



Srednja tehniška in poklicna šola Trbovlje

Šuštarjeva kolonija 7a, 1420 Trbovlje

4. izpitni predmet poklicne mature

Poročilo o razvoju izdelka

Vrsta in ime izobraževalnega programa

Srednje strokovno izobraževanje – tehnik računalništva

Naslov naloge:

Aplikacija za vodenje prijav na dejavnosti v šoli

Mentor: Žiga Podplatnik, prof.

Kandidata: Nik Jukič

Gašper Podbregar

Šolsko leto: 2020/21

IZJAVA O AVTORSTVU

zaključne naloge

Podpisana Gašper Podbregar in Nik Jukič,
rojena 27. 3. 2002 in 16. 1. 2002,

sva avtorja zaključne naloge z naslovom
»Aplikacija za vodenje prijav na dejavnosti v šoli«.

S svojim podpisom zagotavljam, da:

- sva zaključno nalogo izdelal v paru pod mentorstvom prof. Žige Podplatnika in je najino avtorsko delo;
- sva vso uporabljeno literaturo in vsa gradiva, ki sva jih uporabila pri najinem delu, navajala v delu in na koncu zaključnega dela tudi ustrezno navedla;
- je elektronska oblika zaključnega dela, naslov in vsebina identična tiskani obliki zaključne naloge;
- sva končno različico delujočega izdelka v celoti in trajno shranila na medij za zapis/shranjevanje podatkov in ga oddala skupaj s pisnim delom zaključne naloge;
- se strinjava, da se najin izdelek hrani na šoli in se uporablja v učne oz. demonstracijske namene.

V Trbovljah, dne 21. 4. 2021

Podpis avtorja

Zahvala

Zahvaljujeva se najinemu mentorju, profesorju Žigi Podplatniku, za pomoč in podporo pri izdelavi zaključne naloge. Pri vsaki napaki nama je bil pripravljen pomagati, naju spodbujal k rednemu delu ter vodil do končnega rezultata.

Kazalo vsebine

Vsebina

Povzetek	9
1 Uvod	10
1.1 Cilj naloge	10
2 Opis poteka dela	11
2.1 Utemeljitev cilja ter načrt	11
2.2 Izdelava osnovnih temeljev	11
2.3 Izdelava same aplikacije in grafične podobe	11
3 Uporabljena orodja	12
3.1 Ogrodje CodeIgniter	12
3.2 PHP	12
3.3 HTML	13
3.4 CSS	13
3.5 JavaScript	14
3.6 MySQL in phpMyAdmin	14
4 Zgradba	15
4.1 Sestava	15
4.2 Podatkovna baza	16
4.3 Prijava	17
4.3.1 Prijava v aplikacijo	17
4.3.2 Pridobivanje podatkov za dijaka	18
4.3.3 Dodajanje dijakov	19
4.3.4 Spremeni geslo	20
4.4 Dejavnost	21
4.4.1 Prikaz dejavnosti	21
4.4.2 Prijava za dejavnost	22
4.5 Datoteke	24
4.5.1 Izvoz v PDF in CSV	24
5 Aplikacija	25
5.1 Prijava	25
5.2 Dijak	25
5.2.1 Profil dijaka	26

5.2.2	Aktualne dejavnosti dijaka	26
5.2.3	Zaključene dejavnosti dijaka	27
5.2.4	Prijava na dejavnosti dijaka.....	27
5.3	Profesor	28
5.3.1	Profil profesorja.....	28
5.3.2	Aktualne dejavnosti profesorja.....	29
5.3.2.1	Potrdi prijave	29
5.3.2.2	Izbriši prijavo	30
5.3.2.3	Prisotnost	30
5.3.2.4	Udeleženci	31
5.3.2.5	Prijavi dijake.....	31
5.3.3	Razred profesorja	32
5.3.4	Ustvari dejavnost profesor.....	34
5.3.5	Zaključene dejavnosti profesor.....	35
5.4	Administrator.....	35
5.4.1	Profil Administratorja.....	36
5.4.2	Dodajanje dijakov administrator	36
5.4.3	Dodajanje profesorjev administrator	37
5.4.4	Urejanje uporabnikov administrator	37
5.4.5	Izvažanje in uvažanje uporabnikov	38
6	Pogledi.....	39
6.1	Knjižnice	40
7	Izgled	41
7.1	Logotip	41
7.2	Barvna shema	41
7.3	Prvotni izgled	42
7.4	Končni izgled	42
7.5	Pisava.....	43
8	Razprava in zaključek.....	45
9	Spletni viri	46

Kazalo slik

Slika 1: Logotip ogrodja CodeIgniter.....	12
Slika 2: Logotip programskega jezika php.....	12
Slika 3: Logotip označevalnega jezika HTML 5.....	13
Slika 4: Logotip oblikovalnega jezika CSS.....	13
Slika 5: Logotip skriptnega jezika JavaScript	14
Slika 6: Logotip sistema MySQL	14
Slika 7: Logotip orodja phpMyAdmin	14
Slika 8: Logotip spletnega strežnika APACHE.....	14
Slika 9: Sestava aplikacije	15
Slika 10: Slika začetne sheme	16
Slika 11: Slika končne sheme.....	16
Slika 12: Koda izvedena ob prijavi	17
Slika 13: Pridobivanje podatkov dijaka.....	18
Slika 14: Pridobivanje podatkov dijaka.....	18
Slika 15: Dodajanje dijakov	19
Slika 16: Dodajanje dijakov	19
Slika 17: Koda za spremenitev gesla.....	20
Slika 18: Vse funkcije v Prijava	20
Slika 19: Koda za prikaz dejavnosti	21
Slika 20: Poizvedba za dejavnosti	21
Slika 21: Avtomatska prijava	22
Slika 22: Avtomatska prijava	23
Slika 23: Prošnja.....	23
Slika 24: Ustvaritev relacije za prisotnost	23
Slika 25: Koda za generiranje CSV datoteke	24
Slika 26: Koda za generiranje PDF datoteke.....	24
Slika 27: Prijava v aplikaciji.....	25
Slika 28: Domača stran dijak.....	25
Slika 29: Profil dijaka.....	26
Slika 30: Aktualne dejavnosti dijak.....	26
Slika 31: Zaključene dejavnosti dijak.....	27
Slika 32: Prijava na dejavnosti dijak	27
Slika 33: Domača stran profesor	28
Slika 34: Profil profesorja	28
Slika 35: Aktualne dejavnosti profesor	29
Slika 36: Potrdi prijavo profesor	29
Slika 37: Izbriši prijavo profesor.....	30
Slika 38: Urejanje prisotnosti profesor.....	30
Slika 39: Udeleženci dejavnosti profesor	31
Slika 40: Prijavi dijake	31
Slika 41: Razred profesorja	32
Slika 42: Pretekle dejavnosti profesor	32
Slika 43: Izvoz prisotnosti profesor.....	33
Slika 44: Nova dejavnost profesor	34

Slika 45: Zaključene dejavnosti profesor	35
Slika 46: Domača stran profesor	35
Slika 47: Profil administratorja	36
Slika 48: Obrazec dodajanja dijakov	36
Slika 49: Obrazec za dodajanje profesorjev	37
Slika 50: Urejanje uporabnikov administrator	37
Slika 51: Izvoz in uvoz administrator.....	38
Slika 52: Vsi pogledi	39
Slika 53: Zunanji slogi pisav	40
Slika 54: Knjižnice 1	40
Slika 55: Knjižnice 2	40
Slika 56: Izvoz pdf	40
Slika 57: Logotip	41
Slika 58: Barvna shema.....	41
Slika 59: Prvoten izgled	42
Slika 60: Domača stran dijaka.....	42
Slika 61: Pregled računa dijak.....	43
Slika 62: Pisava Poiret One	43
Slika 63: Pisava Varela Round	43
Slika 64: Pisava Raleway	44

Povzetek

Namen naloge je izdelati aplikacijo, s pomočjo katere se dijaki lahko prijavijo na posamezne dejavnosti, ki se odvijajo v šoli. Aplikacija bo pomagala pri organiziranju samih dejavnosti in beleženju prisotnosti. V aplikaciji so trije tipi uporabnikov, in sicer dijak, profesor in administrator.

Dijaki se lahko prijavljajo na dejavnosti in jih spremljajo. Profesorji ustvarjajo dejavnosti, jih nadzirajo ter pomagajo dijakom pri prijavljanju. Administrator lahko poleg vseh naštetih funkcij upravlja tudi z uporabniškimi računi.

1 Uvod

Navodila

Kandidata načrtujeta in razvijeta spletno aplikacijo, ki bo udeležencem izobraževanja v šoli omogočala prijave na dejavnosti v šoli (športni dnevi, projektni dnevi ...). Aplikacija bo imela pogled administratorja, učitelja in dijaka. Dijak se bo lahko prijavil v aplikacijo in izbral med dejavnostmi na šoli. Ko bo enkrat izbiro potrdil, spremembe ne bodo več mogoče (razen v primeru administratorja). V primeru, da bo dijakova prijava sprejeta ali zavrnjena, bo dobil potrdilo oziroma obvestilo. Učitelj bo imel možnost dodajanja dijakov na svojo dejavnost. Prav tako bo potrjeval prijave dijakov na dejavnosti.

