

**AKADEMIJA TEHNIČKO-UMETNIČKIH STRUKOVNIH
STUDIJA BEOGRAD
ODSEK VISOKA ŠKOLA ZA INFORMACIONE I
KOMUNIKACIONE TEHNOLOGIJE**

Web aplikacija „ELEKTRONSKI DNEVNIK”
ZAVRŠNI RAD

Mentor:

Dr Nenad Kojić, dipl. inž.

Student:

Aleksandra Marković 8/18

Beograd, 2021.

**AKADEMIJA TEHNIČKO-UMETNIČKIH STRUKOVNIH
STUDIJA BEOGRAD**

**ODSEK VISOKA ŠKOLA ZA INFORMACIONE I
KOMUNIKACIONE TEHNOLOGIJE**

INTERNET TEHNOLOGIJE

Predmet: Web programiranje - PHP 2

Tema: **Web aplikacija „ELEKTRONSKI DNEVNIK”**

Ocena ____ ()

Članovi komisije:

Sadržaj

1. Uvod	5
2. Radno okruženje	6
3. Organizacija	7
3.1. Stranice	8
Prijava	8
Početna stranica	8
Odeljenje	9
Dnevnik	9
Učenik	10
Dodaj čas	11
Smerovi, predmeti i nastavnici	11
Dodajte novo odeljenje	12
Registracija korisnika	12
Dodajte novog učenika	13
Roditelj	13
Profil	14
4. Kodovi	15
4.1. Controllers	15
OsnovniController	15
CasController	15
DnevnikController	16
LoginController	18
NastavniciController	19
OdeljenjaController	19
PocetnaController	21
PredmetiController	21
ProfilController	22
RegistracijaController	24
RoditeljController	25
SmeroviController	26
UceniciController	27
ZahteviController	31
4.2. Models	32
Cas.php	32
Korisnik.php	33

Menu.php	34
Odeljenje.php	34
Predmet.php	35
Profil.php	36
Roditelj.php	36
Smer.php	38
Ucenik.php	38
Zahtev.php	40
4.3. Views.....	41
Fixed	41
Layouts	44
Pages	44
Pagination	71
Partials	71
4.4. Routes.....	74
Web.php.....	74
4.5. JS	76
Main.js	76
4.6. CSS.....	103
Style.css	103
Responsive.css	115
5. Baza	121
6. Zaključak	122
7. Literatura.....	123

1. Uvod

Web aplikacija „Elektronski dnevnik” rađena u Laravelu, namenjena je nastavnicima u školi za upis časova i ocenjivanje učenika i roditeljima za pregled ocena svoje dece.

Administrator može videti sva odeljenja, predmete, ocene, nastavnike. On unosi korisnike generiše njihova korisnička imena i inicijalne šifre, može unositi nove smerove i predmete. Takođe može da unosi i učenike, da dodeljuje učenike odeljenjima, odeljenskim starešinama, da im dodeljuje predmete. Može vršiti izmenu i brisanje svega navedenog.

Direktor ima pregled svih učenika, predmeta, odeljenja, nastavnika i funkciju izmene pogrešno unete ocene koju prosleđuje nastavnik.

Nastavnik vidi samo svoje predmete i odeljenja na tim predmetima. On unosi ocene, napomene, zaključuje ocene, upisuje časove i odsutne i može poslati zahtev za promenu pogrešno unete ocene.

Odeljenski starešina pored svega navedenog što vrši nastavnik takođe vrši administraciju svog odeljenja. Može da unosi učenike, da im dodeljuje predmete, reguliše izostanke, vidi sve njihove ocene, daje ocenu iz vladanja. Takođe, on dodaje kontakte učenika (roditelje) i kreira njihove naloge.

Roditelj može pristupiti svom nalogu i pregledati ocene i izostanke svog deteta.

Takođe svi korisnici mogu izmeniti svoj profil (korisničko ime, šifru i sliku).

2. Radno okruženje

Izrada koda:

- PhpStorm

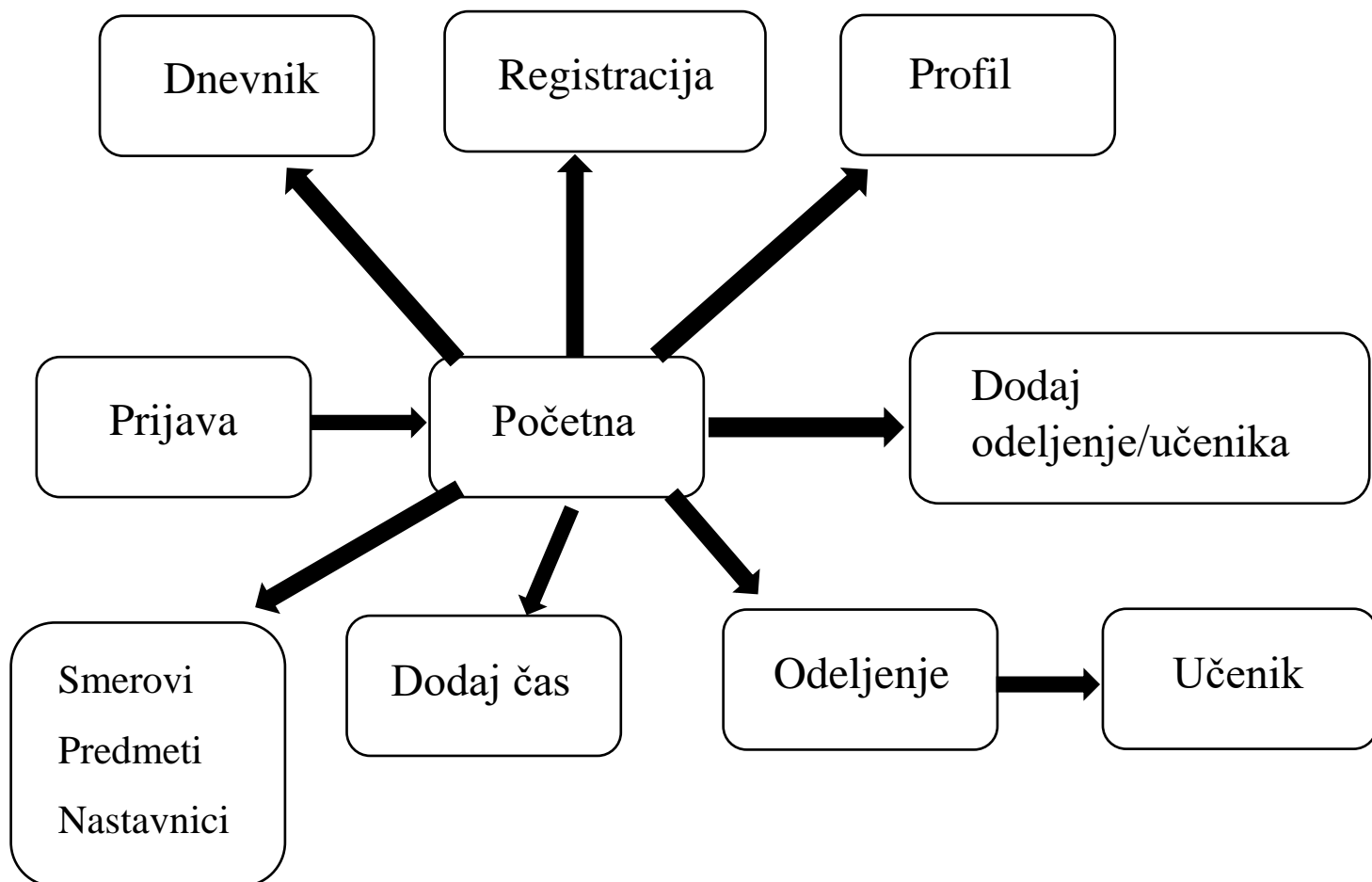
Jezici:

- HTML
- CSS
- PHP – Laravel Framework
- JavaScript
- JQuery

Dodaci:

- FontAwesome
- Google Fonts
- Bootstrap
- Charts.js

3. Organizacija



3.1. Stranice

Prijava



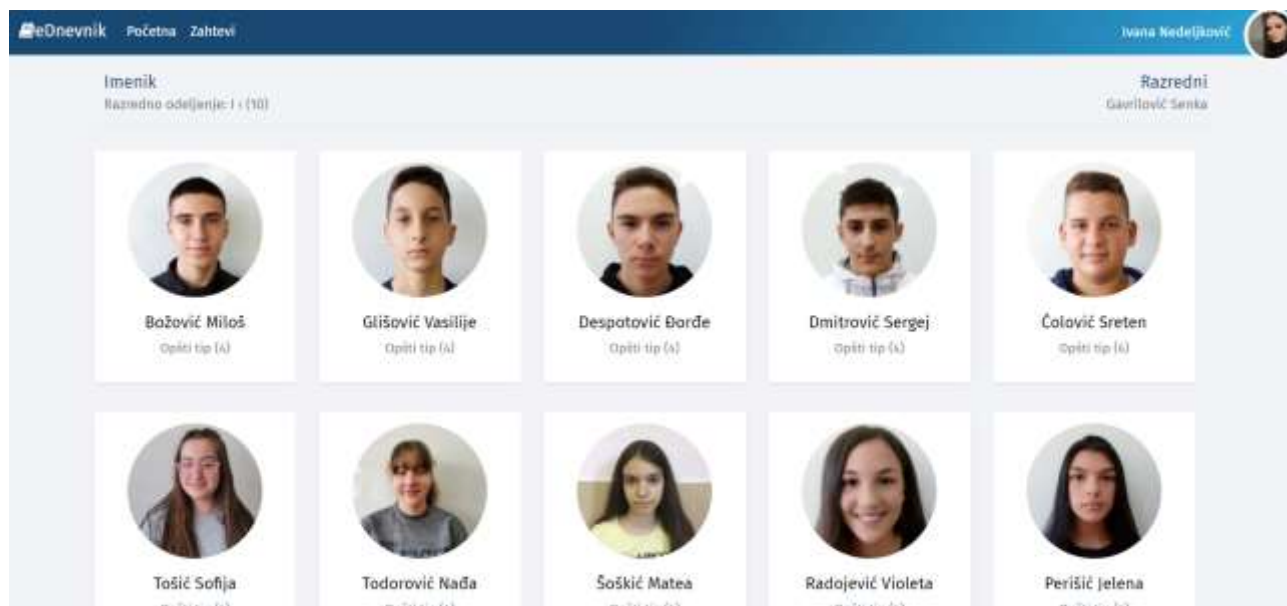
Stranica za prijavu korisnika. Korisnik može biti admin, razredni starešina, nastavnik, direktor i roditelj. Sa ove stranice se može redirektovati na početnu stranicu ili stranicu za roditelja.

Početna stranica



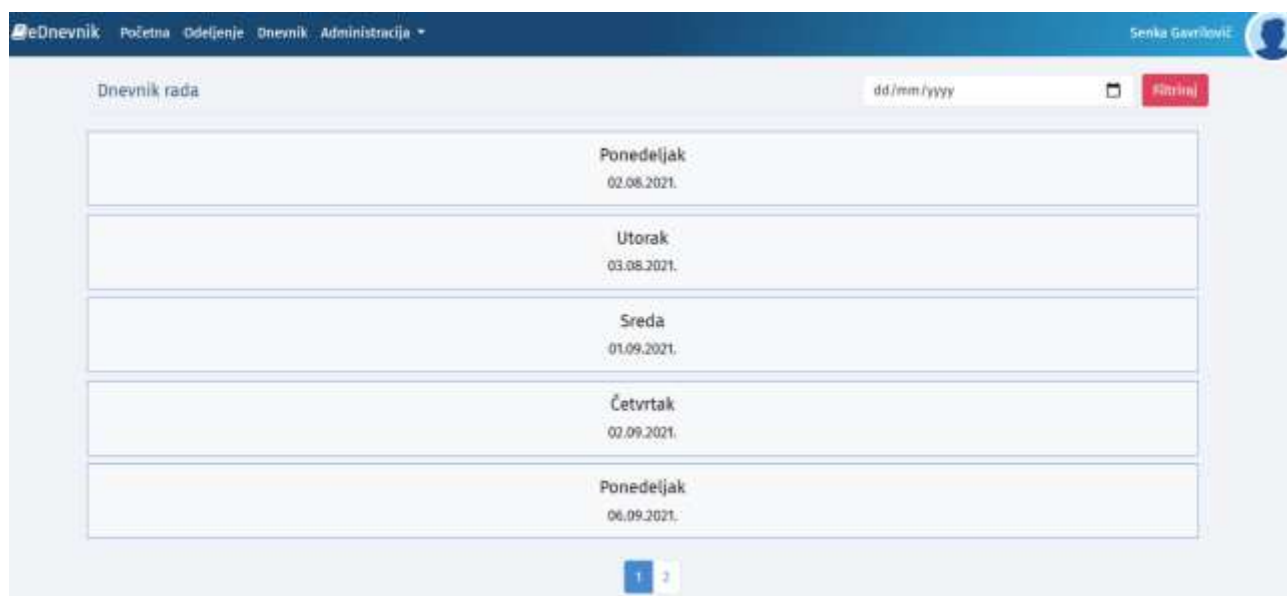
Na početnoj stranici se prikazuje sadržaj u zavisnosti od uloge. Direktor može videti grafike i sva odeljenja, administrator sva odeljenja, nastavnici samo odeljenja kojima predaju.

Odeljenje



Na ovoj stranici se vidi izabrano odeljenje i učenici u tom odeljenju. Klikom na učenika prelazi se na stranicu za određenog učenika.

Dnevnik



Na ovoj stranici se mogu videti svi upisani časovi za određeno odeljenje. Kada se klikne na datum prikažu se svi časovi za taj dan. Nastavnik može izmeniti i brisati časove koje je on upisao. Takođe datumi se mogu filtrirati.

Učenik

The screenshot displays the 'Učenik' (Student) profile page in the 'eDnevnik' system. The page is divided into several sections:

- Header:** Includes navigation links like 'Početna', 'Odešje', 'Dnevnik', and 'Administracija'. The user 'Senka Gavrilović' is logged in.
- Profile Section:** Shows the student's name 'Božović Miloš', a photo, and a button to 'Izmeni vladanje' (Change behavior).
- Grades and Absences Section:** Displays the student's grades for 'Matematika' (Mathematics) and 'Izostanci' (Absences). The average grade for the first semester is 4.5, and for the second semester, it is 0. The number of absences is 3 (3 irregular, 0 justified, 0 unjustified, 3 total).
- Subject Selection Section:** Allows the student to select subjects for the first, second, and third semesters. The subjects listed are 'Matematika', 'Računarstvo i Informatika', and 'Srpski jezik i književnost'.

Copyright © Aleksandra Marković 2021. In Dokumentacija

Na stranici za učenika mogu se videti njegove ocene i izostanci. Admin, razredni starešina i direktor mogu videti ocene iz svih predmeta, dok nastavnik može videti samo ocene iz svog predmeta. Razredni starešina može izmeniti ocenu iz vladanja i regulisati izostanke, takođe on kao i nastavnik mogu dodavati nove ocene i poslati zahtev za promenu pogrešno unete ocene.

U drugom tabu mogu se videti svi predmeti učenika i nastavnici.

U trećem tabu vide se podaci o učeniku i njegovim kontaktima. Razredni starešina i admin mogu dodavati nove kontakte i brisati ih.

Dodaj čas

The screenshot shows the 'Dodaj čas' (Add class) form in the eDnevnik application. The form is titled 'Upišite čas' (Enter class). It contains the following fields and options:

- Nastava jedinica:** A text input field.
- Datum:** A date picker showing '08/09/2021'.
- Redni broj časa:** A text input field.
- Napomena:** A text area for notes.
- Odustni:** A list of checkboxes for selecting teachers. The list includes: Miloš Božović, Vasilije Gilićević, Borđe Despotović, Sergej Dmitrović, Sreten Čolović, Sofija Tošić, Nada Todorović, Matea Šolčić, Violeta Radojević, and Jelena Perišić.

At the bottom of the form is a red button labeled 'Završi' (Finish). The footer of the page shows 'Copyright © Aleksandra Marković 2021' and a link to 'in Dokumentacija'.

Forma za upis časa za određeno odeljenje.

Smerovi, predmeti i nastavnici

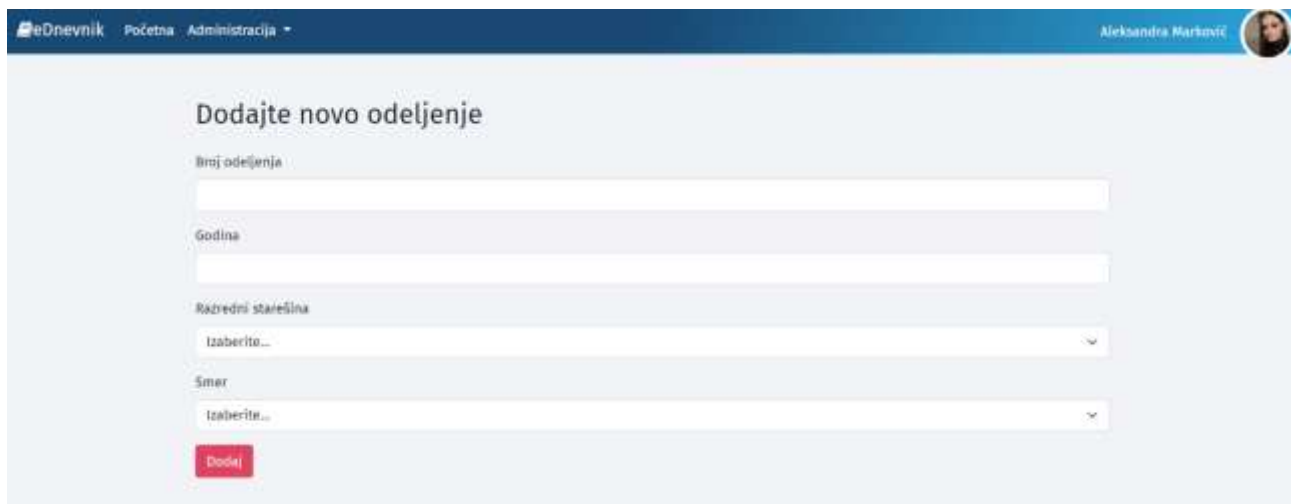
The screenshot shows the 'Dodaj smer' (Add subject) form in the eDnevnik application. The form is titled 'Dodaj smer' (Add subject). It contains the following fields and options:

- Naziv smera:** A text input field.
- Opšti tip:** A dropdown menu with a red button labeled 'Završi' (Finish).
- Opšti tip - Engleski jezik:** A dropdown menu with a red button labeled 'Završi' (Finish).

At the bottom of the form is a red button labeled 'Završi' (Finish). The footer of the page shows 'Copyright © Aleksandra Marković 2021' and a link to 'in Dokumentacija'.

Na ovim stranicama admin može videti sve smerove, predmete i nastavnike. Može dodavati nove smerove i predmete i brisati ih. Takođe može brisati nastavnike.

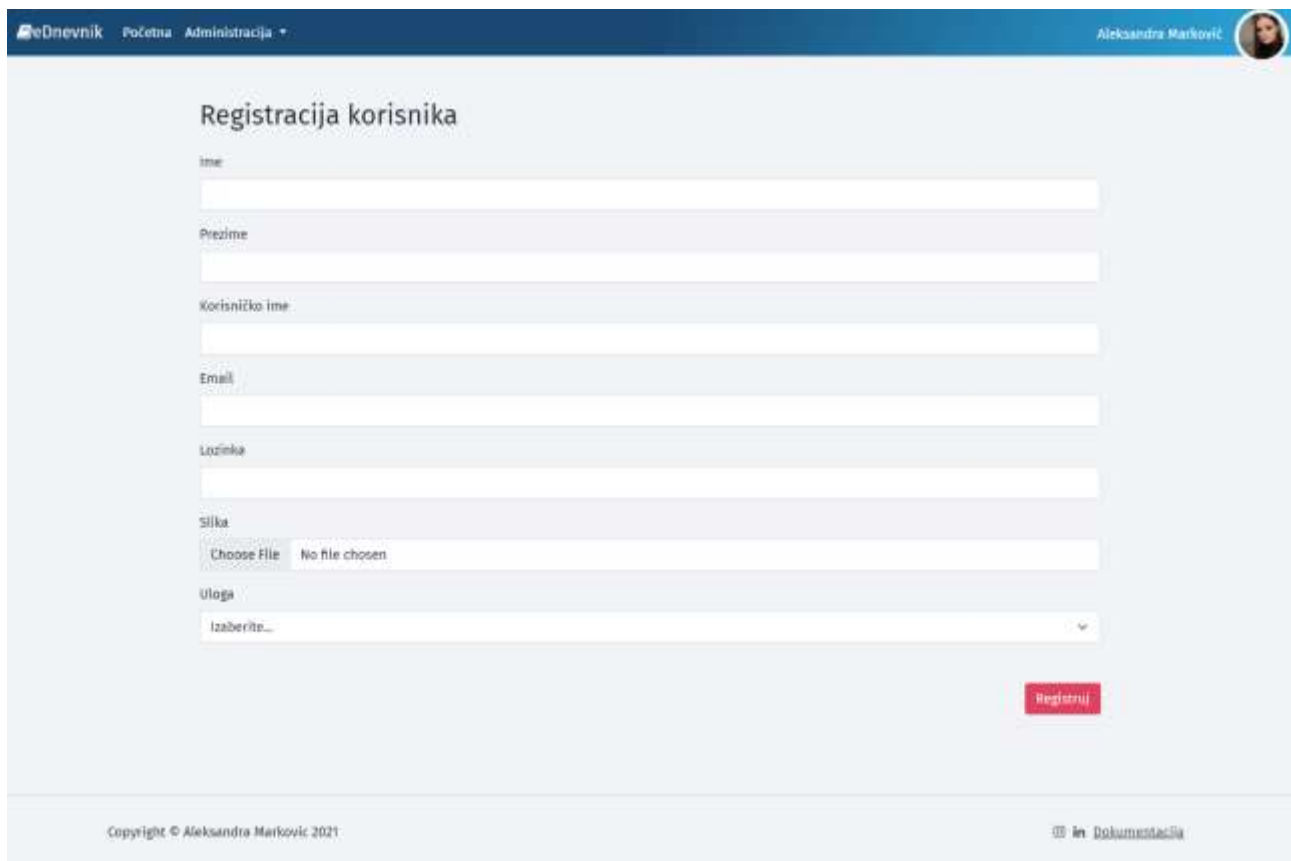
Dodajte novo odeljenje



The screenshot shows the 'Dodajte novo odeljenje' (Add new department) form in the eDnevnik admin interface. The form is located on a light blue background. At the top, there is a dark blue header with the eDnevnik logo, navigation links 'Početna' and 'Administracija', and a user profile for 'Aleksandra Marković'. The form itself has a title 'Dodajte novo odeljenje' and several input fields: 'Broj odeljenja' (Department number), 'Godina' (Year), 'Razredni stalešina' (Class teacher) with a dropdown menu, and 'Smer' (Direction) with a dropdown menu. A red 'Dodaj' (Add) button is at the bottom left of the form.

Forma za dodavanje novog odeljenja koju vidi samo admin. Kada izabere smer prikazuju se svi predmeti na tom smeru i on onda dodeljuje predmete odeljenju.

Registracija korisnika



The screenshot shows the 'Registracija korisnika' (User registration) form in the eDnevnik admin interface. The form is located on a light blue background. At the top, there is a dark blue header with the eDnevnik logo, navigation links 'Početna' and 'Administracija', and a user profile for 'Aleksandra Marković'. The form has a title 'Registracija korisnika' and several input fields: 'Ime' (Name), 'Prezime' (Surname), 'Korisničko ime' (Username), 'Email', 'Lozinka' (Password), 'Slika' (Image) with a 'Choose File' button and 'No file chosen' text, and 'Uloga' (Role) with a dropdown menu. A red 'Registruj' (Register) button is at the bottom right of the form. At the bottom of the page, there is a footer with 'Copyright © Aleksandra Markovic 2021' and a link to 'in Dokumentacija'.

Stranica za registraciju novih korisnika.

Dodajte novog učenika

The screenshot shows the 'Dodajte novog učenika' form in the eDnevnik application. The form is located in the center of the page, with a blue header bar at the top containing the application name and navigation links. The form fields are arranged vertically and include:

- Ime (Name): A text input field.
- Prezime (Surname): A text input field.
- Slika (Photo): A file upload area with a 'Choose File' button and a 'No file chosen' message.
- Matični broj (MBOG) (Matriculation Number): A text input field.
- Adresa (Address): A text input field.
- Mesto rođenja (Place of Birth): A text input field.
- Datum rođenja (Date of Birth): A date picker with a calendar icon.
- Dodaj (Add): A red button at the bottom of the form.

At the bottom of the page, there is a copyright notice: 'Copyright © Aleksandra Marković 2021.' and a link to 'Dokumentacija' (Documentation).

Stranica za dodavanje učenika kojoj mogu pristupiti admin i razredni starešina.

Roditelj

The screenshot shows the 'Roditelj' (Parent) view in the eDnevnik application. The view is for a student named 'Miloš Božović'. The header bar at the top contains the application name and navigation links. The main content area is divided into two sections: 'Ocene i izostanci' (Grades and Absences) and 'Predmeti' (Subjects). The 'Ocene i izostanci' section is active and displays the student's profile picture, name, and grade type. Below the profile picture, there is a table showing the student's grades and absences. The table has four columns: 'Ocene iz vladanja' (Grades from Behavior), 'neregulirani' (Irregular), 'opravdani' (Justified), and 'neopravdani' (Unjustified). The values in the table are: 3, 0, 0, and 3 respectively. The 'Predmeti' section is inactive and shows a dropdown menu for selecting a subject.

