



Univerzitet u Novo Sadu
Tehnički fakultet »Mihajlo Pupin«
Zrenjanin



SEMINARSKI RAD

Informacioni sistemi 2 - Veb aplikacija za biblioteku

Predmetni nastavnik:
prof. dr Biljana Radulović
Asistent:
doc. dr Zoltan Kazi

Student:
Aleksandar Berar IT 78/16

Zrenjanin, 2018

Sadržaj

1 Uvod.....	3
2 Instaliranje alata.....	5
2.1 MySQL.....	5
2.2 PHP.....	5
2.3 Laravel.....	6
3 Implementacija aplikacije.....	7
3.1 Konfigurisanje app.php.....	7
3.2 Konfigurisanje baze podataka.....	7
3.3 Kreiranje modela Knjiga.....	8
3.4 Rutiranje resursa Knjiga.....	8
3.5 Kreiranje KnjigaController.....	8
3.6 Pravljenje Blade podataka.....	12
5 SQL Skripte.....	18
6 Izvršavanje aplikacije.....	20
6.1 Korisničko uputstvo.....	20
7 Zaključak.....	24
8 Literatura i korišćenji web-sajtovi.....	26

1 Uvod

U ovom seminarskom radu će se izvesti serverska aplikacija koja predstavlja veb aplikaciju za biblioteku. To će se postići korišćenjem programskog jezika PHP 7.1 zajedno sa njegovim frejmworkom Laravel verzije 5. Koristiće se operativni sistem Lubuntu koji je vrlo sličan operativnom sistemu Ubuntu, i u pitanju je Linux operativni sistem.

Ovako izvedena aplikacija se može posmatrati kao standardna troslojna arhitektura aplikacije gde PHP predstavlja skriptni serverski sloj, koristi se HTML i CSS za prezentacioni sloj i MySQL baza podataka za sloj podataka pri čemu i sam Laravel frejmork pospešuje u vezi sa tim slojem i veoma olakšava dalji rad sa tim.

Što se tiče metodološke izvedbe ovog seminarskog rada, nakon uvoda će se predstaviti instalacija svih potrebnih alata za razvoj aplikacije, a oni su baza podataka MySQL, alati za programski jezik PHP, composer program i Laravel frejmwork. To će se uraditi korišćenjem aptitude programa iz paketa programa Lubuntu operativnog sistema.

Za testiranje pokrenute aplikacije, primenjivaće se pregledači Firefox i Chrome. A za uređivanje tekstualnih dokumenata koristiće se programi Leafpad, Atom i IntelliJ Idea. Za instaliranje alata će se koristiti emulator terminala Liks-terminal.

Nakon poglavlja koje objašnjava instaliranje svih potrebnih alata, sledi će poglavlje o realizaciji programskog koda u PHP programerskom jeziku, kao i potrebnim konfiguracijama baze podataka i ostalog što ide u sklop Laravel frejmworka.

Kada se bude objasnio razvoj aplikacije, slediće poglavlje sa slikama urađene web aplikacija i to se poglavlje može ujedno posmatrati i kao korisničko uputstvo aplikacije.

Krajnje poglavlje je zaključak gde će se izvesti rezime svog seminarskog rada i daće se dalje mogućnosti za razvoj aplikacije kako bi se poboljšala funkcionalnost.

Cilj ovog rada je da se pokaže kreiranje osnovnog CRUD (create, read, update i delete) modula u Laravel 5.5 aplikaciji. Laravel je popularan open-source PHP MVC frejmwork sa mnogim naprednim funkcionalnostima za razvoj. Laravel verzije 5.5 je izašao skoro.

Ovde će se prikazati unos, ažuriranje, brisanje i pregled sa paginacijom na primeru knjiga u ponudi biblioteke. Na jednostavan način će biti moguće dodavati nove knjige, pregledati knjige, ažurirati ih i brisati knjige iz liste. To će se uraditi jednostavnim primerom korišćenjem Laravel Form builder-a zajedno sa resursnim rutiranjem. Praviće se resursni kontroler za CRUD aplikacije u Laravel 5.5 frejmworku. Tako da će se sve to objasniti korak po korak. Na kraju će se rezultat prikazati preko slika u shodnom poglavlju.

U sledećem poglavlju će se objasniti CRUD operacije nad modelom koji predstavlja knjigu. To je složeniji model u aplikaciji. Jer postoji još modela, a to su autor i izdavač. Na nivou baze podataka, knjiga sadrži ključ koji pokazuje na neki indentifikator od tablica autor i izdavač, jer svaka knjiga ima svog autora i izdavača. I time se ispunjava jedan od uslova izrade seminarskog rada, da se koriste barem dve tablice u bazi podataka i da su u relaciji između sebi. Kod za modela autora i izdavača su veoma slični kao kodovi za model knjige, tako da se on neće izložiti u ovoj dokumentaciji ali će biti priložen izvorni kod sa celokupnim projektom.

