

Univerzitet Singidunum

Tehnički fakultet

Kolokvijum 1

Predmet:

Napredno Softversko Inženjerstvo

Profesor:

Prof. dr Angelina Njeguš

Predmetni asistent:

Petar Biševac

Student:

Božidar Mladenović 2018240254

Beograd, 2022. Godina

1 od 15

1. Uvod.....	3
2. Opis projekta	3
2.1 ERD Dijagram.....	4
3. Postupak kreiranja projekta	5
3.1 Primer Korišćena Laravel Eloquent ORM-a	6
3.2 Primer korišćenja Laravel query builder-a	10
3.2.1 Slike upita.....	11
4. Slike rešenja	13
5. Zaključak.....	15
6. Literatura.....	15

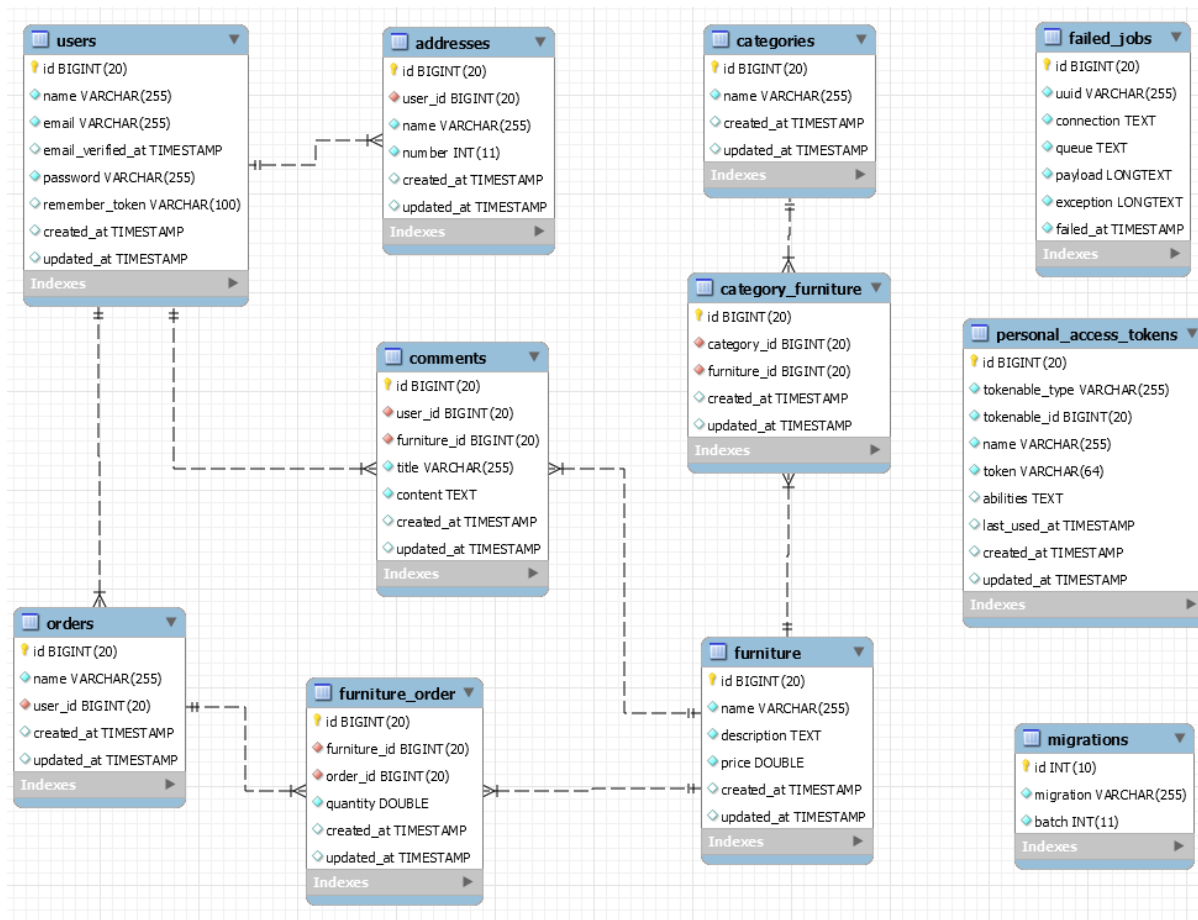
1. Uvod

U ovoj dokumentaciji prikazan je prototip transakcionog sistema za web prodavnicu nameštaja, koji treba da korisnicima omogući pregledanje, odabir i poručivanje nameštaja. U skladu sa trenutnim zahtevima ovog projekta ono što je prikazano i ono što je odrađeno je baza podataka, pravilno kreiranje modela i povezivanje aplikacije sa njom, kao i neki upiti prema njoj za izvlačenje potrebnih informacija. Aplikaciji je potrebno definisati celu poslovnu logiku i korisnički interfejs da bi izašla u produkciju.

