Tallinna Ülikool

Digitehnoloogiate Instituut

Laravel veebiraamistiku juures kasutatavate kolmandate osapoolte administreerimispaneelide võrdlus

Seminaritöö

Autor: Kristo Jürgenson

Juhendaja: Jaagup Kippar

Autor:.....”.....”.....2017

Juhendaja:.....”.....”.....2017

Tallinn 2017

Autorideklaratsioon

Deklareerin, et käesolev seminaritöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

…… ……..(kuupäev)(autor)

Sisukord

Sissejuhatus

Käesoleva seminaritöö eesmärk on võrrelda Laravel veebiraamistiku kolmandate osapoolte administreerimispaneele ning ülesseadistada arenduskeskkond CRM süsteemi loomiseks. Kuna autori bakalaureusetöö on CRM süsteemi loomine autotöökojale, siis tuleb teha eelseadistused ning testid, milliseid tööriistu ja keskkondi oleks kõige lihtsam ja parem kasutada. Seminaritöös annab autor komponentidest ülevaate ning teeb seadistused, et arenduskeskkond püstitada.

Käesolev töö on jagatud kolmeks peatükiks. Esimene peatükk tutvustab kasutusele võetavat veebiraamistikku ja veebimajutuse keskkonda ning annab põhjendused, miks antud komponendid valiti.

Teises peatükis kirjeldatakse töötava veebiserveri ülesseadistamise protsessi. Autor tutvustab, mis komponente on vaja, et Ubuntu operatsiooni süsteemis veebiserver luua. Ta lisab komponentide installeerimise ja ülesseadistamise kirjelduse ning toob välja võimalikud veakohad, mis võivad tekkida. Lisaks sellele Laravel veebiraamistiku ülesseadistamine antud keskkonda.

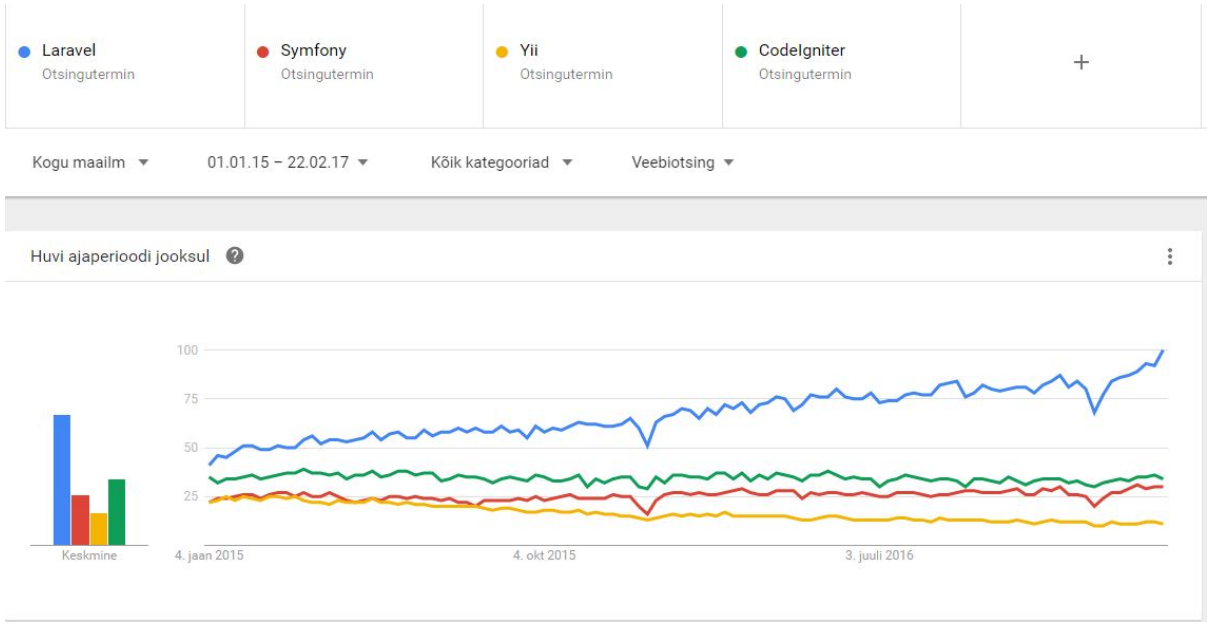
Kolmandas peatükis on Laraveli kolmandate osapoolte administreerimispaneelide installeerimine ning tutvustamine. Antud pakkide võimaluste esile toomine ning nende vahel valida kõige sobivam arenduskeskkonna jaoks.

1. Tutvustused

Siinses peatükis annab autor kiire ülevaate Amazon AWS teenusest ja Laravel veebiraamistikust ning mis eelised on neil teiste sarnaste komponentide ees.

1.1. Laravel

Laravel on avatud lähtekoodiga PHP veebiraamistik, mida on lihtne kasutada ning ülesseadistada. Alates 2015. aastast on Laravel enim kasutatud PHP veebiraamistik (Skvorc, B. 2015) ning Google Trendsi andmetel on selle populaarsus veelgi enam kasvanud (Joonis 1). Töö kirjutamise ajal on kõige uuem versioon 5.3, mis tuli välja 2016 aasta Augustis (Alates 2017. aasta jaanuarist on väljas versioon 5.4).



Joonis 1. Google trends päringute statistika PHP veebiraamistike kohta

Raamistiku loomisel on silmas peetud, et rakenduste arendamine peaks olema lihtne ja nauditav kogemus (Otwell, T. Kuupäev puudub). Sellest tuleneb ka Laraveli populaarus. Lihtsus seisneb selles, et Veebiraamistiku süntaks on kergesti mõistetav ja elegantne. Lisaks on raamistikus levinud ülesannete jaoks olemas tööriistad, mis kiirendavad arendusprotsessi. Nende tööriistade hulka kuuluvad näiteks autentimine, modulaarsus, ruutimine ja migratsioonid.

Lisaks sellele on Laraveliga kaasas Artisan käsurea tööriist, mis pakub palju abistavaid funktsioone rakenduse arendamisel. Võimalik on luua korduvalt jooksvaid skripte või lihtsalt ja kiirelt luua mudeleid ning vaateid. Sellele on omakorda võimalik veel erinevaid käske juurde teha, mis rahuldaksid teatud kasutaja vajadusi.

Eloquent ORM (Object relational mapper) Laravelis implementeerib active record disainimustrit andmebaasiga töötlemiseks. Kõikidel läbi migratsioonide loodud tabelitel on projektis olemas vastav mudel, PHP klassid, mis on omavahel seotud. See võimaldab lihtsat ja kiiret suhtlemist andmebaasi tabelitega ning koodi kirjutamise vajalikkus on minimaalne. Mudel võimaldab olemasolevate käskudega andmete muutmist, salvestamist, lisamist ning luua ühendusi teiste tabelitega.

Lisaks sellele järgib laravel veebiraamistik MVC disaini mustrit. MVC disainimuster ehk Mudel-Vaade-Kontroller on veebiarenduses väga tihti kasutatav disainimuster, kus äriloogika on eraldatud kasutajaliidesest. Rakendus on jagatud kolme komponenti:

- Mudel – täidab andmestruktruuri rolli. Komponent, mis salvestatakse andmebaasi ja defineerib selle ehituse.
- Vaade – kuvab kasutajale visuaali mudelist.
- Kontroller – läbi vaadete uuendab mudeli andmeid. Kasutaja saab vaates muuta mudelit tänu kontrollerile.