Ocene iz vladanja	neregulirani	opravdani	neopravdani	ukupno
3	0	0	3	

Stranica na kojoj roditelj vidi informacije o svom detetu. Ako ime više dece u školi može ih birati u padajućoj listi.


Profil

eDnevnik

Početna

Administracija

Aleksandra Marković



Aleksandra Marković

GLAVNE POSTAVKE

Korisničko ime

aleksandra.markovic

Email

markovic74@gmail.com

Slika

Choose File

No file chosen

Izmeni

BEZBEDNOST

Lozinka

Izmeni lozinku

Profil korisnika koji se može izmeniti.

4. Kodovi

4.1. Controllers

OsnovniController

```
<?php

namespace App\Http\Controllers;

use App\Models\Menu;
use Illuminate\Http\Request;

class OsnovniController extends Controller
{
    public $data;
    private $meniModel;

    public function __construct() {
        $this->meniModel = new Menu();
        $this->data["meni"] = $this->meniModel->dohvatiMeni();
    }
}
```

CasController

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\CasRequest;
use App\Models\Odeljenje;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class CasController extends OsnovniController
{
    private $odeljenjeModel;

    public function index($id) {
        $this->odeljenjeModel = new Odeljenje();
        $this->data['ucenici'] = $this->odeljenjeModel->dohvatiOdeljenje($id);
        return view('pages.main.cas', $this->data);
    }

    public function insert(CasRequest $request) {
        try {
            $predmet = \DB::table('korisnik_predmet')->select("id_predmet")
                ->where("id_korisnik", session('korisnik')->id_korisnik)
                ->first();

            $insert = \DB::table('cas')->insertGetId([
                'opis' => $request->input('opis'),
                'datum' => $request->input('datum'),
            ]);
        } catch (\Exception $e) {
            Log::error($e->getMessage());
        }
    }
}
```

```

        'po_redu' => $request->input('brojCasa'),
        'napomena' => $request->input('napomena'),
        'id_korisnik' => session('korisnik')->id_korisnik,
        'id_predmet' => $predmet->id_predmet,
        'id_odeljenje' => $request->input('id')
    ]);

    if($insert){
        if($request->input('odsutniNiz') != null){
            for($i=1; $i<=count($request->input('odsutniNiz')); $i++){
                \DB::table('odsutni')->insert([
                    'id_cas' => $insert,
                    'id_ucenik' => $i
                ]);
            }
        }
        return "Uspešno upisan čas!";
    }
}
catch(\Exception $e){
    Log::error($e->getMessage());
}
}
}

```

DnevnikController

```

<?php

namespace App\Http\Controllers;

use App\Models\Cas;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class DnevnikController extends OsnovniController
{
    private $casModel;

    public function index($id){
        $this->casModel = new Cas();
        $this->data['datumi'] = $this->casModel->dohvatiDatume($id);
        return view('pages.main.dnevnik', $this->data);
    }

    public function paginacija($id){
        $this->casModel = new Cas();
        $datumi = $this->casModel->dohvatiDatume($id);
        return response()->json($datumi);
    }

    public function filterDatum(Request $request, $id){
        try {
            $this->casModel = new Cas();
            $datumi = $this->casModel->filterDatum($request->input('datum'),
            $id);

            return response()->json($datumi);
        }
        catch(\Exception $e){

```



```

        Log::error($e->getMessage());
    }
}

public function filter(Request $request){
    try {
        $this->casModel = new Cas();
        $casovi = $this->casModel->filter($request->input('datum'));
        return response()->json($casovi);
    }
    catch(\Exception $e){
        Log::error($e->getMessage());
    }
}

public function odustni(Request $request){
    try {
        $this->casModel = new Cas();
        $odsutni = $this->casModel->dohvatiOdsutne($request->input('id'));
        return response()->json($odsutni);
    }
    catch(\Exception $e){
        Log::error($e->getMessage());
    }
}

public function obrisi(Request $request){
    try {
        \DB::table('cas')->where('id_cas', $request->input('id'))->delete();
        return "Uspešno obrisan čas.";
    }
    catch(\Exception $e){
        Log::error($e->getMessage());
    }
}

public function edit(Request $request){
    $this->casModel = new Cas();
    $info = $this->casModel->dohvatiZaCas($request->input('id'));
    return response()->json($info);
}

public function update(Request $request){
    try {
        \DB::table('cas')->where('id_cas', $request->input('id'))->update([
            'opis' => $request->input('opis'),
            'napomena' => $request->input('napomena')
        ]);

        return "Uspešno izmenjen čas.";
    }
    catch(\Exception $e){
        Log::error($e->getMessage());
    }
}
}

```

LoginController

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\LoginRequest;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class LoginController extends OsnovniController
{
    public function index(){
        return view("pages.main.login");
    }

    public function login(LoginRequest $request){
        $kor_ime = $request->input('kor_ime');
        $sifra = md5($request->input('sifra'));

        try {
            $korisnik = \DB::table('korisnici')
                ->where('kor_ime', $kor_ime)
                ->where('sifra', $sifra)
                ->first();

            if($korisnik) {
                $request->session()->put('korisnik', $korisnik);

                if(session('korisnik')->id_uloga == 2){
                    $odeljenje = \DB::table('odeljenje')->select('id_odeljenje
as id_odeljenje')
                        ->where('id_korisnik', session('korisnik')->id_korisnik)
                        ->first();

                    $request->session()->put('odeljenje', $odeljenje);

                    $predmet = \DB::table('korisnik_predmet')->
>select('korisnik_predmet.id_predmet as id_predmet', 'predmet.naziv as naziv')
                        ->join('predmet', 'predmet.id_predmet', '=',
'korisnik_predmet.id_predmet')
                        ->where('korisnik_predmet.id_korisnik',
session('korisnik')->id_korisnik)
                        ->first();

                    $request->session()->put('predmet', $predmet);
                }

                if(session('korisnik')->id_uloga == 4){
                    return "/roditelj";
                }
                else {
                    return "/home";
                }
            }
            else {
                return response(['error'=>true,'errorMsg'=>"Proverite podatke i
pokušajte ponovo."],400);
            }
        }
        catch (\Exception $e){
            Log::error($e->getMessage());
        }
    }
}
```

```

    }
}

public function logout(Request $request){
    if(session('korisnik')->id_uloga == 2){
        $request->session()->pull("odeljenje");
    }
    $request->session()->pull("korisnik");
    return redirect()->route("loginPage");
}
}

```

NastavniciController

```

<?php

namespace App\Http\Controllers;

use App\Models\Korisnik;
use App\Models\Predmet;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class NastavniciController extends OsnovniController
{
    private $nastavniciModel;

    public function index(){
        $this->nastavniciModel = new Korisnik();
        $this->data['nastavnici'] = $this->nastavniciModel->dohvatiNastavnike();
        return view('pages.admin.nastavnici', $this->data);
    }

    public function delete(Request $request){
        try {
            \DB::table('korisnici')->where('id_korisnik', $request->input('id'))->delete();
            $this->nastavniciModel = new Korisnik();
            $nastavnici = $this->nastavniciModel->dohvatiNastavnike();
            return response()->json($nastavnici);
        }
        catch(\Exception $e){
            Log::error($e->getMessage());
        }
    }
}

```

OdeljenjaController

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\OdeljenjeRequest;
use App\Models\Korisnik;
use App\Models\Odeljenje;
use App\Models\Predmet;
use App\Models\Smer;
use Illuminate\Http\Request;

```

```

use Illuminate\Support\Facades\Log;

class OdeljenjaController extends OsnovniController
{
    private $staresinaModel;
    private $smerModel;
    private $predmetModel;
    private $odeljenjeModel;

    public function index(){
        $this->staresinaModel = new Korisnik();
        $this->data['korisnici'] = $this->staresinaModel->dohvatiStaresine();
        $this->smerModel = new Smer();
        $this->data['smerovi'] = $this->smerModel->dohvatiSmerove();
        return view('pages.admin.odeljenja', $this->data);
    }

    public function show($id){
        $this->odeljenjeModel = new Odeljenje();
        $this->data['ucenici'] = $this->odeljenjeModel->dohvatiOdeljenje($id);
        $this->data['podaci'] = $this->odeljenjeModel->dohvatiRazrednog($id);
        return view('pages.ucenici.odeljenje', $this->data);
    }

    public function filterPredmeti(Request $request){
        $this->predmetModel = new Predmet();
        $predmeti = $this->predmetModel->predmetiSmer($request->input('id'));
        return response()->json($predmeti);
    }

    public function insert(OdeljenjeRequest $request){
        try {
            $insert = \DB::table('odeljenje')->insertGetId([
                'broj_odeljenja' => $request->input('broj'),
                'godina' => $request->input('godina'),
                'id_korisnik' => $request->input('razredni'),
                'id_smer' => $request->input('smer')
            ]);

            if($insert) {
                foreach ($request->input('predmeti') as $p) {
                    \DB::table('odeljenje_predmet')->insert([
                        'id_odeljenje' => $insert,
                        'id_predmet' => explode('|', $p)[0],
                        'id_korisnik' => explode('|', $p)[1]
                    ]);
                }
            }

            return "Uspešno dodato odeljenje!";
        } catch (\Exception $e) {
            Log::error($e->getMessage());
        }
    }
}

```

PocetnaController

```
<?php

namespace App\Http\Controllers;

use App\Models\Odeljenje;
use Illuminate\Http\Request;

class PocetnaController extends OsnovniController
{
    private $odeljenjeModel;

    public function index(){
        $this->odeljenjeModel = new Odeljenje();
        if(session('korisnik')->id_uloga == 1 || session('korisnik')->id_uloga
== 5) {
            $this->data['podaci'] = $this->odeljenjeModel-
>dohvatiOdeljenjaPocetna();
        }
        else if(session('korisnik')->id_uloga == 2 || session('korisnik')->id_uloga == 3) {
            $this->data['podaci'] = $this->odeljenjeModel-
>dohvatiOdeljenjaProfesor(session('korisnik')->id_korisnik);
        }

        return view("pages.main.home", $this->data);
    }

    public function direktor(){
        $opravdano = \DB::table('odsutni')->where("opravdano", 1)->get();
        $neopravdano = \DB::table('odsutni')->where("opravdano", 0)->get();
        $neregulisani = \DB::table('odsutni')->where("opravdano", null)->get();
        $odobreni = \DB::table('zahtev')->where('odobreno', 1)->get();
        $neodobreni = \DB::table('zahtev')->where('odobreno', 0)->get();
        return response()->json(
            [
                "opravdano" => $opravdano,
                "neopravdano" => $neopravdano,
                "neregulisani" => $neregulisani,
                "odobreni" => $odobreni,
                "neodobreni" => $neodobreni
            ]
        );
    }
}
```

PredmetiController

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\PredmetRequest;
use App\Models\Predmet;
use App\Models\Smer;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;
```

```

class PredmetiController extends OsnovniController
{
    private $predmetModel;
    private $smerModel;

    public function index(){
        $this->predmetModel = new Predmet();
        $this->data['podaci'] = $this->predmetModel->dohvatiPredmete();
        $this->smerModel = new Smer();
        $this->data['smerovi'] = $this->smerModel->dohvatiSmerove();
        return view('pages.admin.predmeti', $this->data);
    }

    public function insert(PredmetRequest $request){
        try {
            $insert = \DB::table('predmet')->insertGetId([
                'naziv' => $request->input('naziv')
            ]);

            if($insert){
                for($i=1; $i<=count($request->input('smerovi')); $i++){
                    \DB::table('smer_predmet')->insert([
                        'id_predmet' => $insert,
                        'id_smer' => $i
                    ]);
                }

                $this->predmetModel = new Predmet();
                $predmeti = $this->predmetModel->dohvatiPredmete();
                return response()->json($predmeti);
            }
        }
        catch(\Exception $e){
            Log::error($e->getMessage());
        }
    }

    public function delete(Request $request){
        try {
            \DB::table('predmet')->where('id_predmet', $request->input('id'))->delete();

            $this->predmetModel = new Predmet();
            $predmeti = $this->predmetModel->dohvatiPredmete();
            return response()->json($predmeti);
        }
        catch(\Exception $e){
            Log::error($e->getMessage());
        }
    }
}

```

ProfilController

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\LoznikaRequest;
use App\Http\Requests\ProfilRequest;
use App\Models\Profil;

```

```

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class ProfilController extends OsnovniController
{
    private $profilModel;

    public function index(){
        $this->profilModel = new Profil();
        $this->data['profil'] = $this->profilModel->dohvatiProfil(session('korisnik')->id_korisnik);
        return view("pages.main.profil", $this->data);
    }

    public function updateProfil(ProfilRequest $request){
        try {
            move_uploaded_file($_FILES['slikaProfil']['tmp_name'],
                public_path().'/assets/images/'.time().'_'. $_FILES['slikaProfil']['name']);
            $imageName = time().'_'. $_FILES['slikaProfil']['name'];
            \DB::table("korisnici")->where("id_korisnik", session('korisnik')->id_korisnik)->update([
                'kor_ime' => $request->input('kor_imeProfil'),
                'email' => $request->input('emailProfil'),
                'slika' => $imageName,
                'telefon' => $request->input('brojTelefona')
            ]);

            return redirect()->route('profil')->with('success', 'Uspešno ste izmenili profil!');
        }
        catch (\Exception $e){
            Log::error($e->getMessage());
        }
    }

    public function izmeniLozinku(LoznikaRequest $request){
        $lozinka = \DB::table("korisnici")->where('id_korisnik', session('korisnik')->id_korisnik)->select("sifra")->first();

        if($lozinka->sifra == md5($request->input('staraLozinka'))){
            \DB::table("korisnici")->where("id_korisnik", session('korisnik')->id_korisnik)->update([
                'sifra' => md5($request->input('ponovoLozinka'))
            ]);

            return redirect()->route('profil')->with('success', 'Uspešno ste izmenili lozinku!');
        }
        else {
            return redirect()->route('profil')->with('error', 'Pogrešna stara lozinka!');
        }
    }
}

```

RegistracijaController

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\RegistracijaRequest;
use App\Http\Requests\RoditeljRequest;
use App\Models\Predmet;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class RegistracijaController extends OsnovniController
{
    private $predmetiModel;

    public function index(){
        $this->predmetiModel = new Predmet();
        $this->data['predmeti'] = $this->predmetiModel->dohvatiPredmete();
        return view('pages.admin.registracija', $this->data);
    }

    public function insert(RegistracijaRequest $request){
        try {
            if(filesize( $request->file('slikaReg'))){
                move_uploaded_file($_FILES['slikaReg']['tmp_name'],
                public_path().'/assets/images/'.time() . $_FILES['slikaReg']['name']);
                $imageName = time() . $_FILES['slikaReg']['name'];
            }
            else {
                $imageName = 'user.png';
            }

            $insert = \DB::table('korisnici')->insertGetId([
                'ime' => $request->input('imeRegistracija'),
                'prezime' => $request->input('prezimeRegistracija'),
                'kor_ime' => $request->input('korImeReg'),
                'email' => $request->input('emailReg'),
                'sifra' => md5($request->input('sifraReg')),
                'slika' => $imageName,
                'id_uloga' => $request->input('ulogaReg'),
            ]);

            if($insert){
                \DB::table('korisnik_predmet')->insert([
                    'id_korisnik' => $insert,
                    'id_predmet' => $request->input('predmetReg')
                ]);

                return redirect()->route('registracijaForma')->with('success',
                'Uspešno ste registrovali korisnika!');
            }
            else {
                return redirect()->route('registracijaForma')->with('error',
                'Došlo je do greške, pokušajte ponovo!');
            }
        }
        catch (\Exception $e) {
            Log::error($e->getMessage());
        }
    }

    public function insertRoditelj(RoditeljRequest $request){
```



```

        try {
            $insert = \DB::table('korisnici')->insertGetId([
                'ime' => $request->input('ime'),
                'prezime' => $request->input('prezime'),
                'kor_ime' => $request->input('kor_ime'),
                'email' => $request->input('email'),
                'sifra' => md5($request->input('lozinka')),
                'slika' => 'user.png',
                'telefon' => $request->input('broj'),
                'id_uloga' => 4
            ]);

            if($insert){
                \DB::table('roditelj_ucenik')->insert([
                    'id_korisnik' => $insert,
                    'id_ucenik' => $request->input('id')
                ]);

                return "Uspešno dodat kontakt!";
            }
        } catch (\Exception $e) {
            Log::error($e->getMessage());
        }
    }
}

```

RoditeljController

```

<?php

namespace App\Http\Controllers;

use App\Models\Roditelj;
use Illuminate\Http\Request;

class RoditeljController extends OsnovniController
{
    private $roditeljModel;

    public function index(){
        $this->roditeljModel = new Roditelj();
        $this->data['ucenici'] = $this->roditeljModel->dohvatiUcenika();
        return view('pages.roditelj.roditelj', $this->data);
    }

    public function filter(Request $request){
        $this->roditeljModel = new Roditelj();
        $opravdano = $this->roditeljModel->dohvatiIzostanke($request->input('id'), 1);
        $neopravdano = $this->roditeljModel->dohvatiIzostanke($request->input('id'), 0);
        $neregulisani = $this->roditeljModel->dohvatiNeregulisane($request->input('id'));
        $predmeti = $this->roditeljModel->dohvatiPredmeteUcenika($request->input('id'));
        $ocene = $this->roditeljModel->dohvatiOceneAdmin($request->input('predmet'), $request->input('ucenik'));
        $prosek = $this->roditeljModel->dohvatiProsek($request->input('predmet'), $request->input('ucenik'), 1);
    }
}

```

```

        $prosek2 = $this->roditeljModel->dohvatiProsek($request->input('predmet'), $request->input('ucenik'), 2);
        $zakljucna1 = $this->roditeljModel->dohvatiZakljucnuAdmin($request->input('predmet'), $request->input('ucenik'), 1);
        $zakljucna2 = $this->roditeljModel->dohvatiZakljucnuAdmin($request->input('predmet'), $request->input('ucenik'), 2);
        $ucenik = $this->roditeljModel->dohvatiJednogUcenika($request->input('id'));
        return response()->json(
            [
                "opravdano" => $opravdano,
                "neopravdano" => $neopravdano,
                "neregulisani" => $neregulisani,
                "predmeti" => $predmeti,
                "ocene" => $ocene,
                "prosek" => $prosek,
                "prosek2" => $prosek2,
                "zakljucna1" => $zakljucna1,
                "zakljucna2" => $zakljucna2,
                "ucenik" => $ucenik
            ]
        );
    }
}

```

SmeroviController

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\SmerRequest;
use App\Models\Smer;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class SmeroviController extends OsnovniController
{
    private $smerModel;

    public function index(){
        $this->smerModel = new Smer();
        $this->data['podaci'] = $this->smerModel->dohvatiSmerove();
        return view('pages.admin.smerovi', $this->data);
    }

    public function insert(SmerRequest $request){
        try {
            $provera = \DB::table('smer')->where("naziv", $request->input('naziv'))->first();

            if($provera){
                return response(['error'=>true, 'errorMsg'=>"Smer već postoji!"], 400);
            }

            $insert = \DB::table('smer')->insert([
                'naziv' => $request->input('naziv')
            ]);
        }
    }
}

```

```

        if($insert){
            $this->smerModel = new Smer();
            $smerovi = $this->smerModel->dohvatiSmerove();
            return response()->json($smerovi);
        }
    }
    catch(\Exception $e){
        Log::error($e->getMessage());
    }
}

public function delete(Request $request){
    try {
        \DB::table('smer')->where('id_smer', $request->input('id'))->delete();
        $this->smerModel = new Smer();
        $smerovi = $this->smerModel->dohvatiSmerove();
        return response()->json($smerovi);
    }
    catch(\Exception $e){
        Log::error($e->getMessage());
    }
}
}

```

UceniciController

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\IzostanciRequest;
use App\Http\Requests\OcenaRequest;
use App\Http\Requests\OceniRequest;
use App\Http\Requests\UceniciRequest;
use App\Http\Requests\ZahtevRequest;
use App\Models\Odeljenje;
use App\Models\Ucenik;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class UceniciController extends OsnovniController
{
    private $odeljenjeModel;
    private $ucenikModel;

    public function index(){
        $this->odeljenjeModel = new Odeljenje();
        $this->data['odeljenja'] = $this->odeljenjeModel->dohvatiOdeljenja();
        return view('pages.admin.ucenici', $this->data);
    }

    public function show($id){
        $this->ucenikModel = new Ucenik();
        $this->data['ucenik'] = $this->ucenikModel->dohvatiUcenika($id);
        $this->data['predmeti'] = $this->ucenikModel->dohvatiPredmeteUcenika($id);
        $this->data['roditelji'] = $this->ucenikModel->dohvatiKontakte($id);
        $this->data['opravdani'] = $this->ucenikModel->dohvatiIzostanke($id, 1);
    }
}

```

```

        $this->data['neopravdani'] = $this->ucenikModel->dohvatiIzostanke($id,
0);
        $this->data['neregulisani'] = $this->ucenikModel-
>dohvatiNeregulisane($id);
        $this->data['ocene'] = $this->ucenikModel-
>dohvatiOcene(session('korisnik')->id_korisnik, $id);
        $this->data['vrsta'] = $this->ucenikModel->dohvatiVrstu();
        $this->data['prosekPrvo'] = $this->ucenikModel-
>prosekPolug(session('korisnik')->id_korisnik, 1, $id);
        $this->data['prosekDrugo'] = $this->ucenikModel-
>prosekPolug(session('korisnik')->id_korisnik, 2, $id);
        $this->data['zakljucna1'] = $this->ucenikModel-
>dohvatiZakljucnu(session('korisnik')->id_korisnik, $id, 1);
        $this->data['zakljucna2'] = $this->ucenikModel-
>dohvatiZakljucnu(session('korisnik')->id_korisnik, $id, 2);
        return view('pages.ucenici.ucenik', $this->data);
    }

    public function filter(Request $request){
        try {
            $this->ucenikModel = new Ucenik();
            $ocene = $this->ucenikModel->dohvatiOceneAdmin($request-
>input('predmet'), $request->input('id'));
            $prosek = $this->ucenikModel->dohvatiProsek($request-
>input('predmet'), $request->input('id'), 1);
            $prosek2 = $this->ucenikModel->dohvatiProsek($request-
>input('predmet'), $request->input('id'), 2);
            $zakljucna1 = $this->ucenikModel->dohvatiZakljucnuAdmin($request-
>input('predmet'), $request->input('id'), 1);
            $zakljucna2 = $this->ucenikModel->dohvatiZakljucnuAdmin($request-
>input('predmet'), $request->input('id'), 2);
            return response()->json(['ocene' => $ocene, 'prosek' => $prosek,
'prosek2' => $prosek2, 'zakljucna1' => $zakljucna1, 'zakljucna2' =>
$zakljucna2]);
        }
        catch (\Exception $e){
            Log::error($e->getMessage());
        }
    }

    public function insert(UceniciRequest $request){
        try {
            move_uploaded_file($_FILES['slikaUcenika']['tmp_name'],
public_path().'./assets/images/'.time() . $_FILES['slikaUcenika']['name']);
            $imageName = time() . $_FILES['slikaUcenika']['name'];

            if(session('korisnik')->id_uloga == 1){
                $id_odeljenje = $request->input('odeljenjeUcenika');
            }
            else if(session('korisnik')->id_uloga == 2){
                $id = \DB::table('odeljenje')->select('id_odeljenje')
->where('id_korisnik', session('korisnik')->id_korisnik)
->first();

                $id_odeljenje = $id->id_odeljenje;
            }

            $insert = \DB::table('ucenik')->insert([
                'ime' => $request->input('imeUcenika'),
                'prezime' => $request->input('prezimeUcenika'),
                'slika' => $imageName,
                'vladanje' => 5,
                'jmbg' => $request->input('jmbg'),
                'adresa' => $request->input('adresa'),

```

```

        'mesto_rodjenja' => $request->input('mestoRodj'),
        'datum_rodjena' => $request->input('datumRodj'),
        'id_odeljenje' => $id_odeljenje
    ]);

    if ($insert){
        return redirect()->route('ucenici')->with('success', 'Uspešno
ste dodali učenika!');
    }
}
catch (\Exception $e){
    Log::error($e->getMessage());
}
}

public function izostanci(Request $request){
    try {
        for($i=0; $i<count($request->input('nizId')); $i++){
            \DB::table('odsutni')->where('id_odsutni', $request-
>input('nizId')[$i])->update([
                'opravdano' => $request->input('nizOpravdano')[$i]
            ]);
        }

        return "Uspešno regulisan izostanak.";
    }
    catch (\Exception $e){
        Log::error($e->getMessage());
    }
}

public function oceni(OceniRequest $request){
    try {
        $polugodiste = \DB::table('polugodiste')->select('id_polugodiste')
            ->where('datum_od', '<=', Carbon::now())
            ->where('datum_do', '>=', Carbon::now())
            ->first();

        $predmet = \DB::table('korisnik_predmet')->select('id_predmet')
            ->where('id_korisnik', session('korisnik')->id_korisnik)
            ->first();

        $zakljucna = \DB::table('zakljucna_ocena')
            ->where('id_ucenik', $request->input('id'))
            ->where('id_predmet', $predmet->id_predmet)
            ->where('tip_ocene', 1)
            ->get();

        if(count($zakljucna)){
            $idpolugodiste = 2;
        }
        else {
            $idpolugodiste = $polugodiste->id_polugodiste;
        }

        $insert = \DB::table('ocene')->insert([
            'ocena' => $request->input('ocena'),
            'id_ucenik' => $request->input('id'),
            'id_polugodiste' => $idpolugodiste,
            'id_vrsta_ocene' => $request->input('vrsta'),
            'id_predmet' => $predmet->id_predmet,
            'datum' => Carbon::now()
        ]);
    }
}