Uporabnik se bo lahko prijavil v aplikacijo in glede na vlogo imel različen dostop in pravice. Dijak se bo lahko prijavljal na dejavnosti, kar bo moral profesor potrditi. Profesor bo lahko ustvarjal dejavnosti in jih dijakom dodeljeval. Profesorji bodo tudi beležili prisotnost dijakov na dejavnostih. Vsi sezname prijavljenih se bodo lahko izvažali kot PDF ali Excel datoteke. Administrator bo imel vse možne funkcionalnosti. Dodatno bo lahko dodajal druge uporabnike ter dijake samodejno razvrščal po dejavnostih. Izvozil bo lahko sezname prijavljenih v obliki PDF in Excel. Pri posamezni dejavnosti bo lahko za vsak dan dejavnosti zabeležil prisotnost prijavljenih dijakov. Administrator bo prav tako lahko razvrščal uporabnike na dejavnosti in uporabljal orodje samorazvrščanja (možnost, da avtomatsko dodeli dijake, ki niso prijavljeni nikamor). Učitelji, ki bodo označeni kot razredniki, bodo lahko izvozili seznam v obliki PDF in Excel dokumenta (v seznamu bo prikazano, kam je prijavljen posamezni dijak v razredu – po izvedeni dejavnosti bo dodatno lahko pregledal prisotnost). Kandidata izdelata E-R diagram. Kandidata prav tako oblikujeta dokumentacijo in kritično presodita opravljeno delo.

1.1 Cilj naloge

Cilje je ustvariti aplikacijo, ki bo olajšala delo pri organiziranju ter sami izvedbi različnih dejavnosti, ki potekajo v šolah. Omogočala bi prijavo 3 različnih uporabnikov. Dijak, profesor in administrator. Dijak ima možnost poslati prošnjo za prijavo na dejavnost in si ogledati dejavnosti, na katere je prijavljen oziroma so potekle. Profesor ima možnost upravljanja z dejavnostmi, dodajanja oziroma odstranjevanja prijav, izvažanja ter pregledovanja svojega razreda v primeru da je razrednik. Administrator ima poleg vseh naštetih funkcij še dodajanje, urejanje, in uvažanje uporabnikov.

2 Opis poteka dela

2.1 Utemeljitev cilja ter načrt

Najprej sva si zadala cilj, ki ga želiva uresničiti, in nato razmišljala, kako ga bova uresničila. Začela sva z grobo skico spletne strani, dokumentacijo vseh funkcionalnosti ter nadaljevala z grafično podobo in samim programom v ozadju.

2.2 Izdelava osnovnih temeljev

Naslednji korak je bila izdelava relacijskega modela in utemeljitev pravic različnih uporabnikov. Relacijski model se je skozi izdelavo velikokrat spremenil zaradi boljših rešitev, ki sva jih odkrila ob nastalih problemih.

2.3 Izdelava same aplikacije in grafične podobe

Po prvi konzultaciji z mentorjem sva začela z izdelavo same aplikacije. Med tem procesom se je relacijski model velikokrat spremenil zaradi novih potreb oziroma odvečnih delov modela.

3 Uporabljen orodja

3.1 Ogrodje CodeIgniter

Je ogrodje, na katerem temelji sama aplikacija. Uporablja jezik PHP in temelji na tehnologiji »model-view-controller«.

CodeIgniter je eno izmed najbolj popularnih spletnih ogrodjih za dinamične spletne strani. Temelji na programskem jeziku PHP in je odprtokoden. V primerjavi z drugimi ogrodji, ki temeljijo na jeziku PHP, je CodeIgniter najhitrejši in ker je odprtokoden, ni težav z odpravljanjem napak. Izvorna koda je objavljena na spletni strani GitHub.



Slika 1: Logotip ogrodja CodeIgniter

3.2 PHP

PHP je skriptni jezik ustvarjen za razvijanje spletnih aplikacij. Običajno je uporabljen za prikaz spletnih strani, teče pa na strežniku.



Slika 2: Logotip programskega jezika php

3.3 HTML

HTML je standardiziran označevalni jezik, uporabljen za prikazovanje željene vsebine z uporabo spletnih brskalnikov. Po navadi se z njim uporablja tudi tehnologija CSS ter skriptni jezik kot JavaScript.



Slika 3: Logotip označevalnega jezika HTML 5

3.4 CSS

CSS je oblikovalni jezik uporabljen za opisovanje elementov uporabljenih v dokumentu HTML. Z njim opišemo obliko, barvo, slog ter pozicijo elementov.



Slika 4: Logotip oblikovalnega jezika CSS

3.5 JavaScript

JavaScript je skriptni programski jezik, ki sestavlja kar 97 % vseh spletnih strani. Uporablja se ga za prikazovanje in obnašanje interaktivnih spletnih strani.



Slika 5: Logotip skriptnega jezika JavaScript

3.6 MySQL in phpMyAdmin

MySQL je odprtokodni sistem za upravljanje z relacijskimi podatkovnimi bazami. Temelji na poizvedbenem jeziku SQL. PhpMyAdmin je odprtokodni administracijski sistem za lažje upravljanje s sistemom MySQL. Zaradi lažje uporabe sva za upravljanje s podatkovno bazo uporabljala phpMyAdmin. Celotna aplikacija deluje s pomočjo strežniške tehnologije APACHE.



Slika 6: Logotip sistema MySQL



Slika 7: Logotip orodja phpMyAdmin



Slika 8: Logotip spletnega strežnika APACHE

4 Zgradba

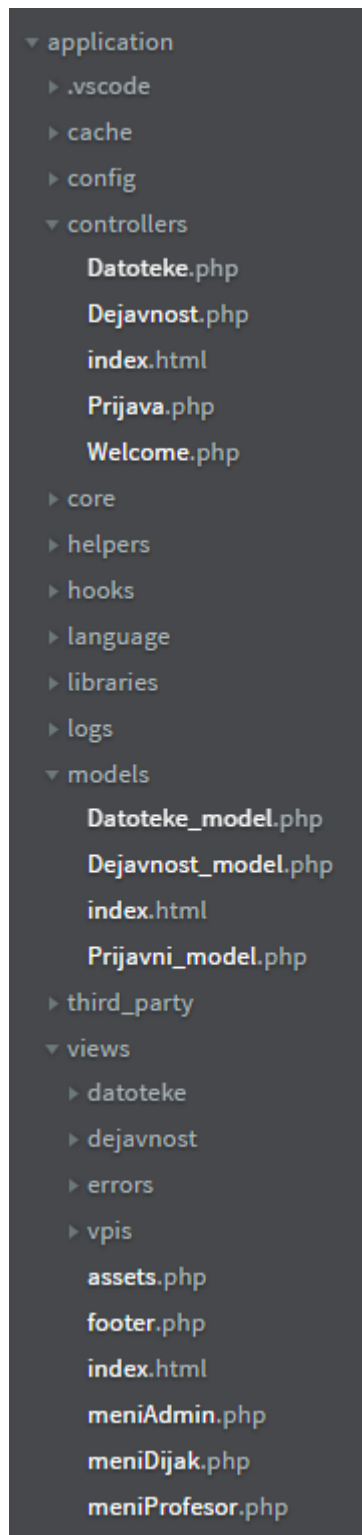
4.1 Sestava

Aplikacije vsebuje MVC tehnologijo, kar pomeni, da je sestavljena iz treh ključnih delov. Prvi del so Controllers oziroma koda, iz katere kličemo Views (pogled) ter Models (koda, kjer pridobimo podatke iz podatkovne baze).

Controllers ter Models sta zaradi različnih funkcij sestavljena iz več datotek.

Views so datoteke, ki vsebujejo izgled same strani. Razdeljene so na 3 menije, za vsak tip uporabnika, ter nogo (footer) strani. Vsebujejo tudi datoteko »assets«, kjer so vsi potrebni klici za zunanje knjižnice.

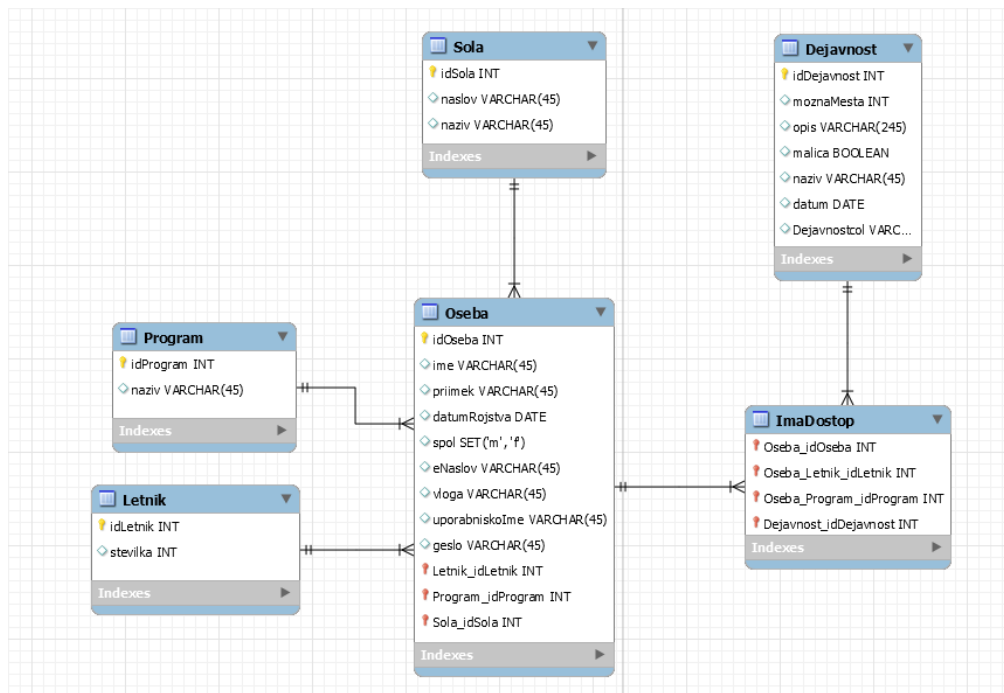
Datoteke, dejavnost in vpis so različni pogledi, glede na katero podstran želimo dostopati.