Võrreldes teiste PHP veebiraamistikega on Laraveli kogukond suurem ja aktiivsem. Lisasid, mida on võimalik juurde installeerida ning kasutada, on rohkem. Dokumentatsioon on kogukas ja õppematerjale ning sellega seonduvaid lehti on mitu. Raamistikuga alustamine on suhteliselt lihtne. Raskete probleemide lahendamine on kergem, sest on võimalik, et kellelgi on see probleem juba esinenud. Samuti on läbi kogukonna lihtne saada vastuseid oma küsimustele.

1.2. Amazon AWS

Amazon AWS (edaspidi AWS) on turvaline pilveteenuse platvorm, mis pakub andmebaasiruumi, infosisu edastamist ja palju teisi teenuseid, mis aitavad ettevõtetel kasvada ja kesta. AWSil on palju erinevaid funktsioone ja võimalusi, mida on võimalik kiirelt

ülesseadistada ning jooksvalt muuta. Näiteks kui esmalt ülesseatud keskkonnal on liiklus liiga tihe, siis on võimalik jõudlust suurendada läbi kontrollpaneeli ning muutused on kohesed. Pakutavad teenused on loodud koos töötama, et luua keerulisi ja skaleeritavaid rakenduseid. AWS pakub lahendusi igale kasutaja vajadusele.

AWS keskkonda, nagu teisi pilveteenuse keskkondi, saab hallata läbi veebibrauseri kontrollpaneeli. Sealt on võimalik valida, milliseid teenuseid kasutaja soovib ning paari hiire vajutusega need teenused käivitada. Käesolevas töös on kasutusel kaks teenust: EC2 ja RDS.

EC2 (Elastic Compute Cloud) pakub skaleeritavat andmetöötlust pilves (Amazon AWS, kuupäev puudub) ning võimaldab ülesseadistada veebimajutuse teenust, mille peale on võimalik omakorda ülesseadistada veebiserver. Ühele EC2 keskkonnale on võimalik jooksutada mitu arenduskeskkonda. Läbi veebis asuva kontrollpaneeli saab määrata, kui palju ruumi läheb vaja ja milline peaks olema jõudlus. Hind sõltub nendest parameetritest ehk mida vähem jõudlust on vaja, seda vähem peab maksma teenuste eest. EC2 on hästi integreeritud teiste teenustega ning väga paindlik kasutaja nõuete suhtes. Alustamine on läbi AWS keskkonna väga lihtne ning hostimise masinat saab kiirelt luua läbi AWS käsureatööriista, mis loob ka turvaelemendid antud instantsi jaoks.

RDS ehk Relational Database Service on teenus, mis teeb lihtsaks relatsiooniliste andmebaaside ülesseadistamise pilvekeskkonda (Amazon AWS, kuupäev puudub). Seda on lihtne hallata läbi AWS juhtimise konsooli ning kõik tehtud muudatused on kohesed ilma seisakuajata. See on turvaline ning vastupidav ehk RDS loob mitu instantsi andmebaasist, et poleks andmekadu ja koormus ei oleks liiga suur.

1.2.1. Võrdlus teiste pilveteenuste pakkujatega

Teenuste ja võimaluste rohkuse tõttu on AWS pigem mõeldud suuremate ettevõtete jaoks. Samuti on võimalus parandada ettevõttes olemasolevate komponentide jõudlust. Antud olukorras ei ole see kõige parem variant, sest AWS pole kõige odavam ega ka kindlam. Näiteks GCE keskkond on sarnaste võimalustega, aga odavam. AWSil on vigu raske välja selgitada. Kui mingi komponent lõpetab töötamise, siis põhjust ei saa tihtipeale teada, sest

suhtlemine toimub läbi klienditeeninduse, kes ei anna teada, kui mingi masin on neil kokku jooksnud.

Kuna on vaja ainult veebiserver kuskile ülesseadistada, siis piisaks ka näiteks DigitalOceani dropletist. Selles keskkonnas on võimalik veebimajutamiseks pilves olevat masinat kasutada ning hind on tunduvalt odavam võrreldes AWSiga. Lisaks on DigitalOceanil kergem eksemplare luua ning üles seadistada. Amazon AWS aga pakub aasta aega tasuta kasutamist ning see on ühtlasi ka põhjus, miks see valiti. Põhiteenuseid saab kasutada ning see annab võimaluse aasta aega testida keskkonda ning arendatava rakenduse vead ja võimalused välja uurida. Lisaks on võimalus salvestatud andmed pärast aastat aega sealt kergelt kätte saada.

2. Serveri ülesseadistamine

Laravel keskkonna seab autor üles Amazon AWS pilvemasinale, mille operatsioonisüsteemiks on Ubuntu 14.04. Antud operatsioonisüsteem on üles seatud nii minimaalsete lisadega kui võimalik. Seda sellepärast, et kõik juurde installeeritavad elemendid oleksid õigetes versioonides ja ebavajalikud rakendused ei kasutaks ära kettaruumi.

Et teha sellest veebiserver, kuhu oleks võimalik ülesseadistada Laravel veebiraamistik, on vaja LAMP komponente. LAMP on avatud lähtekoodiga veebiarenduse platvorm. LAMP nimetus tuleneb sinna kuuluvatest komponentidest, milleks on Linux, Apache, MySql ning PHP. Nimetatud komponendid on vajalikud, et jooksutada veebiserverit. Linux komponent on juba olemas, kuna see tuli kaasa Amazon AWS keskkonna loomisega. Võimalik on installeerida LAMP komponente ka ühe käsu abil. Kuna käesolevas on vaja kindlaid versioone, siis oleks kasulikum kõik eraldi alla laadida.

Et veebiserveri ülesseadistamisega alustada, tuleks esmalt ligi pääseda pilves olevale masinale. Erinevates operatsioonisüsteemides on mitmeid võimalusi ühenduse loomiseks, aga kuna autor kasutab ka Ubuntut, siis sobib terminalis SSH ühenduse loomine. Amazon EC2 masina instantsi loomisega tuli kaasa.pem fail, mis on turvavõti ühenduse loomiseks. Terminalis ühenduse loomiseks tuleks anda vastav käsk:

```
ssh -i {failinimi}.pem ubuntu@{masinaip}
```

Laravel keskkonna ülesseadistamiseks oleks vaja installeerida vastavad komponendid õigete versioonidega:

- PHP >= 5.6.*
- PDO PHP Extension
- Mbstring PHP Extension

- Tokenizer PHP Extension
- XML PHP Extension

2.1. PHP

PHP ehk Hypertext Preprocessor on laialt kasutuses olev avatud lähtekoodiga skriptimise keel, mida enamjaolt kasutatakse veebilehtede arendamiseks (The PHP group, kuupäev puudub). See on mõeldud töötama serveripoolseks skriptimiseks ning on hea tööriist loomaks dünaamilisi ja interaktiivseid veebilehti. Üle 80% internetis olevatest veebilehtedest kasutavad PHPd serveripoolse keelena (W3techs, kuupäev puudub).

Laravel 3.* jaoks sobib PHP versioon 5.6.* ning suurem. PHPd on võimalik kätte saada ametlikust repositooriumist, aga kuna kolmandate osapoolte repositooriumeid uuendatakse tihedamini, siis kasulikum oleks nendest installeerida. Selleks, et PHP saada kolmanda osapoole hoidlast, tuleks esmalt lisada viide hoidlale. Selleks tuleks terminali aknas anda käsk:

```
add-apt-repository ppa:ondrej/php
```

Et operatsiooni süsteem arvestaks selle uue repositooriumi viitega tuleks teha viidete uuendus käsuga:

```
apt-get update
```

Nüüd saab PHP5.6 kätte vastava käsuga:

```
apt-get install -y php5.6 php5.6-mcrypt php5.6-gd
```

Kui kõik õnnestub, peaks PHP 5.6 masinas olema. Kuna Laravel vajab veel PHP lisasid, mis võimaldaksid failidega töötlemist, on vaja jooksutada veel 2 käsku:

```
apt-get install php5.6-xml
apt-get install php5.6-mbstring
```

Et PHP saaks faile lugeda ja töödelda, on vaja neid lisasid. Vahepeal ei tule lisad ametliku PHP installi käigus ning selleks tuleb need eraldi installeerida.

