Extensions:

Ova prosirenja VisualStudia sam ja ubacio da lakse upravljam kodom:

* EditorConfig for VS Code
* HTML Boilerplate
* HTML Class Suggestions
* HTML Snippets
* IntelliSense for CSS class names in HTML
* Laravel 5 Snippets
* Laravel Artisan
* Laravel Blade Snippets
* Laravel Extension Pack
* PHP Debug
* PHP IntelliSense

Jos pogledat koji imaju za CSS i bootstap i tako neki koji ce pomoci

MVC

**VIEW:**

Nalaze se u App\Resources\views. Sluze nam za sve prikaze koje vidimo na nasoj stranici. To je znaci HTML i CSS koji prave nas kostur stranice i sami izgled. Kostur se moze napravit preko ove ekstenzije HTML boilerplate. Svaki view treba nazvati ime.blade.php jer radimo u laravelu (VALJDA)

**RUTE:**

Otvorit ovu rutu App\routes\web.php

Rute predstavljaju putanje do drugih stranica. U njima definiramo kamo ce nas navesti klik na link ili tako nesto

Route::get('/', function () {

return view('welcome');

});

Route::get('/projects', 'ProjectsController@index');

\*Prvom rutom s *'/'* vodimo nas na view *welcome.blade.php*

\*Drugom rutom /projects vodimo nas u metodu tj. funkciju *index* u kontoleru ProjectsController

**KONTROLERI:**

Nalaze se u App\http\controllers.

Novi kontroler pravimo u terminalu :

1. Php artisan make:controller ImeController

U kontroleru pravimo **funkcije** koje obavljaju neke radnje koje vracamo u view!

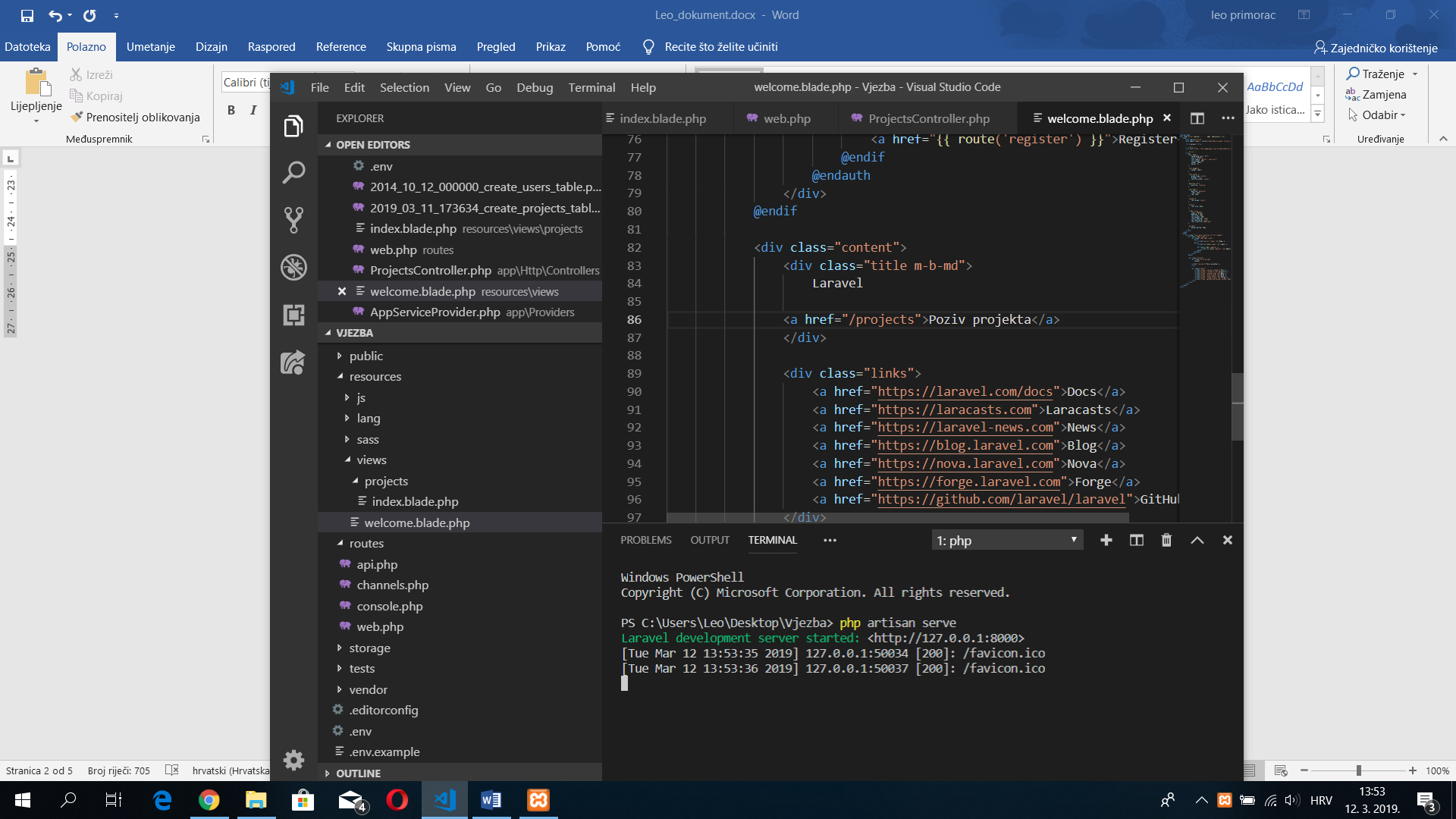
Npr funkcija:

Public function izvuciizbaze(){

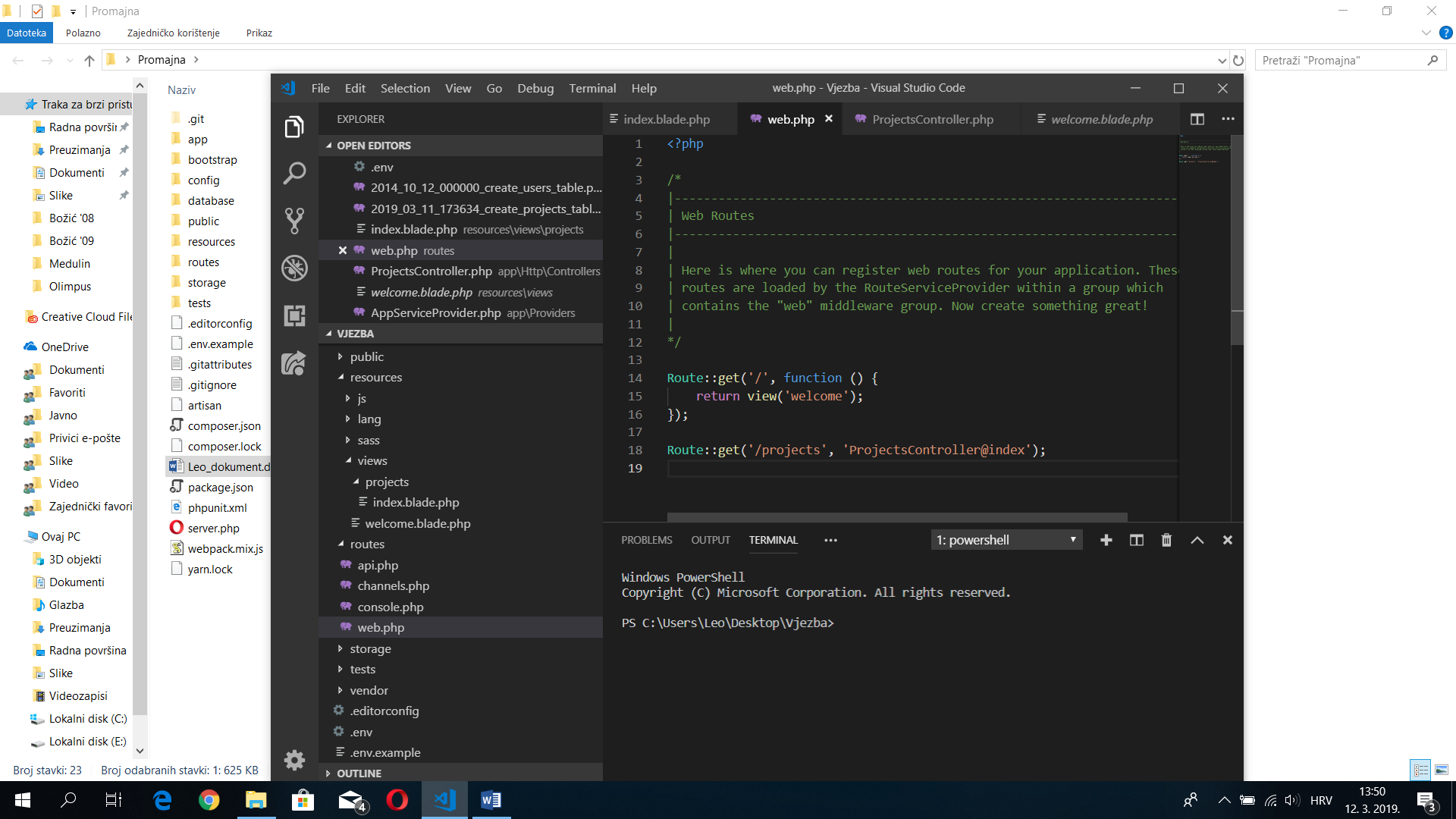
Neki kod}

**Primjer za VMC**

U viewu putem linka cemo se referencirati na routu koja ce pozbvati kontoler.