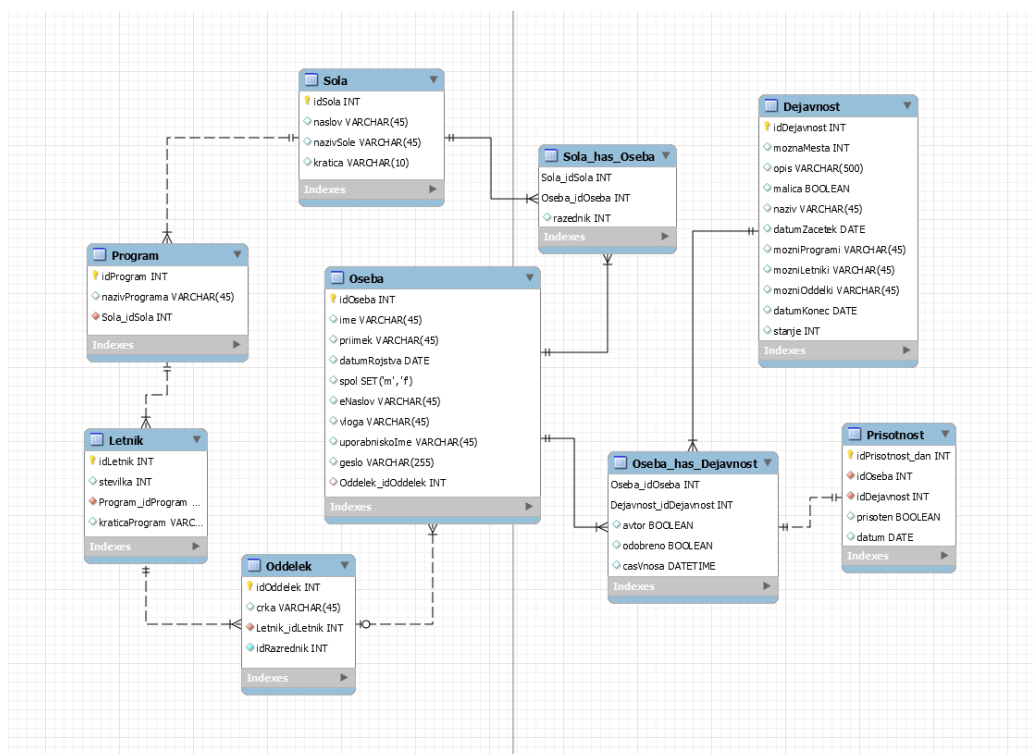
Slika 9: Sestava aplikacije

4.2 Podatkovna baza

Schema podatkovne baze se je skozi izdelavo zelo spremenila zaradi potrebe po dodatnih funkcijah. Med izdelavo aplikacije je shema postala veliko bolj zapletena in nastalo je kar nekaj novih relacij.



Slika 10: Slika začetne sheme



Slika 11: Slika končne sheme

4.3 Prijava

4.3.1 Prijava v aplikacijo

Pri prijavi se izvede spodnja koda. Koda pregleda, ali so podatki pravilni ter nastavi začasne podatke uporabnika.

```
public function prijava_submit(){

    $config_pravila = array(
        array(
            "field">"txt_email",
            "label">"E-poštni naslov",
            "rules">"required"
        ),
        array(
            "field">"txt_geslo",
            "label">"Geslo",
            "rules">"required"
        )
    );

    $this->form_validation->set_rules($config_pravila);

    if($this->form_validation->run()==FALSE){

        $this->login();
    }
    else{

        $podatki = $this->input->post();
        $podatki_array = array(
            "email">$podatki["txt_email"],
            "geslo">hash('sha256', $podatki["txt_geslo"])
        );

        $rezultat = $this->prijavni_model->preveriUporabnika($podatki_array);

        if ($rezultat!=null){

            $this->session->set_userdata("ime", $rezultat["ime"]);
            $this->session->set_userdata("idOseba", $rezultat["idOseba"]);
            $this->session->set_userdata("priimek", $rezultat["priimek"]);
            $this->session->set_userdata("datumRojstva", $rezultat["datumRojstva"]);
            $this->session->set_userdata("apol", $rezultat["apol"]);
            $this->session->set_userdata("eNaslov", $rezultat["eNaslov"]);
            $this->session->set_userdata("vloga", $rezultat["vloga"]);
            $this->session->set_userdata("oddelekID", $rezultat["Oddelek_idOddelek"]);
            $this->session->set_userdata("letnikID", $rezultat["Letnik_idLetnik"]);
            $this->session->set_userdata("programID", $rezultat["Program_idProgram"]);
            $this->session->set_userdata("solaID", $rezultat["Sola_idSola"]);
            $this->session->set_userdata("nazivSole", $rezultat["nazivSole"]);
            $this->session->set_userdata("nazivPrograma", $rezultat["nazivPrograma"]);
            $this->session->set_userdata("stevilka", $rezultat["stevilka"]);
            $this->session->set_userdata("crka", $rezultat["crka"]);
```

Slika 12: Koda izvedena ob prijavi

4.3.2 Pridobivanje podatkov za dijaka

Spodnja koda pridobi potrebne podatke za dijaka za pravilno prikazovanje in delovanje strani.

```
public function fetch_programe(){
    if($this->session->userdata("vloga") != "admin" and $this->session->userdata("vloga") != "profesor"){
        redirect("");
    }
    if($this->input->post('sola_id')){
        echo $this->prijavni_model->fetch_programe($this->input->post("sola_id"));
    }
}

public function fetch_letnike(){
    if($this->session->userdata("vloga") != "admin" and $this->session->userdata("vloga") != "profesor"){
        redirect("");
    }
    if($this->input->post('program_id')){
        echo $this->prijavni_model->fetch_letnike($this->input->post("program_id"));
    }
}

public function fetch_oddelke(){
    if($this->session->userdata("vloga") != "admin" and $this->session->userdata("vloga") != "profesor"){
        redirect("");
    }
    if($this->input->post('letnik_id')){
        echo $this->prijavni_model->fetch_oddelke($this->input->post("letnik_id"));
    }
}
```

Slika 13: Pridobivanje podatkov dijaka

```
public function fetch_programe($idSola){
    $this->db->select("nazivPrograma, idProgram");
    $this->db->from("program");
    $this->db->where("Sola_idSola", $idSola);
    $query = $this->db->get();
    $output = '<option value="">Izberite program</option>';
    foreach($query->result() as $row){
        $output .= '<option value="' . $row->idProgram . '">' . $row->nazivPrograma . '</option>';
    }
    return $output;
}

public function fetch_letnike($idProgram){
    $this->db->select("stevilka, idLetnik");
    $this->db->from("letnik");
    $this->db->where("Program_idProgram", $idProgram);
    $query = $this->db->get();
    $output = '<option value="">Izberite letnik</option>';
    foreach($query->result() as $row){
        $output .= '<option value="' . $row->idLetnik . '">' . $row->stevilka . "</option>";
    }
    return $output;
}

public function fetch_oddelke($idLetnik){
    $this->db->select("crka, idOddelek");
    $this->db->from("oddelek");
    $this->db->where("Letnik_idLetnik", $idLetnik);
    $query = $this->db->get();
    $output = '<option value="">Izberite oddelek</option>';
    foreach($query->result() as $row){
        $output .= '<option value="' . $row->idOddelek . '">' . $row->crka . '</option>';
    }
    return $output;
}
```

Slika 14: Pridobivanje podatkov dijaka

4.3.3 Dodajanje dijakov

Spodaj je prikazana koda dodajanja dijakov. Malo drugačno različico te funkcije ima tudi profesor. Pripravi se seznam seznamov, kjer določimo pravila za vnos podatkov. Nato se vneseni podatki shranijo v ta seznam in se pregledajo. Če so podatki ustrezni, se pokliče funkcija v Models, kjer jih vstavi v podatkovno bazo. V nasprotnem primeru nam te podatke zavrne.

```
public function dodajanje_submit_dijak(){
    if($this->session->userdata("vloga") != "admin"){
        redirect("");
    }

    $config_pravila = array(
        array(
            "field"=>"txt_ime",
            "label"=>"Ime",
            "rules"=>"required"
        ),
        array(
            "field"=>"txt_priimek",
            "label"=>"Priimek",
            "rules"=>"required"
        ),
        array(
            "field"=>"txt_geslo",
            "label"=>"Geslo",
            "rules"=>"required|min_length[5]"
        ),
        array(
            "field"=>"txt_sola",
            "label"=>"Sola",
            "rules"=>"required"
        ),
        array(
            "field"=>"txt_program",
            "label"=>"Program",
            "rules"=>"required"
        ),
        array(
            "field"=>"txt_letnik",
            "label"=>"Letnik",
            "rules"=>"required"
        ),
        array(
            "field"=>"txt_oddelek",
            "label"=>"Oddelek",
            "rules"=>"required"
        ),
        array(
            "field"=>"txt_datum",
            "label"=>"Datum rojstva",
            "rules"=>"required"
        )
    );
    $this->form_validation->set_rules($config_pravila);
    if($this->form_validation->run()==FALSE){
        $this->dodajanjeUporabnikov_dijak();
    }
    else{
        $podatki = $this->input->post();
        $podatki_array = array(
            "ime"=>$podatki["txt_ime"],
            "priimek"=>$podatki["txt_priimek"],
            "datumRojstva"=>$podatki["txt_datum"],
            "spol"=>$podatki["txt_spol"],
            "eNaslov"=>"",
            "vloga"=>"dijak",
            "geslo"=>hash('sha256', $podatki["txt_geslo"]),
            "Oddelek_idOddelek"=>$podatki["txt_oddelek"]
        );
        $izid = $this->prijavni_model->pridobiKratice($podatki["txt_sola"]);
        $prava = $izid[0]["kratica"];
        $podatki_array["eNaslov"] = $this->prijavni_model->sestaviEmail($podatki_array, $prava);
        if($this->prijavni_model->vstavljanjeUporabnika($podatki_array)){
            $IDosebe = $this->prijavni_model->iskanjeID0sebe();

            $podatki_array2 = array(
                "Sola_idSola" => $podatki["txt_sola"],
                "Oseba_idOseba" => $IDosebe["idOseba"]
            );
            if($this->prijavni_model->vstavljanjeSolaHasOseba($podatki_array2)){
                $this->session->set_flashdata("succes", "Dijak je bil uspešno dodan!");
                redirect("prijava/dodajanje-dijak");
            }
            else{
                $this->session->set_flashdata("error", "Dijaka ni bilo mogoče kreirati!");
                redirect("prijava/dodajanje-dijak");
            }
        }
        else{
            $this->session->set_flashdata("error", "Dijaka ni bilo mogoče kreirati!");
            redirect("prijava/dodajanje-dijak");
        }
    }
}
```