2.2. Apache

Kui PHP on installeeritud, on vaja veebiserverit, mis suudaks kuvada projektis olevaid veebilehti ja võtaks vastu päringuid. Unix masinatel on kõige kasutatuim Apache2 veebiserver ning just selle autor paigaldabki.

Apache veebiserver on tarkvaraarenduse koostöö projekt, mille sihtmärk on luua jõuline, funktsiooniderohke avatud lähtekoodiga HTTP veebiserver (HTTP Apache project, kuupäev puudub). Projekti arendavad ja haldavad vabatahtlikud üle maailma.
Apache veebiserveri esimene ametlik versioon tuli välja 1995. aasta aprillis. Veebiserver oli algusest peale väga populaarne ja edukas ning on kasutuses enamus UNIX süsteemides ja seda ka praegu (Netcraft, 2017). Kogukonna poolt ühiselt hallatavaks projektiks muutus see 1999. aastal.

Apache installeerimine on suhteliselt lihtne protsess. Piisab ainult ühest käsust, mis tõmbab repositooriumist tarkvara alla ning peale installeerimist koheselt ka käivitub. Lisaks sellele tuleb käsus viidata, millist PHPd Apache kasutama hakkab.

```
apt-get install apache2 libapache2-mod-php5
```

Kui kõik õnnestub ja terminal näitab, et Apache käivitati, siis seda on võimalik kontrollida läbi veebibrauseri märkides leheks masina IP aadress.

2.3. MySql

Korralikult töötavast veebiserverist on veel puudu andmebaasi haldussüsteem. Valida on mitme haldussüsteemi vahel, mis töötaksid koos Laravel veebiraamistikuga. Näiteks SQLite, PostgreSQL ning MySql. Praeguses olukorras valib autor MySqli, sest tal on sellega kõige rohkem kogemusi.

MySql on üks populaarsematest avatud lähtekoodiga SQL andmebaasi haldussüsteemidest (Oracle, What is MySql?, kuupäev puudub). MySql põhineb relatsioonilisel andmebaasi mudelil, kus andmed on kuvatud eraldi tabelitena. Need tabelid on omavahel seotud määratud reeglite järgi, mis võimaldavad andmebaasis kiiret töötlemist ning andmete täielikkust. Esimest korda avaldati see 1995. aastal ning on kuni tänaseni laialdaselt kasutuses.

Kuna andmete salvestamiseks on olemas eraldi server, ei ole vaja kasutada veebiserveri masinat ning piisab ainult komponendi installeerimisest. Sellest tulenevalt ei pea ka alguses väga skaleeritavuse peale mõtlema. Et saada MySql, mis ühilduks installeeritud PHP versiooniga, tuleb terminalis jooksutada käsk:

```
apt-get install mysql-server php5.6-mysqsl
```

Sellega tulevad kaasa vajalikud draiverid (PDO), et oleks võimalik suhelda SQL keskkonnaga.

2.4. Laravel projekti ülesseadistamine

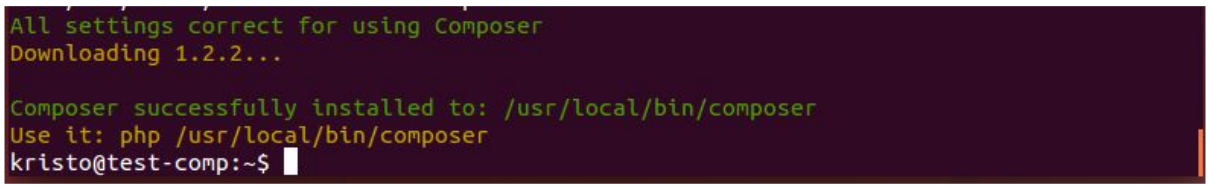
Kui töötav veebiserver on olemas, siis on võimalik ülesseadistada Laravel veebiraamistiku projekt. Selleks on kaks võimalust.

Üks variant oleks otse ametlikust Git repositooriumist tõmmata projekti kaust. Sellega saab peaaegu kõik vajaliku, vaid mõned konfiguratsioonifailid peab ise looma.

Teine variant oleks kasutada Composer tööriista ning läbi selle luua uus projekt. Composer tööriista on lihtsam kasutada, sest kolmandate osapoolte pakkide tõmbamisel paigutatakse need kohe õigetesse kohtadesse ning jääb maha viide, mis pakid on projektis olemas. Et seda kasutada tuleb see esmalt installeerida ning teha sellest globaalne käsk üle terve süsteemi. Selleks tuleks terminalis käivitada käsk.

```
curl -sS https://getcomposer.org/installer | php --
--install-dir=/usr/local/bin --filename=composer
```

Õnnestumise korral saab eduka teate (Joonis 2) ning kirjutades terminali composer kuvatakse käske, mida saab kasutada.

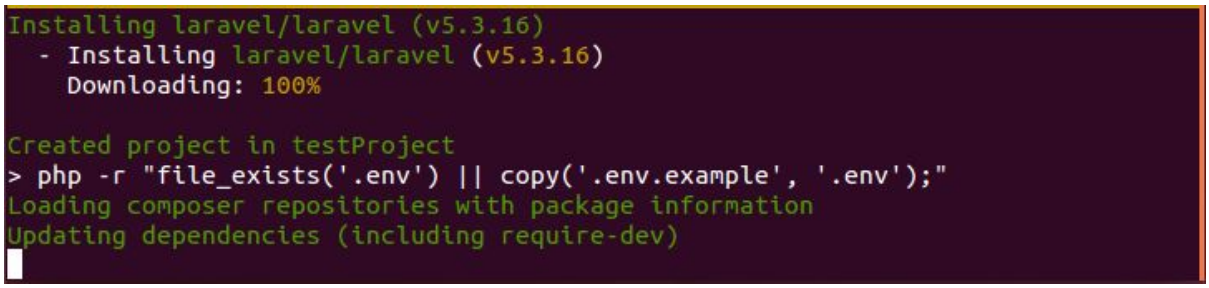


Joonis 2. Composeri edukas install

Kui Composer on olemas ja töötab, siis on võimalik läbi selle luua uus Laravel projekt. Selleks tuleks liikuda Apache projektide kausta, mis asub /var/www/html ning seal käivitada käsk:

```
composer create laravel/laravel {projektinimi} --prefer-dist
```

Hetkel luuakse projekt kõige uuema Laraveli versiooniga. Viimase parameetriga saab määrata, millist versiooni kasutada. Kui käsku käivitada, siis hakkab Composer installeerima kõik vajalikud sõltuvused ning õnnestumise korral annab teate, et projekt loodi ning .env.example failist tehti koopia .env (Joonis 3). Kui .env faili ei loodud, siis tuleks teha koopia näidisfailist ning täita parameetrid õigete andmetega. Eelkõige andmebaasi server, selle kasutaja ning parool.