```

```

        if($insert){
            return "Uspešno ste ocenili učenika.";
        }
    }
    catch (\Exception $e){
        Log::error($e->getMessage());
    }
}

public function izbrisi(Request $request){
    try {
        \DB::table('korisnici')->where('id_korisnik', $request->input('izbrisiRoditelj'))->delete();
        return \App::make('redirect')->back()->with("success", "Uspešno obrisani kontakt.");
    }
    catch (\Exception $e){
        Log::error($e->getMessage());
    }
}

public function zahtev(ZahtevRequest $request){
    try {
        $insert = \DB::table('zahtev')->insert([
            'pogresna_ocena' => $request->input('pogresna'),
            'nova_ocena' => $request->input('nova'),
            'datum_ocene' => $request->input('datum'),
            'id_ucenik' => $request->input('id'),
            'id_korisnik' => session('korisnik')->id_korisnik,
            'id_vrsta_ocene' => $request->input('vrsta'),
        ]);

        if($insert){
            return "Uspešno poslat zahtev!";
        }
    }
    catch (\Exception $e){
        Log::error($e->getMessage());
    }
}

public function zakljuciOcenu(OcenaRequest $request){
    try {
        $predmet = \DB::table('korisnik_predmet')->select('id_predmet')
            ->where('id_korisnik', session('korisnik')->id_korisnik)
            ->first();

        if($request->input('polugodiste') != 0){
            $polugodiste = $request->input('polugodiste');
        }
        else {
            $polugodiste = 2;
        }

        $insert = \DB::table('zakljucna_ocena')->insert([
            'zakljucna_ocena' => $request->input('ocena'),
            'tip_ocene' => $polugodiste,
            'id_ucenik' => $request->input('id'),
            'id_predmet' => $predmet->id_predmet
        ]);

        if($insert){
            return "Uspešno ste zaključili ocenu!";
        }
    }
}

```

```

    }
    catch (\Exception $e){
        Log::error($e->getMessage());
    }
}

public function vladanje(OcenaRequest $request){
    try {
        \DB::table('ucenik')->where('id_ucenik', $request->input('id'))
            ->update([
                'vladanje' => $request->input('ocena')
            ]);

        return "Uspešno izmenjena ocena iz vladanja!";
    }
    catch (\Exception $e){
        Log::error($e->getMessage());
    }
}
}

```

ZahteviController

```

<?php

namespace App\Http\Controllers;

use App\Models\Zahtev;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class ZahteviController extends OsnovniController
{
    private $zahteviModel;

    public function index(){
        $this->zahteviModel = new Zahtev();
        $this->data['zahtevi'] = $this->zahteviModel->dohvatiZahteve();
        return view('pages.direktor.zahtevi', $this->data);
    }

    public function odobri(Request $request){
        try {
            \DB::table('zahtev')->where('id_zahtev', $request->input('zahtev'))
                ->update([
                    'odobreno' => 1
                ]);

            \DB::table('ocene')
                ->where('datum', $request->input('datumZ'))
                ->where('id_ucenik', $request->input('ucenik'))
                ->where('id_vrsta_ocene', $request->input('vrsta'))
                ->update([
                    'ocena' => $request->input('nova')
                ]);

            return redirect()->route('zahtevi')->with('success', 'Zahtev
odobren!');
        }
    }
}

```

```

        catch (\Exception $e){
            Log::error($e->getMessage());
        }
    }

    public function odbij(Request $request){
        try {
            \DB::table('zahtev')->where('id_zahtev', $request->input('zahtev'))
                ->update([
                    'odobreno' => 0
                ]);

            return redirect()->route('zahtevi')->with('error', 'Zahtev odbijen.');
```

4.2. Models

Cas.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Cas extends Model
{
    public function dohvatiDatume($id){
        return \DB::table('cas')->select('datum')
            ->where('id_odeljenje', $id)
            ->groupBy('datum')
            ->paginate(5);
    }

    public function filterDatum($datum, $id){
        return \DB::table('cas')->select('datum')
            ->where('id_odeljenje', $id)
            ->where('datum', $datum)
            ->distinct()
            ->get();
    }

    public function filter($datum){
        return \DB::table('cas')->select('cas.*', 'predmet.naziv',
            'korisnici.ime', 'korisnici.prezime' )
            ->join('predmet', 'predmet.id_predmet', '=', 'cas.id_predmet')
            ->join('korisnici', 'korisnici.id_korisnik', '=', 'cas.id_korisnik')
            ->where('cas.datum', $datum)
            ->get();
    }
}
```



```

    }

    public function dohvatiOdsutne($id){
        return \DB::table('ucenik')->select('ucenik.*')
            ->join('odsutni', 'odsutni.id_ucenik', '=', 'ucenik.id_ucenik')
            ->where('odsutni.id_cas', $id)
            ->distinct()
            ->get();
    }

    public function dohvatiZaCas($id){
        return \DB::table('cas')->select('id_cas', 'opis', 'napomena')
            ->where('cas.id_cas', $id)
            ->get();
    }
}

```

Korisnik.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Korisnik extends Model
{
    public function dohvatiStaresine(){
        return \DB::table('korisnici')->select('korisnici.id_korisnik',
'korisnici.ime', 'korisnici.prezime')
            ->leftJoin('odeljenje', 'korisnici.id_korisnik', '=',
'odeljenje.id_korisnik')
            ->whereNotIn('korisnici.id_korisnik', function ($query){
                $query->select('id_korisnik')
                    ->from('odeljenje');
            })
            ->where('korisnici.id_uloga', 2)
            ->get();
    }

    public function dohvatiNastavnike(){
        return \DB::table('korisnici')->select('korisnici.*', 'predmet.naziv')
            ->leftJoin('korisnik_predmet', 'korisnik_predmet.id_korisnik', '=',
'korisnici.id_korisnik')
            ->leftJoin('predmet', 'korisnik_predmet.id_predmet', '=',
'predmet.id_predmet')
            ->where('korisnici.id_uloga', 2)
            ->orWhere('korisnici.id_uloga', 3)
            ->get();
    }
}

```

Menu.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Menu extends Model
{
    public function dohvatiMeni() {
        return \DB::table("meni")->get();
    }
}
```

Odeljenje.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Odeljenje extends Model
{
    public function dohvatiOdeljenja() {
        return \DB::table('odeljenje')->get();
    }

    public function dohvatiOdeljenje($id){
        return \DB::table('ucenik')->select('ucenik.id_ucenik', 'ucenik.ime',
'ucenik.prezime', 'ucenik.slika', 'smer.naziv')
->join('odeljenje', 'odeljenje.id_odeljenje', '=',
'ucenik.id_odeljenje')
->join('smer', 'smer.id_smer', '=', 'odeljenje.id_smer')
->where('ucenik.id_odeljenje', $id)
->get();
    }

    public function dohvatiRazrednog($id){
        return \DB::table('odeljenje')->select('odeljenje.broj_odeljenja',
'odeljenje.godina', 'korisnici.ime', 'korisnici.prezime')
->join('korisnici', 'korisnici.id_korisnik', '=',
'odeljenje.id_korisnik')
->where('odeljenje.id_odeljenje', $id)
->get();
    }

    public function dohvatiOdeljenjaPocetna(){
        return \DB::table('odeljenje')->select('odeljenje.*', 'korisnici.ime',
'korisnici.prezime')
->join('korisnici', 'korisnici.id_korisnik', '=',
'odeljenje.id_korisnik')
->orderBy('odeljenje.godina', 'asc')
->orderBy('odeljenje.broj_odeljenja', 'asc')
    }
}
```

```

        ->get();
    }

    public function dohvatiOdeljenjaProfesor($id){
        return \DB::table('odeljenje')->select('odeljenje.*', 'predmet.naziv')
            ->join('korisnici', 'korisnici.id_korisnik', '=',
'odeljenje.id_korisnik')
            ->join('odeljenje_predmet', 'odeljenje_predmet.id_odeljenje', '=',
'odeljenje.id_odeljenje')
            ->join('predmet', 'predmet.id_predmet', '=',
'odeljenje_predmet.id_predmet')
            ->where('odeljenje_predmet.id_korisnik', $id)
            ->orderBy('odeljenje.godina', 'asc')
            ->orderBy('odeljenje.broj_odeljenja', 'asc')
            ->get();
    }
}

```

Predmet.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Predmet extends Model
{
    public function dohvatiPredmete(){
        return \DB::table("predmet")->select('id_predmet as id', 'naziv')->
>get();
    }

    public function predmetiSmer($id){
        return \DB::table('predmet')->select('predmet.id_predmet',
'predmet.naziv', 'korisnici.id_korisnik', 'korisnici.ime', 'korisnici.prezime')
            ->join('smer_predmet', 'predmet.id_predmet', '=',
'smer_predmet.id_predmet')
            ->join('korisnik_predmet', 'predmet.id_predmet', '=',
'korisnik_predmet.id_predmet')
            ->join('korisnici', 'korisnici.id_korisnik', '=',
'korisnik_predmet.id_korisnik')
            ->where('smer_predmet.id_smer', $id)
            ->get();
    }
}

```

Profil.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Profil extends Model
{
    public function dohvatiProfil($id){
        return \DB::table("korisnici")->where("id_korisnik", $id)->get();
    }
}
```

Roditelj.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Roditelj extends Model
{
    public function dohvatiUcenika(){
        return \DB::table('ucenik')->select('ucenik.*', 'smer.naziv')
            ->join('odeljenje', 'odeljenje.id_odeljenje', '=',
'ucenik.id_odeljenje')
            ->join('smer', 'smer.id_smer', '=', 'odeljenje.id_smer')
            ->join('roditelj_ucenik', 'roditelj_ucenik.id_ucenik', '=',
'ucenik.id_ucenik')
            ->where('roditelj_ucenik.id_korisnik', session('korisnik')-
>id_korisnik)
            ->get();
    }

    public function dohvatiJednogUcenika($id){
        return \DB::table('ucenik')->select('ucenik.*', 'smer.naziv')
            ->join('odeljenje', 'odeljenje.id_odeljenje', '=',
'ucenik.id_odeljenje')
            ->join('smer', 'smer.id_smer', '=', 'odeljenje.id_smer')
            ->join('roditelj_ucenik', 'roditelj_ucenik.id_ucenik', '=',
'ucenik.id_ucenik')
            ->where('ucenik.id_ucenik', $id)
            ->get();
    }

    public function dohvatiPredmeteUcenika($id){
        return \DB::table('predmet')->select('predmet.*', 'korisnici.*',
'smer.naziv as smer')
            ->join('odeljenje_predmet', 'odeljenje_predmet.id_predmet', '=',
'predmet.id_predmet')
            ->join('odeljenje', 'odeljenje.id_odeljenje', '=',
'odeljenje_predmet.id_odeljenje')
            ->join('ucenik', 'ucenik.id_odeljenje', '=',
'odeljenje.id_odeljenje')
            ->join('smer', 'smer.id_smer', '=', 'odeljenje.id_smer')
```

```

        ->join('korisnici', 'korisnici.id_korisnik', '=',
'odeljenje_predmet.id_korisnik')
        ->where('ucenik.id_ucenik', $id)
        ->get();
    }

    public function dohvatiIzostanke($id, $opravdano){
        return \DB::table('odsutni')
            ->where('opravdano', $opravdano)
            ->where('id_ucenik', $id)
            ->get();
    }

    public function dohvatiNeregulisane($id){
        return \DB::table('odsutni')->select('odsutni.id_odsutni',
'cas.po_redu', 'cas.datum', 'predmet.naziv')
            ->join('cas', 'cas.id_cas', '=', 'odsutni.id_cas')
            ->join('predmet', 'predmet.id_predmet', '=', 'cas.id_predmet')
            ->where('opravdano', null)
            ->where('id_ucenik', $id)
            ->get();
    }

    public function dohvatiProsek($id, $idUcenik, $polugodiste){
        return \DB::table('ocene')->selectRaw('AVG(ocene.ocena) as
prosecnaOcena')
            ->join('vrsta_ocene', 'vrsta_ocene.id_vrsta_ocene', '=',
'ocene.id_vrsta_ocene')
            ->join('korisnik_predmet', 'korisnik_predmet.id_predmet', '=',
'ocene.id_predmet')
            ->where('ocene.id_predmet', $id)
            ->where('ocene.id_ucenik', $idUcenik)
            ->where('ocene.id_polugodiste', $polugodiste)
            ->get();
    }

    public function dohvatiOceneAdmin($id, $idUcenik){
        return \DB::table('ocene')->select('ocene.*', 'vrsta_ocene.vrsta')
            ->join('vrsta_ocene', 'vrsta_ocene.id_vrsta_ocene', '=',
'ocene.id_vrsta_ocene')
            ->where('ocene.id_predmet', $id)
            ->where('ocene.id_ucenik', $idUcenik)
            ->get();
    }

    public function dohvatiZakljucnuAdmin($id, $ucenik, $polugodiste){
        return \DB::table('zakljucna_ocena')->select("zakljucna_ocena.*")
            ->where('zakljucna_ocena.id_predmet', $id)
            ->where('zakljucna_ocena.id_ucenik', $ucenik)
            ->where('zakljucna_ocena.tip_ocene', $polugodiste)
            ->get();
    }
}

```

Smer.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Smer extends Model
{
    public function dohvatiSmerove() {
        return \DB::table('smer')->select("id_smer as id", 'naziv')->get();
    }
}
```

Ucenik.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Ucenik extends Model
{
    public function dohvatiUcenika($id){
        return \DB::table('ucenik')->select('ucenik.*', 'smer.naziv')
            ->join('odeljenje', 'odeljenje.id_odeljenje', '=',
'ucenik.id_odeljenje')
            ->join('smer', 'smer.id_smer', '=', 'odeljenje.id_smer')
            ->where('ucenik.id_ucenik', $id)
            ->get();
    }

    public function dohvatiPredmeteUcenika($id){
        return \DB::table('predmet')->select('predmet.*', 'korisnici.*')
            ->join('odeljenje_predmet', 'odeljenje_predmet.id_predmet', '=',
'predmet.id_predmet')
            ->join('odeljenje', 'odeljenje.id_odeljenje', '=',
'odeljenje_predmet.id_odeljenje')
            ->join('ucenik', 'ucenik.id_odeljenje', '=',
'odeljenje.id_odeljenje')
            ->join('korisnici', 'korisnici.id_korisnik', '=',
'odeljenje_predmet.id_korisnik')
            ->where('ucenik.id_ucenik', $id)
            ->get();
    }

    public function dohvatiKontakte($id){
        return \DB::table('korisnici')->select('korisnici.*')
            ->join('roditelj_ucenik', 'roditelj_ucenik.id_korisnik', '=',
'korisnici.id_korisnik')
            ->where('roditelj_ucenik.id_ucenik', $id)
            ->get();
    }

    public function dohvatiIzostanke($id, $opravdano){
        return \DB::table('odsutni')
```

```

        ->where('opravdano', $opravdano)
        ->where('id_ucenik', $id)
        ->get();
    }

    public function dohvatiNeregulisane($id){
        return \DB::table('odsutni')->select('odsutni.id_odsutni',
'cas.po_redu', 'cas.datum', 'predmet.naziv')
        ->join('cas', 'cas.id_cas', '=', 'odsutni.id_cas')
        ->join('predmet', 'predmet.id_predmet', '=', 'cas.id_predmet')
        ->where('opravdano', null)
        ->where('id_ucenik', $id)
        ->get();
    }

    public function dohvatiOceneAdmin($id, $idUcenik){
        return \DB::table('ocene')->select('ocene.*', 'vrsta_ocene.vrsta')
        ->join('vrsta_ocene', 'vrsta_ocene.id_vrsta_ocene', '=',
'ocene.id_vrsta_ocene')
        ->where('ocene.id_predmet', $id)
        ->where('ocene.id_ucenik', $idUcenik)
        ->get();
    }

    public function dohvatiProsek($id, $idUcenik, $polugodiste){
        return \DB::table('ocene')->selectRaw('AVG(ocene.ocena) as
prosecnaOcena')
        ->join('vrsta_ocene', 'vrsta_ocene.id_vrsta_ocene', '=',
'ocene.id_vrsta_ocene')
        ->join('korisnik_predmet', 'korisnik_predmet.id_predmet', '=',
'ocene.id_predmet')
        ->where('ocene.id_predmet', $id)
        ->where('ocene.id_ucenik', $idUcenik)
        ->where('ocene.id_polugodiste', $polugodiste)
        ->get();
    }

    public function dohvatiOcene($id, $ucenik){
        return \DB::table('ocene')->select('ocene.*', 'vrsta_ocene.vrsta')
        ->join('vrsta_ocene', 'vrsta_ocene.id_vrsta_ocene', '=',
'ocene.id_vrsta_ocene')
        ->join('korisnik_predmet', 'korisnik_predmet.id_predmet', '=',
'ocene.id_predmet')
        ->where('korisnik_predmet.id_korisnik', $id)
        ->where('ocene.id_ucenik', $ucenik)
        ->get();
    }

    public function dohvatiVrstu(){
        return \DB::table('vrsta_ocene')->get();
    }

    public function prosekPolug($id, $polugodiste, $ucenik){
        return \DB::table('ocene')->selectRaw('AVG(ocene.ocena) as
prosecnaOcena')
        ->join('vrsta ocene', 'vrsta ocene.id vrsta ocene', '=',
'ocene.id_vrsta ocene')
        ->join('korisnik_predmet', 'korisnik_predmet.id_predmet', '=',
'ocene.id_predmet')
        ->where('korisnik_predmet.id_korisnik', $id)
        ->where('ocene.id_ucenik', $ucenik)
        ->where('ocene.id_polugodiste', $polugodiste)
        ->get();
    }
}

```

```

    public function dohvatiZakljucnu($id, $ucenik, $polugodiste){
        return \DB::table('zakljucna_ocena')->select("zakljucna_ocena.*")
            ->join('korisnik_predmet', 'korisnik_predmet.id_predmet', '=',
'zakljucna_ocena.id_predmet')
            ->where('korisnik_predmet.id_korisnik', $id)
            ->where('zakljucna_ocena.id_ucenik', $ucenik)
            ->where('zakljucna_ocena.tip_ocene', $polugodiste)
            ->get();
    }

    public function dohvatiZakljucnuAdmin($id, $ucenik, $polugodiste){
        return \DB::table('zakljucna_ocena')->select("zakljucna_ocena.*")
            ->where('zakljucna_ocena.id_predmet', $id)
            ->where('zakljucna_ocena.id_ucenik', $ucenik)
            ->where('zakljucna_ocena.tip_ocene', $polugodiste)
            ->get();
    }
}

```

Zahtev.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Zahtev extends Model
{
    public function dohvatiZahteve(){
        return \DB::table('zahtev')->select('zahtev.*', 'ucenik.ime',
'ucenik.prezime', 'korisnici.ime as imeK', 'korisnici.prezime as prezimeK',
'vrsta_ocene.vrsta', 'predmet.naziv')
            ->join('ucenik', 'ucenik.id_ucenik', '=', 'zahtev.id_ucenik')
            ->join('vrsta_ocene', 'vrsta_ocene.id_vrsta_ocene', '=',
'zahtev.id_vrsta_ocene')
            ->join('korisnici', 'korisnici.id_korisnik', '=',
'zahtev.id_korisnik')
            ->join('korisnik_predmet', 'korisnik_predmet.id_korisnik', '=',
'korisnici.id_korisnik')
            ->join('predmet', 'korisnik_predmet.id_predmet', '=',
'predmet.id_predmet')
            ->where('zahtev.odobreno', null)
            ->get();
    }
}

```


4.3. Views

Fixed

Footer.blade.php

```
<footer id="futer" class="mt-5">
  <div class="container">
    <div class="row pt-3">
      <div class="col-md-4">
        <div class="inner-content">
          <p>Copyright &copy; Aleksandra Markovic 2021</p>
        </div>
      </div>
      <div class="col-md-8">
        <div class="inner-content">
          <ul id="ikonice" >
            <li> <a href="https://www.instagram.com/bekamarkoviic/"
target="_blank"> <i class="fa fa-instagram" aria-hidden="true"></i> </a> </li>
            <li> <a href="https://www.linkedin.com/in/aleksandra-
markovic-3aa730204/" target="_blank"> <i class="fa fa-linkedin" aria-
hidden="true"></i> </a> </li>
            <li> <a href="" target="_blank"> Dokumentacija </a>
</li>
          </ul>
        </div>
      </div>
    </div>
  </div>
</footer>
```

Header.blade.php

```
<head>

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">
  <meta name="description" content=@yield('description')>
  <meta name="keywords" content=@yield('keywords')>
  <meta name="author" content="Aleksandra Markovic">
  <meta name="csrf-token" content="{{ csrf_token() }}">
  <link rel="shortcut icon" href="" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">

  <title>Elektronski dnevnik | @yield('title')</title>

  <link href="{{ asset('assets/bootstrap/css/bootstrap.min.css') }}"
rel="stylesheet">
  <link rel="stylesheet" href="{{ asset('assets/css/style.css') }}">
  <link rel="stylesheet" href="{{ asset('assets/css/responsive.css') }}">

</head>
```

Navigation.blade.php

```
<nav class="navbar fixed-top navbar-expand-lg navbar-light bg-light">
    <div class="container-fluid">
        <a class="navbar-brand" href="{{ route('home') }}"><i class="fa fa-book"
aria-hidden="true" ></i>eDnevnik</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                @foreach($meni as $m)
                    <li class="nav-item">
                        <a class="nav-link" href="{{ route($m->ruta) }}">{{ $m -
> naziv }}</a>
                    </li>
                @endforeach
                @if(session('korisnik')->id_uloga == 5)
                    <li class="nav-item">
                        <a class="nav-link" href="{{ route('zahtevi')
}}>Zahtevi</a>
                    </li>
                @endif
                @if(session('korisnik')->id_uloga == 2)
                    <li class="nav-item">
                        <a class="nav-link" href="{{ route('odeljenje', ['id' =>
session('odeljenje')->id_odeljenje]) }}">Odeljenje</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ route('dnevnik', ['id' =>
session('odeljenje')->id_odeljenje]) }}">Dnevnik</a>
                    </li>
                @endif
                @if(session('korisnik')->id_uloga == 1)
                    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                        <li class="nav-item dropdown">
                            <a class="nav-link dropdown-toggle" href="#"
id="navbarDropdownMenuLink" role="button" data-bs-toggle="dropdown" aria-
expanded="false">
                                Administracija
                            </a>
                            <ul class="dropdown-menu" aria-
labelledby="navbarDropdownMenuLink">
                                <li><a class="dropdown-item" href="{{
route('smerovi') }}">Smerovi</a></li>
                                <li><a class="dropdown-item" href="{{
route('predmeti') }}">Predmeti</a></li>
                                <li><a class="dropdown-item" href="{{
route('odeljenja') }}">Odeljenja</a></li>
                                <li><a class="dropdown-item" href="{{
route('ucenici') }}">Ucenici</a></li>
                                <li><a class="dropdown-item" href="{{
route('nastavnici') }}">Nastavnici</a></li>
                                <li><a class="dropdown-item" href="{{
route('registracijaForma') }}">Registracija korisnika</a></li>
                            </ul>
                        </li>
                    </ul>
                @endif
                @if(session('korisnik')->id_uloga == 2)
                    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
```

```

        <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="#"
id="navbarDropdownMenuLink" role="button" data-bs-toggle="dropdown" aria-
expanded="false">
                Administracija
            </a>
            <ul class="dropdown-menu" aria-
labelledby="navbarDropdownMenuLink">
                <li><a class="dropdown-item" href="{{
route('ucenici') }}">Učenci</a></li>
            </ul>
        </li>
    </ul>
@endif
    <ul class="navbar-nav me-auto mb-2 mb-lg-0 mt-4
ulogovanResponsive">
        <li class="nav-item dropdown">
            <p id="ulogovan">Ulogovani korisnik</p>
            <a class="nav-link dropdown-toggle" href="#"
id="navbarDropdownMenuLink" role="button" data-bs-toggle="dropdown" aria-
expanded="false">
                {{ session('korisnik')->ime }} {{
session('korisnik')->prezime }}
            </a>
            <ul class="dropdown-menu nastavnik" aria-
labelledby="navbarDropdownMenuLink">
                <li><a class="dropdown-item" href="{{
route("profil") }}">Izmeni profil</a></li>
                @if(session()->has('korisnik'))
                <li><a class="dropdown-item" href="{{
route('logout') }}">Odjavi se</a></li>
                @endif
            </ul>
        </li>
    </ul>
</div>
<div class="d-flex" id="divPrijavljen">
    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item dropdown">
            <a class="nav-link " href="#" id="tekstPrijavljen"
role="button" data-bs-toggle="dropdown" aria-expanded="false">
                <span>{{ session('korisnik')->ime }} {{
session('korisnik')->prezime }}</span> slika)}}">
            </a>
            <ul class="dropdown-menu nastavnik" aria-
labelledby="navbarDropdownMenuLink">
                <li><a class="dropdown-item" href="{{
route("profil") }}">Izmeni profil</a></li>
                @if(session()->has('korisnik'))
                <li><a class="dropdown-item" href="{{
route('logout') }}">Odjavi se</a></li>
                @endif
            </ul>
        </li>
    </ul>
</div>
</div>
</nav>

```

Scripts.blade.php

```
<script src="{{ asset('assets/js/jquery.min.js') }}"></script>
<script src="{{ asset('assets/bootstrap/js/bootstrap.min.js') }}"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.5.1/chart.min.js"
integrity="sha512-
Wt1bJGtlnMtGP0dqNFH1x1kLBNpEodaiQ8ZN5JLA5wpc1sUlk/O5uuOMNgvzddzkpvZ9GLyYNa8w2s7r
qiTk5Q==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
<script type="text/javascript">
    const baseUrl = "{{ url('/') }}"
</script>
<script src="{{ asset('assets/js/main.js') }}"></script>
```