Slika 15: Dodajanje dijakov

Slika 16: Dodajanje dijakov

4.3.4 Spremeni geslo

Spodaj je koda za spremembo gesla. Kot pri dodajanju se podatki shranijo v seznam in se preverja korektnost vnosa.

```
public function spremeniGeslo(){
    if($this->session->userdata("vloga") != "admin"){
        redirect("");
    }
    else{
        $config_pravila = array(
            array(
                "field"=>"txt_geslo",
                "label"=>"Geslo",
                "rules"=>"required|min_length[5]"
            )
        );

        $this->form_validation->set_rules($config_pravila);

        if($this->form_validation->run()==FALSE){

            $this->session->set_flashdata("error", "Novo mora biti dolgo vsaj 5 znakov!");
            redirect("prijava/urejanjeUporabnikov");
        }
        else{
            $novoGeslo = hash('sha256', $this->input->post("txt_geslo"));
            $id = $this->input->post("gumb");
            $novoGeslo = hash('sha256', $this->input->post("txt_geslo"));
            $this->db->set("geslo", $novoGeslo);
            $this->db->where('id0seba', $id);
            $this->db->update('0seba');
            $this->session->set_flashdata("succes", "Geslo je bilo spremenjeno");
            redirect("prijava/urejanjeUporabnikov");
        }
    }
}
```

Slika 17: Koda za spremenitev gesla

Spodaj so vse funkcije napisane v datoteki Prijava.

```
class Prijava extends CI_Controller{

    public function __construct() { ... }

    public function login(){ ... }

    public function logout(){ ... }

    public function fetch_programe(){ ... }

    public function fetch_letnike(){ ... }

    public function fetch_oddcelke(){ ... }

    public function prijava_submit(){ ... }

    public function dodajanjeUporabnikov_dijak(){ ... }

    public function dodajanje_submit_dijak(){ ... }

    public function dodajanjeUporabnikov_profesor(){ ... }

    public function dodajanje_submit_profesor(){ ... }

    public function urejanjeUporabnikov(){ ... }

    public function iskanjeUporabnikov(){ ... }

    public function spremeniGeslo(){ ... }

    public function profil(){ ... }

    public function spremeniGesloUporabnik(){ ... }

    public function uporabniki(){ ... }
```

Slika 18: Vse funkcije v Prijavi

4.4 Dejavnost

4.4.1 Prikaz dejavnosti

Spodaj prikazana koda se kliče ob zahtevi za stran, ki prikazuje dejavnosti. Podobna koda se izvede tudi pri drugih uporabnikih, vendar z malo drugačnimi atributi. Najprej se pokliče model za iskanje dejavnosti po podatkovni bazi, potem pa se, zaradi povezav v podatkovni bazi, pri klicu modela rezultat razbije na format uporabljen v dejanskem pogledu.

```
public function mojeDejavnosti(){
    if($this->session->userdata("vloga") != "profesor"){
        redirect("");
    }
    else{
        $rezultat["izbire"] = $this->dejavnost_model->mojeDejavnostiProfesor($this->session->userdata("idOseba"));
        $st=0;
        if($rezultat["izbire"] != NULL){
            foreach($rezultat["izbire"] as $opcija){
                $rezultat["izbire"][$st]["mozneSole"] = explode(";", $opcija["mozneSole"]);
                $rezultat["izbire"][$st]["mozniProgrami"] = explode(";", $opcija["mozniProgrami"]);
                $rezultat["izbire"][$st]["mozniLetniki"] = explode(";", $opcija["mozniLetniki"]);
                $rezultat["izbire"][$st]["mozniOddelki"] = explode(";", $opcija["mozniOddelki"]);

                $key = array_search("", $rezultat["izbire"][$st]["mozneSole"]);
                unset($rezultat["izbire"][$st]["mozneSole"][$key]);

                $key = array_search("", $rezultat["izbire"][$st]["mozniProgrami"]);
                unset($rezultat["izbire"][$st]["mozniProgrami"][$key]);

                $key = array_search("", $rezultat["izbire"][$st]["mozniLetniki"]);
                unset($rezultat["izbire"][$st]["mozniLetniki"][$key]);

                $key = array_search("", $rezultat["izbire"][$st]["mozniOddelki"]);
                unset($rezultat["izbire"][$st]["mozniOddelki"][$key]);
                $st++;
            }
        }
        if($rezultat["izbire"] != NULL){
            for($x=min(array_keys($rezultat["izbire"])); $x<max(array_keys($rezultat["izbire"]))+1; $x++){
                $y=0;
                foreach($rezultat["izbire"][$x]["mozneSole"] as $nekaj){
                    foreach($rezultat["izbire"] as $opcija){
                        if(array_key_exists($y, $rezultat["izbire"][$x]["mozneSole"])){
                            $rezultat["izbire"][$x]["naziv"] . $rezultat["izbire"][$x]["mozneSole"][$y] . "<br>";
                            $rezultat["izbire"][$x]["povezava"][$y] = $this->dejavnost_model->poveziVse($rezultat["izbire"][$x]["mozneSole"][$y],
                                $rezultat["izbire"][$x]["mozniProgrami"],
                                $rezultat["izbire"][$x]["mozniLetniki"],
                                $rezultat["izbire"][$x]["mozniOddelki"]);
                            $y=$y+1;
                        }
                    }
                }
            }
            $this->load->view("dejavnost/moje", $rezultat);
        }
        else{
            $this->load->view("dejavnost/moje", $rezultat);
        }
    }
    else{
        $this->load->view("dejavnost/moje", $rezultat);
    }
}
```

Slika 19: Koda za prikaz dejavnosti

```
public function mojeDejavnostiProfesor($idOseba){
    $this->db->select("*");
    $this->db->from("dejavnost");
    $this->db->join("oseba_has_dejavnost", "dejavnost.idDejavnost=oseba_has_dejavnost.Dejavnost_idDejavnost");
    $this->db->where("oseba_has_dejavnost.oseba_idOseba", $idOseba);
    $this->db->where("avtor", 1);
    $this->db->where("stanje", 1);
    $this->db->order_by('casVnosa', 'DESC');
    $query = $this->db->get();
    return $rezultat = $query->result_array();
}
```

Slika 20: Poizvedba za dejavnosti

4.4.2 Prijava za dejavnost

Ker lahko dijak pošlje samo prošnjo za prijavo na dejavnost, jo mora nato profesor ali administrator potrditi. Spodnja koda prikaže vse prošnje za prijavo glede na to ali je profesor vključen v dejavnost ali ne. Profesor nato lahko prijavo potrdi ali pa zavrne. Aplikacija ima tudi funkcijo avtomatske prijave kjer se vsi dijaki prijavijo ne glede na to ali so poslali prošnjo ali ne

```
public function prosnjeDijakov(){
    if($this->session->userdata("vloga") != "profesor"){
        redirect("");
    }
    else{
        if($this->session->flashdata("idDejavnost") != null){
            $id = $this->session->flashdata("idDejavnost");
        }
        else{
            $id = $this->input->post("gumb");
        }
        $prosnjeUcenecv["prosnje"] = $this->dejavnost_model->pridobiProsnjeDejavnost($this->session->userdata("idOseba"), $id);

        if($prosnjeUcenecv["prosnje"] != NULL){
            for($x=0; $x<sizeof($prosnjeUcenecv["prosnje"]); $x++){
                $sez[$x] = $prosnjeUcenecv["prosnje"][$x]["Dejavnost_idDejavnost"];
            }
            $prosnjeUcenecv["prosnje"] = $this->dejavnost_model->pridobiProsnjeUcenec($sez);
            if($prosnjeUcenecv["prosnje"] != NULL){
                for($x=0; $x<sizeof($prosnjeUcenecv["prosnje"]); $x++){
                    if($prosnjeUcenecv["prosnje"][$x]["odobreno"] != 3){
                        unset($prosnjeUcenecv["prosnje"][$x]);
                    }
                }
                if($prosnjeUcenecv["prosnje"] != NULL){
                    for($x=0; $x<sizeof($prosnjeUcenecv["prosnje"]); $x++){
                        $koncna["prosnje"][$x] = $this->dejavnost_model->dijakiJePoslalProsnjo($prosnjeUcenecv["prosnje"][$x]["Oseba_idOseba"],$prosnjeUcenecv["prosnje"][$x]["Dejavnost_idDejavnost"]);
                    }

                    if($koncna["prosnje"] != NULL){
                        $this->load->view("dejavnost/prosnje", $koncna);
                    }
                }
            }
        }
        else{
            $koncna["prosnje"] = array();
            $this->load->view("dejavnost/prosnje", $koncna);
        }
    }
    else{
        $koncna["prosnje"] = array();
        $this->load->view("dejavnost/prosnje", $koncna);
    }
}
else{
    $koncna["prosnje"] = array();
    $this->load->view("dejavnost/prosnje", $koncna);
}
}
```