2 Instaliranje alata

Za izradu ovog seminarskog rada se koristi operativni sistem Lubuntu koji je veoma sličan operativnom sistemu Ubuntu iz familije Linux operativnih sistema. Sam sistem Lubuntu dolazi sa upravljačem paketa koji se naziva aptitude tako da će se taj program koristiti za instaliranje svih potrebnih alata kako bi se razvila PHP aplikacija.

2.1 MySQL

MySQL je open-source sistem za upravljanjem bazama podataka, koji se uglavnom instalira kao deo popularnog LAMP (Linux, Apache, MySQL, PHP) steka. On koristi relacionalnu bazu podataka i SQL (Structured Query Language) da bi upravljao sa svojim podacima.

Skraćena verzija za instalaciju je prosta: ažurira se indeks paketa, instalira se *mysql-server* paket, i onda se pokreće bezbednosna skripta koja dolazi sa tim.

```
$ sudo apt-get update
$ sudo apt-get install mysql-server
$ mysql_secure_installation
```

Listing 1. Komande za instaliranje MySQL baze podataka

2.2 PHP

Pre nego što se započne sa instalacijom, uvek je dobra ideja da se provere izvori i da li je postojeći softver ažuriran.

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

Listing 2. Komande za ažuriranje i nadogradnju sistema

Svakako se u ovom seminarskom radu koristi server baziran na Ubuntu koji se i pokreće. Pre nego što se instalira Laravel frejmwork, moraju se instalirati ostale komponente koje su neophodne.

Sledeći korak je da se instalira PHP verzije 7.1 zajedno sa još nekoliko dodatnih paketa koji će se pokazati korisnima ako se želi raditi sa Laravel frejmworkom.

```
$ sudo add-apt-repository ppa:ondrej/php
$ sudo apt-get update
$ sudo apt-get install php7.1 php7.1-mcrypt php7.1-xml php7.1-gd php7.1-
opcache php7.1-mbstring php7.1-mysql
```

Listing 3. Komande za instaliranje PHP programskog jezika

Iako Ubuntu ima svoj repozitorijum za PHP, ovde je bolje koristiti repozitorijum od treće stranke zato što se on mnogo češće ažurira. Taj se korak može preskočiti ako se preferira korišćenje Ubuntu-ovog repozitorijuma.

2.3 Laravel

Pre nego što se instalira Laravel frejmwork, potrebno je instalirati program *composer* koji je pomoćni program za rad sa PHP alatima i sa njim će se dalje instalirati Laravel frejmwork i ostali alati koji su potrebni. Composer se instalira korišćenjem komande iz Listinga 3.

```
$ sudo apt-get install composer
```

Listing 4. Komande za instaliranje Laravel frejmworka

Zahtev je da se koristi Laravel frejmwork verzije 5.5 za aplikaciju. Tako da će se koristiti donja komanda, znači da se ona poziva korišćenjem terminalskog emulatora da bi se instalirao Laravel frejmwork i pritom kreirao projekat koji će se pojaviti u otvorenom direktorijumu, a naziv novonastalog direktorijuma sa projektom biće biblioteka.

```
$ composer create-project --prefer-dist laravel/laravel biblioteka
```

Listing 5. Komande za instaliranje Laravel frejmworka

Ovom komandom je uspešno instaliran Laravel frejmwork, nakon što se sačeka da se preuzmu svi podaci sa server i da se nakon toga oni i instaliraju. Takođe je kreiran i podfolder biblioteka u kojem se nalazi struktura Laravel aplikacije.

Kada se to obavilo, potrebno je pomeriti se preko konzole u novonastali folder biblioteka i izvršiti komandu za instaliranje laravelcollective/html composer paketa koji služi za izradu formi zato će se iz terminala pokrenuti komanda iz Listinga 6.

```
$ composer require laravelcollective/html
```

Listing 6. Komande za instaliranje Laravel frejmworka

3 Implementacija aplikacije

U ovom poglavlju ovog seminarskog rada će se pričati o samoj implementaciji aplikacije korišćenjem programerskog jezika PHP sa Laravel frejmworkom i MySQL bazom podataka, gde se u prethodnom poglavlju objasnila instalacija svih tih alata. Pored detaljnog objašnjenja implementacije svih slojeva arhitekture aplikacije, daće se i potrebne konfiguracije za aplikaciju i za bazu podataka u sklopu Laravel frejmworka.