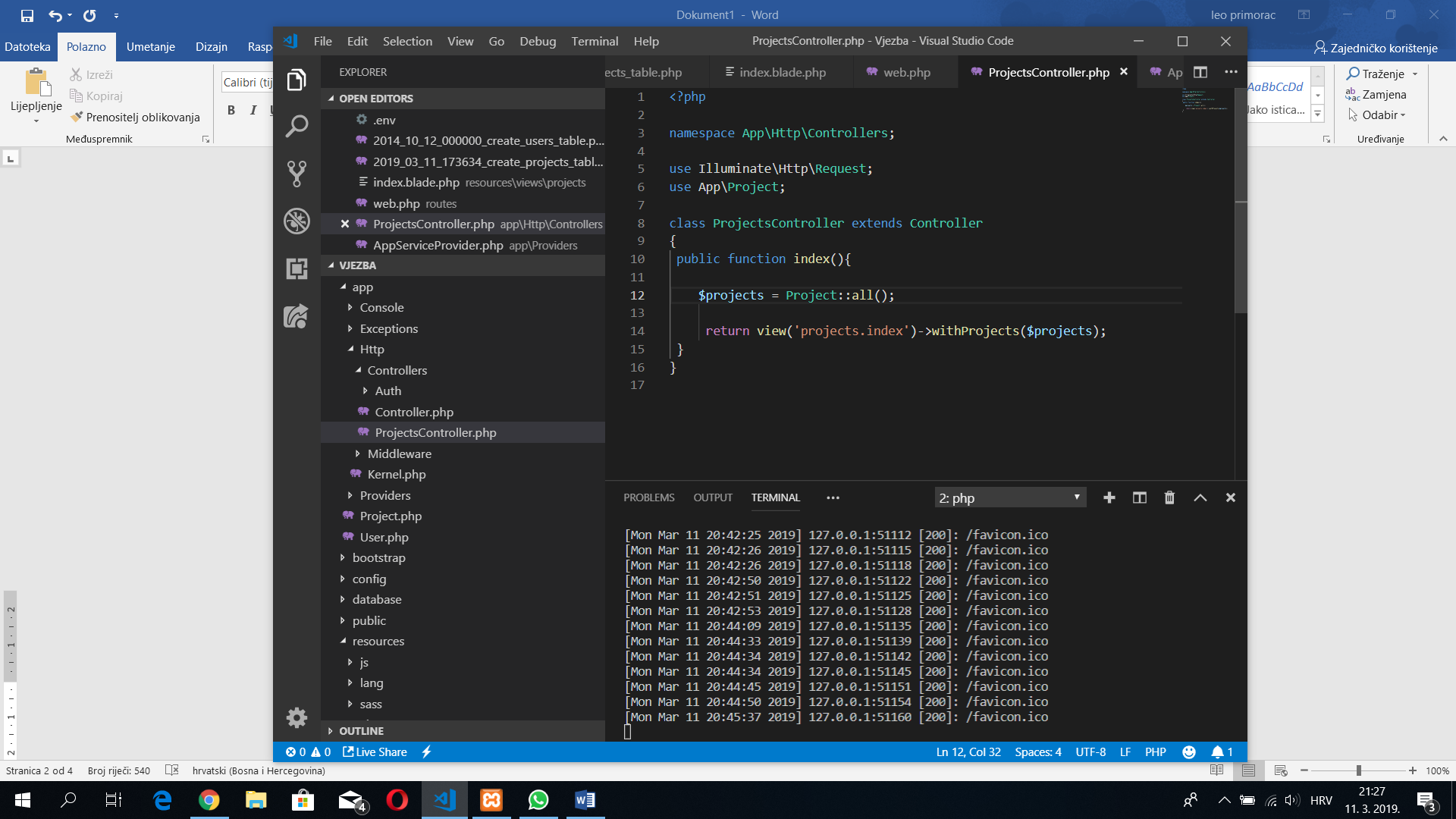
Web.php 🡪 Route



\*Prvom rutom s *'/'* vodimo nas na view *welcome.blade.php*

\*Drugom rutom /projects vodimo nas u metodu tj. funkciju *index* u kontoleru

ProjectsController.php



\*Use \App\Project 🡪 na pocetku navodimo putanju do Modela za bazu da je ne moramo svaki put dole navoditi, kako bismo kasnije jednostavno koristili *Project* kao samostalan direktorij

Klasa se sama generira prilikom pravljenja kontrolera u terminalu s *make:controller*

\*$projects 🡪 Predstavlja varijablu u koju cu pohraniti sve *all()* zapise koje dohvacam '::' iz modela *Project*

\*return view('projects.index')->withProjects($projects); 🡪 vraca '*return'* view koji se nalazi u direktoriju projects i zove se *index.* Takodjer salje parametar Projects '*withProjects'* u koji smo pohranili varijablu *$projects*

**RAD S BAZOM PODATAKA**

.env (file) 🡪 Tu podesimo konekciju na bazu

App\database\migrations 🡪 Tu se nalaze nase migracije za baze (odnosno tabele koje pravimo preko laravela), ovdje mozemo dodavati nove retke i tako dalje

Migracije:

Nacin upisa i stvaranja tablica unutar projekta te njihovo implementiranje u phpmyadmin ili nesto slicno

1. Php artisan migrate -> migracija baze na phpmysql