2. Opis projekta

Kao što je u uvodu ove dokumentacije spomenuto aplikacija i dalje ne poseduje implementiranu poslovnu logiku, ali ona bi u početku glasila ovako: Svaki posetilac sajta mogao bi da pregleda kolekciju nameštaja, da istraži nešto više o nekom komadu koji ga zanima kao i da čita komentare o tom komadu nameštaja. Za funkciju poručivanja nameštaja posetilac bi morao da napravi svoj korisnički nalog sa njegovim osnovnim informacijama, takođe ako bi korisnik želeo da ostavi komentar nekom komadu nameštaja i tako pomogne drugim kupcima u odabiru, pored korisničkog naloga potrebno je i da taj korisnik prethodno naruči i isproba sam komad nameštaja. Aplikacija će takođe posedovati i admin korisnike kao i admin panel koji će služiti za unos nameštaja u bazu, ažuriranje postojećeg nameštaja, brisanje nepotrebnog nameštaja iz baze i osnovna podešavanja sajta. U nastavku prikazan je ERD dijagram.

2.1 ERD Dijagram



(Slika 1. Prikaz ERD dijagrama)

Kao što je prikazano na slici 1 aplikacija sadrži 8 tabela za njeno funkcionisanje i 3 tabele koje su automatski kreirane tehnologijom kojom je ova aplikacija rađena. Imamo tabelu „users“ koja služi za skladištenje podataka o korisniku, povezana je na tabelu „addresses“ gde se čuvaju podaci o adresama korisnika (Može da ima više adresa, veza 1xN), zatim za tabelu „users“ povezana je i tabela „comments“ koja služi da čuva komentare korisnika (Svaki korisnik može da postavi više komentara, veza 1xN), ona je takođe povezana i na tabelu „furniture“ jer jedan komad nameštaja može da ima više komentara opet veza 1xN, tako da za ovu vezu možemo da kažemo da je veza više prema više (MxN), na kraju tabela „users“ je povezana za tabelu „orders“ koja služi za skladištenje porudžbina vezom 1xN, jer svaki korisnik može imati više porudžbina. Tabela „furniture“ koja služi za skladištenje nameštaja veznom tabelom „furniture_order“ je povezana za tabelu „orders“ jer u jednoj porudžbini mogu da se nalaze više komada nameštaja i za jedan komad nameštaja možemo se imati više porudžbina (ponovo veza MxN), takođe tabela „furniture“ je veznom tabelom „category_furniture“ povezana i za tabelu „categories“ koja služi za skladištenje kategorija nameštaja i ovde jedan komad nameštaja može da ima više kategorija, a u jednu kategoriju mogu da spadaju više komada nameštaja (MxN).

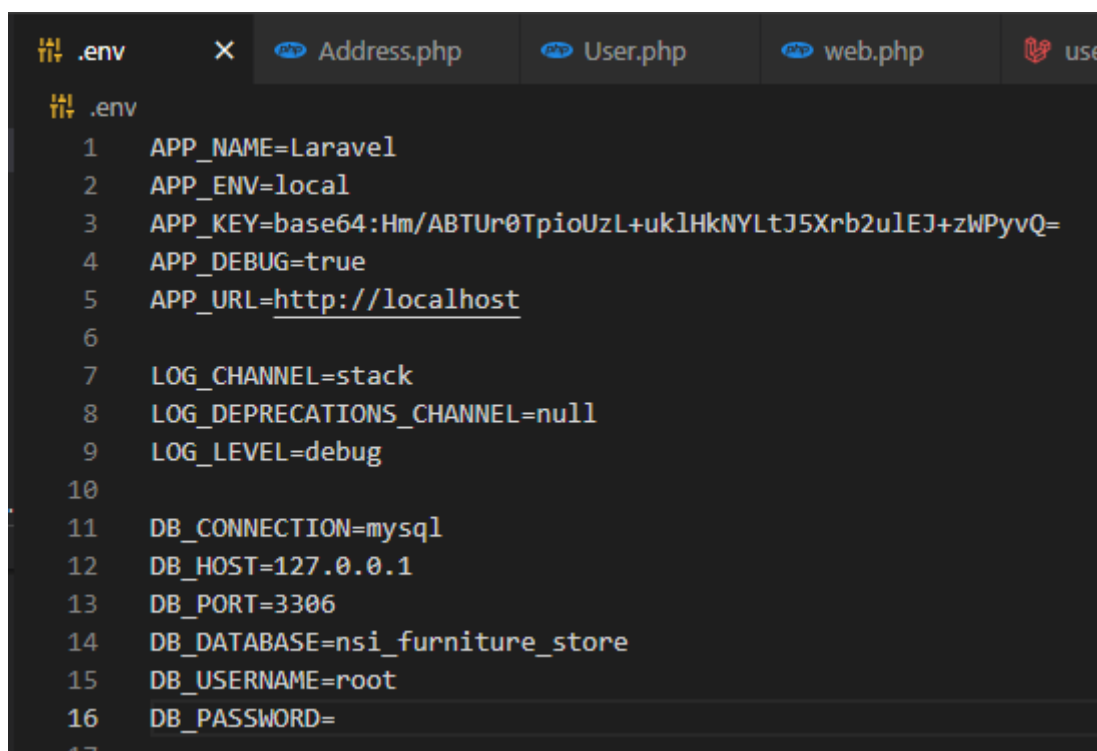
3. Postupak kreiranja projekta

Projekat je pisan u PHP-u to jest za izradu projekta korišćen je php framework „Laravel“ i radi se o MVC arhitekturi projekta, dok je za bazu korišćen MySQL. Sto se tiče razvojnog okruženja za sam kod korišćen je „Visual Studio Code“, a za bazu „MySQL Workbench“.

