

# PRIMENA LARAVEL WEB TEHNOLOGIJA ZA PRAĆENJE POSLOVA SA PRAKTIČNIM PRIMEROM

*Nenad Milenković<sup>1</sup>, Branko Marković<sup>1</sup>*

## **REZIME**

U ovom radu prikazan je način formiranja administratorskog pristupa korišćenjem Laravel korisničkog interface-a i middleware kontrolera u okviru aplikacije za praćenje zadatih poslova koja se aktivno koristi u okviru firme Flash Point System d.o.o.

**Ključne reči:** Međusoftver, Laravel, Korisnički interfejs, Praćenje procesa proizvodnje.

## **APPLICATION OF MODERN WEB TECHNOLOGIES FOR MONITORING BUSINESS WITH A PRACTICAL EXAMPLE**

### **ABSTRAKT**

This paper shows how to create an administrative access using the Larave user interface and middleware controller within the task monitoring application that is actively used within the company Flash Point System doo

**Key words:** Middleware, Laravel,, User interface, Monitoring the production process.

### **1. UVOD**

Programi za vođenje proizvodnje, uz pomoć kojih je moguće sveobuhvatno i efektivno upravljati procesima u preduzeću veoma su rasprostranjeni kao što je na primer PANTHENON ili SAP. Zbog izuzetno visokih cena korišćenje ovakvih programa u okviru mikro i malih preduzeća predstavlja veliki izdatak. Iz navedenog razloga postoji potreba za programima koji su jednostavniji i koriste se samo za zadavanje i praćenjem poslovnih naloga, magacina u zavisnosti od potreba preduzeća.

Za preduzeće **Flash Print System (FPS)** kreirana je jednostavna aplikacija za praćenje poslovnih naloga. Kreiran je na online serveru, program koji će pratiti zadate poslove (poslovne naloge), sa četiri nivoa pristupa: administrator, rukovodilac, zaposleni i prikaz zaposlenima (monitor) u okviru proizvodnje. Online server je pod Linux operativnim sistemom Ubuntu 22.0.LTE. Za realizaciju aplikacije koristi se web tehnologija Laravel 7, a uz pomoć pretraživača korišćenjem IP adrese aplikaciji moguće je pristupiti i pratiti informacije vezane za radne naloge sa bilo kog uređaja i geografske

---

<sup>1</sup> FTN Čačak

tačke. U ovom radu biće prikazan deo aplikacije koji je vezan za korišćenje middleware-a da bi kreirali poseban nivo pristupa Admin korisniku.

## 2. KREIRANJE KORISNIČKOG INTERFEJSA

Prvo je potrebno da se podigne već kreirana virtualna mašina Homestead korišćenjem terminala:

```
~$ cd ~
```

```
~$ cd Homestead
```

a zatim uz pomoć sledećih komandi startujemo virtualnu mašinu (VM):

```
$ vagrant up
```

```
$ vagrant ssh
```

nakon čega selektujemo folder *code/laravel* koji je kreiran za smeštanje projekata. U okviru foldera komandom *new* kreira se nova aplikacija sa nazivom *masterRad* (aplikacija FPS).

```
$ laravel new masterRad
```

Nakon instalacije Laravel aplikacije potrebno je da se instalira Laravel UI paket pozivajući *composer* koristeći terminal. Prvo je potrebno da se nalazimo u folderu projekta, a zatim koristeći sledeću komandu instalira se navedeni laravel paket:

```
$ composer require laravel/ui
```