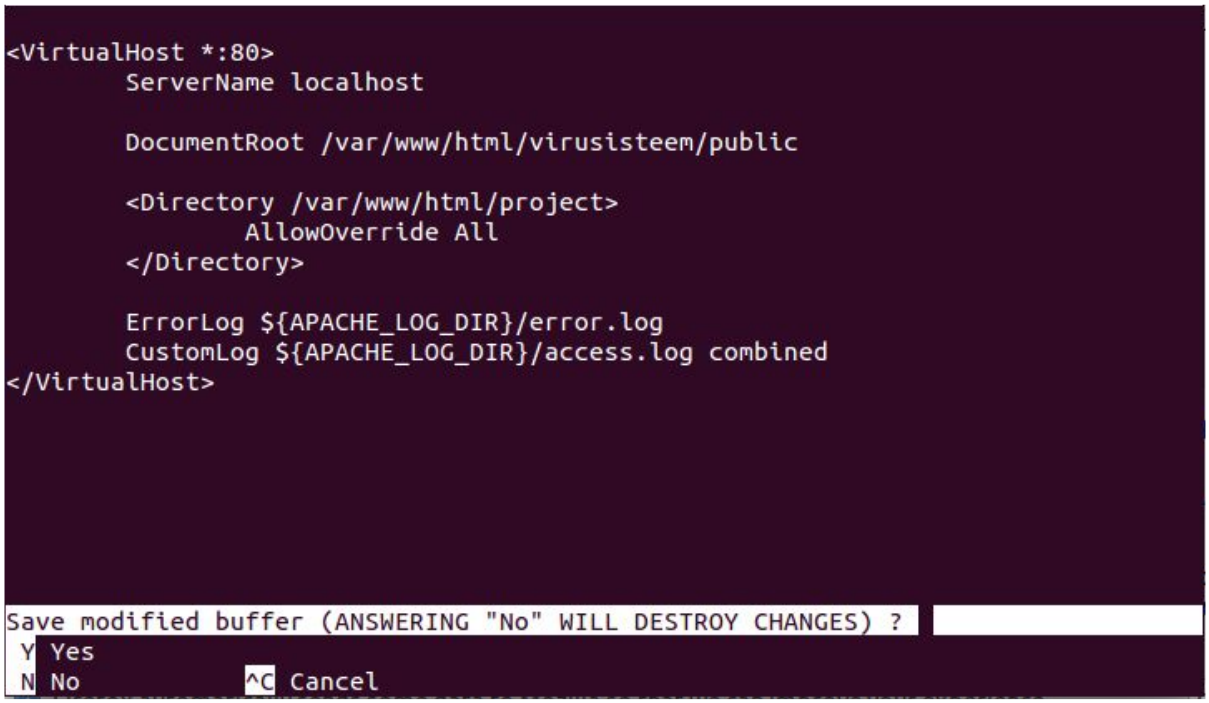


Joonis 3. Laravel projekt loodud

Et veebibrauser saaks kasutada projektis olevaid PHP faile tuleb projektile anda vastavad õigused.

```
chmod -R 777 {projektinimi}
```

Et veebibrauserites projekt ilmuks, tuleb Apaches luua konfiguratsioon, mis viitaks Laraveli projektile. Selleks tuleks olemasolevate lehtede kausta luua uus konfiguratsioon ning see jooksma panna ja vana ära keelata. Lehtede konfiguratsioonifailid asuvad /etc/apache2/sites-availabe. Sinna tuleks luua uus fail .conf laiendiga, mille sisu viitaks index.php failile Laravel projekti publik kaustas (Joonis 4).



Joonis 4. Apache konfiguratsioon Laravel projekti jaoks

Käsud konfiguratsiooni muutmiseks Apaches:

```
sudo a2dissite 000-default.conf
sudo a2ensite laravel.conf
sudo a2enmod rewrite
sudo service apache2 restart
```

Kui toimingud olid edukad, siis veebibrauser peaks kuvama Laravel projekti index.php faili (Joonis 5).

Joonis 5. Laravel index.php

3. Laravel administreerimise paneelid

Siinses peatükis autor installeerib ja seadistab üles Laravel keskkonnale mõeldud kolmandate osapoolte administreerimispaneelid ning testib neid. Testimise protsess koosneb ülesseadistamisest, lisaelementide loomisest ning lõppkasutaja kogemusest. Selle tulemusena teeb autor pakkidest hinnangulise ülevaate, mille lõpus valib parima paki ning kasutab seda alusena CRM rakenduse loomiseks.

Testimise eesmärk on leida pakk, mis võimaldaks arendatava rakenduse esmane töötav variant võimalikult kiirelt avalikuks teha. Selle jaoks oleks vaja leida komponendid, mis võimaldaksid uute moodulite loomist ja nende alla kuuluvate andmete kiiret töötlemist. Samuti on vajalikud komponendid, mis võimalikult väikese ajaga saavad tööle rakenduse, mis parandaks autotöökoja töötajatel nende tööprotsessi.

Laraveli kogukond on üsnagi suur ja aktiivne ning lisaarendusi, mida projekti lisada, on väga palju. Tänu Composerile on kolmandatel osapooltel pakkide jagamine väga lihtsaks tehtud. Veebilehti, kust lisapakke leida, on mitu. Näiteks on olemas Packalyst (http://packalyst.com/), mis on spetsiaalselt mõeldud Laraveli pakkide haldamiseks. Lisaks sellele on olemas Packagist (https://packagist.org/), mis on üldiselt mõeldud kõikide composer pakkide jaoks, aga seal leidub ka palju Laraveli eksklusiivseid pakke. Mõned pakid võivad olla avalikud vaid git repositooriumites.

Antud keskkondades filtreeritakse pakid märksõnade CRM, Admin-Panel, CSM, Admin ning CRUD järgi. Tulemuste põhjal valib autor need pakid, millel on kõige rohkem allalaadimisi ning kõige paremad hinded. Lisaks sellele peab see ühilduma Laravel 5 ja kõrgemate versioonidega.

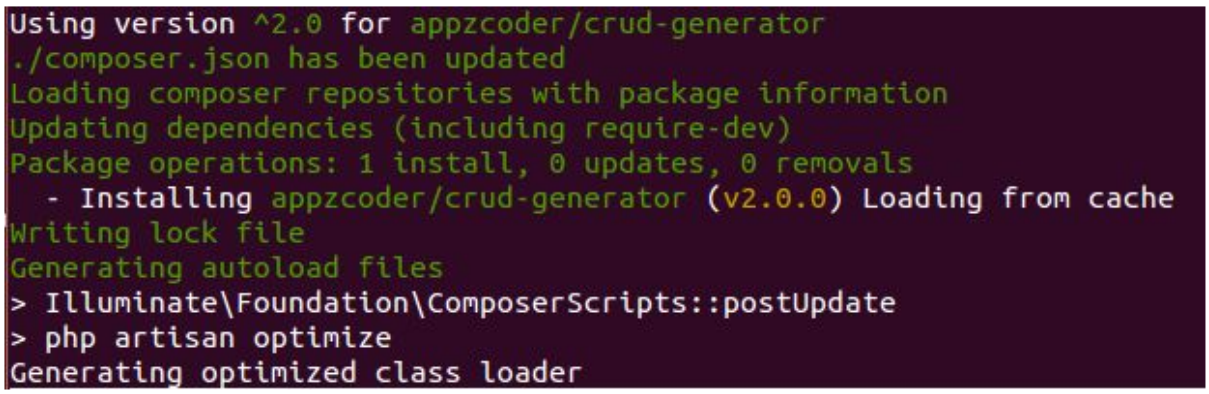
3.1. CrudGenerator

CrudGenerator asub https://github.com/appzcoder/crud-generator.

Et CrudGenerator projekti lisada, tuleb esmalt läbi Composeri lisada sõltuvus. Seda saab teha käsuga:

```
composer require appzcoder/crud-generator
```

Õnnestumise korral lisatakse projektis asuvas composer.json faili uus sõltuvus, mis viitab CrudGeneratori repositooriumisse ning koos sellega installeerib paki projekti vendor kausta. Õnnestumise korral tuleb teade, et pakk on edukalt lisatud (Joonis 6).