Ono što je bilo potrebno za kreiranje ovog projekta jeste naravno server za MySQL i composer uz pomoc kog se pokreće instalacija laravel projekta. MySQL server je preuzet sa Xampp paketom, a u nastavku će detaljnije biti objašnjena instalacija samog projekta:

- Instalacija composer-a na najlakši način se vrši tako što se sa sajta (<https://getcomposer.org/>) preuzme i zatim pokrene instalacija
- Zatim se uz pomoć composer-a preuzima „Laravel installer“ komandom:
composer global require laravel/installer u terminalu
- Uz pomoć „Laravel installer-a“ se kreira novi projekat komandom:
laravel new example-app
- Na kraju se server pokreće komandom:
php artisan serve

Povezivanje baze sa projektom se vrši uz pomoc „.env“ konfiguracionog fajla za laravel aplikaciju i ime šeme baze je „nsi_furniture_store“



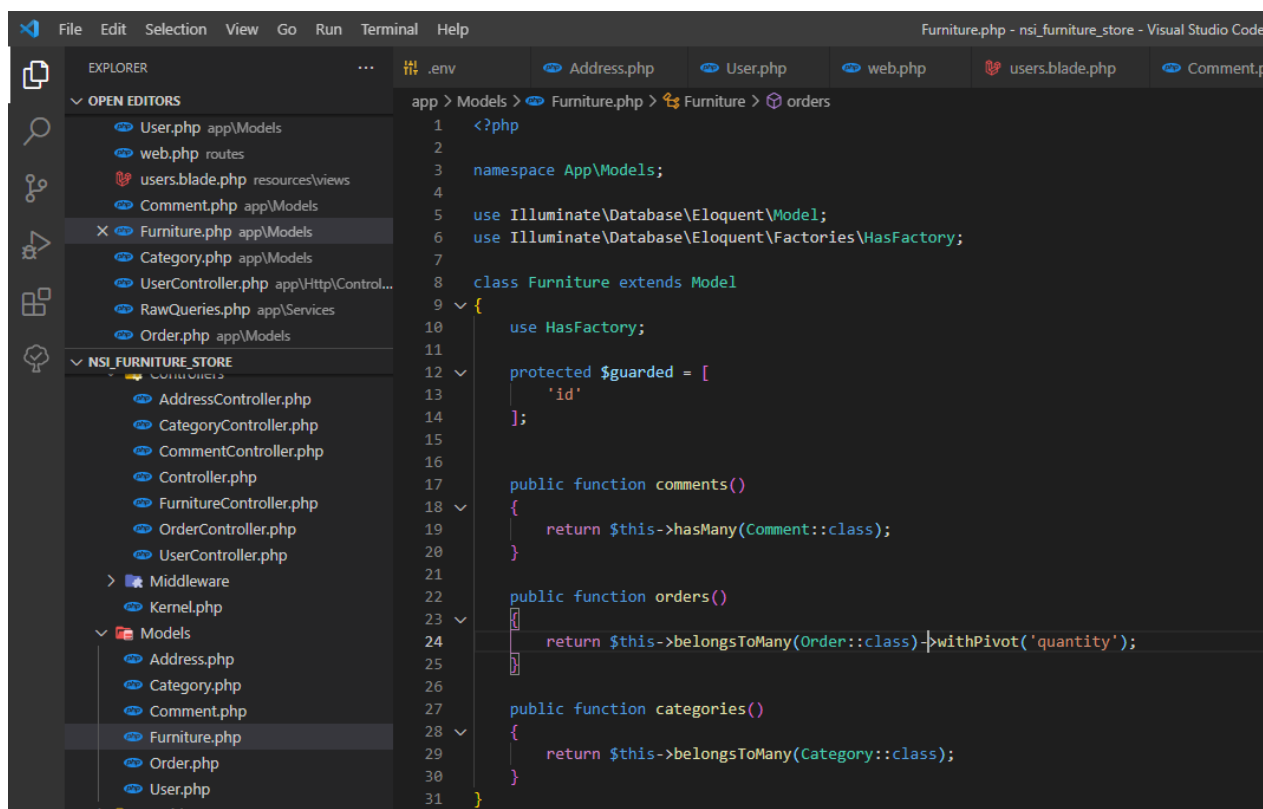
```
.env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:Hm/ABTUr0TpioUzL+uklHkNYLtJ5Xrb2ulEJ+zWPyvQ=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=nsi_furniture_store
15 DB_USERNAME=root
16 DB_PASSWORD=
17
```

(Slika 2. deo „.env“ konfiguracionog fajla)

Što se tiče komunikacije sa bazom ona se vrši uz pomoć ORM-a (object-relational mapper) pod nazivom „Eloquent“ koji je ugrađen u sam laravel i za ručno kreiranje upita korišćen je „Laravel database query builder „.