Layouts

Layout.blade.php

```
<!DOCTYPE html>
<html lang="en">
    @include('fixed.header')
    <body>
        @include('fixed.navigation')
        @yield('content')
        @include('fixed.footer')
        @include('fixed.scripts')
    </body>
</html>
```

LayoutLogin.blade.php

```
<!DOCTYPE html>
<html lang="en">
    @include('fixed.header')
    <body id="loginBody">
        @yield('content')
        @include('fixed.scripts')
    </body>
</html>
```

Pages

Nastavnici.blade.php

```
@extends('layouts.layout')
@section('title') Predmeti @endsection
@section('description') Forma za dodavanje predmeta @endsection
@section('keywords') dnevnik, elektronski, profesor, skola, smer, predmeti
@endsection
@section('content')
    <div class="container containerMargin">
        <div class="row">
            <div class="col-lg-10 m-auto mt-5">
```

```

        <table class="table table-bordered">
            <thead>
                <tr>
                    <th scope="col">Ime i prezime</th>
                    <th scope="col">Predmet</th>
                    <th scope="col"></th>
                </tr>
            </thead>
            <tbody id="ispisNastavnici">
                @foreach($nastavnici as $n)
                    <tr>
                        <td>{{ $n->ime }} {{ $n->prezime }}</td>
                        <td> {{ $n->naziv }} </td>
                        <td class="text-center">
                            <form action="{{ route('obrisiNastavnika') }}"
method="post">
                                @csrf
                                @method('DELETE')
                                <button type="button" class="btn btn-sm btn-
danger izbrisiNastavnika" data-id="{{ $n->id_korisnik }}">
                                    <i class="fa fa-trash" aria-
hidden="true"></i>
                                </button>
                            </form>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>
</div>
</div>
@endsection

```

Odeljenja.blade.php

```

@extends('layouts.layout')
@section('title') Odeljenje @endsection
@section('description') Froma za dodavanje odeljenja @endsection
@section('keywords') dnevnik, elektronski, profesor, odeljenje @endsection
@section('content')
    <div class="container containerMarginal visina">
        <div class="row">
            <div class="col-lg-10 m-auto">
                <h2>Dodajte novo odeljenje</h2>
            </div>
        </div>
        <div class="row">
            <div class="col-lg-10 m-auto">
                <form action="{{ route('dodajOdeljenje') }}" method="post"
class="mt-3" >
                    @csrf
                    <div class="mb-3">
                        <label for="brojOdeljenja" class="form-label">Broj
odeljenja</label>
                        <input type="text" class="form-control"
id="brojOdeljenja" name="brojOdeljenja" />
                        <p class="text-danger" id="brojOdeljenjaG"></p>
                    </div>
                    <div class="mb-3">

```

```

        <label for="godina" class="form-label">Godina</label>
        <input type="text" class="form-control" id="godina"
name="godina" />
        <p class="text-danger" id="godinaG"></p>
    </div>
    <div class="mb-3">
        <label for="razredni" class="form-label">Razredni
starešina</label>
        <select class="form-select" id="razredni"
name="razredni">
            <option value="0">Izaberite...</option>
            @foreach($korisnici as $k)
                <option value="{{ $k->id_korisnik }}">{{ $k->ime
}} {{ $k->prezime }}</option>
            @endforeach
        </select>
        <p class="text-danger" id="razredniG"></p>
    </div>
    <div class="mb-3">
        <label for="smerOdeljenje" class="form-
label">Smer</label>
        <select class="form-select" id="smerOdeljenje"
name="smerOdeljenje">
            <option value="0">Izaberite...</option>
            @foreach($smerovi as $s)
                <option value="{{ $s->id }}">{{ $s->naziv
}}</option>
            @endforeach
        </select>
        <p class="text-danger" id="smerOdeljenjeG"></p>
    </div>
    <div class="mb-3" id="prikaziPredmete">

    </div>
    <button type="button" class="btn mb-5"
id="dodajOdeljenje">Dodaj</button>
    </form>
</div>
</div>
</div>
@endsection

```

Predmeti.blade.php

```

@extends('layouts.layout')
@section('title') Predmeti @endsection
@section('description') Forma za dodavanje predmeta @endsection
@section('keywords') dnevnik, elektronski, profesor, skola, smer, predmeti
@endsection
@section('content')
    <div class="container containerMargina visina">
        <div class="row">
            <div class="col-lg-10 m-auto d-flex flex-column">
                <button type="button" class="btn" id="dodajPredmet">
Dodaj predmet
                <i class="fa fa-plus-square me-1" aria-hidden="true"></i>
                </button>
            </div>
            <div class="col-lg-12" id="formaPredmet">
                <form action="{{ route('dodajSmer') }}" method="post" >

```

```

        <div class="mb-3">
            <input type="text" placeholder="Naziv predmeta"
class="form-control" id="predmet" name="predmet" />
            <p class="text-danger" id="predmetGreska"></p>
        </div>
        @foreach($smerovi as $smer)
            <div class="form-check">
                <input class="form-check-input smerPredmet"
name="smerPredmet[]" type="checkbox" value="{{ $smer->id }}">
                <label class="form-check-label"
for="smerPredmet">
                    {{ $smer->naziv }}
                </label>
                <p class="text-danger"
id="smerPredmetGreska"></p>
            </div>
        @endforeach
        <div class="mb-3">
            <input type="button" class="btn"
id="izvršiPredmet" value="Izvrši">
        </div>
    </form>
</div>
</div>
</div>
<div class="row">
    <div class="col-lg-10 m-auto mt-3">
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th scope="col">Naziv predmeta</th>
                    <th scope="col"></th>
                </tr>
            </thead>
            <tbody id="ispisPredmeti">
                @foreach($podaci as $p)
                    <tr>
                        <td> {{ $p->naziv }} </td>
                        <td class="text-center">
                            <form action="{{ route('izbrisiPredmet') }}"
method="post">
                                @csrf
                                @method('DELETE')
                                <button type="button" class="btn btn-sm
btn-danger izbrisiP" data-id="{{ $p->id }}">
                                    <i class="fa fa-trash" aria-
hidden="true"></i>
                                </button>
                            </form>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>
</div>
</div>
@endsection

```

Registracija.blade.php

```
@extends('layouts.layout')
@section('title') Registracija @endsection
@section('description') Forma za registraciju korisnika @endsection
@section('keywords') dnevnik, elektronski, profesor, registracija, korisnik @endsection
@section('content')
    <div class="container containerMargin1 visina">
        @include('partials.success')
        @include('partials.error')
        <div class="row">
            <div class="col-lg-10 m-auto">
                <h2>Registracija korisnika</h2>
            </div>
        </div>
        <div class="row">
            <div class="col-lg-10 m-auto">
                @include('partials.form', ["action" => 'registracija'])
            </div>
        </div>
    </div>
@endsection
```

Smerovi.blade.php

```
@extends('layouts.layout')
@section('title') Smerovi @endsection
@section('description') Forma za dodavanje smerova @endsection
@section('keywords') dnevnik, elektronski, skola, profesor, smerovi @endsection
@section('content')
    <div class="container containerMargin visina">
        <div class="row">
            <div class="col-lg-10 m-auto d-flex flex-column">
                <button type="button" class="btn" id="dodajSmer">
                    <i class="fa fa-plus-square me-1" aria-hidden="true"></i>
Dodaj smer
                </button>
                <div class="row">
                    <div class="col-lg-12" id="formaSmer">
                        <form action="{ route('dodajSmer') }" method="post"
class="d-flex flex-row forma">
                            <div class="mb-3">
                                <input type="text" placeholder="Naziv smer"
class="form-control" id="smer" name="smer" />
                                <p class="text-danger" id="smerGreska"></p>
                            </div>
                            <div class="mb-3">
                                <input type="button" class="btn" id="izvršiSmer"
value="Izvrši">
                                </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
        <div class="row">
            <div class="col-lg-10 m-auto mt-3">
                <table class="table table-bordered">
                    <thead>
```



```

        <tr>
            <th scope="col">Naziv smera</th>
            <th scope="col"></th>
        </tr>
    </thead>
    <tbody id="ispisSmerovi">
        @foreach($podaci as $p)
            <tr>
                <td> {{ $p->naziv }} </td>
                <td class="text-center">
                    <form action="{{ route('izbrisiSmer') }}"
method="post">
                        @csrf
                        @method('DELETE')
                        <button type="button" class="btn btn-sm btn-
danger izbrisi" data-id="{{ $p->id }}">
                            <i class="fa fa-trash" aria-
hidden="true"></i>
                        </button>
                    </form>
                </td>
            </tr>
        @endforeach
    </tbody>
</table>
</div>
</div>
</div>
@endsection

```

Ucenici.blade.php

```

@extends('layouts.layout')
@section('title') Učenci @endsection
@section('description') Forma za dodavanje ucenika @endsection
@section('keywords') dnevnik, elektronski, profesor, skola, ucenici @endsection
@section('content')
    <div class="container containerMargin">
        <div class="row">
            <div class="col-lg-10 m-auto mt-5">
                @include('partials.success')
                <h2>Dodajte novog učenika</h2>
            </div>
        </div>
        <div class="row">
            <div class="col-lg-10 m-auto">
                <form action="{{ route('dodajUcenika') }}" method="post"
class="mt-3" enctype='multipart/form-data'>
                    @csrf
                    <div class="mb-3">
                        <label for="imeUcenika" class="form-label">Ime</label>
                        <input type="text" class="form-control" id="imeUcenika"
name="imeUcenika" />
                        @error('imeUcenika')
                        <p class="text-danger">
                            {{ $message }}
                        </p>
                        @enderror
                    </div>
                    <div class="mb-3">

```

```

        <label for="prezimeUcenika" class="form-
label">Prezime</label>
        <input type="text" class="form-control"
id="prezimeUcenika" name="prezimeUcenika" />
        @error('prezimeUcenika')
        <p class="text-danger">
            {{ $message }}
        </p>
        @enderror
    </div>
    <div class="mb-3">
        <label for="slikaUcenika" class="form-
label">Slika</label>
        <input type="file" class="form-control"
id="slikaUcenika" name="slikaUcenika" />
        @error('slikaUcenika')
        <p class="text-danger">
            {{ $message }}
        </p>
        @enderror
    </div>
    <div class="mb-3">
        <label for="jmbg" class="form-label">Matični broj
(JMBG)</label>
        <input type="text" class="form-control" id="jmbg"
name="jmbg" />
        @error('jmbg')
        <p class="text-danger">
            {{ $message }}
        </p>
        @enderror
    </div>
    <div class="mb-3">
        <label for="adresa" class="form-label">Adresa</label>
        <input type="text" class="form-control" id="adresa"
name="adresa" />
        @error('adresa')
        <p class="text-danger">
            {{ $message }}
        </p>
        @enderror
    </div>
    <div class="mb-3">
        <label for="mestoRodj" class="form-label">Mesto rođenja
</label>
        <input type="text" class="form-control" id="mestoRodj"
name="mestoRodj" />
        @error('mestoRodj')
        <p class="text-danger">
            {{ $message }}
        </p>
        @enderror
    </div>
    <div class="mb-3">
        <label for="datumRodj" class="form-label">Datum
rođenja</label>
        <input type="date" class="form-control" id="datumRodj"
name="datumRodj" />
        @error('datumRodj')
        <p class="text-danger">
            {{ $message }}
        </p>
        @enderror
    </div>

```

```

        @if(session('korisnik')->id_uloga == 1)
            <div class="mb-3">
                <label for="odeljenjeUcenika" class="form-label">Odeljenje</label>
                <select class="form-select" id="odeljenjeUcenika"
name="odeljenjeUcenika">
                    <option value="0">Izaberite...</option>
                    @foreach($odeljenja as $o)
                        <option value="{{ $o->id_odeljenje }}">
@if($o->godina == 1) I @elseif($o->godina == 2) II @elseif($o->godina == 3) III
@else IV @endif {{ $o->broj_odeljenja }} </option>
                    @endforeach
                </select>
                @error('odeljenjeUcenika')
                <p class="text-danger">
                    {{ $message }}
                </p>
                @enderror
            </div>
            @endif
            <button type="submit" class="btn mb-5"
id="dodajUcenika">Dodaj</button>
        </form>
    </div>
</div>
</div>
@endsection

```

Zahtevi.blade.php

```

@extends('layouts.layout')
@section('title') Zahtevi @endsection
@section('description') Prikaz zahteva direktoru @endsection
@section('keywords') dnevnik, elektronski, profesor, direktor, zahtev
@endsection
@section('content')
    <div class="container containerMargin1 containerMargin2 visina">
        @include('partials.success')
        @include('partials.error')
        @if(count($zahtevi) == 0)
            <div class="row">
                <div class="col-lg-12 visina d-flex justify-content-center
align-items-center" >
                    <h2 class="text-muted nemaZahteva">Trenutno nema
zahteva!</h2>
                </div>
            </div>
        @else
            <div class="row mt-4">
                <div class="col-lg-10 m-auto table-responsive">
                    <table class="table table-bordered">
                        <thead class="text-center">
                            <tr>
                                <th scope="col">Učenik</th>
                                <th scope="col">Nastavnik</th>
                                <th scope="col">Predmet</th>
                                <th scope="col">Pogrešna ocena</th>
                                <th scope="col">Nova ocena</th>
                                <th scope="col">Odobreno</th>
                            </tr>

```

```

        </thead>
        <tbody class="text-center">
        @foreach($zahtevi as $z)
            <tr>
                <td> {{ substr($z->ime, 0, 1) }}. {{ $z->prezime
}} </td>
                <td> {{ substr($z->imeK, 0, 1) }}. {{ $z->
prezimeK }} </td>
                <td> {{ $z->naziv }} </td>
                <td> {{ $z->pogresna_ocena }} </td>
                <td> {{ $z->nova_ocena }} </td>
                <td class="d-flex justify-content-center">
                    <form action="{{ route('odobriZahtev') }}"
method="post">
                        @csrf
                        @method('PUT')
                        <input type="hidden" name="zahtev"
value="{{ $z->id_zahtev }}">
                        <input type="hidden" name="nova"
value="{{ $z->nova_ocena }}">
                        <input type="hidden" name="datumZ"
value="{{ $z->datum_ocene }}">
                        <input type="hidden" name="ucenik"
value="{{ $z->id_ucenik }}">
                        <input type="hidden" name="vrsta"
value="{{ $z->id_vrsta_ocene }}">
                        <button type="submit" class="btn btn-sm
success">
                            <i class="fa fa-check" aria-
hidden="true"></i>
                        </button>
                    </form>
                    <form action="{{ route('odbijZahtev') }}"
method="post">
                        @csrf
                        @method('PUT')
                        <input type="hidden" name="zahtev"
value="{{ $z->id_zahtev }}">
                        <button type="submit" class="btn btn-sm
ms-1">
                            <i class="fa fa-times" aria-
hidden="true"></i>
                        </button>
                    </form>
                </td>
            </tr>
        @endforeach
        </tbody>
    </table>
</div>
</div>
@endif
</div>
@endsection

```

```

@extends('layouts.layout')
@section('title') Čas @endsection
@section('description') Forma za upis casa @endsection
@section('keywords') dnevnik, elektronski, profesor, odeljenja, casovi
@endsection
@section('content')
    <div class="container containerMargina visina">
        <div class="row">
            <div class="col-lg-10 m-auto mt-5">
                <h2>Upišite čas</h2>
            </div>
        </div>
        <div class="row">
            <div class="col-lg-10 m-auto">
                <form action="" method="POST" class="mt-3">
                    <div class="mb-3">
                        <label for="opis" class="form-label">Nastavna
jedinica</label>
                        <input type="text" class="form-control" id="opis"
name="opis" />
                        <p class="text-danger" id="opisGreska"></p>
                    </div>
                    <div class="mb-3">
                        <label for="datumCas" class="form-label">Datum</label>
                        <input type="date" class="form-control" id="datumCas"
name="datumCas" value="{{ date('Y-m-d') }}" />
                        <p class="text-danger" id="datumGreska"></p>
                    </div>
                    <div class="mb-3">
                        <label for="brojCasa" class="form-label">Redni broj
časa</label>
                        <input type="number" class="form-control" id="brojCasa"
name="brojCasa" />
                        <p class="text-danger" id="brojCasaGreska"></p>
                    </div>
                    <div class="mb-3">
                        <label for="napomena" class="form-
label">Napomena</label>
                        <textarea id="napomena" name="napomena" class="form-
control"></textarea>
                    </div>
                    <div class="mb-3">
                        <p>Odustni</p>
                        @foreach($ucenici as $ucenik)
                            <div class="form-check">
                                <input class="form-check-input odsutniCas"
type="checkbox" value="{{ $ucenik->id_ucenik }}" name="odsutniCas">
                                <label class="form-check-label"
for="odsutniCas">
                                    {{ $ucenik->ime }} {{ $ucenik->prezime }}
                                </label>
                            </div>
                        @endforeach
                    </div>
                    <button type="button" class="btn mt-3 mb-3"
id="upisiBtn">Izvrši</button>
                </form>
            </div>
        </div>
    </div>
@endsection

```

```

@extends('layouts.layout')
@section('title') Dnevnik @endsection
@section('description') Prikaz dnevnika i upis casova @endsection
@section('keywords') dnevnik, elektronski, profesor, odeljenja, casovi
@endsection
@section('content')
    <div class="container containerMargina containerDnevnik">
        <div class="row" id="odeljenjeGore">
            <div class="col-lg-6 col-md-6 mt-3">
                <h5 class="card-title mt-2">Dnevnik rada</h5>
            </div>
            <div class="col-lg-6 col-md-6 mt-3 mb-2 d-flex justify-content-end">
                <input type="date" class="form-control w-50" id="casDatum" >
                <input type="button" class="btn ms-2" id="filterDatum"
value="Filtriraj" >
            </div>
        </div>
        <div class="row mb-3" id="datumi">
            @foreach($datumi as $d)
                <div class="col-lg-12">
                    <div class="row datumCas">
                        <a href="#" class="datum dan">
                            @if(date('l', strtotime($d->datum)) == "Monday")
                                Ponedeljak
                            @elseif(date('l', strtotime($d->datum)) ==
"Tuesday")
                                Utorak
                            @elseif(date('l', strtotime($d->datum)) ==
"Wednesday")
                                Sreda
                            @elseif(date('l', strtotime($d->datum)) ==
"Thursday")
                                Četvrtak
                            @elseif(date('l', strtotime($d->datum)) == "Friday")
                                Petak
                            @endif
                        </a>
                        <a href="#" class="datum d">{{ date('d.m.Y.',
strtotime($d->datum)) }}</a>
                    </div>
                    <input type="hidden" class="datumVred" value="{{ $d->datum
}}">
                    <input type="hidden" class="ulogovani" value="{{
session('korisnik')->id_korisnik }}">
                    <div class="row casovi d-flex justify-content-start">
                        </div>
                    </div>
                </div>
            @endforeach
        </div>
        {{ $datumi->links('pagination.bootstrap-4') }}
        <div class="modal fade" id="exampleModalCenter" tabindex="-1"
role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
            <div class="modal-dialog modal-dialog-centered" role="document">
                <div class="modal-content">
                    <div class="modal-header">
                        <h5 class="modal-title"
id="exampleModalLongTitle">Odustni</h5>
                    </div>
                    <div class="modal-body" id="modalBody">

```

```

        </div>
        <div class="modal-footer">
            <button type="button" class="btn" data-bs-
dismiss="modal">Zatvori</button>
        </div>
    </div>
</div>
<div class="modal fade" id="modalIzmeniCas" tabindex="-1" role="dialog"
aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title"
id="exampleModalLongTitle">Izmeni čas</h5>
                <button type="button" class="close btn" data-bs-
dismiss="modal" aria-label="Close">
                    <span aria-hidden="true"><i class="fa fa-times"
aria-hidden="true"></i></span>
                </button>
            </div>
            <div class="modal-body">
                <form action="{{ route('updateCas') }}" method="post">
                    @csrf
                    @method('PUT')
                    <div class="mb-3">
                        <label for="nastavnaJedUpdate" class="form-
label">Nastavna jedinica</label>
                        <input type="text" class="form-control"
id="nastavnaJedUpdate" name="nastavnaJedUpdate"/>
                        <p class="text-danger"
id="nastavnaJedUpdateG"></p>
                    </div>
                    <div class="mb-3">
                        <label for="napomenaUpdate" class="form-
label">Napomena</label>
                        <textarea id="napomenaUpdate"
name="napomenaUpdate" class="form-control"></textarea>
                        <p class="text-danger" id="napomenaUpdateG"></p>
                    </div>
                    <input type="hidden" id="casId" name="casId">
                    <input type="button" class="btn float-end"
id="updateCas" value="Izmeni">
                </form>
            </div>
        </div>
    </div>
</div>
</div>
</div>
@endsection

```

Home.blade.php

```

@extends('layouts.layout')
@section('title') Početna @endsection
@section('description') Početna strana @endsection
@section('keywords') dnevnik, elektronski, profesor, odeljenja @endsection
@section('content')
    @if(session('korisnik')->id uloga == 5)

```

```

        <div class="container containerMargina">
            <div class="row">
                <div class="col-lg-6 mt-5">
                    <canvas id="izostanciChart"></canvas>
                </div>
                <div class="col-lg-6 mt-5">
                    <canvas id="zahteviChart"></canvas>
                </div>
            </div>
        </div>
    @endif
    <div class="container visina @if(session('korisnik')->id_uloga != 5)
containerMargina @endif" id="containerPocetna">
        <div class="row d-flex justify-content-start">
            @foreach($podaci as $o)
                <div class="shadow-sm mt-5 d-flex justify-content-start
odeljenjePocetna">
                    <div class="linija my-auto"></div>
                    <div>
                        <p class="brojOdeljenja">@if($o->godina == 1) I @elseif($o-
>godina == 2) II @elseif($o->godina == 3) III @else IV @endif<span
class="badge">{{ $o->broj_odeljenja }}</span> </p>
                        <h5 class="card-title">@if(session('korisnik')->id_uloga ==
1 || session('korisnik')->id_uloga == 5){ { substr($o->ime, 0, 1) } }. {{ $o-
>prezime }} @elseif(session('korisnik')->id_uloga == 2 || session('korisnik')-
>id_uloga == 3) {{ $o->naziv }} @endif</h5>
                        <h6 class="card-subtitle mb-4 text-muted">@if($o->godina ==
1) I @elseif($o->godina == 2) II @elseif($o->godina == 3) III @else IV @endif
razred srednje skole</h6>
                        <a href="{{ route('odeljenje', ['id' => $o->id_odeljenje])
}}" class="btn-sm linkPocetna" title="Imenik"> <i class="fa fa-user" aria-
hidden="true"></i> </a>
                        <a href="{{ route('dnevnik', ['id' => $o->id_odeljenje]) }}"
class="btn-sm linkPocetna" title="Dnevnik"> <i class="fa fa-book" aria-
hidden="true"></i> </a>
                        <a href="{{ route('formaCas', ['id' => $o->id_odeljenje])
}}" class="btn-sm linkPocetna" title="Dodaj čas"> <i class="fa fa-plus-circle"
aria-hidden="true"></i> </a>
                    </div>
                </div>
            @endforeach
        </div>
    </div>
@endsection

```

Login.blade.php

```

@extends('layouts.layoutLogin')
@section('title') Login @endsection
@section('content')
    <div class="container">
        <div class="row">
            <div class="col-lg-12 d-flex justify-content-center">
                <h1 id="naslov"><i class="fa fa-book" aria-hidden="true"
></i>eDnevnik</h1>
            </div>
        </div>
        <div class="row d-flex justify-content-center" id="tabelaLogin">
            <div class="col-lg-4 col-md-6 col-8 d-flex justify-content-center
align-items-center" id="loginKol">

```



```

        <form action="{{ route('login') }}" method="POST" class="mt-3">
            <div class="mb-3">
                <label for="kor_ime" class="form-label">Korisničko
ime</label>
                <input type="text" class="form-control" id="kor_ime"
name="kor_ime" autocomplete="off"/>
                <p class="text-danger" id="kor_imeGreska"></p>
            </div>
            <div class="mb-3">
                <label for="lozinka" class="form-label">Šifra</label>
                <input type="password" class="form-control" id="lozinka"
name="lozinka" />
                <p class="text-danger" id="lozinkaGreska"></p>
            </div>
            <button type="button" class="btn" id="loginBtn">Prijavi
se</button>
        </form>
    </div>
</div>
</div>
@endsection

```

Profil.blade.php

```

@extends('layouts.layout')
@section('title') Profil @endsection
@section('description') Profil korisnika @endsection
@section('keywords') dnevnik, elektronski, korisnik, odeljenja @endsection
@section('content')
    <div class="container containerMarginal1 visina">
        @include('partials.success')
        @include('partials.error')
        <div class="row shadow-sm" id="redProfil">
            @foreach($profil as $p)
                <div class="col-lg-3 text-center">
                    
                    <h2 class="mt-4 text-center imePrezime">{{ $p->ime }} <br> {{
$p->prezime }}</h2>
                </div>
                <div class="col-lg-8 mt-4">
                    <div class="mt-4">
                        <h3 class="naslovProfil">Glavne postavke</h3>
                    </div>
                    <form action="{{ route('izmeniProfil') }}" method="post"
class="mt-3" enctype='multipart/form-data'>
                        @csrf
                        @method('PUT')
                        <div class="mb-3">
                            <label for="kor_imeProfil" class="form-label">Korisničko
ime</label>
                            <input type="text" class="form-control"
id="kor_imeProfil" name="kor_imeProfil" value="{{ $p->kor_ime }}" />
                            <p class="text-danger" id="kor_imeProfilG"></p>
                            @error('kor_imeProfil')
                                <p class="text-danger">
                                    {{ $message }}
                                </p>
                            @enderror
                        </div>