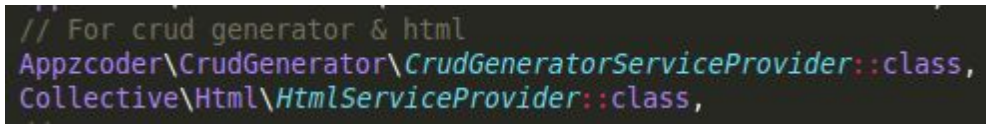
Joonis 6. Edukalt pakk lisatud

Et CrudGenerator korralikult töötaks on lisaks vaja läbi installerida LaravelCollective pakk. Seda lisada käsuga:

```
composer require laravelcollective/html
```

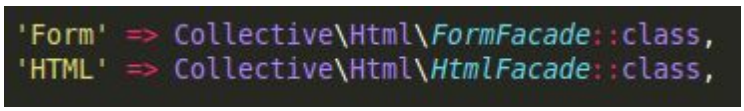
Õnnestumise korral kuvab terminal teate, et pakid on installeeritud.

Kui sõltuvused on lisatud peab projekti konfiguratsiooni failides ära määrama klassid ja aliased, mis tulevad CrudGeneratori ja LaravelCollective installimise käigus. Seda põhjusel, et kui need failid tekivad, siis on nad projektis leitavad ja kasutatavad. Selleks tuleb minna loodud projekti kausta ja sealt valida "/config" kaust. "Config" kaustas on fail nimega "app.php", kus on olemas kõik selle projektiga seotud konfiguratsioonid. Seal tuleb lisada kaks rida "providers" konteinerisse, mis projekti siseselt võimaldab kasutada vastavaid klasse (Joonis 7).



Joonis 7. Providers konteinserisse lisatud read

Kaks rida aliase , mis tuleb lisada “alias” konteinerisse (Joonis 8).

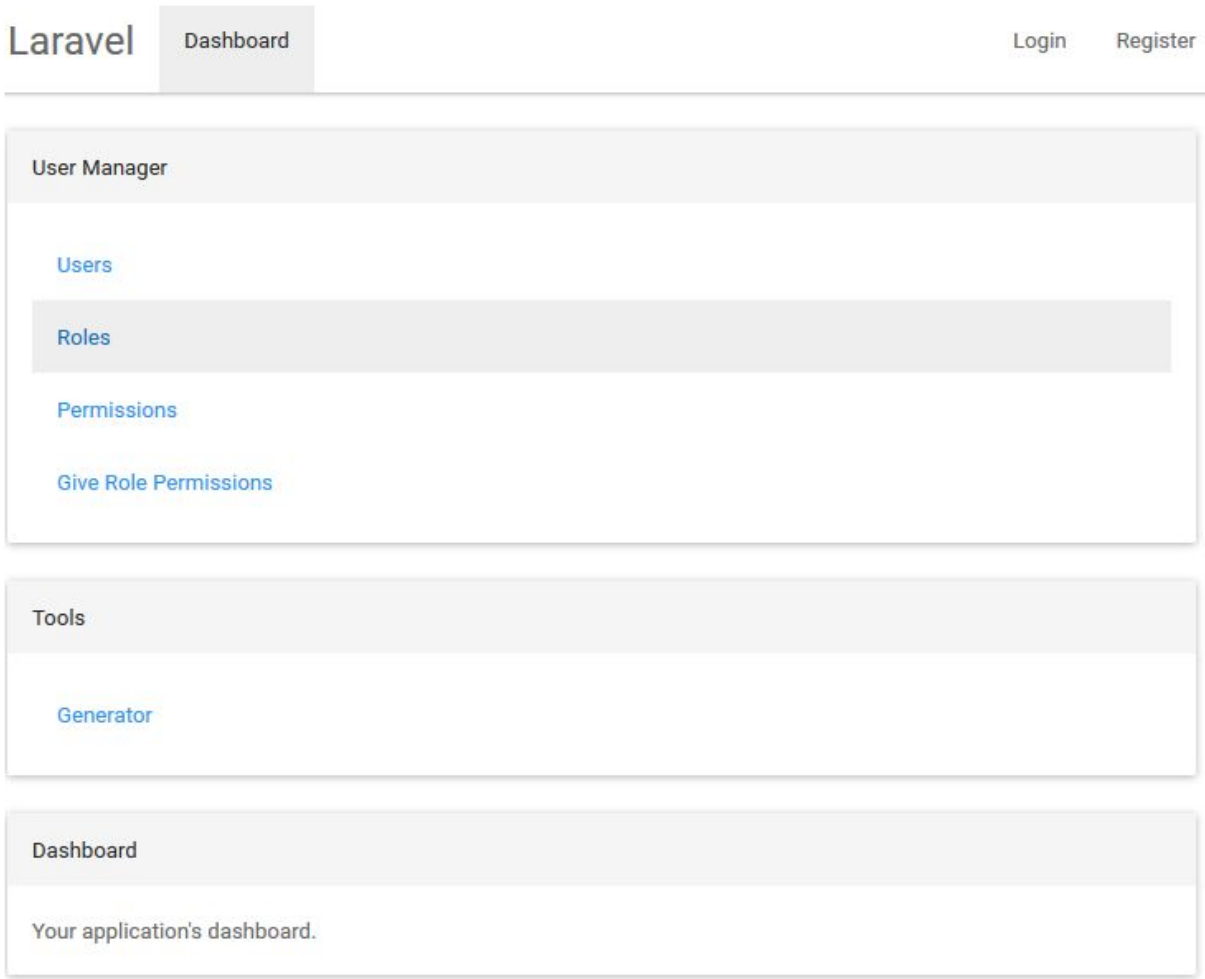


Joonis 8. Alias konteinerisse lisatud read

Viimaks tuleks jooksutada käsk, mis loob uued või muudab olemasolevate konfiguratsioonide parameetreid, et CrudGeneratori tööriistad ja vaated oleksid kättesaadavad ning kasutatavad.

```
composer dump-autoload
php artisan vendor:publish
--provider="Appzcoder\CrudGenerator\CrudGeneratorServiceProvid
er"
```

Kui sõltuvused on installeeritud ja kõik on õnnestunud, siis peaks administreerimise paneeli nägema veebibrauseris “domeen/admin” lehel (Joonis 9).



Joonis 9. CrudGenerator admin leht

CrudGenerator pakub eraldatud administreerimis- ja kasutaja osa. Admini osa pole kohe alguses auntentimisega eraldatud, mis annab võimaluse luua eraldi sissepääsu administraatorile. Suur rõhk on asetatud CRUD tööriistale, mida on võimalik kasutada nii käsureal kui ka läbi graafilise liidese veebibrauseris.

Läbi graafilise liidese on seda üsnagi lihtne kasutada. Saab määrata, milline tabeli ehitus hakkab välja nägema ning see salvestada. Peale salvestamise loob tööriist automaatselt tabelist mudeli, kontrolleri ja vaate. Antud objekte on lihtne luua, aga mooduleid see pakk ei genereeri. Ehk tavakasutaja vaates tuleb kõik käsitsi välja tuua.

Käsureal on CRUD tööriistaga rohkem võimalusi. Seda ainult selle poolest, et kui pole vaja kontrollereid ja vaateid luua, siis need astmed võib vahele jätta. Käsk selle jaoks on muidu suhteliselt pikk ja lohakas. Käsurea tööriistale saab ette anda ka “.json” faili, mille põhjal

genereerib vastavad elemendid. Loodud vaated kasutavad ainult külgriba ja pärise malli ning midagi muud juures pole.

Pakk tuleb kaasa minimaalse ülesehituse ja ainult CRUD tööriistaga. Seda on üsna lihtne kasutada - eriti selle graafilist liidest. Küll aga on elementide kuvamine ja üksteistest eraldamine suhteliselt aeganõudev tegevus. Menüüsid peale administraatori vaates olevate kasutajate ja rollide menüüdele lisaks pole. Andmete töötlemiseks peab arendama juurde autentimise osa, et tavakasutajatel poleks võimalik tähtsaid andmeid muuta.

