

Tehnicka dokumentacija

1. Preuzimanje composera <https://getcomposer.org/>
 2. Pokretanje Xamppa i kreiranje baze podata na localhost/phpmyadmin
 3. Pokretanje instalacije laravel projekta:

```
composer create-project laravel/laravel laravel-app
```
 4. Koriscenje alata Blueprint za kreiranje modela i kontrolera. Blueprint nam automatski pravi popunjene modele i popunjene kontrolere, pravi nam popunjene migracije sa nasim vrednostima koje smo pisali u Blueprint fajlu.
- Komanda za instaliranje Blueprinta:
- ```
composer require -W --dev laravel-shift/blueprint
```
5. Koristicemo Breeze za Autentifikaciju, da bi mogli da se Loginujemo i Registrujemo. Pomocu Breeza i middleware, u web.php cemo da namestiti da middleware na ‘auth’ kako bi mogli samo ulogovani korisnici da prate sadrzaj na nasem sajtu.
- Komanda za instaliranje Breeze:
- ```
composer require laravel/breeze --dev
```
- ```
php artisan breeze:install
```
6. Napravicemo Seedere i popunicemo nase tabele sa podacima rucno preko komandi:
- Komanda za kreiranje Seedera: `php artisan make:seeder UserSeeder`
- Komanda za pokretanje Seedera: `php artisan db:seed`
- Komanda za ponovno pokretanje svih Migracija: `php artisan migrate:fresh -seed`
7. Koristimo ChatGPT alat za kreiranje random blejdova koje cemo koristiti za CRUD operacije nad tabelama: kupacs,fakturas,dobavljaes i robas. Za CRUD operacije cemo napraviti blejdove za svaku od ovih tabela, a blejdovi ce biti : index.blade.php, show.blade.php, edit.blade.php i create.blade.php.

8. Dolazimo do greske prilikom prikazivanja Tabele Fakturas, jer uzimamo primarni kljuc i podatke iz tabele Kupacs. Da bi smo resili taj problem, moramo izmeniti dva podatka u dve putanje: App\Http\Request\FakturaStoreRequest.php i App\Http\Request\FakturaUpdateRequest.php.

```
public function rules(): array
{
 return [
 'datum' => ['required', 'date'],
 'ukupno' => ['required', 'integer'],
 'kolicina' => ['required', 'integer'],
 'cena' => ['required', 'integer'],
 'kupac_id' => ['required', 'integer', 'exists:kupacs,id'],
];
}
```

U ovom ozначенom delu, zbog Blueprinta, u exists ce nam stajati Kupac, gde mi dolazimo do greske jer se nasra tabela ne zove Kupac nego se zove kupacs. Nakon ove izmene, moci cemo da u uzmemmo podatke iz tabele kupac.

9. Projekat koristi GitHub Actions za automatsko testiranje. Svaki put kada koristimo push ili pull request, github automatski istalira php i larave ldependencije, priprema nas .env fajl, pokreće migracije, pokreće Pint i pokreće sve testove.

Cela ova konfiguracija se nalazi u: .github/workflows/laravel.yml

10. Imamo Unit test koji proverava poslovnu logiku, bez baze i bez HTTP zahteva. U ovom projektu Unit se koristi za model Faktura koji proverava da li se iznos fakture ispravno racuna.

Primer poslovnog pravila: ukupno: kolicina x cena

Implementiramo je u model Faktura:

Public function izracunajUkupno()

```
{
```

```
 Return $this->kolicina x $this->cena
```

```
}
```

To tesiramo: tests/Unit/FakturaUnitTest.php

Test nam proverava da ako je kolicina 4 i cena 250, ukupno mora biti 1000.

11. Imamo Feature testove koji proveravaju celokupan sistem: rute, kontrolere, validaciju, baze podataka, prikaz view-ova, testiranje kompetnog CRUD procesa (Kupce,Fakture,Robu,Dobavljace).

Testovi se nalazi u: tests/Feature/Http/Controlles

Posto su rute zasticene autentifikacijom, testovi koriste kod kako bi simulirali ulogovanog korisnika:

```
$this->actingAs(User::factory(->create()));
```

Pokretanje testova: php artisan test

12. Ulogujemo se na GitHub, napravimo Repozitorijum, ne cekiramo Add README i Add .gitignore i kreiramo repozitorijum.

GitHub ce nam dati URL, npr:

<https://github.com/username/gradjevinska-firma-laravel.git>

Kasnije otvorimo folder svog Laravel projekta,

proveravamo da li .git postoji: ls -a ,

a ako ne postoji,inicilizujemo git: git init .

Dodajemo fajlove u git:

git add.

git commit -m "Initial Laravel project"

Povezi projekat sa GitHub-om:

git branch -M main

git remote add origin <https://github.com/username/gradjevinska-firma-laravel.git>

git push -u origin main