3.1 Konfigurisanje app.php

Nakon što su se uspešno instalirali svi paketi, potrebno je otvoriti fajl *config/app.php* koji se nalazi u baznog folderu *biblioteka* i dodati servisne omogućivače i alijase.

```
'providers' => [  
    ....  
    Collective\Html\HtmlServiceProvider::class,  
],  
'aliases' [  
    ....  
    'Form' => Collective\Html\FormFacade::class,  
    'Html' => Collective\Html\HtmlFacade::class,  
],
```

Listing 7. Uređivanje fajla app.php

3.2 Konfigurisanje baze podataka

U ovom koraku će se namestiti konfiguracija baze podataka, na primer, ime baze podataka, šifra, i tako dalje, za aplikaciju koja koristi Laravel frejmwork 5.5. Stoga mora da se koriguje fajl *.env*.

```
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1  
DB_PORT=3306  
DB_DATABASE=biblioteka  
DB_USERNAME=root  
DB_PASSWORD=root
```

Listing 8. Uređivanje fajla .env

3.3 Kreiranje modela Knjiga

Nakon što je kreirana tablica *knjigas* treba da se kreira model za “knjige” tablicu tako što će se pokrenuti donja komanda i time će se stvoriti novi model:

```
$ php artisan make:model Knjiga
```

Listing 9. Komanda za kreiranje modela Knjiga

Ako se to uradilo, onda će se pronaći podatak *app/Knjiga.php* i u taj podatak će se nalepiti sadržaj iz Listinga 10.

```
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class Knjiga extends Model
{
    protected $fillable = [
        'naslov', 'godina_objave', 'cena', 'izdavac_id', 'autor_id'
    ];
}
```

Listing 10. Sadržaj fajla Knjiga.php

3.4 Rutiranje resursa Knjiga

U ovom koraku je potrebno dodati resursnu putanju za CRUD aplikacije za Knjiga. Tako da će se otvoriti *routes/web.php* fajl i dodaće se sledeća ruta.

```
Route::resource('knjigas', 'KnjigaController');
```

Listing 11. Komanda za rutiranje

3.5 Kreiranje KnjigaController

Sada je potrebno da se namesti novi kontroler pod nazivom KnjigaController. Zato se izvršava donja komanda da bi se kreirao novi kontroler. Ovo je resursni kontroler.

```
$ php artisan make:controller KnjigaController -resource
```

Listing 12. Komanda za kreiranje kontrolera KnjigaController

Nakon što je izvršena gornja komanda moći će da se pronađe novi fajl na putanji *app/Http/Controllers/KnjigaController.php*.

Unutar ovog kontrolera će se kreirati sedam metoda po osnovnoj vrednosti koji će se dole navesti:

1. index()
2. create()
3. store()
4. show()
5. edit()
6. update()
7. destroy()

Dole će se navesti kod koji treba da se prekopira u taj fajl.

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use DB;
use App\Knjiga;
use Illuminate\Support\Facades\Input;
class KnjigaController extends Controller
{
    public function index(Request $request) {
        $search = $request->get('search');
        if ($search == '') {
            $knjigas = DB::table('knjigas')
                ->join('autors', 'knjigas.autor_id', '=',
'autors.id')
                ->join('izdavacs', 'knjigas.izdavac_id', '=',
'izdavacs.id')
                ->select('knjigas.*', 'autors.prezime as autor_id',
'izdavacs.naziv as izdavac_id')
                ->paginate(5);
        } else {
            $knjigas = DB::table('knjigas')
                ->join('autors', 'knjigas.autor_id', '=',
'autors.id')
                ->join('izdavacs', 'knjigas.izdavac_id', '=',
'izdavacs.id')
                ->select('knjigas.*', 'autors.prezime as autor_id',
```

```

'izdavacs.naziv as izdovac_id')
    ->where('knjigas.naslov', 'like', '%'.$search.'%')
    ->paginate(5);
}
$stampaj = Input::get('stampaj');
return view('knjigas.index',compact('knjigas', 'stampaj'))
    ->with('i', (request()->input('page', 1) - 1) * 5);
}
public function create()
{
    return view('knjigas.create');
}
public function store(Request $request)
{
    DB::beginTransaction();
    request()->validate([
        'naslov' => 'required',
        'godina_objave' => 'required',
        'cena' => 'required',
        'autor_id' => 'required',
        'izdovac_id' => 'required'
    ]);
    try {
        Knjiga::create($request->all());
        DB::commit();
        return redirect()->route('knjigas.index')
            ->with('success','Knjiga je uspesno sacuvana.');
```