```

```

        <div class="mb-3">
            <label for="emailProfil" class="form-
label">Email</label>
            <input type="text" class="form-control" id="emailProfil"
name="emailProfil" value="{{ $p->email }}" />
            <p class="text-danger" id="emailProfilGreska"></p>
            @error('emailProfil')
            <p class="text-danger">
                {{ $message }}
            </p>
            @enderror
        </div>
        @if(session('korisnik')->id_uloga == 4)
            <div class="mb-3">
                <label for="brojTelefona" class="form-label">Broj
telefona</label>
                <input type="text" class="form-control"
id="brojTelefona" name="brojTelefona" value="{{ $p->telefon }}" />
                <p class="text-danger" id="brojTelefonaG"></p>
            </div>
            @endif
            <div class="mb-3">
                <label for="slikaProfil" class="form-
label">Slika</label>
                <input type="file" class="form-control" id="slikaProfil"
name="slikaProfil" />
                @error('slikaProfil')
                <p class="text-danger">
                    {{ $message }}
                </p>
                @enderror
            </div>
            <input type="submit" class="btn float-end" id="updateProfil"
value="Izmeni" />
        </form>
    @endforeach
    <div class="bezbednost">
        <h3 class="naslovProfil">Bezbednost</h3>
    </div>
    <div class="mt-3 mb-3">
        <label for="izmeniLozinku" class="form-
label">Lozinka</label>
        <button type="button" class="btn btn-sm mt-2 mb-3 ms-2"
id="izmeniLozinku">Izmeni lozinku</button>
    </div>
    <form action="{{ route('izmeniLozinku') }}" method="post"
class="mt-3" id="formaLozinka">
        @csrf
        @method('PUT')
        <div class="mb-3">
            <label for="staraLozinka" class="form-label">Stara
lozinka</label>
            <input type="password" class="form-control"
id="staraLozinka" name="staraLozinka" />
            @error('staraLozinka')
            <p class="text-danger">
                {{ $message }}
            </p>
            @enderror
        </div>
        <div class="mb-3">
            <label for="novaLozinka" class="form-label">Nova
lozinka</label>
            <input type="password" class="form-control"

```

```

id="novaLozinka" name="novaLozinka" />
        @error('novaLozinka')
        <p class="text-danger">
            {{ $message }}
        </p>
        @enderror
    </div>
    <div class="mb-3">
        <label for="ponovoLozinka" class="form-label">Ponovite
lozinku</label>
        <input type="password" class="form-control"
id="ponovoLozinka" name="ponovoLozinka" />
        @error('ponovoLozinka')
        <p class="text-danger">
            {{ $message }}
        </p>
        @enderror
    </div>
    <button type="submit" class="btn mb-4 float-end"
id="updateLozinka">Izmeni</button>
    </form>
</div>
</div>
</div>
@endsection

```

Roditelj.blade.php

```

@extends('layouts.layout')
@section('title') Roditelj @endsection
@section('description') Prikaz učenika roditelju @endsection
@section('keywords') dnevnik, elektronski, roditelj, skola, odeljenje
@endsection
@section('content')
    <div class="container visina containerMargina">
        <div class="row mb-4">
            <div class="col-lg-12 col-md-12 col-11 m-auto mt-3
padajucaRoditelj">
                <label for="ucenikRoditelj" class="form-label
labelUcenik">Učenik</label>
                <select id="ucenikRoditelj" name="ucenikRoditelj" class="form-
select">
                    @foreach($ucenici as $ucenik)
                        <option value="{{ $ucenik->id ucenik }}">{{ $ucenik->ime
}} {{ $ucenik->prezime }}</option>
                    @endforeach
                </select>
            </div>
        </div>
        <div class="row ucenik shadow-sm">
            <div class="col-lg-3 text-center levo" id="roditeljLevo">
            </div>
            <div class="col-lg-9">
                <ul class="nav nav-tabs navigacija">
                    <li class="nav-item">
                        <a class="nav-link linkTab active linkOcene" aria-
current="page" href="#" id="linkOcene">Ocene i izostanci</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link linkTab linkPredmeti" href="#"

```

```

id="linkPredmeti">Predmeti</a>
        </li>
    </ul>
    <div class="ocene">
        <div>
            <div class="col-lg-12">
                <select name="predmetiR" id="predmetiR" class="form-
select mt-4 mb-3">

                    </select>
                </div>
            </div>
            <div class="w-100 text-center mt-5 izaberitePredmet">
                <h5 class="fw-normal">Izaberite predmet za prikaz
ocena.</h5>

            </div>
            <div class="d-flex justify-content-between polugodista"
id="oceneRoditelj" >

            </div>
            <div class="d-flex justify-content-start prosek mt-3">
                <div class="d-flex justify-content-start me-3"
id="prosekPrvoR">

                </div>
                <div class="d-flex justify-content-start"
id="prosekDrugoR">

                </div>
            </div>
            <div class="mt-5 sakrij" id="zakljucnaNaslov">
                <h5>Zaključne ocene</h5>
            </div>
            <div class="d-flex justify-content-start"
id="zakljucneRoditelj">

            </div>
            <div class="izostanciNaslov mt-5">
                <h5>Izostanci</h5>
            </div>
            <div class="d-flex justify-content-start flex-wrap mt-3 mb-
4" id="izostanciRoditelj">

            </div>
        </div>
    <div class="predmeti">
        {{--
            <div class="row">--}}
            <div class="col-lg-12">
                <table class="table" id="tabelaPredmeti">
                    <thead>
                        <tr>
                            <th scope="col">Predmet</th>
                            <th scope="col">Nastavnik</th>
                        </tr>
                    </thead>
                    <tbody id="predmetiRoditelj">

                    </tbody>
                </table>
            </div>
        {{--
            </div>--}}
    </div>
</div>
</div>
</div>
@endsection

```

Odeljenje.blade.php

```
@extends('layouts.layout')
@section('title') Odeljenje @endsection
@section('description') Prikaz odeljenja u skoli @endsection
@section('keywords') dnevnik, elektronski, profesor, skola, odeljenje
@endsection
@section('content')
    <div class="container containerMargina visina">
        <div class="row" id="odeljenjeGore">
            @foreach($podaci as $p)
                <div class="col-lg-6 col-md-6 odeljenjeResponsive mt-3">
                    <h5 class="card-title">Imenik</h5>
                    <h6 class="card-subtitle mb-3 text-muted">Razredno odeljenje:
@if($p->godina == 1) I @elseif($p->godina == 2) II @elseif($p->godina == 3) III
@else IV @endif<span class="badge text-muted">{{ $p->broj_odeljenja }}</span>
                    <h6>{{ count($ucenici) }}</h6>
                </div>
                <div class="col-lg-6 col-md-6 odeljenjeResponsive mt-3 text-end">
                    <h5 class="card-title">Razredni</h5>
                    <h6 class="card-subtitle mb-3 text-muted">{{ $p->prezime }} {{
$p->ime }}</h6>
                </div>
            @endforeach
        </div>
        <div class="row djaci">
            @foreach($ucenici as $u)
                <div class="mt-2 shadow-sm ucenikOdeljenje">
                    <div class="m-auto text-center">
                        <a href="{{ route('ucenik', ['id' => $u->id_ucenik])
}}"> prezime }} {{ $u->ime }}"> </a>
                    </div>
                    <div class="ucenikIme text-center">
                        <a href="{{ route('ucenik', ['id' => $u->id_ucenik])
}}">{{ $u->prezime }} {{ $u->ime }}</a>
                        <p class="text-muted smerOdeljenje">{{ $u->naziv }}
(4)</p>
                    </div>
                </div>
            @endforeach
        </div>
    </div>
@endsection
```

Ucenik.blade.php

```
@extends('layouts.layout')
@section('title') Učenik @endsection
@section('description') Prikaz ucenika @endsection
@section('keywords') dnevnik, elektronski, profesor, odeljenje, ucenik
@endsection
@section('content')
    <div class="container visina ucenikContainer">
        @include('partials.success')
        <div class="row ucenik shadow-sm">
            <div class="col-lg-3 text-center levo">
                @foreach($ucenik as $u)
                    <div class="slikaIme">
                        
        <div class="imeSmer">
            <p class="ime mt-2">{{ $u->prezime }} {{ $u->ime
        }}</p>

            <p class="text-muted">{{ $u->naziv }} (4) </p>
        </div>
    </div>
    <div>
        @if(session('korisnik')->id_uloga == 2 && $u-
>id_odeljenje == session('odeljenje')->id_odeljenje)
            <form class="prikaziVladanje mt-4" action="{{
route('vladanje') }}" method="post">
                @csrf
                @method('PUT')
                <div class="w-75 m-auto d-flex mb-2 justify-
content-start divVladanje">
                    <input type="text" class="form-control form-
control-sm text-center" id="vladanje" name="vladanje" value="{{ $u->vladanje
        }}" />
                    <p class="text-danger" id="vladanjeG"></p>
                    <input type="button" class="btn btn-sm w-75
ms-2" id="vladanjeBtn" value="Izmeni vladanje">
                </div>
            </form>
        @else
            <div class="w-75 m-auto d-flex mb-3 justify-content-
start divTekstVladanje vucenik">
                <p class="tekstVladanje">Ocena iz vladanja</p>
                <p class="ocenaVladanje">{{ $u->vladanje }}</p>
            </div>
        @endif
        @if(session('korisnik')->id_uloga == 2 ||
session('korisnik')->id_uloga == 3)
            <a type="button" class="btn btn-sm w-75
@if(count($zakljucna2)) sakrij @endif" id="prikaziOcenu" href="#ModalOcena"
data-bs-toggle="modal">
                <i class="fa fa-plus-square me-1" aria-
hidden="true"></i>
                Dodaj ocenu
            </a>
            <a type="button" class="btn btn-sm w-75 mt-1
@if(count($zakljucna2)) sakrij @endif" id="prikaziZakljucnu"
href="#ModalZakljucna" data-bs-toggle="modal">
                <i class="fa fa-graduation-cap me-1" aria-
hidden="true"></i>
                Zaključi ocenu
            </a>
            <a type="button" class="btn btn-sm w-75 mt-1 mb-3"
id="prikaziZahtev" href="#ModalFormaZahtev" data-bs-toggle="modal">
                <i class="fa fa-flag me-1" aria-
hidden="true"></i>
                Pogrešna ocena
            </a>
        @endif
        @if(session('korisnik')->id_uloga == 1 ||
(session('korisnik')->id_uloga == 2 && $u->id_odeljenje == session('odeljenje')-
>id_odeljenje))
            <div class="sakrij mb-3 divKontakt">
                <a class="btn btn-sm w-50 mt-2 mb-4
btnDodajKontakt" href="#ModalFormaRoditelj" data-bs-toggle="modal"><i class="fa
fa-plus-square me-1" aria-hidden="true"></i> Dodaj kontakt</a>
            </div>
        @endif
    </div>

```

```

        @endforeach
    </div>
    <div class="col-lg-9">
        <ul class="nav nav-tabs navigacija">
            <li class="nav-item">
                <a class="nav-link linkTab active linkOcene" aria-
current="page" href="#" id="linkOcene">Ocene i izostanci</a>
            </li>
            <li class="nav-item">
                <a class="nav-link linkTab linkPredmeti" href="#"
id="linkPredmeti">Predmeti</a>
            </li>
            <li class="nav-item">
                <a class="nav-link linkTab linkInfo" href="#"
id="linkInfo">Informacije</a>
            </li>
        </ul>
        <div class="ocene">
            @foreach($ucenik as $u)
                @if(session('korisnik')->id_uloga == 1 ||
session('korisnik')->id_uloga == 5 || (session('korisnik')->id_uloga == 2 && $u-
>id_odeljenje == session('odeljenje')->id_odeljenje))
                    <div>
                        <div class="col-lg-12">
                            <select class="form-select mt-4 mb-3"
id="predmetOcena" name="predmetOcena">
                                @if(session('korisnik')->id_uloga == 2)
                                    <option value="{{
session('predmet')->id_predmet }}" selected>{{ session('predmet')->naziv
}}</option>
                                @foreach($predmeti as $p)
                                    @if(session('predmet')->
id_predmet != $p->id_predmet)
                                        <option value="{{ $p->
id_predmet }}">{{ $p->naziv }}</option>
                                    @endif
                                @endforeach
                                <input type="hidden" id="idpredmet"
value="{{ session('predmet')->id_predmet }}">
                                @else
                                    <option
value="0">Izaberite...</option>
                                    @foreach($predmeti as $p)
                                        <option value="{{ $p->id_predmet
}}">{{ $p->naziv }}</option>
                                    @endforeach
                                @endif
                                <input type="hidden" id="idpredmet"
value="">
                            </select>
                        </div>
                    </div>
                    <div class="w-100 text-center mt-5"
izaberitePredmet">
                        <h5 class="fw-normal">Izaberite predmet za
prikaz ocena.</h5>
                    </div>
                @endif
            @endforeach
            <div class="d-flex justify-content-between polugodista" >
                <div class="col-lg-6 mt-4">
                    <div class="naslovPolugodiste mb-4">
                        <h5 class="polugodiste">I Polugodište</h5>
                    </div>

```

```

        <div class="w-100 d-flex justify-content-between">
            <div class="polug1 d-flex w-50">
                @foreach($ocene as $ocena)
                    @if($ocena->id_polugodiste == 1)
                        <div title="{{ $ocena->vrsta }}"
class="ocena">
                            <span>{{ $ocena ->ocena }}</span>
                        </div>
                    @endif
                @endforeach
            </div>
        </div>
    </div>
    <div class="col-lg-6 mt-4">
        <div class="naslovPolugodiste mb-4">
            <h5 class="polugodiste">II Polugodište</h5>
        </div>
        <div class="w-100 d-flex justify-content-between">
            <div class="polug2 d-flex w-50">
                @foreach($ocene as $ocena)
                    @if($ocena->id_polugodiste == 2)
                        <div title="{{ $ocena->vrsta }}"
class="ocena">
                            <span>{{ $ocena ->ocena }}</span>
                        </div>
                    @endif
                @endforeach
            </div>
        </div>
    </div>
    <div class="d-flex justify-content-start prosek mt-3">
        <div class="prosekPrvo d-flex justify-content-start me-
3">
            @foreach($prosekPrvo as $p)
                <div class="srednja">
                    <p>Srednja ocena (I Polugodište)</p>
                </div>
                <div class="srednjaOcena">
                    <p class="fw-bold">{{ round($p-
>prosecnaOcena, 2) }}</p>
                </div>
            @endforeach
        </div>
        <div class="prosekDrugo d-flex justify-content-start">
            @foreach($prosekDrugo as $p)
                <div class="srednja">
                    <p>Srednja ocena (II Polugodište)</p>
                </div>
                <div class="srednjaOcena">
                    <p class="fw-bold">{{ round($p-
>prosecnaOcena, 2) }}</p>
                </div>
            @endforeach
        </div>
    </div>
    <div class="mt-5" id="zakljucnaNaslov">
        <h5>Zaključne ocene</h5>
    </div>
    <div class="d-flex justify-content-start"
id="zakljucneUcenik">
        <div class="zakljucnaPolugodiste me-3 mt-2">
            <h6>I Polugodište</h6>
            @if(count($zakljucna1))

```



```

                                @foreach($zakljucna1 as $z)
                                @switch($z->zakljucna_ocena)
                                @case(1)
                                <p>nedovoljan ({{$z->
>zakljucna_ocena}})</p>
                                @break
                                @case(2)
                                <p>dovoljan ({{$z->
>zakljucna_ocena}})</p>
                                @break
                                @case(3)
                                <p>dobar ({{$z->
>zakljucna_ocena}})</p>
                                @break
                                @case(4)
                                <p>vrlo dobar ({{$z->
>zakljucna_ocena}})</p>
                                @break
                                @case(5)
                                <p>odličan ({{$z->
>zakljucna_ocena}})</p>
                                @break
                                @endswitch
                                @endforeach
                                @else
                                <p>&ndash;</p>
                                @endif
                            </div>
                            <div class="zakljucnaPolugodiste mt-2">
                                <h6>II Polugodište</h6>
                                @if(count($zakljucna2))
                                    @foreach($zakljucna2 as $z)
                                    @switch($z->zakljucna_ocena)
                                    @case(1)
                                    <p>nedovoljan ({{$z->
>zakljucna_ocena}})</p>
                                    @break
                                    @case(2)
                                    <p>dovoljan ({{$z->
>zakljucna_ocena}})</p>
                                    @break
                                    @case(3)
                                    <p>dobar ({{$z->
>zakljucna_ocena}})</p>
                                    @break
                                    @case(4)
                                    <p>vrlo dobar ({{$z->
>zakljucna_ocena}})</p>
                                    @break
                                    @case(5)
                                    <p>odličan ({{$z->
>zakljucna_ocena}})</p>
                                    @break
                                    @endswitch
                                    @endforeach
                                    @else
                                    <p>&ndash;</p>
                                    @endif
                                </div>
                            </div>
                            <div class="izostanciNaslov mt-5">
                                <h5>Izostanci</h5>
                            </div>
                            <div class="d-flex justify-content-start flex-wrap mt-3 mb-

```

```

4">
    <div class="col-lg-3 col-md-3 col-6 mt-3 d-flex align-
items-center">
        <div class="izostanci neregulisani">
            <span>{{ count($neregulisani) }}</span>
        </div>
        <div class="d-flex align-items-center">
            <p class="text-center mt-2 text-muted ms-
2">neregulisani</p>
        </div>
    </div>
    <div class="col-lg-3 col-md-3 col-6 mt-3 d-flex align-
items-center">
        <div class="izostanci opravdani">
            <span>{{ count($opravdani) }}</span>
        </div>
        <div class="d-flex align-items-center">
            <p class="text-center mt-2 text-muted ms-
2">opravdani</p>
        </div>
    </div>
    <div class="col-lg-3 col-md-3 col-6 mt-3 d-flex align-
items-center">
        <div class="izostanci neopravdani">
            <span>{{ count($neopravdani) }}</span>
        </div>
        <div class="d-flex align-items-center">
            <p class="text-center mt-2 text-muted ms-
2">neopravdani</p>
        </div>
    </div>
    <div class="col-lg-3 col-md-3 col-6 mt-3 d-flex align-
items-center">
        <div class="izostanci ukupno">
            <span>{{ count($neregulisani) +
count($opravdani) + count($neopravdani) }}</span>
        </div>
        <div class="d-flex align-items-center">
            <p class="text-center mt-2 text-muted ms-
2">ukupno</p>
        </div>
    </div>
</div>
@if(count($neregulisani) > 0 && session('korisnik')-
>id_uloga == 2 && $u->id_odeljenje == session('odeljenje')->id_odeljenje)
    <div class="izostanciNaslov mt-5">
        <h5>Neregulisani izostanci</h5>
    </div>
    <div id="neregulisaniIzostanci">
        <form action="{{ route('izostanci') }}"
method="post" class="mt-4">
            @csrf
            @method('PUT')
            @foreach($neregulisani as $n)
                <div class="d-flex justify-content-start
flex-wrap mb-3">
                    <div class="datumNeregulisani">
                        <p>{{ date('d.m.Y.', strtotime($n-
>datum)) }} {{ $n->naziv }}</p>
                    </div>
                    <div class="casNeregulisani">
                        <p>{{ $n->po_redu }}. čas</p>
                    </div>
                    <div class="w-25 neregulisaniSelect">

```

```

selectOpravdano" name="selectOpravdano">
    <select class="form-select mb-3 mt-1"
value="izaberite">Izaberite...</option>
    <option
value="1">Opravdano</option>
    <option
value="0">Neopravdano</option>
    </select>
</div>
<input type="hidden" class="idOdsutni"
value="{{ $n->id_odsutni }}">
    </div>
@endforeach
<p class="text-danger"
id="selectOpravdanoG"></p>
    <div class="col-lg-3 mb-3">
        <input class="btn" type="button"
id="regIzostanke" value="Izvrši">
    </div>
</form>
</div>
@endif
</div>
<div class="predmeti">
    <div class="col-lg-12 table-responsive">
        <table class="table" id="tabelaPredmeti">
            <thead>
                <tr>
                    <th scope="col">Predmet</th>
                    <th scope="col">Nastavnik</th>
                </tr>
            </thead>
            <tbody>
                @foreach($predmeti as $p)
                    <tr>
                        <td class="nazivPredmeta">{{ $p->naziv
}}</td>
                        <td>{{ $p->prezime }} {{ $p->ime }}</td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>
</div>
<div class="informacije">
    <div class="col-lg-6">
        <div class="mb-3 naslovInfo">
            <h4 class="fw-normal">Podaci o učeniku</h4>
        </div>
        @foreach($ucenik as $u)
            <div class="d-flex">
                <span class="fw-bold w-50">Datum rođenja</span>
                <p class="ms-2">{{ date('d.m.Y.', strtotime($u-
>datum_rodjena)) }}</p>
            </div>
            <div class="d-flex">
                <span class="fw-bold w-50">Mesto rođenja</span>
                <p class="ms-2">{{ $u->mesto_rodjenja }}</p>
            </div>
            <div class="d-flex">
                <span class="fw-bold w-50">Adresa</span>
                <p class="ms-2">{{ $u->adresa }}</p>
            </div>
        @endforeach
    </div>
</div>

```

```

        <div class="d-flex">
            <span class="fw-bold w-50">JMBG</span>
            <p class="ms-2">{{ $u->jmbg }}</p>
        </div>
    @endforeach
</div>
<div class="col-lg-6">
    <div class="mb-3 naslovInfo">
        <h4 class="fw-normal">Kontakti</h4>
    </div>
    @foreach($roditelji as $r)
        <div class="d-flex">
            <span class="fw-bold w-50">Ime</span>
            <p class="ms-2">{{ $r->prezime }} {{ $r->ime
}}</p>

            </div>
            <div class="d-flex">
                <span class="fw-bold w-50">Telefon</span>
                <p class="ms-2">{{ substr($r->telefon, 0, 3)
}}/{{ substr($r->telefon, 3, 3) }}-{{ substr($r->telefon, 6, 3) }}</p>
            </div>
            <div class="d-flex">
                <span class="fw-bold w-50">Email</span>
                <p class="ms-2">{{ $r->email }}</p>
            </div>
            <div class="d-flex justify-content-end
borderBottom">
                @if(session('korisnik')->id_uloga == 1 ||
(session('korisnik')->id_uloga == 2 && $u->id_odeljenje == session('odeljenje')->id_odeljenje))
                    <form action="{{ route('izbrisi') }}"
method="post" id="formaKontakt">
                        @csrf
                        @method('DELETE')
                        <input type="hidden"
name="izbrisiRoditelj" value="{{ $r->id_korisnik }}">
                        <button type="submit" class="btn btn-sm
mb-2">
                            Izbrisi
                        </button>
                    </form>
                @endif
            </div>
        @endforeach
    </div>
</div>
<div class="modal fade" id="ModalFormaZahtev" tabindex="-1"
role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered"
role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title"
id="exampleModalLongTitle">Pošaljite zahtev</h5>
                <button>
            </div>
            <div class="modal-body">
                <form action="{{ route('zahtev') }}"
method="POST" class="mt-3">
                    <div class="mb-3">
                        <label for="pogresnaOcena" class="form-
label">Pogrešna ocena</label>
                        <input type="text" class="form-control"
id="pogresnaOcena" name="pogresnaOcena" />

```

```

        <p class="text-danger"
id="pogresnaOcenaG"></p>
    </div>
    <div class="mb-3">
        <label for="novaOcena" class="form-
label">Nova ocena</label>
        <input type="text" class="form-control"
id="novaOcena" name="novaOcena" />
        <p class="text-danger"
id="novaOcenaG"></p>
    </div>
    <div class="mb-3">
        <label for="datumOcene" class="form-
label">Datum ocene</label>
        <input type="date" class="form-control"
id="datumOcene" name="datumOcene" />
        <p class="text-danger"
id="datumOceneG"></p>
    </div>
    <div class="mb-3">
        <label for="vrstaOceneZ" class="form-
label">Vrsta ocene</label>
        <select class="form-select"
id="vrstaOceneZ" name="vrstaOceneZ">
            <option
value="0">Izaberite...</option>
            @foreach($vrsta as $v)
                <option value="{{ $v-
id_vrsta_ocene }}">{{ $v->vrsta }}</option>
            @endforeach
        </select>
        <p class="text-danger"
id="vrstaOceneZG"></p>
    </div>
</form>
</div>
<div class="modal-footer">
    <button type="button" class="btn"
id="zahtev">Pošalji</button>
    <button type="button" class="btn" data-bs-
dismiss="modal">Odustani</button>
</div>
</div>
</div>
<div class="modal fade" id="ModalFormaRoditelj" tabindex="-1"
role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered"
role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title"
id="exampleModalLongTitle">Pošaljite zahtev</h5>
                <button type="button" class="close btn" data-bs-
dismiss="modal" aria-label="Close">
                    <span aria-hidden="true"><i class="fa fa-
times" aria-hidden="true"></i></span>
                </button>
            </div>
            <div class="modal-body" >
                @include('partials.form', ["action" =>
'dodajKontakt'])
            </div>
        </div>
    </div>
</div>