2. Php artisan migrate:fresh -> da bismo ubacili neke promjene na phpmysql (brise sve tablice i opet ih ucitava s tim promjenama)
3. Php artisan make:migration create\_ImeTablice\_table-> pravljenje nove tablice u bazi (nakon ovog uradit 2. naredbu)
4. Php artisan migrate:rollback ->brise zadnju migraciju

Modeli:

1. Php artisan make:model ImeZaModel -> Pravljenje modela koji predstavlja neki podatak iz neke tabliceu bazi (nalazit ce se u app direktoriju)
2. Php artisan tinker -> „kao igraliste za laravel“, neki mod za upsi podataka
3. App\ImeModela::all() -> '::' sluze nam za dohvacanje necega, 'all()' sluzi da dohvatimo sve podatke

-||- ::first() 🡪> da dohvatimo prvi

::latest()->first() 🡪 da dohatimo obrnutim redoslijedom

::all()[0] 🡪 dohvacanje prvog zapisa (ako zelimo drugi onda 1, treci 2 itd.

:: all()[0]->title 🡪 dohvacanje naslova prvog ( tako mozemo bilo koji redak u tablici

:: all()->map->title 🡪 „skenira“ ih sve s map i dohvaca njihove naslove

Unos podataka u bazu:

1. Udjemo u tinker ( korak 2. u Modelima) \* znak >>> je nas upis ,a => je ispis programa

**>>> $project = new App\Project;** -> Varijabla koja ce pohranjivat zapise koje unesemo u title i description jer sam njih stavio kao jedine elemente u bazi te one njegove automatske timestampove