3.1 Primer Korišćena Laravel Eloquent ORM-a

Da bi se koristio Eloquent ORM za tabelu iz koje se izvlače, upisuju, ažuriraju i brišu podaci potrebno je imati model, primer jednog modela prikazan je na slici ispod



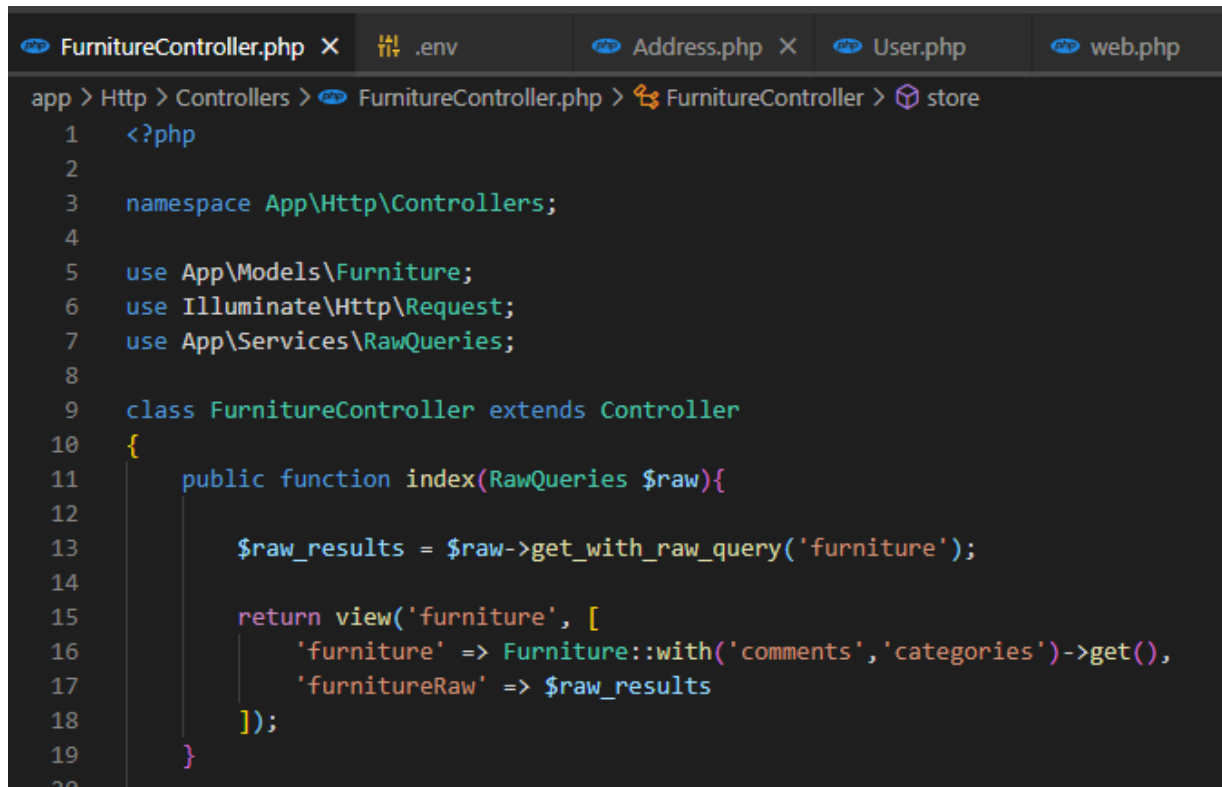
(Slika 3. Prikaz „Furniture“ modela)

Na slici 3 je prikazan Eloquent model za tabelu „furniture“ iz baze, jako je prost i u njemu se nalaze funkcije za povezivanje modela i tabela, pa tako na slici su prikazane 3 veze, veza modela „Furniture“ sa modelima „Comment“, „Order“ i „Category“. Veza sa komentarima je veza 1xN pa se koristi „hasMany“ što znači da jedan komad nameštaja ima više komentara u modelu „Comment“ postoji veza ka ovom modelu nazad uz pomoć veze „belongsTo“ (Slika 4) što znači da jedan komentar pripada jednom komadu nameštaja. Veze „belongsToMany“ se koriste kada postoji MxN veza između modela i tabela. Na liniji 24 se pored veze „belongsToMany“ vidi i funkcija „withPivot“ uz pomoć koje dohvatamo dodatne podatke iz vezne tabele, pivot je u laravelu vezna tabela, tako da modeli za vezne tabele nisu potrebni aplikaciji.

```
public function user()
{
    return $this->belongsTo(User::class);
}
```

(Slika 4. Veza „belongsTo“ u Comment modelu)

Korišćenje ovih veza to jest pozivanje ovih funkcija se odvija u kontrolerima na sledećoj slici biće prikazana funkcija za dohvaćanje podataka iz baze, a posle toga i ceo jedan takav kontroler u kome se nalaze i funkcije za unos, ažuriranje i brisanje.