```

```

        </div>
    </div>
    <div class="modal fade" id="ModalOcena" tabindex="-1"
role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
        <div class="modal-dialog modal-dialog-centered"
role="document">
            <div class="modal-content">
                <div class="modal-header">
                    <h5 class="modal-title"
id="exampleModalLongTitle">Ocenite učenika</h5>
                    <button type="button" class="close btn" data-bs-
dismiss="modal" aria-label="Close">
                        <span aria-hidden="true"><i class="fa fa-
times" aria-hidden="true"></i></span>
                    </button>
                </div>
                <div class="modal-body" >
                    <form action="{{ route('ocena') }}"
method="post">
                        <input type="text" class="form-control w-
100" id="ocenaVred" name="ocenaVred" placeholder="Unesite ocenu">
                        <p class="text-danger" id="ocenaGreska"></p>
                        <select class="form-select" id="vrstaOcene"
name="vrstaOcene">
                            <option value="0">Izaberite...</option>
                            @foreach($vrsta as $v)
                                <option value="{{ $v->id_vrsta_ocene
}}">{{ $v->vrsta }}</option>
                            @endforeach
                        </select>
                        <p class="text-danger" id="vrstaGreska"></p>
                        <input type="button" class="btn"
id="btnOcena" value="Oceni">
                    </form>
                </div>
            </div>
        </div>
    </div>
    <div class="modal fade" id="ModalZakljucna" tabindex="-1"
role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
        <div class="modal-dialog modal-dialog-centered"
role="document">
            <div class="modal-content">
                <div class="modal-header">
                    <h5 class="modal-title"
id="exampleModalLongTitle">Zaključite ocenu</h5>
                    <button type="button" class="close btn" data-bs-
dismiss="modal" aria-label="Close">
                        <span aria-hidden="true"><i class="fa fa-
times" aria-hidden="true"></i></span>
                    </button>
                </div>
                <div class="modal-body" >
                    <form action="{{ route('zakljucnaOcena') }}"
method="post">
                        <input type="text" class="form-control w-
100" id="zakljucnaOcena" name="zakljucnaOcena" placeholder="Zaključite ocenu">
                        <p class="text-danger"
id="zakljucnaGreska"></p>
                        <select class="form-select mb-3
@if(count($zakljucnal)) sakrij @endif" id="polug" name="polug">
                            <option value="0">Izaberite...</option>
                            <option value="1">Prvo
polugodište</option>

```


Success.blade.php

```
@if(session()->has('success'))
    <div class="container mt-4">
        <div class="alert alert-success" role="alert">
            {{ session('success')}}
        </div>
    </div>
@endif
```

Form.blade.php

```
<form action="{{ route($action) }}" method="post" class="mt-3"
enctype='multipart/form-data'>
    @csrf
    <div class="mb-3">
        <label for="imeRegistracija" class="form-label">Ime</label>
        <input type="text" class="form-control" id="imeRegistracija"
name="imeRegistracija" />
        <p class="text-danger" id="imeRegistracijaG"></p>
        @error('imeRegistracija')
        <p class="text-danger">
            {{ $message}}
        </p>
        @enderror
    </div>
    <div class="mb-3">
        <label for="prezimeRegistracija" class="form-label">Prezime</label>
        <input type="text" class="form-control" id="prezimeRegistracija"
name="prezimeRegistracija" />
        <p class="text-danger" id="prezimeRegistracijaG"></p>
        @error('prezimeRegistracija')
        <p class="text-danger">
            {{ $message}}
        </p>
        @enderror
    </div>
    <div class="mb-3">
        <label for="korImeReg" class="form-label">Korisničko ime</label>
        <input type="text" class="form-control" id="korImeReg" name="korImeReg"
/>
        <p class="text-danger" id="korImeRegG"></p>
        @error('korImeReg')
        <p class="text-danger">
            {{ $message}}
        </p>
        @enderror
    </div>
    <div class="mb-3">
        <label for="emailReg" class="form-label">Email</label>
        <input type="text" class="form-control" id="emailReg" name="emailReg" />
        <p class="text-danger" id="emailRegG"></p>
        @error('emailReg')
        <p class="text-danger">
            {{ $message}}
        </p>
        @enderror
    </div>
    <div class="mb-3">
        <label for="sifraReg" class="form-label">Lozinka</label>
```



```

        <input type="password" class="form-control" id="sifraReg"
name="sifraReg" />
        <p class="text-danger" id="sifraRegG"></p>
        @error('sifraReg')
        <p class="text-danger">
            {{ $message }}
        </p>
        @enderror
    </div>
    @if($action == 'registracija')
        <div class="mb-3">
            <label for="slikaReg" class="form-label">Slika</label>
            <input type="file" class="form-control" id="slikaReg"
name="slikaReg" />
        </div>
        <div class="mb-3">
            <label for="predmetReg" class="form-label">Predmet</label>
            <select class="form-select" id="predmetReg" name="predmetReg">
                <option value="0">Izaberite...</option>
                @foreach($predmeti as $p)
                    <option value="{{ $p->id }}">{{ $p->naziv }}</option>
                @endforeach
            </select>
            @error('predmetReg')
            <p class="text-danger">
                {{ $message }}
            </p>
            @enderror
        </div>
        <div class="mb-3">
            <label for="ulogaReg" class="form-label">Uloga</label>
            <select class="form-select" id="ulogaReg" name="ulogaReg">
                <option value="0">Izaberite...</option>
                <option value="3">Nastavnik</option>
                <option value="2">Razredni starešina</option>
            </select>
            @error('ulogaReg')
            <p class="text-danger">
                {{ $message }}
            </p>
            @enderror
        </div>
    @else
        <div class="mb-3">
            <label for="brojTelefona" class="form-label">Broj telefona</label>
            <input type="text" class="form-control" id="brojTelefona"
name="brojTelefona" />
            <p class="text-danger" id="brojTelefonaG"></p>
        </div>
    @endif
    <div class="mt-5 d-flex justify-content-end">
        @if($action == 'registracija')
            <button type="submit" class="btn mb-5" id="registruj">Registruj</button>
        @else
            <button type="button" class="btn mb-5"
id="registrujRoditelj">Dodaj</button>
        @endif
    </div>
</form>

```

4.4. Routes

Web.php

```
<?php

use Illuminate\Support\Facades\Route;

/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/

Route::get('/', [\App\Http\Controllers\LoginController::class, 'index'])->
>name('loginPage');
Route::post('/login', [\App\Http\Controllers\LoginController::class, 'login'])->
>name('login');
Route::get('/logout', [\App\Http\Controllers\LoginController::class, 'logout'])->
>name('logout');

Route::get('/home', [\App\Http\Controllers\PocetnaController::class, 'index'])->
>name('home')->middleware('profesor');
Route::get('/home/direktor', [\App\Http\Controllers\PocetnaController::class,
'direktor'])->>name('direktor');

Route::get('/dnevnik/{id}', [\App\Http\Controllers\DnevnikController::class,
'index'])->>name('dnevnik')->middleware('profesor');
Route::get('/dnevnik/{id}/filter',
[\App\Http\Controllers\DnevnikController::class, 'filter'])->
>name('dnevnikFilter')->middleware('profesor');
Route::get('/dnevnik/{id}/paginacija',
[\App\Http\Controllers\DnevnikController::class, 'paginacija'])->
>name('paginacija')->middleware('profesor');
Route::get('/dnevnik/{id}/filterDatum',
[\App\Http\Controllers\DnevnikController::class, 'filterDatum'])->
>name('filterDatum')->middleware('profesor');
Route::get('/dnevnik/{id}/odsutni',
[\App\Http\Controllers\DnevnikController::class, 'odustni'])->
>name('dnevnikOdustni')->middleware('profesor');
Route::delete('/dnevnik/{id}/obrisi',
[\App\Http\Controllers\DnevnikController::class, 'obrisi'])->>name('obrisiCas')-
>middleware('profesor');
Route::get('/dnevnik/{id}/edit',
[\App\Http\Controllers\DnevnikController::class, 'edit'])->>name('edit')-
>middleware('profesor');
Route::put('/dnevnik/update', [\App\Http\Controllers\DnevnikController::class,
'update'])->>name('updateCas')->middleware('profesor');

Route::get('/cas/{id}', [\App\Http\Controllers\CasController::class, 'index'])->
>name('formaCas')->middleware('profesor');
Route::post('/cas/{id}/insert', [\App\Http\Controllers\CasController::class,
'insert'])->>name('dodajCas')->middleware('profesor');

Route::get('/profil', [\App\Http\Controllers\ProfilController::class, 'index'])->
>name('profil')->middleware('korisnik');
Route::put('/profil/izmeni', [\App\Http\Controllers\ProfilController::class,
```

```
'updateProfil']->name('izmeniProfil')->middleware('korisnik');
Route::put('/profil/izmeniLozinku',
[\App\Http\Controllers\ProfilController::class, 'izmeniLozinku'])-
>name('izmeniLozinku')->middleware('korisnik');

Route::get('/smerovi', [\App\Http\Controllers\SmeroviController::class,
'index'])->name('smerovi')->middleware('admin');
Route::post('/smerovi/insert', [\App\Http\Controllers\SmeroviController::class,
'insert'])->name('dodajSmer')->middleware('admin');
Route::delete('/smerovi/delete',
[\App\Http\Controllers\SmeroviController::class, 'delete'])-
>name('izbrisiSmer')->middleware('admin');

Route::get('/predmeti', [\App\Http\Controllers\PredmetiController::class,
'index'])->name('predmeti')->middleware('admin');
Route::post('/predmeti/insert',
[\App\Http\Controllers\PredmetiController::class, 'insert'])-
>name('dodajPredmet')->middleware('admin');
Route::delete('/predmeti/delete',
[\App\Http\Controllers\PredmetiController::class, 'delete'])-
>name('izbrisiPredmet')->middleware('admin');

Route::get('/nastavnici', [\App\Http\Controllers\NastavniciController::class,
'index'])->name('nastavnici')->middleware('admin');
Route::delete('/nastavnici/delete',
[\App\Http\Controllers\NastavniciController::class, 'delete'])-
>name('obrisiNastavnika')->middleware('admin');

Route::get('/ucenici', [\App\Http\Controllers\UceniciController::class,
'index'])->name('ucenici')->middleware('adminRazredni');
Route::post('/ucenici/insert', [\App\Http\Controllers\UceniciController::class,
'insert'])->name('dodajUcenika')->middleware('adminRazredni');

Route::get('/odeljenja', [\App\Http\Controllers\OdeljenjaController::class,
'index'])->name('odeljenja')->middleware('admin');
Route::get('/odeljenja/filter',
[\App\Http\Controllers\OdeljenjaController::class, 'filterPredmeti'])-
>name('odeljenjaPredmet')->middleware('admin');
Route::post('/odeljenja/insert',
[\App\Http\Controllers\OdeljenjaController::class, 'insert'])-
>name('dodajOdeljenje')->middleware('admin');

Route::get('/registracija',
[\App\Http\Controllers\RegistracijaController::class, 'index'])-
>name('registracijaForma')->middleware('adminRazredni');
Route::post('/registracija/insert',
[\App\Http\Controllers\RegistracijaController::class, 'insert'])-
>name('registracija')->middleware('adminRazredni');
Route::post('/roditelj', [\App\Http\Controllers\RegistracijaController::class,
'insertRoditelj'])->name('dodajKontakt')->middleware('adminRazredni');

Route::get('/odeljenje/{id}', [\App\Http\Controllers\OdeljenjaController::class,
'show'])->name('odeljenje')->middleware('profesor');
Route::get('/ucenik/{id}', [\App\Http\Controllers\UceniciController::class,
'show'])->name('ucenik')->middleware('profesor');
Route::get('/ucenik/{id}/filter',
[\App\Http\Controllers\UceniciController::class, 'filter'])-
>name('ucenikFilter')->middleware('profesor');
Route::put('/ucenik/izostanci', [\App\Http\Controllers\UceniciController::class,
'izostanci'])->name('izostanci')->middleware('razredni');
Route::post('/ucenik/ocena', [\App\Http\Controllers\UceniciController::class,
'oceni'])->name('ocena')->middleware('profesor');
Route::delete('/ucenik/izbrisi',
[\App\Http\Controllers\UceniciController::class, 'izbrisi']->name('izbrisi')-
```

```

>middleware('adminRazredni');
Route::post('/ucenik/zahtev', [\App\Http\Controllers\UceniciController::class,
'zahtev'])->name('zahtev')->middleware('profesor');
Route::post('/ucenik/zakljucna',
[\App\Http\Controllers\UceniciController::class, 'zakljuciOcenu'])->
>name('zakljucnaOcena')->middleware('profesor');
Route::put('/ucenik/vladanje', [\App\Http\Controllers\UceniciController::class,
'vladanje'])->name('vladanje')->middleware('razredni');

Route::get('/zahtevi', [\App\Http\Controllers\ZahteviController::class,
'index'])->name('zahtevi')->middleware('direktor');
Route::put('/zahtevi/odobri', [\App\Http\Controllers\ZahteviController::class,
'odobri'])->name('odobriZahtev')->middleware('direktor');
Route::put('/zahtevi/odbij', [\App\Http\Controllers\ZahteviController::class,
'odbij'])->name('odbijZahtev')->middleware('direktor');

Route::get('/roditelj', [\App\Http\Controllers\RoditeljController::class,
'index'])->name('roditelj')->middleware('roditelj');
Route::get('/roditelj/filter', [\App\Http\Controllers\RoditeljController::class,
'filter'])->name('roditeljFilter')->middleware('roditelj');

```

4.5. JS

Main.js

```

const token = $('meta[name="csrf-token"]').attr('content');

$.ajaxSetup({
  headers: {
    'X-CSRF-TOKEN': token
  }
})

$(document).ready(function () {
  $("#loginBtn").click(login);
  $("#izvrstiSmer").click(dodajSmer);
  $(".izbrisi").click(izbrisiSmer);
  $(".izbrisiP").click(izbrisiPredmet);
  $("#izvrstiPredmet").click(dodajPredmet);
  $("#smerOdeljenje").change(filterPredmet);
  $("#dodajOdeljenje").click(dodajOdeljenje);
  $(".datum").click(dohvatiCasove);
  $("#upisiBtn").click(upisiCas);
  // $("#predmetOcena").change(dohvatiOcene);
  $("#regIzostanke").click(regulisiIzostanke);
  $("#btnOcena").click(oceni);
  $("#zahtev").click(zahtev);
  $("#btnZakljuci").click(zakljuciOcenu);
  $("#registrujRoditelj").click(dodajKontakt);
  $("#vladanjeBtn").click(vladanje);
  $("#ucenikRoditelj").change(roditelj);
  $("#predmetiR").change(dohvatiOceneRoditelj);
  $(".izbrisiNastavnika").click(izbrisiNastavnika);
  $("#filterDatum").click(filterDatum);
  $(".page-link").click(loadMore);
  $("#updateCas").click(updateCas);

  if(window.location.href == baseUrl + "/home"){

```

```

        dohvatiIzostankeDirektor();
    }

    if(window.location.href == baseUrl + "/roditelj"){
        r();
    }

    for(let i=0; i<1000; i++){
        if(window.location.href == baseUrl + `/ucenik/${i}`){
            if(document.getElementById("predmetOcena").value == "0"){
                $(".izaberitePredmet").show();
                $(".polugodista").removeClass('d-flex');
                $(".prosek").removeClass('d-flex');
                $("#zakljucneUcenik").removeClass('d-flex');
                $("#zakljucnaNaslov").addClass('sakrij');
                $(".polugodista").hide();
                $(".prosek").hide();
                $("#zakljucneUcenik").hide();

                $("#predmetOcena").change(dohvatiOcene);
            }
            else {
                $(".izaberitePredmet").hide();
                $("#predmetOcena").change(dohvatiOcene);
            }
        }
    }

    $("#formaLozinka").hide();
    $("#izmeniLozinku").click(function (e) {
        e.preventDefault();
        $("#formaLozinka").slideToggle();
    });

    $("#formaSmer").hide();
    $("#dodajSmer").click(function (e) {
        e.preventDefault();
        $("#formaSmer").slideToggle();
    });

    $("#formaPredmet").hide();
    $("#dodajPredmet").click(function (e) {
        e.preventDefault();
        $("#formaPredmet").slideToggle();
    });

    $(".predmeti").hide();
    $(".informacije").hide();

    linkovi('.linkPredmeti', '.predmeti', '.ocene', '.informacije');
    linkovi('.linkInfo', '.informacije', '.ocene', '.predmeti');
    linkovi('.linkOcene', '.ocene', '.predmeti', '.informacije');

    $(".ucenikOdeljenje").hover(function () {
        $(this).addClass('shadow-lg')
    }, function () {
        $(this).removeClass('shadow-lg')
    })

    $('#btnModal').modal();
})

function linkovi(id, prikazi, sakrij1, sakrij2){
    $(id).click(function (e) {

```

```

        e.preventDefault();
        $(prikazi).show();
        $(sakrij1).hide();
        $(sakrij2).hide();
        $('.linkTab').removeClass('active');
        $(this).addClass('active');

        if(id == ".linkInfo"){
            $(".btnDodajKontakt").parent().removeClass('sakrij');
            $(".prikaziVladanje").addClass('sakrij');
            $("#prikaziZahtev").addClass('sakrij');
            $("#prikaziOcenu").addClass('sakrij');
            $("#prikaziZakljucnu").addClass('sakrij');
            $(".tekstVladanje").addClass('sakrij');
            $(".ocenaVladanje").addClass('sakrij');
        }
        else if(id == ".linkOcene"){
            $(".prikaziVladanje").removeClass('sakrij');
            $("#prikaziZahtev").removeClass('sakrij');
            $(".btnDodajKontakt").parent().addClass('sakrij');
            $("#prikaziOcenu").removeClass('sakrij');
            $("#prikaziZakljucnu").removeClass('sakrij');
            $(".tekstVladanje").removeClass('sakrij');
            $(".ocenaVladanje").removeClass('sakrij');
        }
        else {
            $(".prikaziVladanje").addClass('sakrij');
            $("#prikaziZahtev").addClass('sakrij');
            $(".btnDodajKontakt").parent().addClass('sakrij');
            $("#prikaziOcenu").addClass('sakrij');
            $("#prikaziZakljucnu").addClass('sakrij');
            $(".tekstVladanje").addClass('sakrij');
            $(".ocenaVladanje").addClass('sakrij');
        }
    })
}

function login() {
    var kor_ime, sifra, reKorIme;
    var validate = true;

    kor_ime = document.getElementById('kor_ime').value;
    sifra = document.getElementById('lozinka').value;
    reKorIme = /^[a-zA-Z0-9_\-\.]{5,}$/;

    if (kor_ime == "") {
        document.getElementById("kor_imeGreska").innerHTML = "Korisničko ime je obavezno.";
        validate = false;
    } else if (!reKorIme.test(kor_ime)) {
        document.getElementById("kor_imeGreska").innerHTML = "Korisničko ime nije ispravno.";
        validate = false;
    } else {
        document.getElementById("kor_imeGreska").innerHTML = "";
    }

    if (sifra == "") {
        document.getElementById("lozinkaGreska").innerHTML = "Šifra je obavezna.";
        validate = false;
    } else {
        document.getElementById("lozinkaGreska").innerHTML = "";
    }
}

```

```

    }

    if (validate) {
        $.ajax({
            url: baseUrl + "/login",
            method: "POST",
            data: {
                kor_ime: kor_ime,
                sifra: sifra,
                _token : token
            },
            success: function (data) {
                window.location.replace(baseUrl + data)
            },
            error: function (xhr, status, err) {
                if(xhr.status == 400){
                    alert(xhr.responseJSON.errorMsg);
                }
            }
        })
    }
}

function dodajSmer(){
    var naziv, reNaziv;
    var validate = true;

    naziv = document.getElementById("smer").value;
    reNaziv = /^[A-ZŠĐŽČĆ][a-zšđžćć]{2,}\s?([a-zšđžćć]{2,}\s?)*$/;

    if (naziv == "") {
        document.getElementById("smerGreska").innerHTML = "Naziv je obavezan.";
        validate = false;
    } else if (!reNaziv.test(naziv)) {
        document.getElementById("smerGreska").innerHTML = "Naziv može sadržati samo slova i mora početi velikim slovom.";
        validate = false;
    } else {
        document.getElementById("smerGreska").innerHTML = "";
    }

    if(validate){
        $.ajax({
            url: baseUrl + "/smerovi/insert",
            method: "POST",
            data: {
                naziv: naziv,
                _token : token
            },
            success: function (data) {
                alert("Uspešno ste dodali smer!");
                ispisi(data);
            },
            error: function (xhr, status, err) {
                if(xhr.status == 400){
                    alert(xhr.responseJSON.errorMsg);
                }
            }
        })
    }
}

function izbrisiSmer(){
    var id = this.dataset.id;

```

```

$.ajax({
  url: baseUrl + "/smerovi/delete",
  method: "delete",
  data: {
    id : id
  },
  success: function (data) {
    alert("Uspešno ste izbrisali smer.");
    ispisi(data);
  },
  error: function (xhr, status, err) {
    if(xhr.status == 400){
      alert(xhr.responseJSON.errorMsg);
    }
  }
})
}

function izbrisiNastavnika(){
  var id = this.dataset.id;

  $.ajax({
    url: baseUrl + "/nastavnici/delete",
    method: "delete",
    data: {
      id : id
    },
    success: function (data) {
      alert("Uspešno ste izbrisali nastavnika.");
      ispisiNastavnike(data);
    },
    error: function (xhr, status, err) {
      if(xhr.status == 400){
        alert(xhr.responseJSON.errorMsg);
      }
    }
  })
}

function ispisiNastavnike(data){
  let html = '';

  data.forEach(el => {
    html += `<tr>
      <td>${el.ime} ${el.prezime}</td>
      <td> ${el.naziv} </td>
      <td class="text-center">
        <form action="${baseUrl + "/nastavnici/delete"}"
method="delete">
          <button type="button" class="btn btn-sm btn-danger
izbrisiNastavnika" data-id="${el.id_korisnik}">
            <i class="fa fa-trash" aria-hidden="true"></i>
          </button>
        </form>
      </td>
    </tr>`
  })

  $("#ispisNastavnici").html(html);
  $(".izbrisiNastavnika").click(izbrisiNastavnika);
}

function ispisi(data){

```



```

let html = '';

data.forEach(el => {
    html += `<tr>
        <td> ${el.naziv} </td>
        <td class="text-center">
            <form action="${baseUrl + "/smerovi/delete"}"
method="delete">
                <button type="button" class="btn btn-sm btn-danger
izbrisi" data-id="${el.id}">
                    <i class="fa fa-trash" aria-hidden="true"></i>
                </button>
            </form>
        </td>
    </tr>`
})

$("#ispisSmerovi").html(html);
$(".izbrisi").click(izbrisiSmer);
}

function izbrisiPredmet() {
    var id = this.dataset.id;

    $.ajax({
        url: baseUrl + "/predmeti/delete",
        method: "delete",
        data: {
            id : id
        },
        success: function (data) {
            alert("Uspešno ste izbrisali predmet.");
            ispisiPredmete(data);
        },
        error: function (xhr, status, err) {
            if(xhr.status == 400){
                alert(xhr.responseJSON.errorMsg);
            }
        }
    })
}

function ispisiPredmete(data) {
    let html = '';

    data.forEach(el => {
        html += `<tr>
            <td> ${el.naziv} </td>
            <td class="text-center">
                <form action="${baseUrl + "/smerovi/delete"}"
method="delete">
                    <button type="button" class="btn btn-sm btn-danger
izbrisiP" data-id="${el.id}">
                        <i class="fa fa-trash" aria-hidden="true"></i>
                    </button>
                </form>
            </td>
        </tr>`
    })

    $("#ispisPredmeti").html(html);
    $(".izbrisiP").click(izbrisiPredmet);
}

```

```

function dodajPredmet(){
    var naziv, smer, reNaziv;
    var smerovi = [];
    var validate = true;
    naziv = document.getElementById('predmet').value;
    reNaziv = /^[A-ZŠĐŽČĆ][a-zšđžčć]{2,}\s?([a-zšđžčć]{2,}\s?)*$/;
    smer = $(".smerPredmet:checked");
    for (let i=0; i<smer.length; i++){
        smerovi.push(smer[i].value)
    }

    if (naziv == "") {
        document.getElementById("predmetGreska").innerHTML = "Naziv je obavezan.";
        validate = false;
    } else if (!reNaziv.test(naziv)) {
        document.getElementById("predmetGreska").innerHTML = "Naziv može sadržati samo slova i mora početi velikim slovom.";
        validate = false;
    } else {
        document.getElementById("predmetGreska").innerHTML = "";
    }

    if (smerovi.length == 0) {
        document.getElementById("smerPredmetGreska").innerHTML = "Smer je obavezan.";
        validate = false;
    } else {
        document.getElementById("smerPredmetGreska").innerHTML = "";
    }

    if(validate){

        $.ajax({
            url: baseUrl + "/predmeti/insert",
            method: "POST",
            data: {
                naziv: naziv,
                smerovi : smerovi,
                _token : token
            },
            success: function (data) {
                alert("Uspešno ste dodali predmet!");
                ispisipredmete(data);
            },
            error: function (xhr, status, err) {
                if(xhr.status == 400){
                    alert(xhr.responseJSON.errorMsg);
                }
            }
        })
    }
}

function filterPredmet(){
    var id = this.value;

    $.ajax({
        url: baseUrl + "/odeljenja/filter",
        method: "get",
        data: {
            id : id
        },
    },