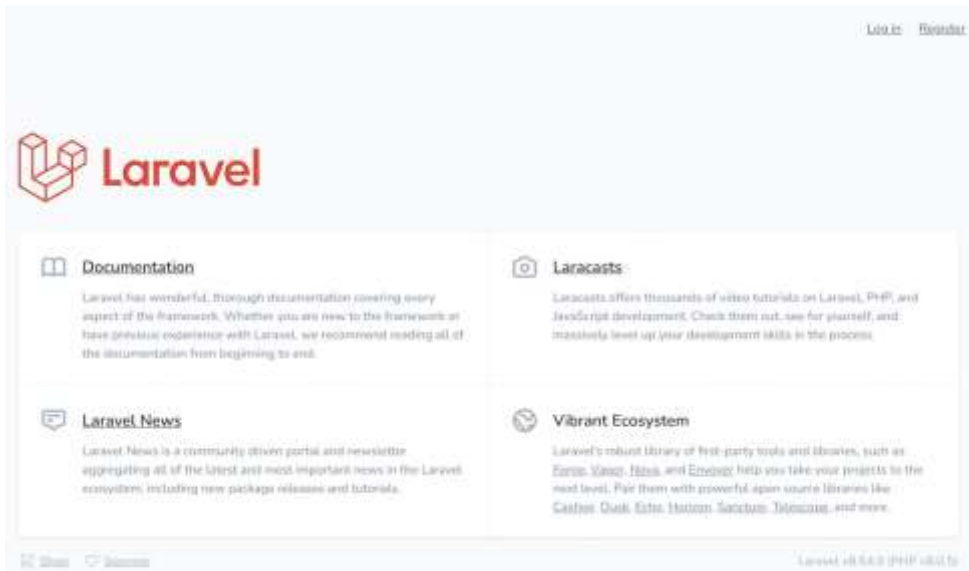
Zatim se instalira Laravel UI paket za kreiranje Auth scaffold-a koristeći Bootstrap verziju 5. pokretanjem sledeće komande:

```
$ php artisan ui bootstrap --auth
```

Instalaciju npm web paketa pokreće se upotrebom sledećih komandi:

```
$ npm install & npm run dev
```

Nakon instalacije potrebno je proveriti ispravnost aplikacije startovanjem servera *php artisan serve* i na zadatoj URL lokaciji *master.test* pojavljuje se prozor sa dobrodošlicom u Laravel-u u kome se nalaze dodatne opcije u gornjem desnom uglu za prijavljivanje i registraciju novog korisnika slika 2.1



Slika 2.1 Ekran dobrodošlice u aplikaciji

Kada se klikne na opciju *register* pojavljuje se stranica za registraciju korisnika sa unapred predefinisanim poljima slika 2.2

The image shows a registration form titled 'Register' in a light gray header. The form has four input fields: 'Name', 'E-Mail Address', 'Password', and 'Confirm Password'. Each field has a small icon on the right side: a calendar icon for Name, a mail icon for E-Mail Address, and a key icon for both Password and Confirm Password. Below the input fields is a blue button with a white user icon and the text 'Register'.

Slika 2.2 Ekran za registraciju novih korisnika

Klikom na *login* opciju pojavljuje se stranica za prijavljivanje korisnika kao na slici 2.3

Slika 2.3 Ekran za prijavljivanje korisnika

### 3. KREIRANJE PRISTUPA UZ POMOĆ MIDDLEWARE-A

Da bi se napravio pristup za administratora potrebno je da se kreira Middleware kontroler i funkcionalnosti u okviru User kontrolera i modela.

**Middleware (međusoftver)** je softver koji interaguje između aplikacije i mreže. Middleware se koristi za preusmjeravanje, a može biti korišćen kako bi omogućio samo autentifikovanim korisnicima pristup datoj ruti. Laravel dolazi s `auth middleware-om`, koji je definisan u `Illuminate\Auth\Middleware\Authenticate`. S obzirom na to da je Middleware već registrovan u HTTP jezgru, sve što je potrebno napraviti je pridružiti middleware definiciji rute:

```
Route::get('profile',function(){
    //Ulaz dozvoljen samo autenticiranim korisnicima}->middleware('auth');
```

Naravno, ukoliko se u rutama koristi kontroler što je slučaj vezan za aplikaciju, moguće je pozvati middleware metodu iz kontrolerovog konstruktora, umesto direktnog pridruživanja definiciji rute:

```
public function construct() {
    $this->middleware('auth');
}
```