The image shows a code editor with a dark theme. At the top, there are tabs for 'FurnitureController.php', '.env', 'Address.php', 'User.php', and 'web.php'. The 'FurnitureController.php' tab is active. Below the tabs, the breadcrumb path is 'app > Http > Controllers > FurnitureController.php > FurnitureController > store'. The code is as follows:

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Furniture;
6  use Illuminate\Http\Request;
7  use App\Services\RawQueries;
8
9  class FurnitureController extends Controller
10 {
11     public function index(RawQueries $raw){
12
13         $raw_results = $raw->get_with_raw_query('furniture');
14
15         return view('furniture', [
16             'furniture' => Furniture::with('comments','categories')->get(),
17             'furnitureRaw' => $raw_results
18         ]);
19     }
20 }
```

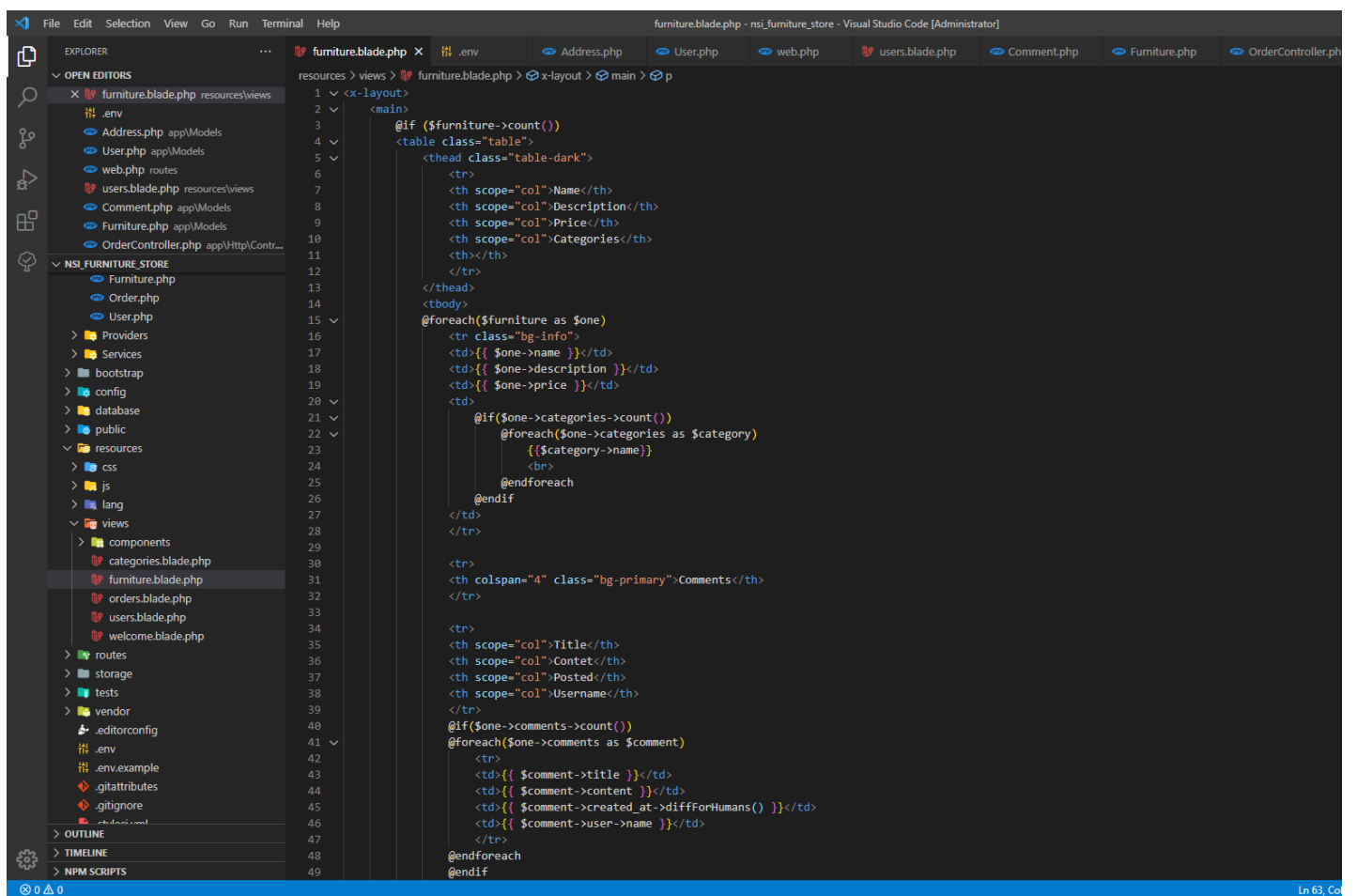
(Slika 5. Prikaz „index“ funkcije u kontroleru)

Na slici 5 je prikazan deo „FurnitureController-a“ tačnije „index“ funkcija kontrolera, koja kako se vidi u return naredbi vraća pogled (view) sa nekim podacima to jest šalje se niz podataka. Prvi element ovog niza jesu podaci o komadima nameštaja uz pomoć Furniture modela, na liniji 16 se poziva Furniture klasa i dodeljuje joj se funkcija „with“ koja specifikira da se traže komadi nameštaja sa njihovim komentarima i kategorijama (argumenti ove funkcije su veze iz modela) i uz pomoć funkcije „get“ se izvlače iz baze i šalju se na view. Pored with i get funkcija postoje i mnogo drugih koje čine ovaj sistem još moćnijim neke od njih su map, pluck, where (slika 6) itd. Drugi argument ovog niza „furnitureRaw“ je dohvaćanje podataka uz pomoć „Laravel query buildera“ o kome će biti reči u nastavku ove dokumentacije. Kada se sve ovo prosledi na view onda se uz pomoć „blade template engine-a“ jednostavnim ulančavanjem sve ovo prikazuje (Slika 7).