*=> App\Project {#2921} 🡪 ispise ovo*

**>>> $project->title = 'MOj projekt' 🡪 u var project u title unosimo neki naziv**

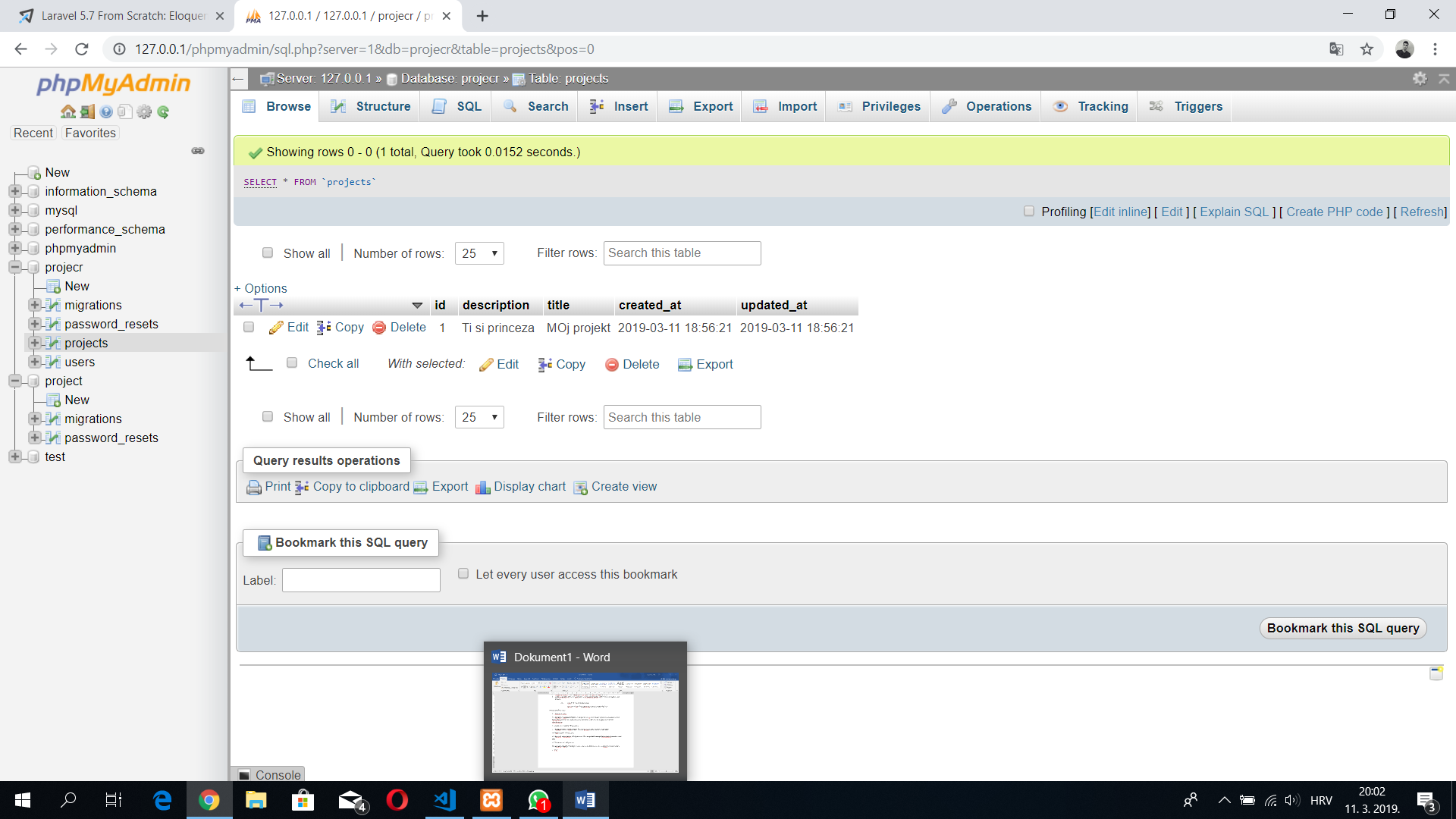
*=> "MOj projekt" 🡪 ispise ovo*

**>>> $project->description ='Ti si princeza' 🡪 u var projekt kao opis (description) unosimo neki opis**

*=> "Ti si princeza" 🡪ispise ovo*

**>>> $project->save(); 🡪** zadnje sto unesemo da pohranimo sve ovo u project varijabli u bazu

*=> true*



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dohvaćanje podataka:

* koristit cemo 3. korak iz modela: \* znak >>> je nas upis, a => je ispis programa

**>>> App\Project::all()->map->title**

*=> Illuminate\Support\Collection {#2927*

*all: [*

*"MOj projekt",*

*"Novi projekt 2",] }*

**>>> App\Project::all()[0]**

*=> App\Project {#2926*

*id: 1,*

*description: "Ti si princeza",*

*title: "MOj projekt",*

*created\_at: "2019-03-11 18:56:21",*

*updated\_at: "2019-03-11 18:56:21",*

*}*

**>>> App\Project::all()[0]->title**

*=> "MOj projekt"*