3.2. Laraadmin

Asub aadressil www.laraadmin.com.

LaraAdmini projekti lisamiseks tuleb terminalis Composer käsk jooksutada, mis installeeriks praeguse kõige stabiilseima versiooni.

```
composer require "dwij/laraadmin:1.0.40"
```

Sellega saab vajaliku paki koos kõikide sõltuvustega, mida on LaraAdminil töötamiseks vaja, kätte. Kui failid on olemas, siis tuleb konfiguratsiooni kaustas, rakenduse failis lisada viide LAProvider klassile. Selleks tuleb projektis leida “/config” kaust ning sealt “app.php” fail. Seal tuleb lisada providers konteinerisse vastav rida:

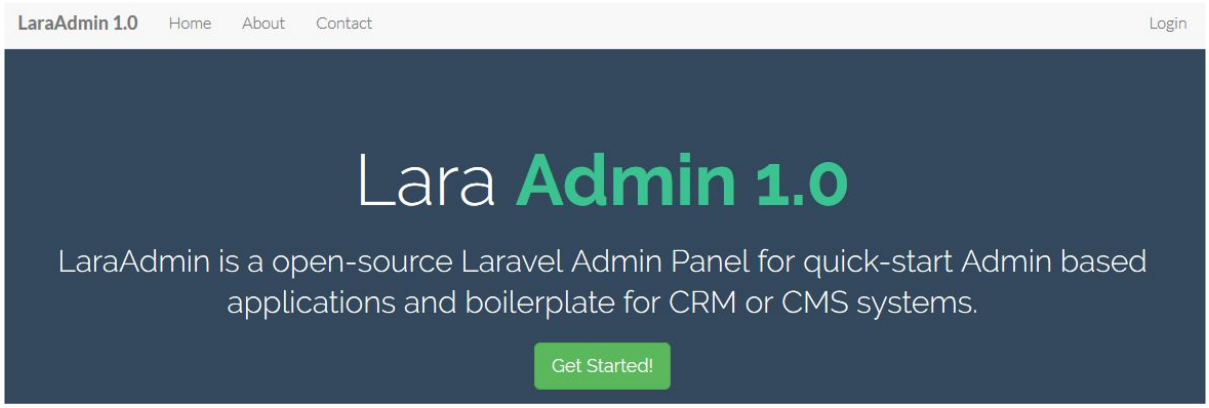
```
Dwij\Laraadmin\LAProvider::class
```

Kui viide on märgitud, siis on võimalik terminalis jooksutada käsk:

```
php artisan la:install
```

Antud käsk omakorda käivitab LaraAdmini ülesseadistamise protsessi. Protsess peaks pärima .env konfiguratsiooni faili muutmise kohta. Kuna see peaks juba eelnevalt muudetud olema, siis võib vastuseks anda ‘no’. Edasi peaks olema teave nelja projektis oleva faili salvestamise kohta. Aga kuna projekt peaks olema nullist üles seatud, siis nende salvestamine ei ole vajalik, kuna neid faile ei ole muudetud.

Viimane etapp enne ülesseadistamist on super admin kasutaja loomine. Selle täitmisel käivitub paigaldus protsess ning kõik vajalikud moodulid ja vaated luuakse ära. Õnnestumise korral peaks veebibrauser kuvama esilehena LaraAdmin tervitus lehte (Joonis 10). Administreerimise paneelile saab leheaadressile "/admin" lõppu panemisel.



Joonis 10. LaraAdmin admin leht

LaraAdmin on suhteliselt mahukas pakk ning paigalduse protsess on lihtne ja kiire. Kohe on olemas administraatori osa, mis on eraldatud tavakasutajatest. Administreerimise paneel sisaldab palju alguses olevaid mooduleid näiteks organisatsioonid, töötajad, kliendid. Uute moodulite ja elementide loomine on väga kiirelt teostatav läbi veebibrauseri kui ka käsurea. LaraAdmini osa kasutab Laravelist teistsuguseid migratsiooni faile. Väljade ehitused, mis ette antakse, on teistsugused. Näiteks uue migratsiooni loomiseks piisab ainult käsust:

```
php artisan la:migration klient
```

Luuakse uus migratsiooni fail "klient" teiste migratsioonide juurde. Sellel tuleb ära määrata, mis andmeid mudelis vaja oleks ning migratsiooni failid saab andmebaasi salvestada käsuga:

```
php artisan migrate
```

See loob vastava kliendi objekti ehk tabeli andmebaasi, aga kasutaja vaatest on see praegu puudu. CRUD elemendid on puudu, et saaks luua ja muuta kliendi objekte. Selle loomiseks piisab käsust:

```
php artisan la:crud Klients
```

Antud käsk loob mudeli, vaate, kontrolleri ning kasutaja vaatesse luuakse uus moodul nende andmete töötlemiseks.

LaraAdminil on kohe alguses kasutaja vaade olemas ning väga kergelt kasutatavad tööriistad, mis aitavad arendusprotsessile kaasa. Arendamine on üsna lihtne, sest väga palju pole vaja muuta. Saab hästi ära kasutada neid elemente, mis on ette antud. Uute moodulite loomisel on probleemiks, et mingi kindel mall on ette antud. Seda on võimalik muuta LaraAdmini abiklassides, mis asuvad projektis "/vendor/dwij". Siis tuleks otsida seda klassi, mida oleks vaja muuta. LaraAdmin võimaldab veel läbi veebibrauseri muuta mudeleid ja menüü punkte. Mõned elemendid tunduvad poolikud, näiteks vaadete muutmine veebibrauseris. Aga üldiselt neid super admin kasutajale pole vaja näidata ning koos sellega kaasneb ainult peitmise vaev. Ette antud vaate malli pole väga palju vaja muuta, sest kohe on olemas tabel objektist ning vastavad bootstrap modalid nende objektide töötlemiseks.

3.3. Rapyd

Asub veebilehel www.rapyd.com.

Kõige hilisem Rapyd versioon on 2.2.* ning see töötab kõige stabiilsemalt Laravel versioon 5.2.*. Et seda projekti lisada, tuleb läbi Composer installeerida vastav pakk.

```
composer require zode/rapyd: 2.2.*
```

Kui Composer on failid installeerinud, siis tuleb config kaustas app.phps viidata provider klassile. Selleks tuleb php failis providers konteinerisse lisada rida:

```
Zofe\Rapyd\RapydServiceProvider::class,
```

Lõpuks tuleb failid projektis avaldada käsuga:

```
php artisan vendor:publish
```

Rapyd on CRUD tööriist, mis aitab luua vidinaid andmete töötlemiseks ja kuvamiseks. Paari lihtsa koodireaga on võimalik kuvada vorme või ridu vaadetes. Nende loomine on väga kiire ja lihtne. Kontroller klassis peab looma meetodi, mis tagastab antud objekti või vaate. Näiteks kui luua vorm mingist mudelist, siis esmalt tuleb luua uus vorm antud mudelist ning hiljem määrata, mis andmeid on võimalik vormis täita. Rohkem pole vaja, sest tööriistad andmete töötlemiseks on olemas ja rohkem meetodeid või koodi pole vaja kirjutada vormi jaoks.

```php
public function formExample(){
    $form = \DataForm::source(Client::find(1));

    $form->add('title','Title', 'text')->rule('required|min:5');
    $form->add('body','Body', 'redactor');

    //belongs to many (field name must be the relation name)
            $form->add('cars','Cars','checkboxgroup')->options(Cars::lists('name', 'id')->all());

    $form->submit('Save');

    $form->saved(function () use ($form) {
        $form->message("ok record saved");
        $form->link("/rapyd-demo/form","back to the form");
    });

    return view('rapyd::demo.form', compact('form'));
}
```

Koodinäide 1. Rapyd vormi loomine

Et kasutaja näeks antud vormi, siis tuleb see vaatesse veel lisada.