```

} catch (Exception $e) {
    DB::rollback();
    return redirect()->route('knjigas.index')
        ->with('error','Knjiga nije sacuvana.');
```

```

}
public function show($id)
{
    $knjiga = Knjiga::find($id);
    return view('knjigas.show',compact('knjiga'));
}
public function edit($id)
{
    $knjiga = Knjiga::find($id);
    return view('knjigas.edit',compact('knjiga'));
}

```

```

}
public function update(Request $request, $id)
{
    DB::beginTransaction();
    request()->validate([
        'naslov' => 'required',
        'godina_objave' => 'required',
        'cena' => 'required',
        'autor_id' => 'required',
        'izdavac_id' => 'required'
    ]);
    try {
        Knjiga::find($id)->update($request->all());
        DB::commit();
        return redirect()->route('knjigas.index')
            ->with('success','Knjiga je uspesno
azurirana.');
```

```

    } catch (Exception $e) {
        DB::rollback();
        return redirect()->route('knjigas.index')
            ->with('error','Knjiga nije uspesno
azurirana.');
```

```

    }
}
public function destroy($id)
{
    DB::beginTransaction();
    try {
        Knjiga::find($id)->delete();
        DB::commit();
        return redirect()->route('knjigas.index')
            ->with('success','Knjiga je uspesno
obrisana.');
```

```

    } catch (Exception $e) {
        DB::rollback();
        return redirect()->route('knjigas.index')
            ->with('error','Knjiga nije uspesno
obrisana.');
```

```

    }
}
}
}

```

Listing 13. Kod KnjigaController

3.6 Pravljenje Blade podataka

Sada sleduje poslednji korak pri izradi implementacije aplikacije sa Laravel frejmworkom. U ovom koraku će se praviti blade fajlovi. Što znači da će se uglavnom praviti “layout” šablonski fajlovi i onda će se namestiti novi folder “knjigas”, i onda će se praviti blade fajlovi od CRUD aplikacije.

Što znači da se konačno trebaju namestiti sledeći blade podaci:

1. layout.blade.php
2. index.blade.php
3. show.blade.php
4. form.blade.php
5. create.blade.php
6. edit.blade.php

Tako da će se izlistati sadržaj tako nameštenih fajlova.

```
<!DOCTYPE html>
<html>
<head>
  <title>Veb aplikacija za Biblioteku</title>
  <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.0.0-alpha/css/bootstrap.css" rel="stylesheet">
</head>
<body>
<div class="container">
  @yield('content')
</div>
</body>
</html>
```

Listing 14. Kod resources/views/layout.blade.php

Kod iz Listinga 14. predstavlja sadržaj fajla layout.blade.php koji služi kao sadržilac mustre u koju se ostale mustre dodavaju kao sadržaj. A to se već može primetiti iz sledećeg listinga, iz prve linije koda se jasno vidi da se tako poziva referencom na ovaj fajl, *layout*.

```
@extends('layout')
@section('content')
<div class="row">
```

```

<div class="col-lg-12 margin-tb">
  <div class="pull-left">
    <h2>Veb aplikacija za Biblioteku</h2>
  </div>
  @if (!$stampaj)
    <div class="pull-right">
      <a class="btn btn-success"
href="{{ route('knjigas.create') }}"> Dodati Novu Knjigu</a>
      <a class="btn btn-info" href="/"> Meni</a>
      <a class="btn btn-primary" href="/knjigas?
stampaj=true">Štampaj</a>
    </div>
  @endif
</div>
@if (!$stampaj)
  {!! Form::open(['method'=>'GET', 'route' =>
['knjigas.index'], 'class'=>'navbar-form navbar-
left', 'role'=>'search']) !!}
  <div class="input-group custom-search-form">
    <input type="text" class="form-control" name="search"
placeholder="Pretraži po Naslovu...">
    <span class="input-group-btn">
      <button class="btn btn-default-sm" type="submit">
        <i class="fa fa-search">i
      </button>
    </span>
  </div>
@endif
</div>
@if ($message = Session::get('success'))
<div class="alert alert-success">
  <p>{{ $message }}</p>
</div>
@endif
<table class="table table-bordered">
  <tr>
    <th>No</th>
    <th>Naslov</th>
    <th>Godina Objave</th>
    <th>Cena</th>
    <th>Autor</th>
    <th>Izdavac</th>
  </tr>

```