Slika: Koda za prikaz prošenj

```
public function avtomatskaPrijava(){
    if($this->session->userdata("vloga") != "admin"){
        redirect("");
    }
    else{
        $id = $this->input->post("gumb");
        $oddelki = $this->dejavnost_model->avtomatskaPrijavaPridobiLetnike($id);
        $moznaMesta = $this->dejavnost_model->avtomatskaPrijavaPridobiMoznaMesta($id);
        $moznaMesta = $moznaMesta[0]["moznaMesta"];
        $oddelki = explode(";", $oddelki[0]["mozniOdelki"]);
        unset($oddelki[count($oddelki)-1]);
        $dijakiKiUstrezajoPrijava = $this->dejavnost_model->avtomatskaPrijavaPridobiDijake($oddelki, $moznaMesta);
        if($dijakiKiUstrezajoPrijava == null){
            $this->session->set_flashdata("succes", "Vsi dijaki so že prijavljeni");
            redirect("dejavnost/vseDejavnosti");
        }
        else{
            foreach($dijakiKiUstrezajoPrijava as $dijak){
                $relacija = array(
                    "Oseba_idOseba" => $dijak["idOseba"],
                    "Dejavnost_idDejavnost" => $id,
                    "avtor" => 0,
                    "odobreno" => 1
                );
                $this->dejavnost_model->avtomatskaPrijavaUstvariRelacijo($id, $relacija);

                $this->dejavnost_model->potrditevUstvariPrisotnost($relacija["Dejavnost_idDejavnost"],$relacija["Oseba_idOseba"]);
            }
            $this->session->set_flashdata("succes", "Dijaki so bili uspešno prijavljeni");
            redirect("dejavnost/vseDejavnosti");
        }
    }
}
```

Slika 21: Avtomatska prijava

```

public function avtomatskaPrijavaPridobiletne($id){
    $this->db->select("mozniOddeleki");
    $this->db->from("dejavnost");
    $this->db->where("idDejavnost", $id);
    $query = $this->db->get();
    return $rezultat = $query->result_array();
}

public function avtomatskaPrijavaPridobiMoznaMesta($id){
    $this->db->select("moznaMesta");
    $this->db->from("dejavnost");
    $this->db->where("idDejavnost", $id);
    $query = $this->db->get();
    return $rezultat = $query->result_array();
}

public function avtomatskaPrijavaPridobiDijake($oddelki, $moznaMesta){
    #Create where clause
    $this->db->distinct();
    $this->db->select('oseba_idOseba');
    $this->db->from('oseba_has_dejavnost');

    $where_clause = $this->db->get_compiled_select();

    #Create main query
    $this->db->distinct();
    $this->db->select('idOseba');
    $this->db->from('oseba');
    $this->db->where("'idOseba' NOT IN ($where_clause)", NULL, FALSE);
    $this->db->where_in("Oddelek_idOddelek", $oddelki);
    $this->db->limit($moznaMesta);
    $query = $this->db->get();
    return $rezultat = $query->result_array();
}

public function avtomatskaPrijavaUstvariRelacijo($id, $relacija){
    $this->db->set("moznaMesta", "moznaMesta-1", FALSE);
    $this->db->where("idDejavnost", $id);
    $this->db->update("dejavnost");

    return $this->db->insert("oseba_has_dejavnost", $relacija);
}

```

Slika 22: Avtomatska prijava

```

public function dijaKiJePoslalProsnjo($idOseba, $idDejavnost){
    $this->db->select("*");
    $this->db->from("oseba_has_dejavnost");
    $this->db->join("oseba", "oseba.idOseba=oseba_has_dejavnost.Oseba_idOseba", "inner");
    $this->db->join("dejavnost", "dejavnost.idDejavnost=oseba_has_dejavnost.Dejavnost_idDejavnost", "inner");
    $this->db->where("Dejavnost_idDejavnost", $idDejavnost);
    $this->db->where("Oseba_idOseba", $idOseba);
    $query = $this->db->get();
    return $rezultat = $query->row_array();
}

```

Slika 23: Prošnja

```

public function potrditevUstvariPrisotnost($idDejavnost, $idDijaka){
    $this->db->select("datumZacetek, datumKonec");
    $this->db->from("dejavnost");
    $this->db->where("idDejavnost", $idDejavnost);
    $query = $this->db->get();
    $rezultat = $query->result_array();

    $datumi = array();
    $zacetek = strtotime($rezultat[0]["datumZacetek"]);
    $konec = strtotime($rezultat[0]["datumKonec"]);
    $step = '+1 day';
    $output_format = 'Y/m/d';
    while( $zacetek <= $konec ) {
        $datumi[] = date($output_format, $zacetek);
        $zacetek = strtotime($step, $zacetek);
    }

    foreach($datumi as $datum){
        $prisotnost = array(
            'idOseba' => $idDijaka,
            'idDejavnost' => $idDejavnost,
            'datum' => $datum
        );

        $this->db->insert("prisotnost", $prisotnost);
    }
}

```

Slika 24: Ustvaritev relacije za prisotnost

4.5 Datoteke

4.5.1 Izvoz v PDF in CSV

Spodaj sta sliki kode za generiranje PDF in CSV datoteke.

```
public function izvoziPrijavljeneCSV(){
    if($this->session->userdata("vloga") != "admin" and $this->session->userdata("vloga") != "profesor"){
        redirect("");
    }
    else{
        $id = $this->input->post("gumb");
        $file_name = 'student_details_on_' . date('Ymd') . '.csv';
        header("Content-Description: File Transfer");
        header("Content-Disposition: attachment; filename=$file_name");
        header("Content-Type: application/csv;");
        $student_data = $this->datoteke_model->prijavljeni($id);
        $file = fopen('php://output', 'w');

        $header = array("Dejavnost", "Ime", "Priimek", "Letnik", "Oddelek", "Program", "Sola", "id0seba");
        fputcsv($file, $header);
        foreach ($student_data->result_array() as $key => $value)
        {
            $value = mb_convert_encoding($value, "Windows-1252", "UTF-8");
            fputcsv($file, $value);
        }
        fclose($file);
        exit;
    }
}
```

Slika 25: Koda za generiranje CSV datoteke

```
public function izvoziPrijavljenePDF(){
    if($this->session->userdata("vloga") != "admin" and $this->session->userdata("vloga") != "profesor"){
        redirect("");
    }
    else{
        $id = $this->input->post("gumb");
        $idDejavnost = $id;
        $html_content = '<head>
<style>
table {
border-collapse: collapse;
width: 100%;
}

th, td {
text-align: left;
padding: 8px;
}

tr:nth-child(even){background-color: #f2f2f2}

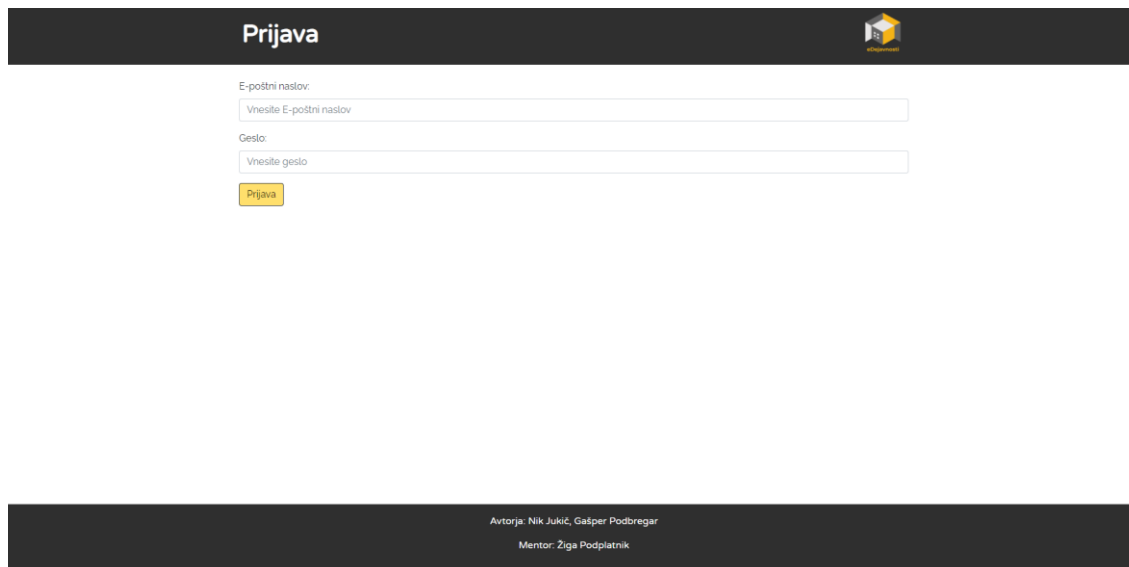
th {
background-color: #374045;
color: white;
}
</style>
</head><h3 align="center">Seznam prijavljenih na dejavnost</h3>';
        $html_content .= $this->datoteke_model->prijavljeniPDF($idDejavnost);
        $this->pdf->loadHtml($html_content);
        $this->pdf->render();
        $this->pdf->stream("". $idDejavnost . ".pdf", array("Attachment"=>0));
    }
}
```

Slika 26: Koda za generiranje PDF datoteke

5 Aplikacija

5.1 Prijava

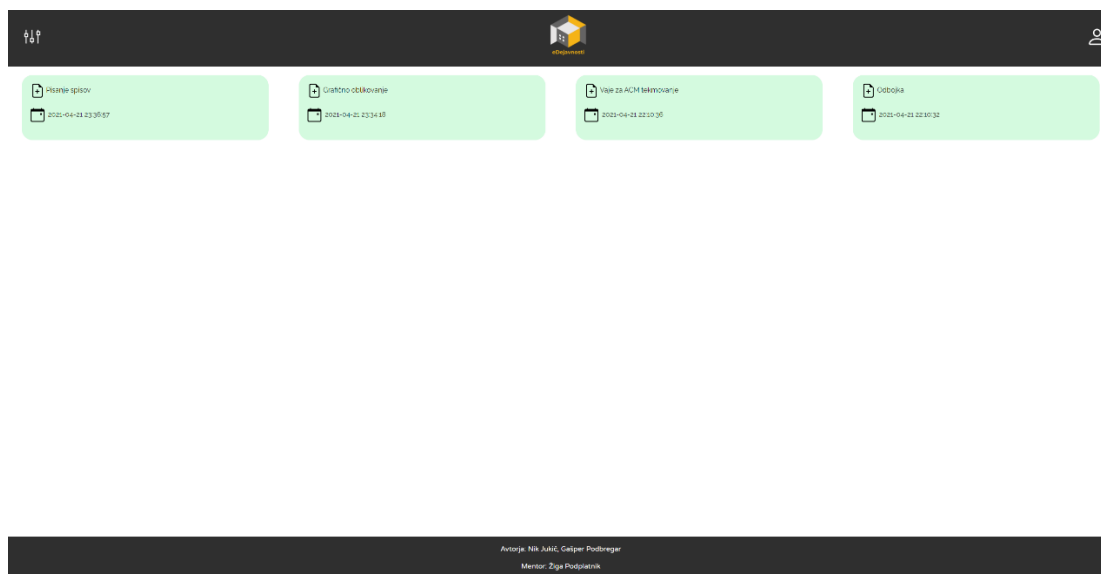
Uporabnik se na spodnji spletni strani prijavi v aplikacijo z danim uporabniškim imenom oziroma E – poštnim naslovom ter geslom.