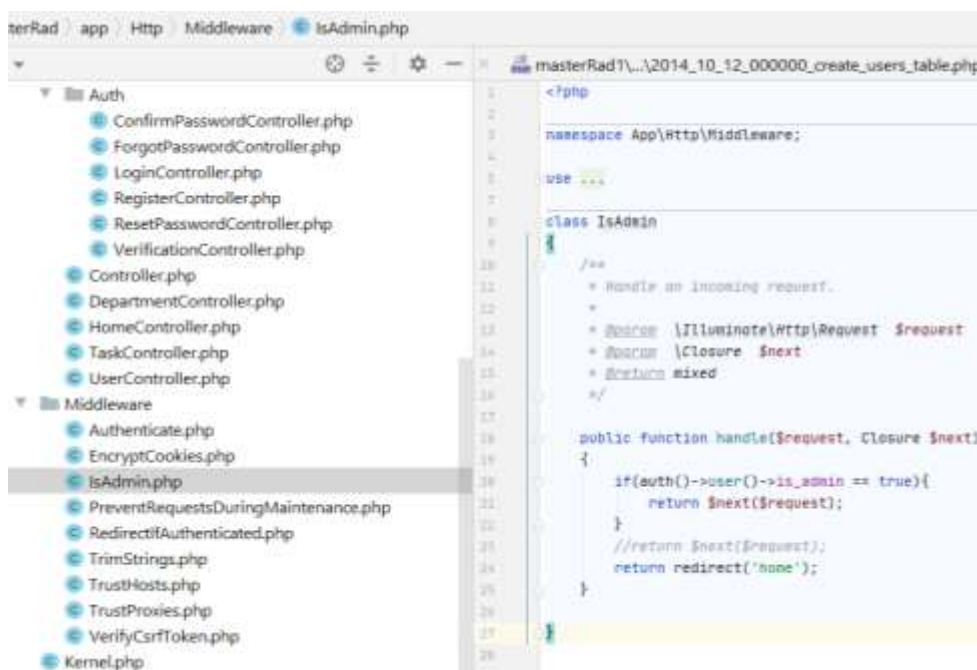
Laravel dolazi sa nekim ugrađenim međusoftverima, ali pored njih korisnik može sam definisati dodatne međusoftvere za izvođenje raznih funkcionalnosti. Ukoliko je potrebno da definišemo međusoftver, to je moguće učiniti pomoću Artisan alata. U slučaju ovog rada biće definisan `IsAdmin` middleware koji će se koristiti za kontrolu pristupa Administratora koristeći tabelu `User` (model `USER`) na sledeći način:

```
$ php artisan make:middleware IsAdmin
```

Nakon izvršenja komande u folderu `app/Http/Middleware` biće kreiran fajl `IsAdmin.php` sa klasom `IsAdmin`, kao što je prikazano na slici 3.1.

Nakon definisanja međusoftvera, treba obezbediti njegovu logiku. Logika je individualna u zavisnosti od toga kakvu ulogu će taj međusoftver imati. U ovom slučaju u okviru funkcije *handle* unet je uslov u kojem se proverava da li je korisnik administrator to jest da li je *is\_admin = true*, ako nije vraća ga na početnu *Home* straicu uz pomoć komande za redirekciju.

```
return redirect('home');
```



Slika 3.1: Kreiran međusoftver *isAdmin* u Laravel-u

Najkarakterističniji primer je međusoftver koji presreće i beleži korisniške zahteve, beleži vreme korisničkog pristupa aplikaciji, beleži IP adresu sa koje je korisnik došao i tome slično. Middleware sa kojim Laravel dolazi su:

- **EncryptCookies** - Kolačići igraju važnu ulogu tokom korisničke sesije u web aplikacijama;
- **RedirectIfAuthenticated** - Zadužen je za proveru da li je korisnik aplikacije prošao autentikaciju;
- **TrimStrings** - Njegova uloga je automatsko brisanje nehotice dodatih razmaka ili praznog prostora;
- **VerifyCsrfToken** (CSRF - Cross-Site Request Forgery) - Ima za zadatak da zaštiti stranicu od zlonamernih upada. Za svaku korisničku sesiju u aplikaciji Laravel automatski generiše token. Ovaj token se koristi kako bi korisnik dokazao da je on taj koji šalje zahteve u aplikaciji;