```

        @if (!$stampaj)
        <th width="280px">Akcija</th>
        @endif
    </tr>
    @foreach ($knjigas as $knjiga)
    <tr>
        <td>{{ ++$i }}</td>
        <td>{{ $knjiga->naslov }}</td>
        <td>{{ $knjiga->godina_objave }}</td>
        <td>{{ $knjiga->cena }}</td>
        <td>{{ $knjiga->autor_id }}</td>
        <td>{{ $knjiga->izdavac_id }}</td>
        @if (!$stampaj)
        <td>
            <a class="btn btn-info" href="{ route('knjigas.show',
$knjiga->id) }">Pokazi</a>
            <a class="btn btn-primary" href="{ route('knjigas.edit',
$knjiga->id) }">Uredi</a>
            {!! Form::open(['method' => 'DELETE', 'route' =>
['knjigas.destroy', $knjiga->id], 'style'=>'display:inline']) !!}
            {!! Form::submit('Obrisi', ['class' => 'btn btn-danger'])
!!}

            {!! Form::close() !!}
        </td>
        @endif
    </tr>
    @endforeach
</table>
{!! $knjigas->links() !!}
@endsection

```

Listing 15. Kod resources/views/knjigas/index.blade.php

Iz Listinga 15. se vidi celokupan kod fajla index.blade.php koji se odnosi na domen Knjiga. Taj fajl je jako bitan jer će se preko njega pristupati tabelarnom prikazu, tj. indeks stranici za model Knjiga, što će se videti u poglavlju korisničkog uputstva.

```

@extends('layout')
@section('content')
<div class="row">
    <div class="col-lg-12 margin-tb">
        <div class="pull-left">
            <h2> Prikazi Knjigu</h2>

```

```

        </div>
        <div class="pull-right">
            <a class="btn btn-primary"
href="{{ route('knjigas.index') }}"> Nazad</a>
        </div>
    </div>
</div>
<div class="row">
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Naslov:</strong>
            {{ $knjiga->naslov}}
            <br/>
            <strong>Godina Objave:</strong>
            {{ $knjiga->godina_objave}}
            <br/>
            <strong>Cena:</strong>
            {{ $knjiga->cena}}
            <br/>
            <strong>Autor:</strong>
            {{ $knjiga->autor_id}}
            <br/>
            <strong>Izdavac:</strong>
            {{ $knjiga->izdavac_id}}
            <br/>
        </div>
    </div>
</div>
</div>
@endsection

```

Listing 16. Kod resources/views/knjigas/show.blade.php

U Listingu 16. se vidi ceo kod fajla show.blade.php za model Knjiga. Ovo je kod sekcije koji služi za prikazivanje pojedinačne knjige.

```

<div class="row">
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Naslov:</strong>
            {!! Form::text('naslov', null, array('placeholder' =>
'Naslov','class' => 'form-control')) !!}
            <strong>Godina Objave:</strong>
            {!! Form::text('godina_objave', null, array('placeholder'

```

```

=> 'Godina Objave','class' => 'form-control')) !!}
    <strong>Cena:</strong>
    {!! Form::text('cena', null, array('placeholder' =>
'Cena','class' => 'form-control')) !!}
    <strong>Autor:</strong>
    {!! Form::text('autor_id', null, array('placeholder' =>
'Autor','class' => 'form-control')) !!}
    <strong>Izdavac:</strong>
    {!! Form::text('izdavac_id', null, array('placeholder' =>
'Izdavac','class' => 'form-control')) !!}
    </div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12 text-center">
    <button type="submit" class="btn btn-
primary">Posalji</button>
</div>
</div>

```

Listing 17. Kod resources/views/knjigas/form.blade.php

Kod forme modela Knjiga je predstavljen u Listingu 17. To znači da je ovo sadržaj fajla za unos nove knjige.

```

@extends('layout')
@section('content')
<div class="row">
    <div class="col-lg-12 margin-tb">
        <div class="pull-left">
            <h2>Dodaj Novu Knjigu</h2>
        </div>
        <div class="pull-right">
            <a class="btn btn-primary"
href="{{ route('knjigas.index') }}"> Nazad</a>
        </div>
    </div>
</div>
@if (count($errors) < 0)
<div class="alert alert-danger">
    <strong>Ups!</strong> Vas unos nije dobar.<br><br>
</div>
@endif
{!! Form::open(array('route' =>
'knjigas.store','method'=>'POST')) !!}

```



```
@include('knjigas.form')
{!! Form::close() !!}
@endsection
```

Listing 18. Kod resources/views/knjigas/create.blade.php

U Listingu 18. je dat celokupan kod fajla create.blade.php za model Knjiga. To je bitan fajl koji služi za dodavanje nove knjige. U kodu se vidi da se preko direktive `@include` ubacuje sadržaj iz prethodnog listinga, Listinga 17., koji je vezan za unosnu formu.