Slika 27: Prijava v aplikaciji

5.2 Dijak

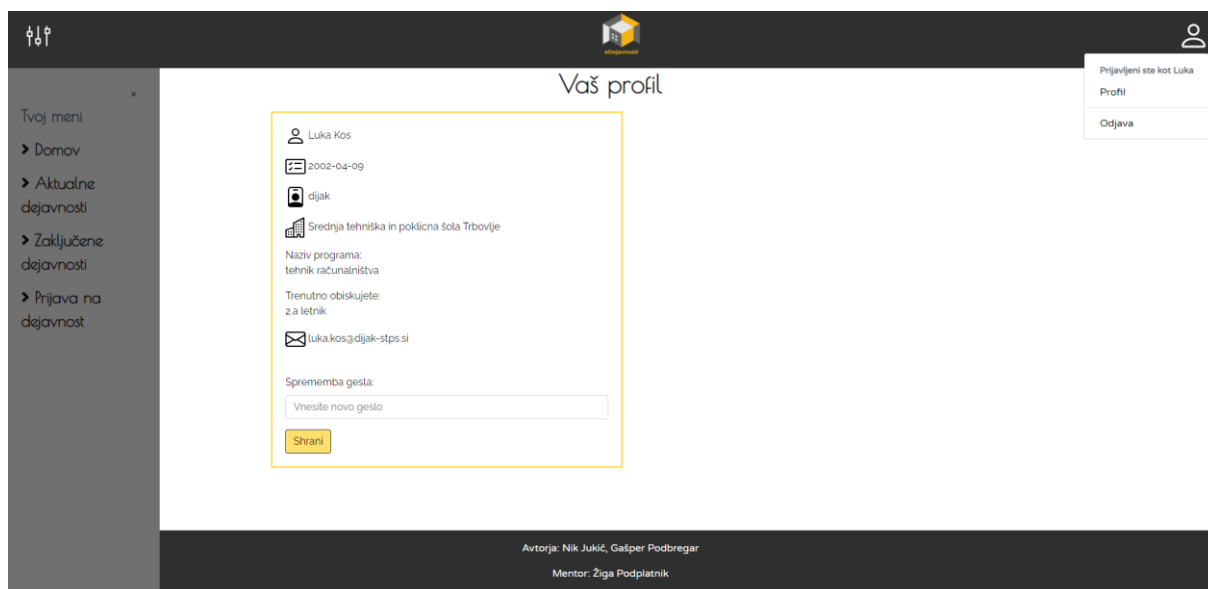
Če se uporabnik prijavi kot dijak ga čaka spodnja spletna stran kjer lahko vidi dejavnosti na katere se je prijavil in ali so bile prošnje za prijavo sprejete ali zavrnjene.



Slika 28: Domača stran dijak

5.2.1 Profil dijaka

Dijak ima možnost pregleda svojih podatkov. Lahko si tudi spremeni svoje geslo.



Slika 29: Profil dijaka

5.2.2 Aktualne dejavnosti dijaka

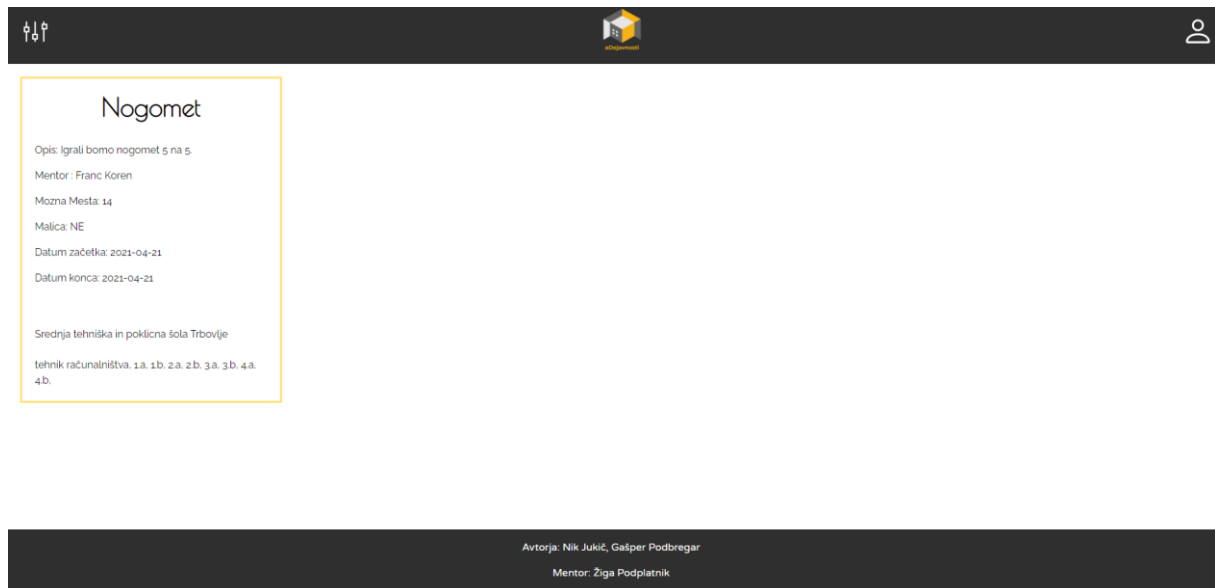
Tukaj se prikažejo vse dejavnosti na katere je dijak prijavljen in se še niso zaključile oziroma sploh začele.



Slika 30: Aktualne dejavnosti dijaka

5.2.3 Zaključene dejavnosti dijaka

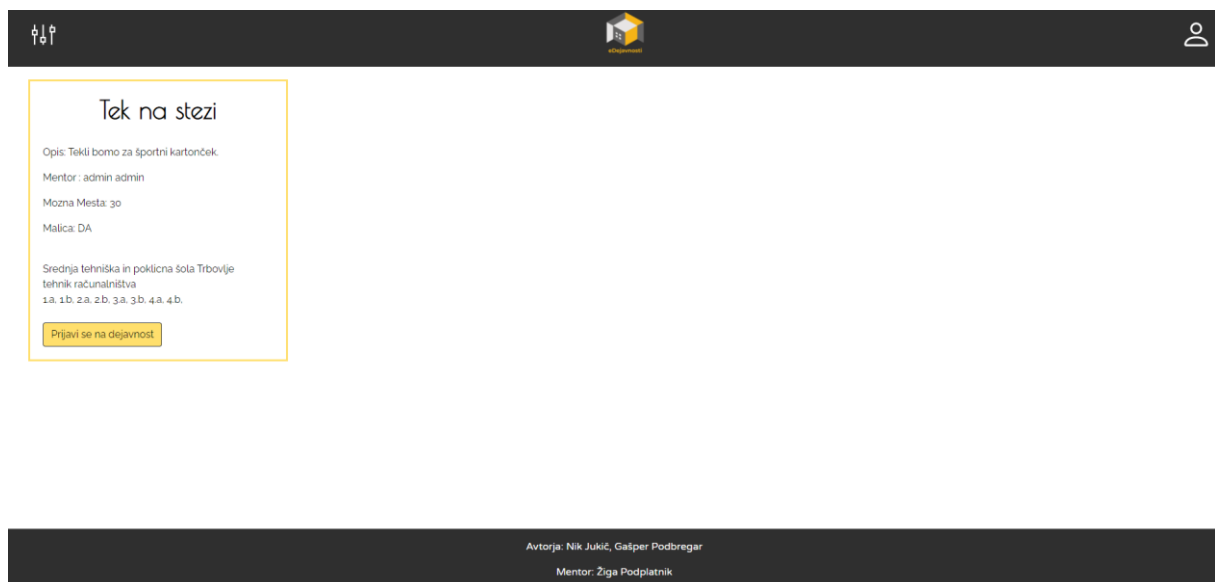
Ta pogled omogoča dijaku pregledati pretekle dejavnosti na katerih je sodeloval.