```

```

        success: function (data) {
            ispisiPredmeteSmer(data);
        },
        error: function (xhr, status, err) {
            if(xhr.status == 400){
                alert(xhr.responseJSON.errorMsg);
            }
        }
    })
}

function ispisiPredmeteSmer(data) {
    let html = '<p class="mt-3 naslovPredmeti">Predmeti</p>';

    data.forEach(el => {
        html += '<div class="form-check">
                    <input class="form-check-input
predmetiOdeljenje" name="predmetiOdeljenje[]" type="checkbox"
value="{el.id_predmet}|{el.id_korisnik}">
                    <label class="form-check-label"
for="predmetiOdeljenje">
                        <span class="text-
danger">{el.naziv}</span> | {el.ime} {el.prezime}
                        </label>
                    </div>'
    })

    html += '<p class="text-danger" id="predmetiOdeljenjeG"></p>';
    $('#prikaziPredmete').html(html);
}

function dodajOdeljenje() {
    var broj, godina, razredni, smer, predmet, reBroj, reGodina;
    var validate = true;
    var predmeti = [];

    broj = document.getElementById('brojOdeljenja').value;
    godina = document.getElementById('godina').value;
    razredni = document.getElementById('razredni').value;
    smer = document.getElementById('smerOdeljenje').value;
    predmet = $(".predmetiOdeljenje:checked");
    for (let i=0; i<predmet.length; i++){
        predmeti.push(predmet[i].value)
    }

    reBroj = /^[1-9]+$//;
    reGodina = /^[1-4]+$//;

    if (broj == "") {
        document.getElementById("brojOdeljenjaG").innerHTML = "Broj odeljenja je obavezan.";
        validate = false;
    } else if (!reBroj.test(broj)) {
        document.getElementById("brojOdeljenjaG").innerHTML = "Dozvoljeni su samo brojevi.";
        validate = false;
    } else {
        document.getElementById("brojOdeljenjaG").innerHTML = "";
    }

    if (godina == "") {
        document.getElementById("godinaG").innerHTML = "Godina je obavezna.";
        validate = false;
    } else if (!reGodina.test(godina)) {

```

```

        document.getElementById("godinaG").innerHTML = "Dozvoljeni su samo
brojevi od 1 do 4.";
        validate = false;
    } else {
        document.getElementById("godinaG").innerHTML = "";
    }

    if (razredni == "0") {
        document.getElementById("razredniG").innerHTML = "Razredni starešina je
obavezan.";
        validate = false;
    } else {
        document.getElementById("razredniG").innerHTML = "";
    }

    if (smer == "0") {
        document.getElementById("smerOdeljenjeG").innerHTML = "Smer je
obavezan.";
        validate = false;
    } else {
        document.getElementById("smerOdeljenjeG").innerHTML = "";
    }

    if (predmet.length == 0) {
        document.getElementById("predmetiOdeljenjeG").innerHTML = "Predmeti su
obavezni.";
        validate = false;
    } else {
        document.getElementById("predmetiOdeljenjeG").innerHTML = "";
    }

    if(validate){
        $.ajax({
            url: baseUrl + "/odeljenja/insert",
            method: "post",
            data: {
                broj: broj,
                godina: godina,
                razredni:razredni,
                smer:smer,
                predmeti:predmeti,
                _token : token
            },
            success: function (data) {
                alert(data);
                window.location.reload();
            },
            error: function (xhr, status, err) {
                if(xhr.status == 400){
                    alert(xhr.responseJSON.errorMsg);
                }
            }
        })
    }
}

function dohvatiCasove(e){
    e.preventDefault();
    var datum = $(this).parent().next().val();
    var ulogovani = $(this).parent().next().next().val();
    var link = $(this);
    var id = window.location.href.split('/')[6];

```

```

$.ajax({
  url: baseUrl + "/dnevnik/" + id + "/filter",
  method: "get",
  data: {
    datum: datum
  },
  success: function (data) {
    ispisiCasove(data);
  },
  error: function (xhr, status, err) {
    if(xhr.status == 400){
      alert(xhr.responseJSON.errorMsg);
    }
  }
})

function ispisiCasove(data){
  let html = '';

  data.forEach(el => {
    html += `<div class="d-flex justify-content-start kolonaCasovi mt-3
mb-3">
      <div class="casRed">
        <p>${el.po_redu}. čas</p>
      </div>
      <div class="casOpis">
        <p class="ms-1 predmetCas">${el.naziv}</p>
        <span class="ms-1
nastavnaJedinica">${el.opis}</span>
        <p class="ms-1 mt-1 text-muted">${el.prezime}
${el.ime}</p>`
        if(ulogovani == el.id_korisnik){
          html += `<a href="#" class="btn-sm linkPocetna
linkObrisiCas float-end ms-1" title="Obriši čas"> <i class="fa fa-trash" aria-
hidden="true"></i> </a>
          <a href="#modalIzmeniCas" data-bs-
toggle="modal" class="btn-sm linkPocetna linkIzmeniCas float-end ms-1"
title="Izmeni čas"> <i class="fa fa-pencil" aria-hidden="true"></i> </a>`
        }
        html += `<a href="#exampleModalCenter" data-bs-
toggle="modal" class="btn-sm linkPocetna linkOdsutni float-end" title="Odsutni">
<i class="fa fa-user-times" aria-hidden="true"></i> </a>
        <input type="hidden" class="idcas" value="${el.id_cas}">
      </div>
    </div>`
  })

  link.parent().parent().find('.casovi').html(html);
  $('.linkOdsutni').click(dohvatiOdsutne)
  $('.linkObrisiCas').click(izbrisiCas)
  $('.linkIzmeniCas').click(ispisiFormuCas);
}

function dohvatiOdsutne(e){
  e.preventDefault()
  var id = $(this).next().val();

  $.ajax({
    url: baseUrl + "/dnevnik/" + id + "/odsutni",
    method: "get",
    data: {
      id : id
    },

```

```

        success: function (data) {
            ispisiOdsutne(data);
        },
        error: function (xhr, status, err) {
            if(xhr.status == 400){
                alert(xhr.responseJSON.errorMsg);
            }
        }
    })
}

function ispisiOdsutne(data) {
    let html = '';
    data.forEach(el => {
        html += `<p>${el.ime} ${el.prezime}</p>`
    })

    if(data.length > 0){
        $("#modalBody").html(html);
    }
    else {
        $("#modalBody").html("Nema odsutnih.");
    }
}

function izbrisiCas(e) {
    e.preventDefault();
    var id = $(this).next().next().next().val();

    $.ajax({
        url: baseUrl + "/dnevnik/" + id + "/obrisi",
        method: "delete",
        data: {
            id : id
        },
        success: function (data) {
            alert(data);
            window.location.reload();
        },
        error: function (xhr, status, err) {
            if(xhr.status == 400){
                alert(xhr.responseJSON.errorMsg);
            }
        }
    })
}

function ispisiFormuCas() {
    var id = $(this).next().next().val();

    $.ajax({
        url: baseUrl + "/dnevnik/" + id + "/edit",
        method: "get",
        data: {
            id : id
        },
        success: function (data) {
            data.forEach(el => {
                $("#nastavnaJedUpdate").val(el.opis);
                $("#napomenaUpdate").val(el.napomena);
                $("#casId").val(el.id_cas);
            })
        },
        error: function (xhr, status, err) {

```

```

        if(xhr.status == 400){
            alert(xhr.responseJSON.errorMsg);
        }
    }
})
}

function updateCas(){
    var opis, napomena, id;
    var validate = true;

    opis = $("#nastavnaJedUpdate").val();
    napomena = $("#napomenaUpdate").val();
    id = $("#casId").val();

    if(opis == ""){
        $("#nastavnaJedUpdateG").html('Nastavna jedinica je obavezna.');
```

validate = false;

```
    }
    else {
        $("#nastavnaJedUpdateG").html('');
    }

    if(validate){
        $.ajax({
            url: baseUrl + "/dnevnik/update",
            method: "put",
            data: {
                id : id,
                opis : opis,
                napomena : napomena
            },
            success: function (data) {
                alert(data);
                window.location.reload();
            },
            error: function (xhr, status, err) {
                if(xhr.status == 400){
                    alert(xhr.responseJSON.errorMsg);
                }
            }
        })
    }
}

function upisiCas() {
    var id, opis, datum, brojCasa, odsutni, napomena, reBroj;
    var odsutniNiz = [];
    var validate = true;

    opis = document.getElementById('opis').value;
    datum = document.getElementById('datumCas').value;
    brojCasa = document.getElementById('brojCasa').value;
    napomena = document.getElementById('napomena').value;
    odsutni = $(".odsutniCas:checked");
    for (let i = 0; i < odsutni.length; i++) {
        odsutniNiz.push(odsutni[i].value)
    }
    id = window.location.href.split('/')[6];
    reBroj = /^[0-9]+$//;

    if (opis == "") {
        document.getElementById("opisGreska").innerHTML = "Opis je obavezan.";
```

```

        validate = false;
    } else {
        document.getElementById("opisGreska").innerHTML = "";
    }

    if (datum == "") {
        document.getElementById("datumGreska").innerHTML = "Datum je obavezan.";
        validate = false;
    } else {
        document.getElementById("datumGreska").innerHTML = "";
    }

    if (brojCasa == "") {
        document.getElementById("brojCasaGreska").innerHTML = "Redni broj časa
je obavezan.";
        validate = false;
    } else if (!reBroj.test(brojCasa)){
        document.getElementById("brojCasaGreska").innerHTML = "Dozvoljeni su
samo brojevi.";
        validate = false;
    } else {
        document.getElementById("brojCasaGreska").innerHTML = "";
    }

    if(validate){
        $.ajax({
            url: baseUrl + "/cas/" + id + "/insert",
            method: "post",
            data: {
                id: id,
                opis: opis,
                datum: datum,
                brojCasa: brojCasa,
                odsutniNiz: odsutniNiz,
                napomena: napomena,
                _token: token
            },
            success: function (data) {
                alert(data);
                window.location.reload();
            },
            error: function (xhr, status, err) {
                if (xhr.status == 400) {
                    alert(xhr.responseJSON.errorMsg);
                }
            }
        })
    }
}

function dohvatiOcene(){
    var predmet = this.value;
    var idpredmet = document.getElementById('idpredmet').value;
    var id = window.location.href.split('/')[6];

    $.ajax({
        url: baseUrl + "/ucenik/" + id + "/filter",
        method: "get",
        data: {
            id : id,
            predmet: predmet
        },
        success: function (data) {
            ispisiOcene(data, ".polugodista");
        }
    });
}

```



```

opravdano = $('.selectOpravdano');
for(let i=0; i<opravdano.length; i++){
    nizOpravdano.push(opravdano[i].value);
}

for(let i=0; i<nizOpravdano.length; i++){
    if(nizOpravdano[i] == 'izaberite'){
        $('#selectOpravdanoG').html("Izaberite vrednost!")
        validate = false;
    }
    else {
        $('#selectOpravdanoG').html("")
    }
}

if(validate){
    $.ajax({
        url: baseUrl + "/ucenik/izostanci",
        method: "put",
        data: {
            nizId : nizId,
            nizOpravdano : nizOpravdano
        },
        success: function (data) {
            alert(data);
            window.location.reload();
        },
        error: function (xhr, status, err) {
            if(xhr.status == 400){
                alert(xhr.responseJSON.errorMsg);
            }
        }
    })
}
}

function oceni(){
    var ocena, vrsta, reBroj;
    var validate = true;
    var id = window.location.href.split('/')[6];
    ocena = document.getElementById('ocenaVred').value;
    vrsta = document.getElementById('vrstaOcene').value;
    reBroj = /^[1-5]$/;

    if (ocena == "") {
        document.getElementById("ocenaGreska").innerHTML = "Ocena je obavezna.";
        validate = false;
    } else if (!reBroj.test(ocena)) {
        document.getElementById("ocenaGreska").innerHTML = "Dozvoljeni su samo
brojevi od 1 do 5.";
        validate = false;
    } else {
        document.getElementById("ocenaGreska").innerHTML = "";
    }

    if (vrsta == "0") {
        document.getElementById("vrstaGreska").innerHTML = "Vrsta ocene je
obavezna.";
        validate = false;
    } else {
        document.getElementById("vrstaGreska").innerHTML = "";
    }
}

```

```

    if (validate){
        $.ajax({
            url: baseUrl + "/ucenik/ocena",
            method: "post",
            data: {
                ocena : ocena,
                vrsta : vrsta,
                id : id,
                _token : token
            },
            success: function (data) {
                alert(data);
                window.location.reload();
            },
            error: function (xhr, status, err) {
                if(xhr.status == 400){
                    alert(xhr.responseJSON.errorMsg);
                }
            }
        })
    }
}

function zahtev(){
    var pogresna, nova, datum, vrsta, reBroj;
    var id = window.location.href.split('/')[6];
    var validate = true;

    pogresna = document.getElementById('pogresnaOcena').value;
    nova = document.getElementById('novaOcena').value;
    datum = document.getElementById('datumOcena').value;
    vrsta = document.getElementById('vrstaOcena').value;
    reBroj = /^[1-5]$/;

    if(pogresna == ""){
        document.getElementById('pogresnaOcenaG').innerHTML = 'Pogrešna ocena je obavezna!'
        validate = false;
    } else if (!reBroj.test(pogresna)){
        document.getElementById('pogresnaOcenaG').innerHTML = 'Dozvoljeni su samo brojevi od 1 do 5!'
        validate = false;
    } else {
        document.getElementById('pogresnaOcenaG').innerHTML = ''
    }

    if(nova == ""){
        document.getElementById('novaOcenaG').innerHTML = 'Nova ocena je obavezna!'
        validate = false;
    } else if (!reBroj.test(nova)){
        document.getElementById('novaOcenaG').innerHTML = 'Dozvoljeni su samo brojevi od 1 do 5!'
        validate = false;
    } else {
        document.getElementById('novaOcenaG').innerHTML = ''
    }

    if(datum == ""){
        document.getElementById('datumOcenaG').innerHTML = 'Datum je obavezan!'
        validate = false;
    } else {

```

```

        document.getElementById('datumOceneG').innerHTML = ''
    }

    if(vrsta == "0"){
        document.getElementById('vrstaOceneZG').innerHTML = 'Vrsta ocene je obavezna!'
        validate = false;
    } else {
        document.getElementById('vrstaOceneZG').innerHTML = ''
    }

    if(validate){
        $.ajax({
            url: baseUrl + "/ucenik/zahtev",
            method: "post",
            data: {
                pogresna : pogresna,
                nova : nova,
                datum : datum,
                vrsta : vrsta,
                id : id,
                _token : token
            },
            success: function (data) {
                alert(data);
                window.location.reload();
            },
            error: function (xhr, status, err) {
                if(xhr.status == 400){
                    alert(xhr.responseJSON.errorMsg);
                }
            }
        })
    }
}

function zakljuciOcenu(){
    var ocena, polugodiste;
    var id = window.location.href.split('/')[6];
    var validate = true;

    ocena = document.getElementById('zakljucnaOcena').value;
    reBroj = /^[1-5]$/;

    if (ocena == "") {
        document.getElementById("zakljucnaGreska").innerHTML = "Ocena je obavezna.";
        validate = false;
    } else if (!reBroj.test(ocena)) {
        document.getElementById("zakljucnaGreska").innerHTML = "Dozvoljeni su samo brojevi od 1 do 5.";
        validate = false;
    } else {
        document.getElementById("zakljucnaGreska").innerHTML = "";
    }

    if(document.getElementById('polug') != undefined){
        polugodiste = document.getElementById('polug').value;
    }

    if(validate){
        $.ajax({
            url: baseUrl + "/ucenik/zakljucna",
            method: "post",

```

```

        data: {
            ocena : ocena,
            polugodiste : polugodiste,
            id : id,
            _token : token
        },
        success: function (data) {
            alert(data);
            window.location.reload();
        },
        error: function (xhr, status, err) {
            if(xhr.status == 400){
                alert(xhr.responseJSON.errorMsg);
            }
        }
    })
}

function dodajKontakt(){
    var ime, prezime, kor_ime, email, lozinka, broj, reIme, rePrezime, reKorIme,
    reEmail, reBroj;
    var id = window.location.href.split('/')[6];
    var validate = true;

    ime = document.getElementById('imeRegistracija').value;
    prezime = document.getElementById('prezimeRegistracija').value;
    kor_ime = document.getElementById('korImeReg').value;
    email = document.getElementById('emailReg').value;
    lozinka = document.getElementById('sifraReg').value;
    broj = document.getElementById('brojTelefona').value;
    reIme = /^[A-ZŠĐŽČĆ][a-zšđžčć]{1,13}$/;
    rePrezime = /^[A-ZŠĐŽČĆ][a-zšđžčć]{1,30}\s?)/;
    reKorIme = /^[A-Za-z0-9\.\_]{3,20}$/;
    reEmail = /^[a-z][a-z\d\_\.\-]+\@[a-z\d](\.[a-z]{2,4})+$/;
    reBroj = /^[0-9]+$/;

    if (ime == "") {
        document.getElementById("imeRegistracijaG").innerHTML = "Ime je obavezno.";
        validate = false;
    } else if (!reIme.test(ime)) {
        document.getElementById("imeRegistracijaG").innerHTML = "Ime nije u ispravnom formatu";
        validate = false;
    } else {
        document.getElementById("imeRegistracijaG").innerHTML = "";
    }

    if (prezime == "") {
        document.getElementById("prezimeRegistracijaG").innerHTML = "Prezime je obavezno.";
        validate = false;
    } else if (!rePrezime.test(prezime)) {
        document.getElementById("prezimeRegistracijaG").innerHTML = "Prezime nije u ispravnom formatu";
        validate = false;
    } else {
        document.getElementById("prezimeRegistracijaG").innerHTML = "";
    }

    if (kor_ime == "") {
        document.getElementById("korImeRegG").innerHTML = "Korisničko ime je obavezno.";
    }
}

```

```

        validate = false;
    } else if (!reKorIme.test(kor_ime)) {
        document.getElementById("korImeRegG").innerHTML = "Korisničko ime nije u
ispravnom formatu";
        validate = false;
    } else {
        document.getElementById("korImeRegG").innerHTML = "";
    }

    if (email == "") {
        document.getElementById("emailRegG").innerHTML = "Ime je obavezno.";
        validate = false;
    } else if (!reEmail.test(email)) {
        document.getElementById("emailRegG").innerHTML = "Email nije u ispravnom
formatu";
        validate = false;
    } else {
        document.getElementById("emailRegG").innerHTML = "";
    }

    if (broj == "") {
        document.getElementById("brojTelefonaG").innerHTML = "Broj je
obavezan.";
        validate = false;
    } else if (!reBroj.test(broj)) {
        document.getElementById("brojTelefonaG").innerHTML = "Broj nije u
ispravnom formatu";
        validate = false;
    } else {
        document.getElementById("brojTelefonaG").innerHTML = "";
    }

    if (lozinka == "") {
        document.getElementById("sifraRegG").innerHTML = "Lozinka je obavezna.";
        validate = false;
    } else if (lozinka.length < 8) {
        document.getElementById("sifraRegG").innerHTML = "Lozinka mora imati
minimalno 8 karaktera";
        validate = false;
    } else {
        document.getElementById("sifraRegG").innerHTML = "";
    }

    if(validate){
        $.ajax({
            url: baseUrl + "/roditelj",
            method: "post",
            data: {
                ime : ime,
                prezime : prezime,
                kor_ime : kor_ime,
                email : email,
                lozinka : lozinka,
                broj : broj,
                id : id
            },
            success: function (data) {
                alert(data);
                window.location.reload();
            },
            error: function (xhr, status, err) {
                if(xhr.status == 400){
                    alert(xhr.responseJSON.errorMsg);
                }
            }
        });
    }

```



```

    }
    })
}

function vladanje(){
    var id = window.location.href.split('/')[6];
    var ocena = document.getElementById('vladanje').value;
    reBroj = /^[1-5]$/;
    var validate = true;

    if(ocena == ""){
        document.getElementById('vladanjeG').innerHTML = 'Ocena je obavezna!'
        validate = false;
    } else if (!reBroj.test(ocena)){
        document.getElementById('vladanjeG').innerHTML = 'Dozvoljeni su samo
brojevi od 1 do 5!'
        validate = false;
    } else {
        document.getElementById('vladanjeG').innerHTML = ''
    }

    if(validate){
        $.ajax({
            url: baseUrl + "/ucenik/vladanje",
            method: "put",
            data: {
                id : id,
                ocena : ocena
            },
            success: function (data) {
                alert(data);
                window.location.reload();
            },
            error: function (xhr, status, err) {
                if(xhr.status == 400){
                    alert(xhr.responseJSON.errorMsg);
                }
            }
        })
    }
}

function r(){
    var id = $("#ucenikRoditelj option:selected").val()
    var idpredmet = $("#predmetiR").val();

    $.ajax({
        url: baseUrl + "/roditelj/filter",
        method: "get",
        data: {
            id : id,
            idpredmet : idpredmet
        },
        success: function (data) {
            ispisiIzostanci(data)
            ispisiLevoRoditelj(data.ucenik)
            ispisiPredmeteRoditelj(data.predmeti)
            ispisiPadajucuListu(data.predmeti)
        },
        error: function (xhr, status, err) {
            if(xhr.status == 400){
                alert(xhr.responseJSON.errorMsg);
            }
        }
    })
}

```

```

    }
  })
}

function ispisiLevoRoditelj(data) {
  let html = "";

  data.forEach(el => {
    html += `<div class="slikaIme">
      <img class="img-fluid mb-2 mt-4 slika" src=
"assets/images/${el.slika}">
      <div class="imeSmer">
        <p class="ime mt-2">${el.prezime} ${el.ime}</p>
        <p class="text-muted">${el.naziv} (4) </p>
      </div>
    </div>
    <div class="w-75 m-auto d-flex mb-3 justify-content-start
divTekstVladanje vroditelj">
      <p class="tekstVladanje">Ocena iz vladanja</p>
      <p class="ocenaVladanje">${el.vladanje}</p>
    </div>`
  })

  $("#roditeljLevo").html(html);
}

function dohvatiOceneRoditelj() {
  var predmet = this.value;
  var ucenik = $("#ucenikRoditelj").val();

  $.ajax({
    url: baseUrl + "/roditelj/filter",
    method: "get",
    data: {
      predmet: predmet,
      ucenik: ucenik
    },
    success: function (data) {
      if (predmet != "0") {
        $(".izaberitePredmet").hide();
        $("#oceneRoditelj").addClass('d-flex');
        $("#prosekPrvoR").addClass('d-flex');
        $("#prosekDrugoR").addClass('d-flex');
        $("#zakljucneRoditelj").addClass('d-flex');
        $("#zakljucnaNaslov").removeClass('sakrij');
        ispisiOcene(data, "#oceneRoditelj");
        ispisiProsek(data.prosek, "#prosekPrvoR");
        ispisiProsek(data.prosek2, "#prosekDrugoR");
        ispisiZakljucnu(data, "#zakljucneRoditelj");
      }
      else {
        $(".izaberitePredmet").show();
        $("#oceneRoditelj").removeClass('d-flex');
        $("#prosekPrvoR").removeClass('d-flex');
        $("#prosekDrugoR").removeClass('d-flex');
        $("#zakljucneRoditelj").removeClass('d-flex');
        $("#zakljucnaNaslov").addClass('sakrij');
        $("#oceneRoditelj").hide();
        $("#prosekPrvoR").hide();
        $("#prosekDrugoR").hide();
        $("#zakljucneRoditelj").hide();
      }
    }
  },

```

```

        error: function (xhr, status, err) {
            if(xhr.status == 400){
                alert(xhr.responseJSON.errorMsg);
            }
        }
    })
}

function roditelj(){
    var id = this.value;

    $.ajax({
        url: baseUrl + "/roditelj/filter",
        method: "get",
        data: {
            id : id
        },
        success: function (data) {
            ispisiIzostanci(data)
            ispisiLevoRoditelj(data.ucenik)
            ispisiPredmeteRoditelj(data.predmeti)
            ispisiPadajucuListu(data.predmeti)
        },
        error: function (xhr, status, err) {
            if(xhr.status == 400){
                alert(xhr.responseJSON.errorMsg);
            }
        }
    })
}

function ispisiIzostanci(data) {

    let html = `<div class="col-lg-3 col-md-3 col-6 mt-3 d-flex align-items-center">

                <div class="izostanci neregulisani">
                    <span>${data.neregulisani.length}</span>
                </div>
                <div class="d-flex align-items-center">
                    <p class="text-center mt-2 text-muted ms-2">neregulisani</p>
                </div>
            </div>
            <div class="col-lg-3 col-md-3 col-6 mt-3 d-flex align-items-center">

                <div class="izostanci opravdani">
                    <span>${data.opravdano.length}</span>
                </div>
                <div class="d-flex align-items-center">
                    <p class="text-center mt-2 text-muted ms-2">opravdani</p>
                </div>
            </div>
            <div class="col-lg-3 col-md-3 col-6 mt-3 d-flex align-items-center">

                <div class="izostanci neopravdani">
                    <span>${data.neopravdano.length}</span>
                </div>
                <div class="d-flex align-items-center">
                    <p class="text-center mt-2 text-muted ms-2">neopravdani</p>
                </div>
            </div>
            <div class="col-lg-3 col-md-3 col-6 mt-3 d-flex align-

```

```

items-center">
                <div class="izostanci ukupno">
                    <span>${data.neopravdano.length +
data.opravdano.length + data.neregulisani.length}</span>
                </div>
                <div class="d-flex align-items-center">
                    <p class="text-center mt-2 text-muted ms-
2">ukupno</p>
                </div>
            </div>`

    $("#izostanciRoditelj").html(html)
}

function ispisiPredmeteRoditelj(data){
    let html = '';

    data.forEach(el => {
        html += `<tr>
                <td class="nazivPredmeta">${el.naziv}</td>
                <td>${el.prezime} ${el.ime}</td>
            </tr>`
    })

    $("#predmetiRoditelj").html(html);
}

function ispisiPadajucuListu(data){
    let html = '<option value="0">Izaberite predmet...</option>';

    data.forEach(el => {
        html += `<option value='${el.id_predmet}'>${el.naziv}</option>`
    })

    $("#predmetiR").html(html)
}

function charts(id, labels, label, data, background, border){
    if(document.getElementById(id) != null){
        var ctx = document.getElementById(id).getContext('2d');
        var myChart = new Chart(ctx, {
            type: 'bar',
            data: {
                labels: labels,
                datasets: [{
                    label: label,
                    data: data,
                    backgroundColor: background,
                    borderColor: border,
                    borderWidth: 1
                }]
            },
            options: {
                scales: {
                    yAxes: [{
                        ticks: {
                            beginAtZero: true
                        }
                    }]
                }
            }
        });
    }
}