```
@extends('layout')
@section('content')
<div class="row">
  <div class="col-lg-12 margin-tb">
    <div class="pull-left">
      <h2>Uredi Knjigu</h2>
    </div>
    <div class="pull-right">
      <a class="btn btn-primary"
href="{{ route('knjigas.index') }}"> Nazad</a>
    </div>
  </div>
</div>
@if (count($errors) > 0)
<div class="alert alert-danger">
  <strong>Ups!</strong> Vas unos nije dobar.<br><br>
</div>
@endif
{!! Form::model($knjiga, ['method' => 'PATCH', 'route' =>
['knjigas.update', $knjiga->id]]) !!}
@include('knjigas.form')
{!! Form::close() !!}
@endsection
```

Listing 19. Kod resources/views/knjigas/edit.blade.php

U Listingu 19. je dat ceo kod za podatak edit.blade.php za model Knjiga. U pitanju je kod koji služi za ažuriranje modela Kniga. U kodu se vidi da se preko direktive `@include` ubacuje sadržaj iz prethodnog listinga, Listinga 17., koji je vezan za unosnu formu, što se u ovom slučaju koristi za uređivanje, odnosno, ažuriranje pojedinačnog reda iz tablice knjigas.

5 SQL Skripte

Potrebno je da se izvrše neke komande unutar mysql konzole da bi se namestila baza podataka i neophodne tablice.

```
$ mysql -u root -p
Enter your password: root

mysql> CREATE DATABASE biblioteka;

mysql> USE biblioteka;

mysql>
CREATE TABLE IF NOT EXISTS authors (
  id int(5) NOT NULL AUTO_INCREMENT,
  ime varchar(50) DEFAULT NULL,
  prezime varchar(50) DEFAULT NULL,
  godina_rodjenja int(5) DEFAULT NULL,
  drzava varchar(50) DEFAULT NULL,
  created_at datetime DEFAULT NULL,
  updated_at datetime DEFAULT NULL,
  PRIMARY KEY(id)
);

mysql>
CREATE TABLE IF NOT EXISTS izdavacs (
  id int(5) NOT NULL AUTO_INCREMENT,
  naziv varchar(50) DEFAULT NULL,
  godina_nastanka int(5) DEFAULT NULL,
  drzava varchar(50) DEFAULT NULL,
  created_at datetime DEFAULT NULL,
  updated_at datetime DEFAULT NULL,
  PRIMARY KEY(id)
);

mysql>
CREATE TABLE IF NOT EXISTS knjigas (
  id int(5) NOT NULL AUTO_INCREMENT,
  naslov varchar(50) DEFAULT NULL,
  godina_objave int(5) DEFAULT NULL,
  cena float(7,2) DEFAULT NULL,
  created_at datetime DEFAULT NULL,
  updated_at datetime DEFAULT NULL,
  izdavac_id int(5),
  autor_id int(5),
```

```
PRIMARY KEY(id),  
FOREIGN KEY (izdavac_id) REFERENCES izdavacs(id),  
FOREIGN KEY (autor_id) REFERENCES autors(id)  
);
```

Listing 20. Komande u mysql programu

6 Izvršavanje aplikacije

Sada je sve spremno da bi se aplikacija pokrenula sa izvršavanjem. Da bi se ova CRUD aplikacija pokrenula, potrebno je izvršiti donju navedenu komandu.

```
$ php artisan serve
```

Listing 21. Komanda za pokretanje izvršavanja aplikacije

Nakon pokretanja aplikacije, sa određenim pretraživačem se može pristupiti njoj tako što će se posetiti donji link iz Listinga 22.

```
http://localhost:8000/
```

Listing 22. Link ka glavnom meniju aplikacije

```
http://localhost:8000/knjigas
```

Listing 23. Link ka aplikaciji za CRUD za model Knjiga

```
http://localhost:8000/izdavacs
```

Listing 24. Link ka aplikaciji za CRUD za model Izdavač

