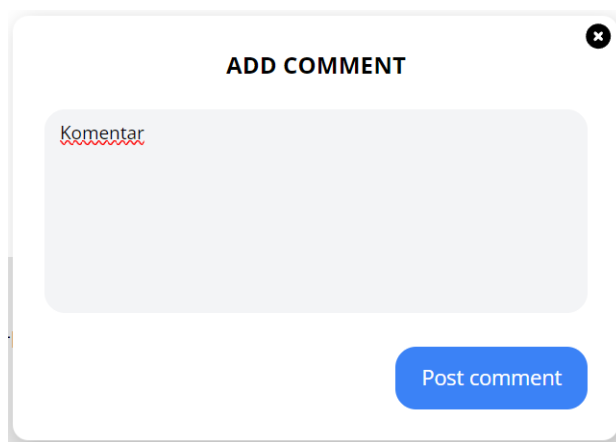
Ovo je tekst za editovani post

✓ ✗

Slika 29 Editovanje post-a



Slika 30 Brisanje post-a

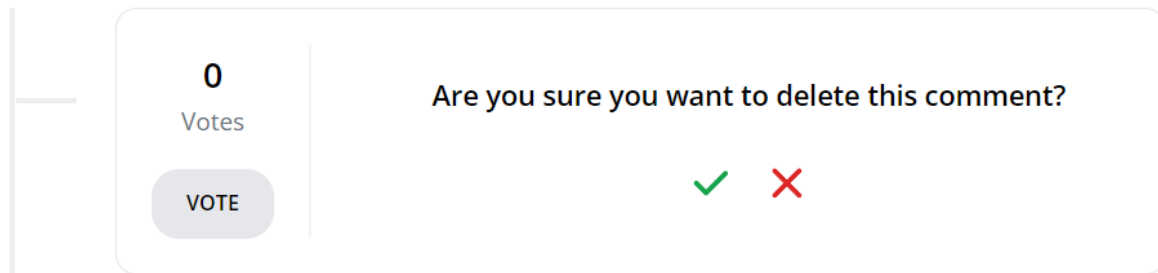


ADD COMMENT

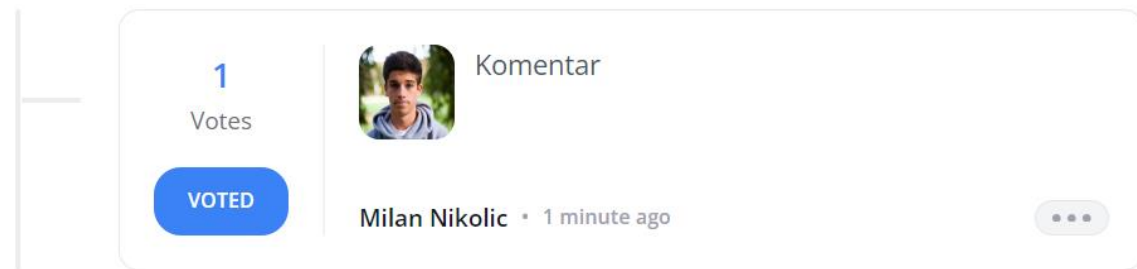
Komentar

Post comment

Slika 31 Kreiranje komentara



Slika 32 Brisanje komentara



Slika 33 Glasanje komentara

CRUD operacije nad postovima i komentarima su odrađene koristeći DAO klase, odnosno u Laravelu zvane Repozitorijume. Metode za izvršavanje operacija su ranije navedene.