```
.env Address.php User.php web.php users.blade.php Comment.php
app > Http > Controllers > OrderController.php > OrderController > destroy

2
3 namespace App\Http\Controllers;
4
5
6 use App\Models\Order;
7 use Illuminate\Http\Request;
8 use App\Services\RawQueries;
9
10 class OrderController extends Controller
11 {
12     public function index(RawQueries $raw){
13
14         $raw_results = $raw->get_with_raw_query('orders');
15
16         return view('orders', [
17             'orders' => Order::where('user_id', '=', 1)->with('furniture')->get(),
18             'ordersRaw' => $raw_results
19         ]);
20     }
21 }
```

(Slika 6. Primer „where“ funkcije)



(Slika 7. Izgled jedne „view“ stranice)


```
19
20
21 public function store(Request $request)
22 {
23     $request->validate([
24         'name'=>'required|string',
25         'description'=>'required|string',
26         'price'=>'required|numeric'
27     ]);
28
29     $furniture = new Furniture([
30         'name' => $request->get('name'),
31         'description' => $request->get('description'),
32         'price' => $request->get('price')
33     ]);
34     $furniture->save();
35
36     return redirect('/')->with('success', 'Furniture created.');
```

```
37
38
39 public function update(Request $request, $id)
40 {
41     $request->validate([
42         'name'=>'required|string',
43         'description'=>'required|string',
44         'price'=>'required|numeric'
45     ]);
46
47     $furniture = Furniture::find($id);
48     $furniture->name = $request->get('name');
49     $furniture->description = $request->get('description');
50     $furniture->price = $request->get('price');
51     $furniture->save();
52
53     return redirect('/')->with('success', 'Furniture updated.');
```

```
54
55
56 public function destroy($id)
57 {
58     $furniture = Furniture::find($id);
59     $furniture->delete();
60
61     return redirect('/')->with('success', 'Furniture removed.');
```

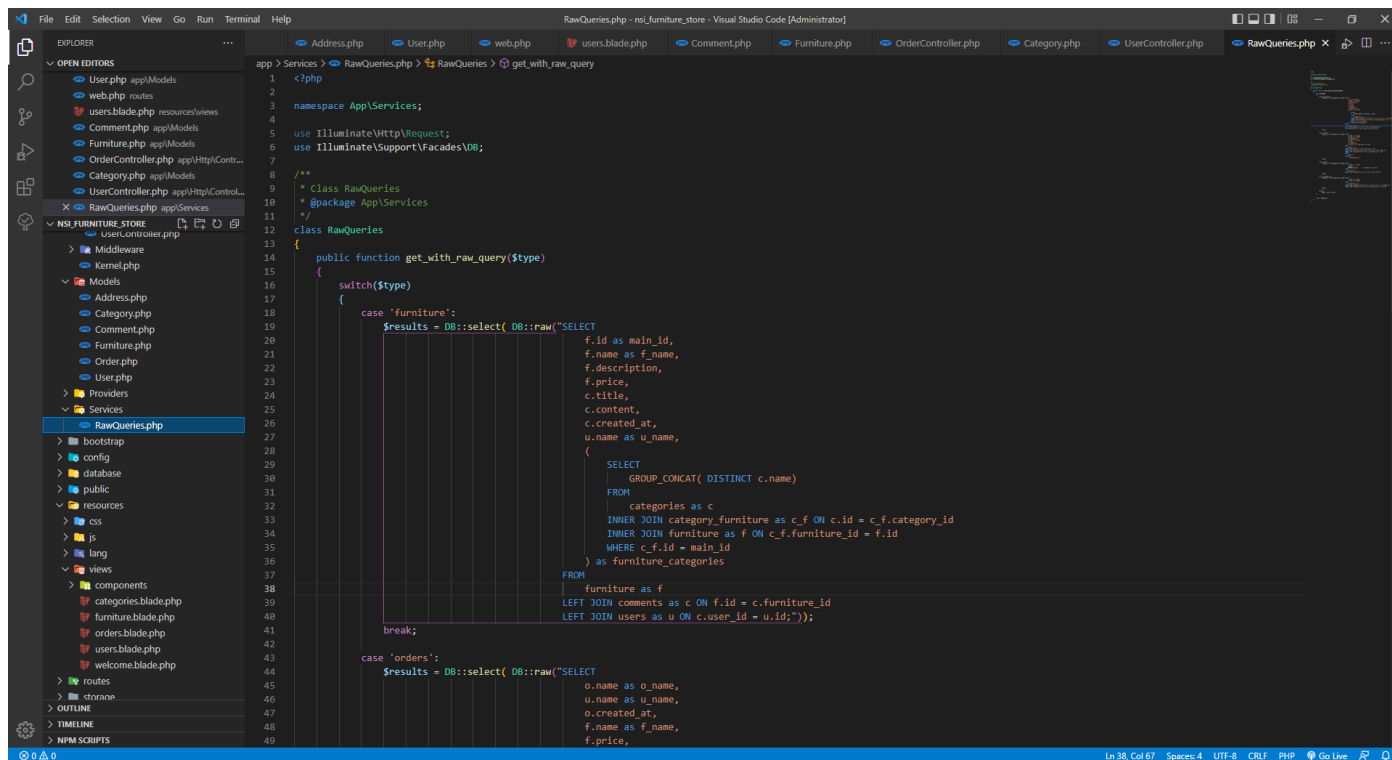
```
62
63 }
64
```

(Slika 8. Ostatak koda „FurnitureController-a“)

Na Slici 8 je prikazan ostatak koda FurnitureController-a to jest prikazane su funkcije za unos, ažuriranje i brisanje podataka iz baze. U praksi je najbolje držati kontrolere što čistijim pa se kod većih projekata preporučuje da se kodovi vezani za ove operacije prebace u servise i u kontrolerima samo da se pozivaju. Takođe za svaku validaciju koje se nalaze na početku funkcija za ažuriranje i unos postoji poseban folder koji se zove „Requests“ u kojima se kreiraju fajlovi za validaciju i ovaj kod se prebacuje tamo.

3.2 Primer korišćenja Laravel query builder-a

Kada Eloquent nije dovoljan i kada u aplikaciji postoje jako kompleksni upiti, laravel nudi rešenje kreiranja upita uz pomoć query builder-a, u nastavku biće prikazano njegovo korišćenje.



(Slika 9. Prikaz servisa u kome se koristi query builder)

U ovom projektu postoji servis koji se naziva „RawQueries“ u kome se nalaze neki upiti, ovaj servis je samo tu radi prikazivanja i drugog načina dohvaćanja podataka i vraćanjem na sliku 5, gde je spomenut drugi argument niza, ovo je taj drugi argument znači servis vraća niz podataka iz baze koji smo dohvatili uz pomoć upita i to smešta u promenljivu „raw_results“ i to prosleđuje pogledu pod ključem „furnitureRaw“.