```
http://localhost:8000/autors
```

Listing 25. Link ka aplikaciji za CRUD za model Autor

6.1 Korisničko uputstvo

Veb Aplikacija za Biblioteku

KNJIGE AUTORI IZDAVACI

Slika 1. Glavni meni aplikacije

Na Slici 1. je predstavljen glavni meni aplikacije gde se može izabrati neki od modela aplikacije pritiskom na dugme. Ti modelu si KNJIGE, AUTORI i IZDAVACI i oni se nalaze kao dugmad odmah ispod naslova. Klikom na ta dugmad se vodi na linkove koji su navedeni u prethodna tri Listinga.

Pretrazi po Nazivu...

No	Naziv	Godina Nastanka	Drzava	Akcija
1	Springer Publishing	1950	USA	Pokazi Uredi Obrisi
2	Laguna	1998	Srbija	Pokazi Uredi Obrisi
3	O'Reilly Media	1978	USA	Pokazi Uredi Obrisi

Slika 2. Tabelarni prikaz izdavača

Na Slici 2. je dat tabelarni prikaz izdavača koji se nalazi na linku /izdavacs. Preko dugmeta *Dodati Novog Izdavaca* se vrši pravljenje novog izdavača. Klikom na dugme *Meni* se vraća na glavni meni. A dugme *Štampaj* daće tabelarni prikaz shodan za štampanje.

Za svaki red modela postoje dugmad *Pokazi*, *Uredi* i *Obrisi* što su standardne CRUD operacije sa kojima je moguće upravljati modelom.

Dodaj Novog Izdavaca

Nazad

Naziv:

Godina Nastanka:

Drzava:

Posalji

Slika 3. Izgled stranice za dodavanje novog izdavača

Na Slici 3. je dat izgled stanice za dodavanje novog izdavača gde se unose sve potrebne vrednosti. Ukoliko je sve ispravno, dodaće se novi red i vratiće se rutiranjem na indeks stranicu modela.

Pretrazi po Prezimenu...

No	Ime	Prezime	Godina Rodjenja	Drzava	Akcija
1	Kristofer	Bisop	1959	Engleska	Pokazi Uredi Obrisi
2	Trevor	Hasti	1953	Juzna Afrika	Pokazi Uredi Obrisi
3	Nele	Karajlic	1962	Bosna	Pokazi Uredi Obrisi
4	Vanja	Bulic	1947	Srbija	Pokazi Uredi Obrisi
5	Fredrik	Bakman	1981	Svedska	Pokazi Uredi Obrisi

Slika 4. Tabelarni prikaz autora

Na Slici 4. je dat tabelarni prikaz autora koji se nalazi na linku /autors. Preko dugmeta *Dodati Novog Autora* se vrši pravljenje novog autora. Klikom na dugme *Meni* se vraća na glavni meni. A dugme *Štampaj* daće tabelarni prikaz shodan za štampanje.

Za svaki red modela postoje dugmad *Pokaži*, *Uredi* i *Obriši* što su standardne CRUD operacije sa kojima je moguće upravljati modelom.

Slika 5. Izgled stranice za dodavanje novog autora

Na Slici 5. je dat izgled stanice za dodavanje novog autora gde se unose sve potrebne vrednosti. Ukoliko je sve ispravno, dodaće se novi red i vratiće se rutiranjem na indeks stranicu modela.

No	Naslov	Godina Objave	Cena	Autor	Izdavac	Akcija
1	Moja baka vam se izvinjava	2018	1150	Bakman	Laguna	Pokazi Uredi Obriši
2	Covek po imenu Uve	2017	1200	Bakman	Laguna	Pokazi Uredi Obriši
3	Dusanova kletva	2018	780	Bulic	Laguna	Pokazi Uredi Obriši
4	Solunska 28	2018	1150	Karajlic	Laguna	Pokazi Uredi Obriši
5	The Elements of Statistical Learning	2009	2499	Hasti	Springer Publishing	Pokazi Uredi Obriši

Slika 6. Tabelarni prikaz knjige

Na Slici 4. je dat tabelarni prikaz knjiga koji se nalazi na linku /knjigas. Preko dugmeta *Dodati Novu Knjigu* se vrši pravljenje nove knjige. Klikom na dugme *Meni* se vraća na glavni meni. A dugme *Štampaj* daće tabelarni prikaz shodan za štampanje.

Za svaki red modela postoje dugmad *Pokaži*, *Uredi* i *Obriši* što su standardne CRUD operacije sa kojima je moguće upravljati modelom.

Web aplikacija za Biblioteku

No	Naslov	Godina Objave	Cena	Autor	Izdavac
1	Moja baka vam se izvinjava	2018	1150	Bakman	Laguna
2	Covek po imenu Uve	2017	1200	Bakman	Laguna
3	Dusanova kletva	2018	780	Bulic	Laguna
4	Solunska 28	2018	1150	Karajlic	Laguna
5	The Elements of Statistical Learning	2009	2499	Hasti	Springer Publishing

Slika 7. Prikaz rezultata Štampaj za model Knjiga

Na Slici 7. je dat prikaz rezultata štampanja modela knjiga. To je stranica koja se dobija ukoliko se klikne da dugme *Štampaj* koje se nalazi na indeks stranici modela koja se pokazala na Slici 6. u ovom radu.