Rapyd sobiks suurepäraselt kokku mingi administreerimis paneeli disainimalliga või isegi lisana kuskile juurde pandud. Ainuüksi selle kasutamisega tuleks veel palju lisasid juurde teha ning see pole veel kokkusobiv Laravel 5.2 uutemate versioonidega.

3.4. Admin-LTE

Asub http://packalyst.com/packages/package/acacha/admin-lte-template-laravel ning seda projekti on võimalik lisada käsuga:

```
composer require "acacha/admin-lte-template-laravel:4.*"
```

Peale failide edukat installeerimist tuleb konfiguratsiooni failide kaustas viidata provider klassile ning ära märkida alias. Selleks tuleb config kaustas app.php failis providers konteinerisse lisada rida:

```
Acacha\AdminLTETemplateLaravel\Providers\AdminLTETemplateSer
viceProvider::class,
```

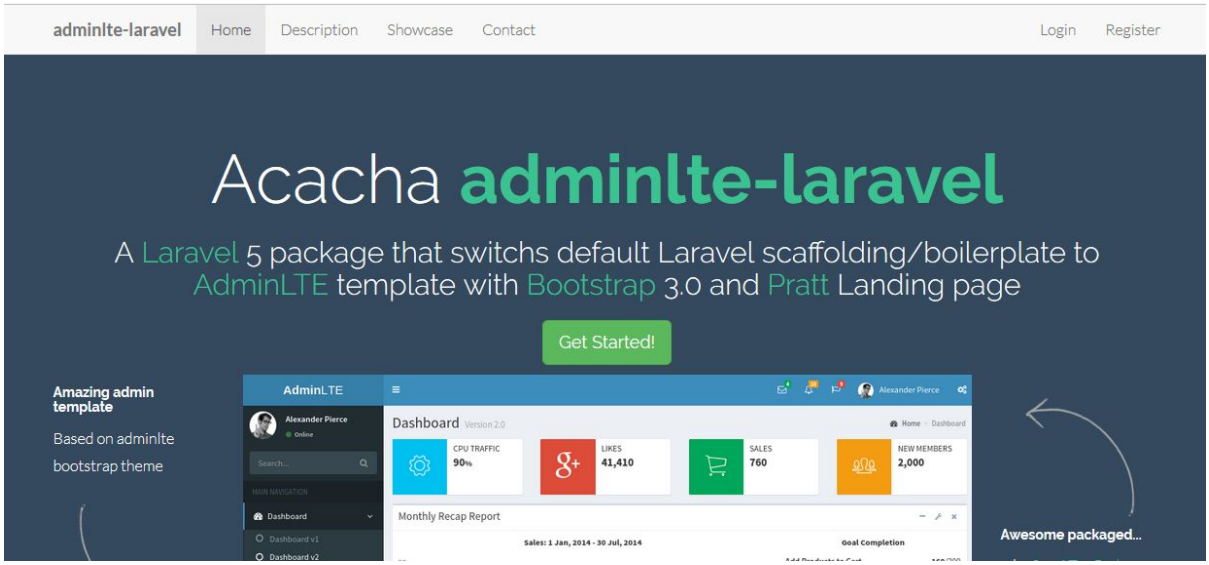
Ning aliaste konteinerisse lisada:

```
'AdminLTE'                                                    =>
Acacha\AdminLTETemplateLaravel\Facades\AdminLTE::class,
```

Viimaks tuleb lisa failid projektis avaldada käsuga.

```
php artisan vendor:publish
```

Õnnestumise korral peaks veebibrauser kuvama Admin-LTE maandumislehte (Joonis 11).



Joonis 11. Admin-LTE maandumisleht

Admin-LTE on disainimall administreerimis paneeli loomise jaoks. Põhirõhk on asetatud sellele, kuidas elemendid on paigutatud. Arendaja peab lihtsalt andmed õigetesse elementidesse lisama ning eraldama malli osa autentimisega. Admin-LTE kasutab Laraveli standardseid käske vaadete ning mudelise loomiseks. Ainuke vahe vaadete puhul on see, et kasutab kaasa tulnud elemente, kui on projektis avaldatud. Näiteks kui külgriba on projekti lisatud, siis kõik uued vaated kasutavad seda.

Antud pakk kasutab kaasa tulnud route.php faili suunamiseks õigetele kontroll klassidele.
Et sinna õiged suunamised tekiksid, peab kasutama Admin-LTE pakiga kaasa tulnud käske. Need on väga sarnased Artisan standard käskudele, aga ainuke vahe on süntaks ja see, et loodavad failid lähevad õigetesse kaustadesse õigete lisadega.

Lisatud objektede vahel on võimalik ainult soovitud asju kasutada. Näiteks on võimalik kasutada projektis ainult külgriba või päist.
Antud pakk peale disaini elementide väga palju võimalusi juurde ei anna. Kui oleks mingi tööriista kogum, mis kiirendaks moodulite loomise protsessi, siis oleks pakk väga kasulik.

3.5. Backpack

Asub lehel http://packalyst.com/packages/package/backpack/base.
Projekti lisamiseks läbi Composer tuleb jooksutada käsk:

```
composer require backpack/base
```

Järgmisena provider klass lisada app.php faili, mis asub projektis config kaustas.

```
Backpack\Base\BaseServiceProvider::class,
```

Ning lõpuks tuleb elemendid projektis avalikuks teha:

```
php artisan vendor:publish
--provider="Backpack\Base\BaseServiceProvider"
php artisan vendor:publish
--provider="Prologue\Alerts\AlertsServiceProvider"
php artisan migrate
```

Et saada registreerimise ja admin osa lehti kasutajale kuvada, tuleb veel luua vaated ning teha need läbi route.php nähtavaks.

Backpack on administreerimispaneeli malli sarnane, mis sarnaneb Admin-LTEga. Backpackiga alustamine on natuke rohkem aeganõudvam ning sama kehtib ka uute moodulite loomisel. Artisan kasutab standardseid käske uute elementide loomiseks. Malli välimust saab muuta base.php failis, mis asub kaustas /config/backpack. Backpack tuleb kaasa teatamissüsteemiga, mis reaalajas võimaldab näidata sõnumeid. Dokumentatsioon on suhteliselt puudulik. Esmakordne ülesseadistamine võib tekitada muresid sellega, et pole kirjas, kuhu mis failid tekivad ja kuidas neid kasutama peaks. Visuaalselt on väga kasulik, aga sellega asi piirdub, sest mingeid funktsioone ja võimalusi pole juurde antud.

3.6. Sobiva paneeli valimine

Kõige kasulikum pakk peaks lihtsustama moodulite loomist ja sisaldama autentimist tavakeskkonna ja administreerimise keskkonna vahel. Samuti peaks see genereerima võimalikult palju elemente, et arendaja saaks esialgse töötava keskkonna püsti kiirelt ja lihtsalt ning hiljem arendada selle peale võimalusi juurde. Pakid võiks töötada ka kõige uuemate Laravel versioonidega. Tabel 1 näitab, mis võimalusi pakid pakuvad ning mis versioonidega need töötavad.