```

```

}

function dohvatiIzostankeDirektor() {
    $.ajax({
        url: baseUrl + "/home/direktor",
        method: "get",
        success: function (data) {
            charts(
                "izostanciChart",
                ["Neregulisani", "Opravdani", "Neopravdani", "Ukupno"],
                'Izostanci',
                [
                    data.neregulisani.length,
                    data.opravdano.length,
                    data.neopravdano.length,
                    data.opravdano.length + data.neopravdano.length +
data.neregulisani.length
                ],
                [
                    'rgba(54, 162, 235, 0.2)',
                    'rgba(75, 192, 192, 0.2)',
                    'rgba(255, 99, 132, 0.2)',
                    'rgba(153, 102, 255, 0.2)'
                ],
                [
                    'rgba(54, 162, 235, 1)',
                    'rgba(75, 192, 192, 1)',
                    'rgba(255,99,132,1)',
                    'rgba(153, 102, 255, 1)'
                ]
            );
            charts(
                "zahteviChart",
                ["Odobreni", "Neodobreni", "Ukupno"],
                'Zahtevi za izmenu ocene',
                [
                    data.odobreni.length,
                    data.neodobreni.length,
                    data.odobreni.length + data.neodobreni.length
                ],
                [
                    'rgba(75, 192, 192, 0.2)',
                    'rgba(255, 99, 132, 0.2)',
                    'rgba(153, 102, 255, 0.2)'
                ],
                [
                    'rgba(75, 192, 192, 1)',
                    'rgba(255,99,132,1)',
                    'rgba(153, 102, 255, 1)'
                ]
            );
        },
        error: function (xhr, status, err) {
            if(xhr.status == 400){
                alert(xhr.responseJSON.errorMsg);
            }
        }
    })
}

function filterDatum() {
    var datum = document.getElementById('casDatum').value;
    var id = window.location.href.split('/')[6];

```



```

function getDates(page) {
    const caller = arguments.callee.caller.name;
    var id = window.location.href.split('/')[6];

    $.ajax({
        url: baseUrl + "/dnevnik/" + id + "/paginacija",
        method: "get",
        data: {
            page : page
        },
        dataType: "json",
        success: function (response) {
            console.log(response)
            if(response.data.length > 0){
                ispisiDatum(response.data);
                if(caller == 'loadMore'){
                    changeActivePageLink(response.current_page);
                }
                $(".pagination").show();
            }
            else {
                // $("#products").html("<div class='row mb-5'><div class='col-12 d-flex justify-content-center'><h2>No products in this category!</h2></div></div>");
                $(".pagination").hide();
            }
        },
        error: function (xhr, status, err) {
            console.log(xhr.status)
        }
    })
}

function changeActivePageLink(currentPage) {
    $(".page-item").removeClass('active');
    $(".#link" + currentPage).parent().addClass('active');
}

```

4.6. CSS

Style.css

```

@import url('https://fonts.googleapis.com/css2?family=Fira+Sans&display=swap');

body {
    font-family: 'Fira Sans', sans-serif;
    background-color: #f1f3f7;
}

table {
    background-color: #FFFFFF;
}

.containerMargina {
    margin-top: 60px;
}

```

```
}

.containerMarginal {
    margin-top: 100px;
}

#loginBody {
    background-image: url("../images/pozadinal.jpg");
    background-repeat: no-repeat;
    background-size: cover;
    background-attachment: fixed;
}

#tabelaLogin {
    min-height: 50vh;
    margin-top: 10px;
}

#loginKol {
    background-color: rgba(0, 0, 0, 0.6);
    color: #ffffff;
}

#loginKol form {
    width: 80%;
}

#loginBtn {
    margin-top: 20px;
    background-color: #f8f9fa;
    color: #212529;
}

#loginBtn:hover {
    background-color: #f9fafb;
}

#naslov {
    margin-top: 15vh;
    font-size: 4vw;
    color: #ffffff;
}

#knjiga {
    margin-top: 18vh;
    width: 20%;
}

.navbar {
    background: linear-gradient(
        120deg
        , #194872 50%, #2daadf);
}

.navbar-brand {
    color: #FFFFFF !important;
}

.dropdown-menu {
    background: #194872;
}

.dropdown-menu li a {
    color: #FFFFFF;
}
```



```
}

.nastavnik {
    background: #FFFFFF;
}
.nastavnik li a {
    color: black;
}

.nav-link {
    color: #FFFFFF!important;
}

#ikonice {
    list-style-type: none;
    display: inline-flex;
    flex-direction: row;
    float: right;
}

#ikonice li {
    margin-right: 10px;
}
#ikonice li a{
    color: #4a4a4a;
}

#listaProfil {
    position: absolute;
}

.profilnaSlika {
    width: 70%;
    border-radius: 50%;
}

#futer {
    width: 100%;
    padding: 1%;
    border-top: 1px solid #c2c2c2;
}

#formaLozinka {
    position: static;
}

.visina {
    min-height: 80vh;
}

#slikaPrijavljen {
    border-radius: 50%;
    position: absolute;
    top: 0;
    margin-left: 15px;
    width: 65px;
    height: 65px;
}

#divPrijavljen {
    margin-right: 5%;
    position: absolute;
    top: 0;
```

```
    right: 0;
}

#tekstPrijavljen {
    margin-top: 6%;
}

#dodajSmer {
    margin-top: 25px;
    margin-bottom: 15px;
    width: 15%;
}

#dodajPredmet {
    margin-top: 25px;
    margin-bottom: 15px;
    width: 20%;
}

#izvrsiSmer {
    margin-left: 10%;
}

#smer {
    width: 25vw;
}

.ucenikOdeljenje {
    background-color: #ffffff;
    padding: 1%;
    width: 18%;
    margin-right: 2%;
    margin-bottom: 2%;
}

.ucenikSlika {
    border-radius: 50%;
}

.ucenikIme {
    margin: 15px 0px;
}

.ucenikIme a {
    color: black;
    text-decoration: none;
    font-size: 1.25vw;
}

#odeljenjeGore {
    border-bottom: 1px solid #dce2e6;
    margin-bottom: 20px;
}

#odeljenjeGore div h2 {
    font-size: 1.1vw;
}

#odeljenjeGore div p {
    font-size: 0.9vw;
}

.badge {
```

```
    color: black;
    padding: 0px;
    font-weight: normal;
}

.ocene .row {
    background-color: #ffffff;
}

.ocene {
    padding-left: 2%;
}

.odeljenjePocetna {
    padding: 1%;
    background-color: #ffffff;
    width: 23%;
    margin-right: 2%;
}

.linija {
    width: 4px;
    height: 95%;
    margin-right: 6%;
    background-color: #478ad4;
    border-radius: 2px;
}

.brojOdeljenja {
    font-size: 20px;
    font-weight: bold;
    color: #478ad4;
}

.brojOdeljenja span {
    color: #478ad4;
    font-weight: bold;
}

.card-title, .card-subtitle {
    font-weight: normal;
}

.card-title {
    color: #194872;
}

.linkPocetna {
    background-color: #eef4f8;
    text-decoration: none;
    color: #478ad4;
    border-color: #d1elec;
    margin-right: 1.5%;
}

.linkPocetna:hover {
    background-color: #478ad4;
    color: #ffffff;
}

.kolonaCasovi {
    background-color: #ffffff;
    width: 23%;
    margin-right: 1%;
}
```

```
.predmetCas {
  margin-bottom: 8px!important;
}

.nastavnaJedinica {
  color: #194872;
  font-size: 1.5vw;
}

.casRed {
  height: 100%;
  display: flex;
  align-items: center;
  border-right: 1px solid #dce2e6;
  padding-right: 10px;
}

.casOpis {
  margin-left: 10px;
  margin-top: 10px;
  padding-bottom: 3%;
  width: 75%;
}

.datumCas {
  width: 100%;
  padding: 1% 0%;
  text-align: center;
  background-color: #f8f9fa;
  margin-bottom: 10px;
  border: 0.5px solid #478ad4;
}

.datumCas:hover {
  background-color: #478ad4;
}

.datumCas:hover a {
  color: #FFFFFF;
}

.datumCas a {
  color: black;
  text-decoration: none;
}

.linkOdsutni {
  margin-right: 0%!important;
}

#prazno {
  min-height: 50vh;
}

#prazno p {
  font-size: 4vw;
}

.izostanci {
  width: 25%;
  /*margin: auto;*/
  text-align: center;
  border-radius: 6px;
```

```

        font-size: 20px;
        font-weight: bold;
        padding: 3% 0%;
    }

    .neregulisani {
        border: 4px solid #478ad4;
        color: #478ad4;
    }

    .neregulisani a {
        text-decoration: none;
    }

    .opravdani {
        border: 4px solid #6eb981;
        color: #6eb981;
    }

    .neopravdani {
        border: 4px solid #db2b2e;
        color: #db2b2e;
    }

    .ukupno {
        border: 4px solid #7743a0;
        color: #7743a0;
    }

    .ocena {
        width: 12%;
        background-color: #3a668c;
        padding: 1% 0%;
        text-align: center;
        font-size: 18px;
        color: #FFFFFF;
        font-weight: 700;
        margin-right: 1%;
        margin-bottom: 10px;
    }

    #tabelaPredmeti, .informacije {
        background-color: #ffffff;
    }

    .neregulisaniSelect {
        width: 100% !important;
        margin-left: 0 !important;
        margin-top: 5px;
    }

    .selectOpravdano {
        width: 100% !important;
    }

    .informacije {
        padding-top: 2%;
        padding-left: 2%;
    }

    .informacije .col-lg-6 {
        float: left;
    }

```

```
#formaZahtev {
  background-color: #ffffff;
  margin-top: 20px;
}

.zakljucna {
  width: 15%;
  background-color: #de445f;
  padding: 1% 0%;
  text-align: center;
  font-size: 18px;
  color: #ffffff;
  font-weight: bold;
  margin-left: 50%;
  margin-bottom: 10px;
}

.sakrij {
  display: none;
}

.ucenik {
  background-color: #ffffff;
}

.levo {
  border-right: 1px solid #dee2e6;
}

.ucenik .col-lg-9 {
  padding: 0%!important;
}

.slika {
  border-radius: 50%;
  width: 60%;
}

.ime {
  font-size: 1.3vw;
  margin-bottom: 0px!important;
}

.linkTab {
  color: #7a8690!important;
}

.linkTab.active {
  color: black!important;
}

.tekstVladanje {
  background-color: #eef4f8;
  color: #3a668c;
  border-color: #d1e1ec;
  width: 80%;
  padding: 1.5% 0;
  margin-bottom: 0!important;
  border-radius: 4px 0 0 4px;
}

.ocenaVladanje {
  background-color: #3a668c;
  color: #fff;
}
```

```
    font-weight: 700;
    width: 20%;
    padding: 1.5% 0;
    margin-bottom: 0 !important;
    border-radius: 0 4px 4px 0;
}

.srednja, .datumNeregulisani {
    background-color: #e9ecef;
    width: 70%;
    min-height: 5vh;
    display: flex;
    justify-content: flex-end;
    align-items: center;
    border-radius: 4px 0 0 4px;
}

.srednja {
    width: 89.5%;
}

.datumNeregulisani {
    width: 90%;
    justify-content: flex-start;
}

.srednjaOcena, .casNeregulisani {
    background-color: #f8f9fa;
    width: 10%;
    min-height: 5vh;
    display: flex;
    justify-content: center;
    align-items: center;
    border-radius: 0 4px 4px 0;
}

.srednjaOcena p, .srednja p, .casNeregulisani p, .datumNeregulisani p {
    margin-bottom: 0px !important;
}

#neregulisaniIzostanci {
    width: 95%;
}

.datumNeregulisani p {
    margin-left: 1vw;
}

.srednja p {
    margin-right: 1vw;
}

#ocenaVred, #zakljucnaOcena {
    width: 40%;
}

.navigacija {
    background-color: #f1f3f7;
}

.predmeti {
    padding-left: 2%;
}
```

```
}

.naslovRoditelj {
  margin-top: 30px;
}

#redProfil {
  background-color: #FFFFFF;
}

.ucenikContainer {
  margin-top: 110px;
}

#predmetOcena, #predmetiR {
  width: 95%;
}

.naslovInfo {
  border-bottom: 1px solid #dce2e6;
  width: 90%;
}

.naslovPolugodiste {
  border-top: 1px solid #478ad4;
  width: 90%;
}

.polugodiste {
  color: #478ad4;
  text-transform: uppercase;
  font-weight: 700;
  font-size: 16px;
  margin-top: 5px;
}

.naslovPredmeti {
  margin-bottom: 5px !important;
}

.smerOdeljenje {
  font-size: 1vw;
  margin-top: 5px;
}

.naslovProfil {
  color: #194872;
  font-weight: 700;
  text-transform: uppercase;
  font-size: 1.3vw;
}

.bezbednost {
  margin-top: 80px;
  border-top: 1px solid #dce2e6;
}

.bezbednost h3 {
  margin-top: 20px;
}

#redProfil .col-lg-3 {
  border-right: 1px solid #dce2e6;
}
```



```
#redProfil .col-lg-8 {
  padding-left: 3%;
}

.imePrezime {
  font-size: 2vw;
  color: #194872;
  font-weight: normal;
}

.dan {
  font-size: 1.3vw;
}

.d {
  font-size: 1.1vw;
  margin-top: 5px;
}

.page-item.active .page-link {
  background-color: #478ad4;
  border-color: #478ad4;
}

.page-link {
  color: #478ad4;
}

.btn {
  background-color: #de445f;
  color: #FFFFFF;
}

.btn:hover {
  background-color: #dc3545;
  color: #FFFFFF;
}

.close.btn {
  background-color: #FFFFFF;
  color: black;
}

.success {
  background-color: #28a745;
}

.success:hover {
  background-color: #157347;
}

.nemaZahteva {
  font-size: 5vw;
  font-weight: normal;
}

#tabelaPredmeti thead tr th {
  color: #194872;
  text-transform: uppercase;
}

.nazivPredmeta {
  color: #3366bb;
}
```

```
}

#prikaziZakljucnu, #prikaziZahtev {
  background-color: #eef4f8;
  color: #3a668c;
  border-color: #d1e1ec;
}

#prikaziZakljucnu:hover, #prikaziZahtev:hover {
  background-color: #3a668c;
  color: #fff;
}

#zakljucnaNaslov, .izostanciNaslov {
  border-top: 1px solid #478ad4;
  width: 95%;
}

#zakljucnaNaslov h5, .izostanciNaslov h5 {
  color: #478ad4;
  text-transform: uppercase;
  font-weight: 700;
  font-size: 16px;
  margin-top: 5px;
}

.zakljucnaPolugodiste {
  width: 46.7%;
  background: #f8f9fa;
  padding: 10px;
  border-radius: 4px;
  border: 2px solid transparent;
}

.zakljucnaPolugodiste:hover {
  background: #e9ecef;
}

.zakljucnaPolugodiste h6 {
  color: #194872;
  font-weight: 700;
  font-size: 16px;
}

.zakljucnaPolugodiste p {
  font-size: 18px;
  color: #343a40;
  margin-bottom: 0px !important;
}

.prosekPrvo, .prosekDrugo, #prosekPrvoR, #prosekDrugoR {
  width: 46.7%;
}

.padajucaRoditelj {
  padding: 0 !important;
}

.labelUcenik {
  font-size: 18px;
  color: #194872;
  font-weight: 700;
  text-transform: uppercase;
}
```

```

#ulogovan {
    margin: 0!important;
    color: #c4ecff;
}

.ulogovanResponsive {
    display: none;
}

.izaberitePredmet {
    min-height: 12vh;
    padding-top: 5%;
}

.borderBottom {
    width: 90%;
    border-bottom: 1px solid #dce2e6;
    margin-bottom: 10px;
}

```

Responsive.css

```

@media screen and (max-width:1024px){
    #dodajSmer, #dodajPredmet{
        width: 30%;
    }

    #slikaPrijavljen {
        display: none;
    }
}

@media screen and (max-width:992px){
    .imePrezime {
        font-size: 3vw;
        margin-bottom: 30px;
    }

    .imePrezime br {
        display: none;
    }

    #redProfil .col-lg-3 {
        border-right: none;
    }

    #redProfil .col-lg-8 {
        margin-top: 0!important;
        border-top: 1px solid #dce2e6;
    }

    .naslovProfil {
        font-size: 1.8vw;
    }

    .odeljenjePocetna {
        width: 31%;
        margin-right: 2%;
    }
}

```

```
.kolonaCasovi {
    width: 48%;
    margin-right: 2%;
}

.ulozovanResponsive {
    display: block;
}

#divPrijavljen {
    display: none !important;
}

.levo {
    display: flex;
    justify-content: flex-start;
    margin-bottom: 20px;
}

.slikaIme {
    width: 50%;
    display: flex;
    justify-content: flex-start;
}

.imeSmer {
    width: 60%;
    padding-top: 10%;
    margin-left: 5%;
    text-align: left;
}

.ime {
    font-size: 2vw;
}

.dugmici div, .dugmici a {
    width: 50% !important;
}

.polugodista {
    flex-direction: column;
}

.divKontakt {
    width: 100% !important;
    margin-top: 30%;
    margin-left: 35%;
}

.btnDodajKontakt {
    width: 100% !important;
}

.vroditelj {
    width: 40% !important;
}

.vucenik {
    margin-top: 10% !important;
}

.informacije {
```

```

        flex-direction: column;
        align-items: center;
    }

    .informacije .col-lg-6 {
        width: 90%!important;
        margin-top: 20px;
    }
}

@media screen and (max-width:768px){
    #futer .container .row .col-md-4, #futer .container .row .col-md-8 {
        display: flex;
        justify-content: center;
    }

    #ikonice {
        padding: 0!important;
    }

    .odeljenjePocetna {
        width: 48%;
        margin-right: 2%;
    }

    #odeljenjeGore .col-md-6{
        display: flex;
        justify-content: center!important;
    }

    #odeljenjeGore .odeljenjeResponsive{
        display: flex;
        flex-direction: column;
        align-items: center!important;
    }

    .ucenikIme a {
        font-size: 2vw;
    }

    .smerOdeljenje {
        font-size: 1.8vw;
    }

    .datumCas {
        padding: 2% 0%;
    }

    .dan {
        font-size: 2.5vw;
    }

    .d {
        font-size: 2vw;
    }

    .nastavnaJedinica {
        font-size: 4vw;
    }

    #navbar {
        display: flex;
        flex-direction: column;
    }
}

```

```
.ucenikOdeljenje {
  width: 23%;
  margin-right: 2%;
}

}

@media screen and (max-width:480px){
  #smer {
    width: 50vw;
  }

  #dodajSmer, #dodajPredmet{
    width: 50%;
  }

  .imePrezime {
    font-size: 8vw;
  }

  .naslovProfil {
    font-size: 4.5vw;
  }

  .containerMarginal {
    margin-top: 50px!important;
  }

  #futer {
    margin-top: 0!important;
  }

  #containerPocetna .row {
    justify-content: center!important;
  }

  .odeljenjePocetna {
    width: 80%;
    padding-left: 2%;
    padding-bottom: 2%;
  }

  .kolonaCasovi {
    width: 80%;
    margin-right: 2%;
  }

  .casovi {
    justify-content: center!important;
  }

  .datumCas {
    margin-left: 0.5%;
  }

  .odeljenjePocetna:last-child {
    margin-bottom: 20px;
  }

  .containerMargina2 {
    margin-top: 80px!important;
  }
}
```

```
.dan {
    font-size: 4vw;
}

.d {
    font-size: 3.2vw;
}

.nastavnaJedinica {
    font-size: 5.3vw;
}

#casDatum {
    width: 65%!important;
}

.ucenikOdeljenje {
    width: 48%;
    margin-right: 2%;
}

.ucenikIme a {
    font-size: 3.8vw;
}

.smerOdeljenje {
    font-size: 3vw;
}

.djaci {
    padding-left: 2%;
}

.levo {
    flex-direction: column;
    align-items: center;
}

.slikaIme {
    width: 100%;
    display: flex;
    justify-content: flex-start;
}

.imeSmer {
    text-align: left;
}

.ime {
    font-size: 5vw;
}

.divVladanje, .divTekstVladanje, #prikaziOcenu, #prikaziZakljucnu,
#prikaziZahtev {
    width: 75%!important;
}

.divKontakt {
    margin-top: 20%;
    margin-left: 0%;
}

.prosek, #zakljucneUcenik, #zakljucneRoditelj {
    flex-direction: column;
```

```
        margin-top: 0!important;
    }

    .prosekPrvo, .prosekDrugo, #prosekPrvoR, #prosekDrugoR,
    .zakljucnaPolugodiste {
        width: 90%;
        margin-bottom: 10px;
    }

    .naslovPolugodiste {
        width: 95% !important;
    }

    .datumNeregulisani {
        width: 75% !important;
        margin-left: 2%;
    }

    .casNeregulisani {
        width: 20% !important;
    }

    .divTekstVladanje {
        margin-top: 10px!important;
    }
}
```


5. Baza

tblkorisnik_predmet_cas id_cas int(10) ime varchar(200) datum date po_reda int(10) napomena varchar(500) id_korisnik int(10) id_predmet int(10) id_odejenje int(10)	tblkorisnik_predmet_korisnik_predmet id_korisnik_predmet int(10) id_korisnik int(10) id_predmet int(10)	tblkorisnik_predmet_smer_predmet id_smer_predmet int(10) id_smer int(10) id_predmet int(10)	tblkorisnik_predmet_ocena id_ocena int(10) godina int(10) datum date id_ucenik int(10) id_predmet int(10) id_polugodiste int(10) id_vrsta_ocene int(10)	tblkorisnik_predmet_ucenik id_ucenik int(10) ime varchar(100) prezime varchar(100) sila varchar(500) vladaje int(10) prez varchar(10) adresa varchar(150) mesto_rodenja varchar(500) datum_rodenja date id_odejenje int(10)
tblkorisnik_predmet_proseki id_proseki int(10) id_ucenik int(10) proseki double(10,2) polugodiste int(10)	tblkorisnik_predmet_odosudni id_odosudni int(10) id_ucenik int(10) id_cas int(10) opisodane tinyint(1)	tblkorisnik_predmet_predmet id_predmet int(10) naziv varchar(100)	tblkorisnik_predmet_zahtev id_zahtev int(10) polgodiste_ocena int(10) nova_ocena int(10) datum_ocene date id_ucenik int(10) id_korisnik int(10) id_vrsta_ocene int(10) odosudeno tinyint(1)	tblkorisnik_predmet_korisnik id_korisnik int(10) ime varchar(50) prezime varchar(50) kor_ime varchar(50) sila varchar(50) email varchar(100) slike varchar(500) lozinka varchar(100) id_kloga int(10)
tblkorisnik_predmet_roditelj_ucenik id_roditelj_ucenik int(10) id_korisnik int(10) id_ucenik int(10)	tblkorisnik_predmet_meci id_meci int(10) naziv varchar(50) tula varchar(50)	tblkorisnik_predmet_polugodiste id_polugodiste int(10) polugodiste int(10) datum_od date datum_do date	tblkorisnik_predmet_sloga id_sloga int(10) naziv varchar(50)	tblkorisnik_predmet_odejenje id_odejenje int(10) broj_odejenja int(10) godina varchar(10) id_korisnik int(10) id_smer int(10)
	tblkorisnik_predmet_smer id_smer int(10) naziv varchar(500)	tblkorisnik_predmet_vrsta_ocena id_vrsta_ocene int(10) vrsta varchar(150)		

6. Zaključak

Korišćenjem web aplikacije „Elektronski dnevnik“ u školi se u potpunosti izbacuju iz upotrebe dnevnicu rada u fizičkom obliku. Omogućeni su svi unosi koji su ranije rađeni ručno i samim tim omogućena je velika ušteda vremena za sve korisnike ove aplikacije (nastavnici, direktor, razredne starešine). Takođe, veoma je bitno napomenuti i mogućnost uvida ocena i izostanaka od strane roditelja koji se na sistem prijavljuju pomoću korisničkog imena i lozinke koje dodeljuje razredni starešina ili administrator. Roditelj može videti samo ocene za svoje dete, čime je informisanje roditelja olakšano, jer je u svakom trenutku moguće pristupiti podacima bez dolaska u školu.

Administrator je unosio korisnike (nastavnike, razredne starešine i roditelje), generisao njihova korisnička imena i inicijalne šifre, uneo smerove i predmete. Takođe je unosio i učenike, dodelio ih odeljenjima i dodelio im predmete. Po potrebi je vršio izmene.

Direktor je imao pregled svih učenika, predmeta, odeljenja, nastavnika i odobravao ili odbijao zahteve za promenu pogrešno unete ocene.

Nastavnik je unosio ocene, napomene, zaključivao ocene, upisivao časove i odsutne i po potrebi slao zahtev za promenu pogrešno unete ocene.

Odeljenski starešina pored svega navedenog što može da radi nastavnik takođe je vršio administraciju svog odeljenja. Unosio učenike, dodelio im predmete, regulisao izostanke, imao pregled svih ocena i ocenjivao vladanje. Dodavao im je kontakte (roditelje) i kreirao njihove naloge. Roditelj pristupa svom nalogu i ima uvid u ocene i izostanke svog deteta.

7. Literatura

[1] <https://laravel.com/>

[2] <https://getbootstrap.com/>

[3] <https://www.chartjs.org/docs/latest/>

[4] Mastering Laravel - Christofer John Pecoraro Laravel: Up & Running – Matt Stauffer