```
57         @if (count($furnitureRaw))
58             @foreach($furnitureRaw as $raw)
59                 <p>Furniture:</p>
60                 <p>Name: {{ $raw->f_name }}</p> <br>
61                 <p>Description: {{ $raw->description }}</p> <br>
62                 <p>Price: {{ $raw->price }}</p> <br>
63                 <p>Categories: {{ $raw->furniture_categories }}</p>
64                 <p>Comment:</p>
65                 <p>Title: {{ $raw->title }}</p> <br>
66                 <p>Content: {{ $raw->content }}</p> <br>
67                 <p>Posted At: {{ $raw->created_at }}</p> <br>
68                 <p>User Name: {{ $raw->u_name }}</p> <br>
69             @endforeach
70         @endif
71     </main>
72 </x-layout>
```

(Slika 10. Nastavak pogleda za komade nameštaja gde se prikazuju podaci uz pomoć „furnitureRaw“)

3.2.1 Slike upita

```
$results = DB::select( DB::raw("SELECT
    f.id as main_id,
    f.name as f_name,
    f.description,
    f.price,
    c.title,
    c.content,
    c.created_at,
    u.name as u_name,
    (
        SELECT
            GROUP_CONCAT( DISTINCT c.name)
        FROM
            categories as c
        INNER JOIN category_furniture as c_f ON c.id = c_f.category_id
        INNER JOIN furniture as f ON c_f.furniture_id = f.id
        WHERE c_f.id = main_id
    ) as furniture_categories
FROM
    furniture as f
LEFT JOIN comments as c ON f.id = c.furniture_id
LEFT JOIN users as u ON c.user_id = u.id;"));
```

(Slika 11. Prikaz upita za dohvaćanje nameštaja sa komentarima i kategorijama)

```
$results = DB::select( DB::raw("SELECT
    o.name as o_name,
    u.name as u_name,
    o.created_at,
    f.name as f_name,
    f.price,
    f_o.quantity,
    f.price * f_o.quantity AS total
FROM
    orders as o
INNER JOIN users as u ON o.user_id = u.id
INNER JOIN furniture_order as f_o ON o.id = f_o.order_id
INNER JOIN furniture as f ON f_o.furniture_id = f.id
WHERE
    u.id = 1
ORDER BY
    o.created_at"));
```

(Slika 12. Prikaz upita za dohvaćanje porudžbina sa podacima o korisniku i stavkama porudžbine)

```
$results = DB::select( DB::raw("SELECT
    u.name as u_name,
    u.email,
    CONCAT(a.name, ' ', a.number) as address
FROM
    users as u
    INNER JOIN addresses as a ON u.id = a.user_id"));
```

(Slika 13. Prikaz upita za dohvaćanje korisnika i njihovih adresa)