Tabel 1. Pakkide võimalused ja sobivused

	Backpack	Laraadmin	Rapyd	Admin-LTE	CrudGenerator
CRUD		x	x		x
Teavitused	x				
Disainimall	x	x		x	x
Vaadete genereerimine	x	x			
Administreerimise osa		x		x	x
Autentimine		x			
Moodulite genereerimine	x	x		x	
5.1.*	x	x	x	x	x
5.2.*	x	x	x	x	x
5.3.*	x	x		x	x
5.4.*	x	x		x	x

Tabelis on välja toodud vajalikud aspektid, mida oleks vaja CRM süsteemi loomiseks.
Kõik andministreerimisepaneelid pakuvad väga erinevaid võimalusi, mis kiirendaksid CRM keskkonna ülesseadistamist. Admin-LTE ja Backpack pakuvad disainimalli väikeste lisadega. Admin-LTE võimaldab moodulite lihtsamat genereerimist ning Backpack pakub teavitussüsteemi. CRUD-generator ja rapyd on tööriistad CRUD elementide loomiseks. Disain on minimaalne, aga andmete töötlemine on tehtud väga lihtsaks. Nendega saab kontroller klassides luua elemendid, mida vaadetest kuvada. Antud tööriistu on võimalik ka kombineerida, andes juurde rohkem võimalusi. CrudGenerator pakub minimaalselt disaini, aga mingit eraldi autentimist pole. Sobiks hästi kokku näiteks Admin-LTE pakiga.

Antud pakkide vahel pakub LaraAdmin kõige rohkem võimalusi. Ülesseadistamine on väga lihtne ning peale seadistamist on praktiliselt kohe olemas töötav lahendus. Esialgsed andmed ja mõned vaated tuleks ära kustutada, kuna need sisaldavad näidisandmeid, mida pole vaja. Moodulite loomine on lihtne ning moodulite vaated, mis luuakse, on praktiliselt töötavad lehed. Lisaks pakub LaraAdmin läbi veebibrauseri palju muudatusi ning mõningaid arendusi. Tekstiredaktor on olemas, mis võimaldab vaadete muutmist, aga selleks peab projektis kirjutamise ja muutmise õigused olemas olema. Erinevate rollide puhul tuleks veel täiendusi teha, et kõik kasutajad ei saaks ligipääsu ebavajalikule infole. Kuna see pakk pakub kõige rohkem ja kasulikumaid võimalusi, siis sellest saab alusmall bakalaureusetöö tarbeks.

Kokkuvõte

Käesoleva seminaritöö eesmärgiks on võrrelda Laravel veebiraamistikule pakutavate kolmandate osapoolte administreerimispaneele. Et paneele üldse võrrelda pidi autor ülesseadistama Laravel projekti. Kuna autori bakalaureusetöö on CRM lahenduse loomine autotöökoja tarbeks, siis ülesseadistatav projekt võiks täita arenduskeskkonna rolli.
Administreerimispaneelide võdluse tulemusel soovib autor leida pakki, mis kiirendaks CRM süsteemi loomist. Et selleni jõuda, seadistab autor üles veebimajutuse keskkonna ning paneb seal jooksma veebiserveri.

Käesoleva töö käigus õppis autor palju uut juurde:

- Laravel veebiraamistikuga töötamist.
- Laravel veebiraamistiku ülesseadistamist veebimajutuse keskkonda.
- Composer tööriista kasutamist.
- Composeriga lisa pakkide installeerimist ning olemasolevate pakkide uuendamist.
- Erinevatest PHP veebiraamistikkudest ja nende omadustest.

Algselt ei õnnestunud Laravel projekti kuvamine veebiserveris. Standartselt tuleks projektis teha “migrations” ja “config” kaustad kõikide õigustega ehk projekti kaustas saavad kõik faile käivitada, luua ning muuta. “Migrations” ja “config” kaustade õiguste andmisest ei piisanud, sest projekti ikka ei kuvatud ning selle parandamiseks andis autor projekti kaustale kõik õigused.
Administreerimispaneelide testimise käigus esines mõningatel pakkidel andmete vahetuse viga. Kuna MySql pakiga tulevad kaasa draiverid andmebaasiga suhtlemiseks, siis mõndadel pakkidel oli vaja ka SqlLite draivereid. Isegi kui projekt ei kasutanud SqlLite andmebaasi, siis mõningad faile kasutati seal. mis võimaldas suhtlemist MySql andmebaasiga.
Testimise käigus selgus, et LaraAdmin administreerimispaneel on olemasolevatest pakkidest kõige kasulikum, mille peale CRM süsteemi ehitada. See võimaldab kiiret moodulite loomist ning koos moodulitega luuakse vaated ja kontrollerid, mida saaks juba rakenduses kasutada. Ülesseadistamine on väga lihtne ning see on avatud lähtekoodiga ehk olemasolevat

funktsionaalsust saab veel muuta, kui vajadus selleks tekib. LaraAdminit saab alusmall autori bakalaureusetöö koostamisel.

Kasutatud kirjandus

Amazon. (kuupäev puudub). Amazon AWS. Loetud aadressil https://aws.amazon.com/

Amazon. (kuupäev puudub). Amazon AWS EC2. Loetud aadressil https://aws.amazon.com/ec2/?ft=n

Amazon. (kuupäev puudub). Amazon AWS RDS. Loetud aadressil https://aws.amazon.com/rds/?ft=n

Admin-LTE. (kuupäev puudub). Admint-LTE. Loetud aadressil https://github.com/acacha/adminlte-laravel

Apache. (kuupäev puudub). About Apache. Loetud aadressil https://httpd.apache.org/ABOUT_APACHE.html

Backpack. (kuupäev puudub). Backpack documentation. Loetud aadressil https://laravel-backpack.readme.io/docs

Composer. (kuupäev puudub). Composer introduction. Loetud aadressil https://getcomposer.org/doc/00-intro.md

CrudGenerator. (kuupäev puudub). CrudGenerator. Loetud aadressil https://packagist.org/packages/appzcoder/crud-generator

LaraAdmin. (kuupäev puudub). LaraAdmin documentation. Loetud aadressil http://laraadmin.com/docs/1.0

Laravel. (Kuupäev puudub). Laravel framework. Loetud aadressil https://laravel.com/

Laravel. (kuupäev puudub). Artisan CLI. Loetud aadressil https://laravel.com/docs/5.0/artisan

Netcraft. (2017). Webserver survey. Loetud aadressil https://news.netcraft.com/archives/category/web-server-survey/

Oracle. (kuupäev puudub). What is MySql? Loetud aadressil http://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html

PHP. (kuupäev puudub). What is PHP? Loetud aadressil http://php.net/manual/en/intro-whatis.php

Rapyd. (kuupäev puudub). What is Rapyd. Loetud aadressil http://rapyd.com/post/what-is-rapyd

Sitepoint. (2015). Best PHP frameworks. Loetud aadressil https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/

Social compare. (uuendatud 2017). PHP frameworks comparison. Loetud aadressil http://socialcompare.com/en/comparison/php-frameworks-comparison

Surguy, M. (2013). History of Laravel PHP framework. Loetud aadressil https://maxoffsky.com/code-blog/history-of-laravel-php-framework-eloquence-emerging/

Ubuntu. (kuupäev puudub). Apache, MySql and PHP. Loetud aadressil https://help.ubuntu.com/community/ApacheMySQLPHP

Ubuntu. (kuupäev puudub). HTTPD. Loetud aadressil https://help.ubuntu.com/lts/serverguide/httpd.html

Ubuntu. (kuupäev puudub). MySql. Loetud aadressil https://help.ubuntu.com/lts/serverguide/mysql.html

W3techs. (kuupäev puudub). *Usage of server-side programming languages for websites.* Loetud aadressil https://w3techs.com/technologies/overview/programming_language/all

Wikipedia. (kuupäev puudub). Apache HTTP server. Loetud aadressil https://en.wikipedia.org/wiki/Apache_HTTP_Server

Wikipedia. (kuupäev puudub). LAMP. Loetud aadressil https://en.wikipedia.org/wiki/LAMP_(software_bundle)#MySQL_and_alternatives