U fajlu *Kernel.php* koji se nalazi na lokaciji *app/Http* u promenljivoj *\$routeMiddleware* u nizu je potrebno dodati middleware rutu da bi bila vidljiva u aplikaciji.

```
protected $routeMiddleware [  
    .....,  
    'is_admin' > \App\Http\Middleware\IsAdmin::class,  
];
```

U fajlu *route/web.php* definiše se ruta vezana za administratorski pristup aplikaciji. Koristeći klasu *Auth::routes()*; definiše se da je obavezno prijavljivanje na aplikaciju. U sledećim linijama koda kreira se middleware grupa u kojoj se kroz putanju proverava da li je *is\_admin = true*. U slučaju pozitivnog ishoda preusmerava se na stranu *dashboard*.

```
...  
// ADMIN ROUTES  
Route::middleware('is_admin')->prefix('admin')->group(function(){  
    // Dashboard  
    Route::get('/', 'App\Http\Controllers\HomeController@dashboard')->  
        name('admin.dashboard');
```

U listing 3.1 prikazan je kontroler login strane aplikacije koji se nalazi na lokaciji *app/Http/Controllers/Auth*. Ovaj kontroler sadrži klasu *loginController* koja nasleđuje klasu *Controller* u kojoj se nalazi funkcija *login(Request \$request)* i u kojoj je unet uslov preko koga se proverava da li je u pitanju administratorski nalog ili neki drugi.

```
<?php  
namespace App\Http\Controllers\Auth;  
use App\Http\Controllers\Controller;  
use App\Providers\RouteServiceProvider;  
use Illuminate\Foundation\Auth\AuthenticatesUsers;  
use Illuminate\Http\Request;  
  
class LoginController extends Controller  
{  
    use AuthenticatesUsers;  
    protected $redirectTo = RouteServiceProvider::HOME;  
    public function __construct()  
    {  
        $this->middleware('guest')->except('logout');  
    }  
    public function login(Request $request)
```

```

{
    $input $request->all();

    $this->validate($request, [
        'email' > 'required/email',
        'password' > 'required',
        // if status ! true return message
    ]);
    if(auth()->attempt(array('email' > $input['email'], 'password' >
$input['password'])))
    {
        if(auth()->user()->is_admin 1) {
            return redirect()->route('admin.dashboard');
        }else{
            return redirect()->route('home');
        }
    }else{
        return redirect()->route('login')
            ->with('error', 'Email-Address And Password Are Wrong. ');
    }
}
}
}

```

Listing 3.1 *loginController.php*

#### 4. ZAKLJUČAK

U ovom radu prikazana je upotreba međusoftvera (middleware) u okviru Laravel korisničkog interfejsa koja omogućava autentifikacija administratoru aplikacije. Kroz rad je prikazana instalacija nove aplikacije kao i instalacija korisničkog interfejsa u okviru Laravel-a, a zatim integracija i podešavanje međusoftver kontrolera preko koga se usmerava *http* ruta u zavisnosti od nivoa pristupa. Ovakav metod kreiranja administratorskog nivoa u aplikaciji omogućava da u okviru iste tabele (tabele *User*) bilo kom korisniku dodeli mogućnost da poseduje administratorski nivo. Navedeni sistem pristupa realizovan je u okviru aplikacije koja se aktivno koristi u okviru štamparije Flash Point System d.o.o.

## **5. LITERATURA**

[1] Nenad Milenković, Primena savremenih web tehnologija za praćenje poslova sa praktičnim primerom, Master Rad, 2022.

[2] Laracast, <https://laracasts.com/> , 19.09.2022.год

[3] Laravel authentication with Laravel UI, <https://laravel.com/docs/8.x/authentication>, 19.09.2022.год

[4] Laravel middleware, <https://laravel.com/docs/8.x/middleware> , 20.09.2022.год