```
$results = DB::select( DB::raw("SELECT
    c.name as c_name,
    f.name as f_name
FROM
    categories as c
    INNER JOIN category_furniture as c_f ON c.id = c_f.category_id
    INNER JOIN furniture as f ON c_f.furniture_id = f.id;"));
```

(Slika 14. Prikaz upita za dohvaćanje kategorija i komada nameštaja koji tim kategorijama pripadaju)

Naravno pored select komande postoje i insert, update i delete.

4.Slike rešenja

Name	Description	Price	Categories
Table	Nullam interdum lectus urna. non porttitor eros laoreet at. Praesent pretium ligula metus. Ut consequat maximus erat, nec sodales est egestas vitae. Integer tincidunt, sem sed tempor congue, ipsum turpis fermentum velit, sit amet feugiat tortor nulla id turpis. In et condimentum elit. Curabitur sit amet lorem in metus ullamcorper tristique in et felis. Pellentesque luctus massa velit, consectetur maximus tellus volutpat in. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Praesent tortor orci, interdum ac accumsan et, porta congue diam. Maecenas in placerat nunc. Praesent ultricies finibus enim sed pulvinar. Mauris purus metus, sagittis non pellentesque sed, pellentesque vel lacus.	50	Category1
Comments			
Title	Content	Posted	Username
Comment 1	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur a eros porta, porttitor eros fermentum, scelerisque quam. Cras sit amet tellus pulvinar, imperdiet	1 day ago	Bozidar
Comment3	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur a eros porta, porttitor eros fermentum, scelerisque quam. Cras sit amet tellus pulvinar, imperdiet	1 day ago	TestUser
Chair	Vivamus dolor ipsum, porttitor sit amet dignissim vitae, commodo quis nulla. Morbi id lectus aliquet, posuere purus nec, blandit leo. Interdum et malesuada fames ac ante ipsum primis in faucibus. Ut ipsum massa, egestas et maximus eu, scelerisque a arcu. Donec pharetra eros non arcu sollicitudin mollis. Pellentesque eleifend aliquet ipsum, ac aliquet turpis volutpat quis. Vivamus maximus nulla ut efficitur aliquet. Vestibulum aliquam enim eget elit sodales, non blandit tortor vulputate.	10	Category1
Comments			
Title	Content	Posted	Username
Comment 2	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur a eros porta, porttitor eros fermentum, scelerisque quam. Cras sit amet tellus pulvinar, imperdiet	1 day ago	Bozidar
Bed	Ut pretium consequat neque, eget semper libero vulputate vel. Mauris pretium ac tortor et egestas. Integer maximus maximus velit, quis congue dui tincidunt ut. Proin aliquam dui scelerisque tellus fringilla pulvinar. Fusce pellentesque, mi quis consectetur egestas, nunc odio dictum nisl, a porta enim nibh sodales ante. Nulla cursus nisl vitae accumsan euismod. Vestibulum pharetra nulla vel tortor placerat consectetur.	80	Category2
Comments			
Title	Content	Posted	Username
Shelf	Curabitur ultricies hendrerit erat sed tincidunt. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Fusce mollis sagittis mauris in condimentum. Cras tempor euismod justo, sit amet auctor dolor elementum ac. Mauris maximus lectus at rhoncus gravida. Phasellus tincidunt ut risus in dapibus	30	Category2
Comments			
Title	Content	Posted	Username

(Slika 15. Trenutni izgled „furniture“ stranice)

Name	Email	Addresses
Bozidar	test@test.com	Branka Djurica 10 Alekse Markisica 19
TestUser	test2@test.com	

User:
Name: Bozidar
Email: test@test.com
address: Branka Djurica 10

User:
Name: Bozidar
Email: test@test.com
address: Alekse Markisica 19

(Slika 16. Trenutni izgled „users“ stranice)

Home Furniture Users Orders Categories		
Name		User Name
Order1		Bozidar
Items		
Name	Price	Quantity
Table	50	2
Chair	10	4
Wardrobe	100	1
Order2		Bozidar
Items		
Name	Price	Quantity
Order:		
Name: Order1		
User: Bozidar		
Item:		
Name: Table		
Price: 50		
Quantity: 2		
Total: 100		

(Slika 17. Trenutni izgled „orders“ stranice)

Home Furniture Users Orders Categories	
Category	Furniture
Category1	Table Chair
Category2	Bed Shelf
Category3	Wardrobe
Category: Category1	
Furniture: Table	
Category: Category1	
Furniture: Chair	
Category: Category2	
Furniture: Bed	
Category: Category2	
Furniture: Shelf	
Category: Category3	
Furniture: Wardrobe	

(Slika 18. Trenutni izgled „categories“ stranice)

5. Zaključak

Aplikacija je i dalje u razvoju i potrebno je dosta toga dodati i izmeniti, dalji koraci u razvoju su: potrebno je srediti poslovnu logiku u kodu, kreirati korisnički interfejs, očistiti kontrolere dodatnim servisima i ubaciti sisteme za plaćanje.

6. Literatura

- Laravel.com, Dostupno na: <https://laravel.com/>
- Eloquent ORM, Dostupno na: <https://laravel.com/docs/9.x/eloquent>
- Laravel query builder, Dostupno na: <https://laravel.com/docs/9.x/queries>
- Laravel blade template engine, Dostupno na: <https://laravel.com/docs/9.x/blade#main-content>
- Materijali sa predavanja i vežbi