7 Zaključak

U ovom seminarskom radu se prikazala jedna aplikacija sa CRUD sposobnostima korišćenjem PHP programskog jezika sa Laravel 5.5 frejvorkom i MVC pristupom.

Zahtev za izradu seminarskog rada je pravljenje sajta tematike po izboru tako da se odabrala tema veb aplikacija za biblioteku. Ispunjenu su svi uslovi za izradu:

- Čitanje, pisanje i brisanje podataka iz baze putem HTML formi (kroz objektno orijentisano programiranje)
- Povezivanje između više entiteta (bar dva)
- Filtriranje podataka putem polja za pretragu
- I još mnogi drugi.

Pored toga, koristio se poseban frejmvork, Laravel 5.5, što je dodalo poboljšanu i napredniju funkcionalnosti.

Što se tiče metodološke uređenosti rada, rad je započet sa uvodom, nakon čega je sledilo poglavlje o instalaciji potrebnih alata za izradu aplikacije. Alati koji su bili potrebni i koji su se koristili su PHP, MySQL, Laravel, Composer. Navedeni su i alati koji su korišćeni za izradu, operativni sistem Linux Ubuntu, a od tekstualnih editora i IDE su se koristili leafpad, atom, IntelliJ Idea, i terminalski emulator Liks-emulator.

Nakon što je navršeno poglavlje o instaliranju alata, započeto je poglavlje o kompletnoj izradi podataka za model Knjiga. Pored ovog modela, u sistemu postoji i drugi modeli, Autor i Izdavač, koji su u relaciji sa modelom Knjiga. Ali je model Knjiga predstavljen zato što je složeniji i izloženo je mnogoštvo koda putem raznih listinga gde je predstavljen celokupan MVC pristup koji je u srži Laravel 5.5 frejmvorka.

Sa obrzirom da se koristila MySQL baza podataka, to je bilo i naredno poglavlje, gde se opisala konekcija sa njom i kreiranje baze podataka zajedno sa neophodnim tablicama koje odgovaraju model klasama.

Krajnje poglavlje razrade je nazvano Izvršavanje aplikacije gde su se iznele komande i linkovi za pregled aplikacije kao i slike koje predstavljaju rad sa urađenom aplikacijom.

Što se tiče daljih poboljšanja aplikacije, vrlo je lako proširiti modela Kniga, Autor i Izdavač jer se oni trenutno sastoje od samo nekolicine polja tako da bi se njihovim proširavanjem dobio puniji sadržaj koji bi bio interesantniji za posetioce web sajta. Pored toga, moguće bi bilo dodati i druge modele, na primer, korisničke komentare, što bi dalje povuklo izradu korisničkih ulogu

gde bi pored anonimnog posetioca web sajta, postojao i registrovan posetilac sa privilegijom da komentariše na knjige. Još jedna moguća uloga u aplikaciji bi mogao biti administrator koji bi imao nadzornu ulogu i dodatne privilegije kao što su CRUD operacije nad registrovanim korisnicima.

Autor ovog rada je po prvi put bio u dodiru sa Laravel frejmworkom, a tokom slušanja predmeta Internet alati i servisi, i sa programskim jezikom PHP. Njegovo generalno mišljenje je da nije oduševljen samim programskim jezikom PHP, ali se frejmwork Laravel 5.5 pokazao kao krajnje solidan i mnogo je olakšao rad sa programskim jezikom PHP. Interesantno je da je potrebno iskucavati vrlo malo serverskog koda i modelskih klasa da bi aplikacija bila funkcionalna i da iako se koristi upravljanje bazom podataka, sam Laravel frejmwork obezbeđuje opnu tako da nema mnogo dodira sa SQL jezikom, što je pogodno za programere koji se bave serverskim programiranjem, i to delom koji je na sredini, između prezentacionog sloja i sloja podataka.

8 Literatura i korišćenji web-sajtovi

- [1] Filipović P., Razvoj Web aplikacija, (2003).
- [2] Vaughan T., Majstor za multimediju, (2002).
- [3] Williams E. H., Web aplikacije i baze podataka, (2003).
- [4] Cordoso J., Semantic Web Services, Processes and Applications, (2006).
- [5] Cedeno N., Internet i njegovi alati, (1996).
- [6] <https://itsolutionstuff.com/>
- [7] Milutinović V., Infrastructure for Electronic Business on the Internet (2001).