Slika 31: Zaključene dejavnosti dijak

5.2.4 Prijava na dejavnosti dijaka

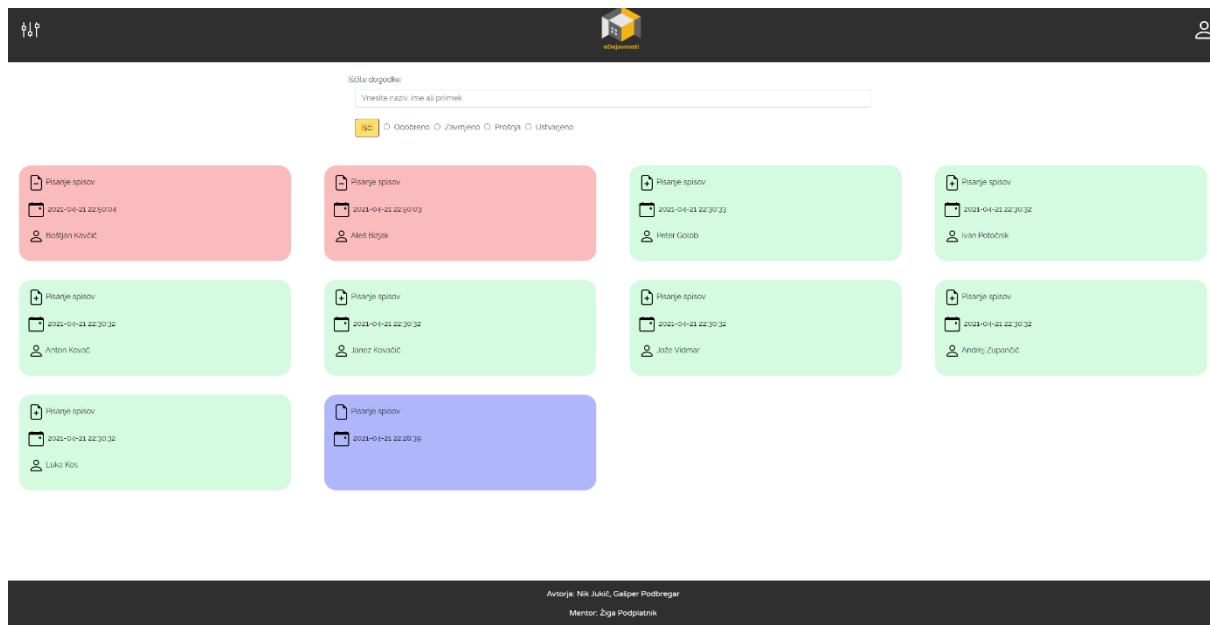
Dijak se lahko prijavi na dejavnosti



Slika 32: Prijava na dejavnosti dijak

5.3 Profesor

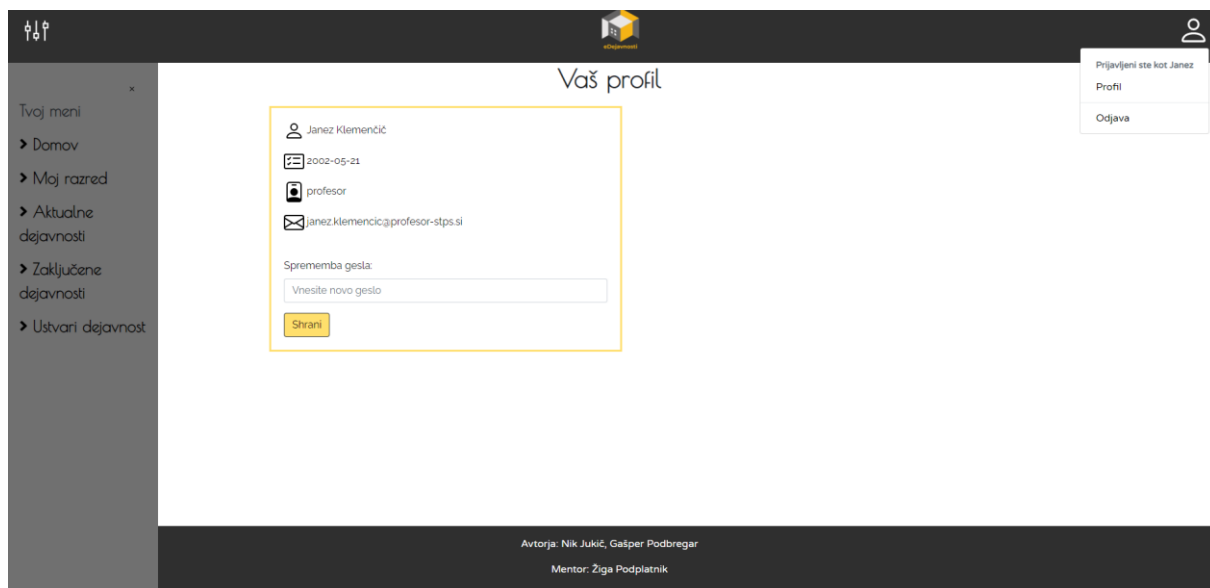
Ker imata profesor in administrator malo več sporočil imata tudi sistem za razvrščanje ter iskanje.



Slika 33: Domača stran profesor

5.3.1 Profil profesorja

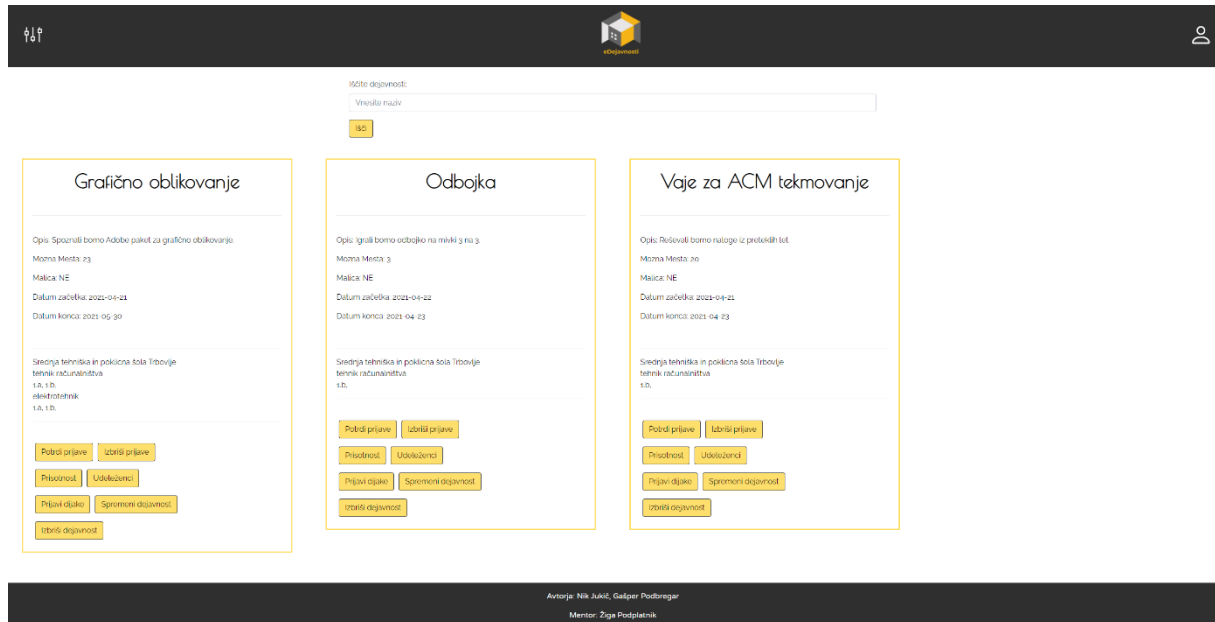
Kot dijak ima profesor enako funkcijo glede profila.



Slika 34: Profil profesorja

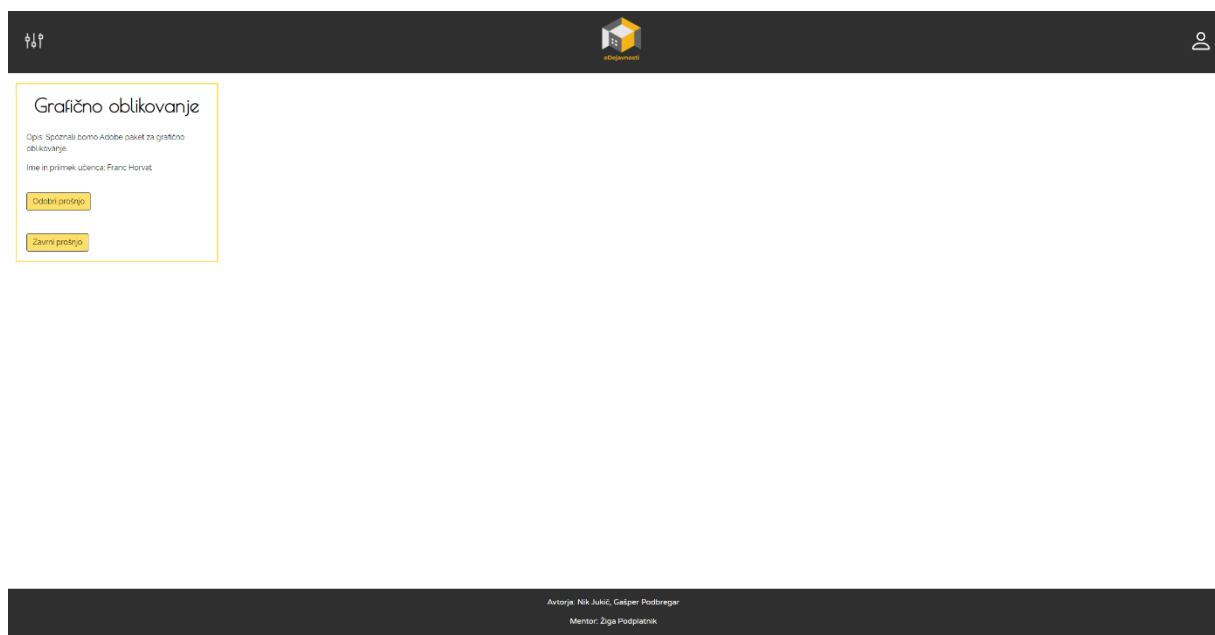
5.3.2 Aktualne dejavnosti profesorja

Profesor lahko vidi dejavnosti ki se dogajajo oziroma se bodo še zgodile, jih izbriše oziroma spremeni ter naredi naslednje stvari:



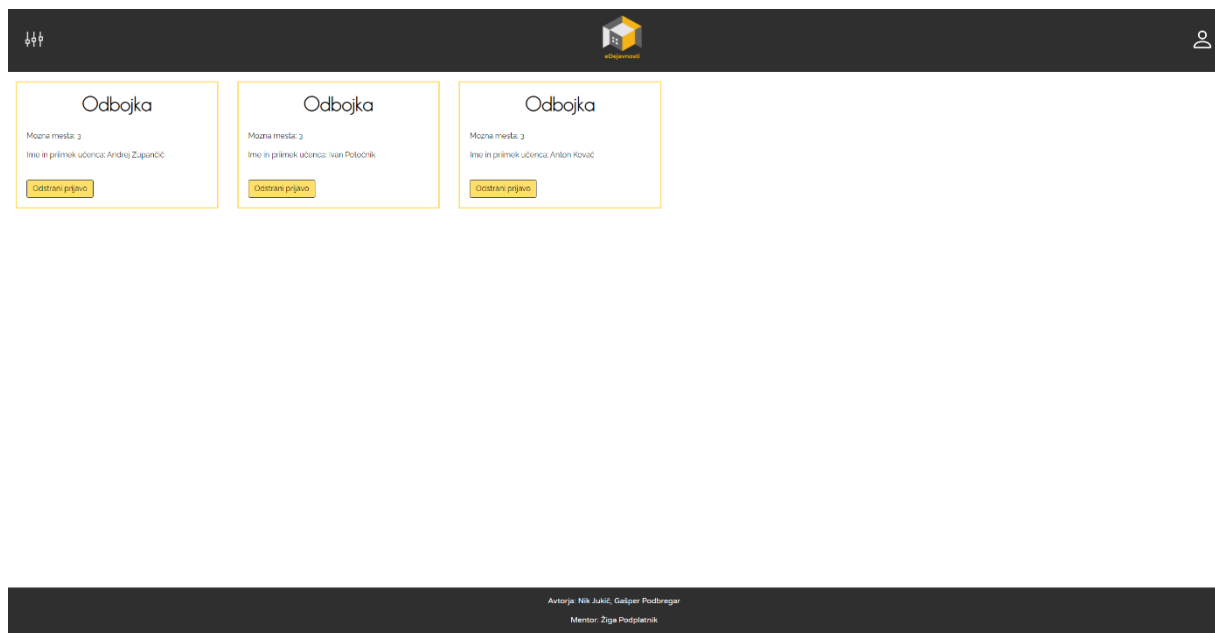
Slika 35: Aktualne dejavnosti profesor

5.3.2.1 Potrdi prijavo



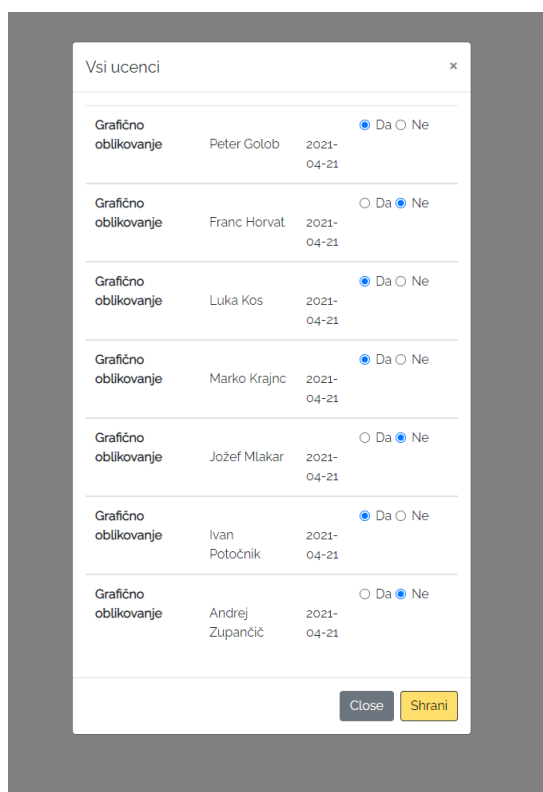
Slika 36: Potrdi prijavo profesor

5.3.2.2 Izbriši prijavo



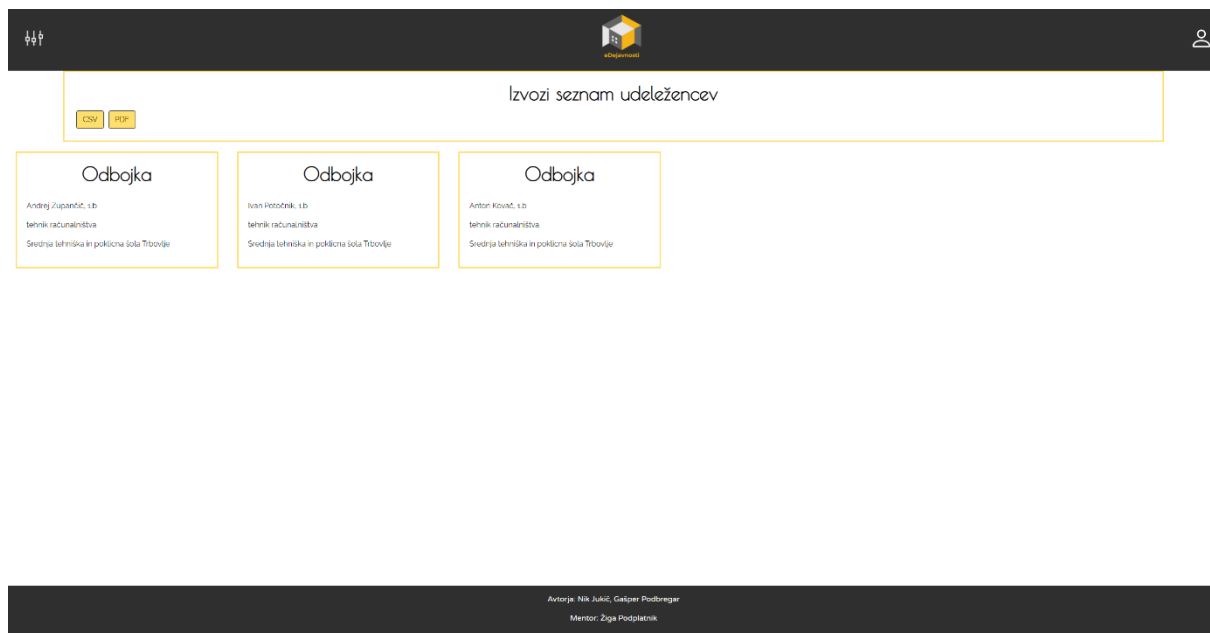
Slika 37: Izbriši prijavo profesor

5.3.2.3 Prisotnost



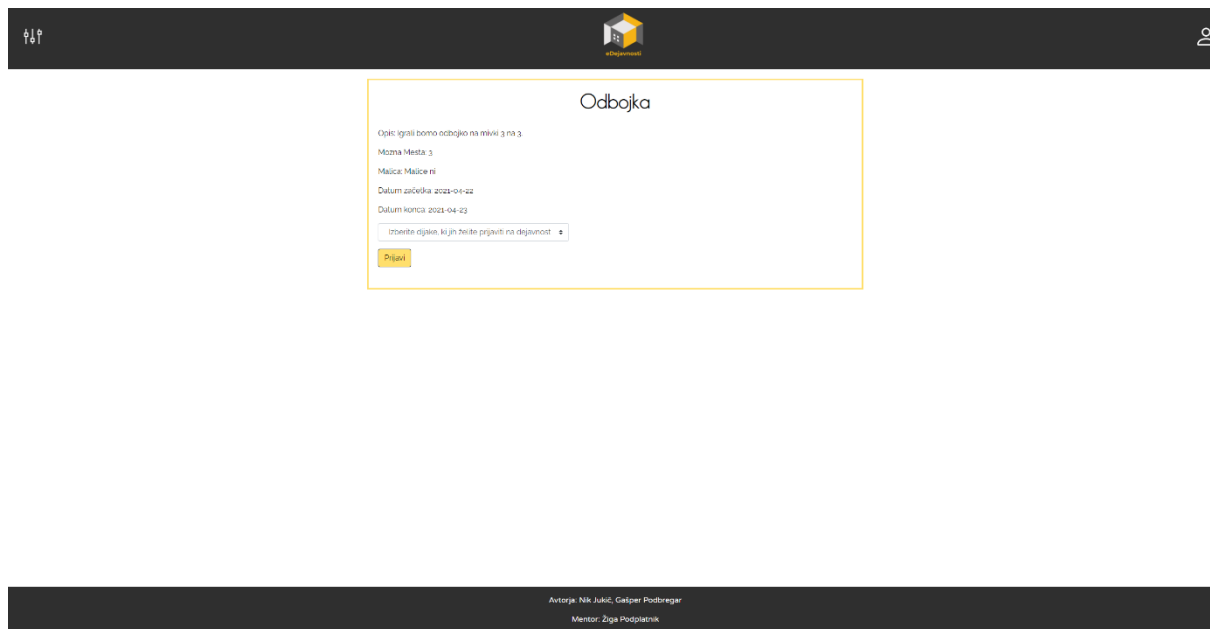
Slika 38: Urejanje prisotnosti profesor

5.3.2.4 Udeleženci



Slika 39: Udeleženci dejavnosti profesor

5.3.2.5 Prijavi dijake



Slika 40: Prijavi dijake

5.3.3 Razred profesorja

Profesor ima lahko vlogo razrednika in v podstrani Moj razred vidi dijake ter izvozi tiste ki so trenutno prijavljeni na vsaj eno dejavnost. Lahko tudi vidi pretekle dejavnosti ter tudi te izvozi v PDF.

Izvozi prijave:

CVS PDF Preglej prisotnost

Ivan Potočnik, 1b
tehnik računalništva

Jožef Mlakar, 1b
tehnik računalništva

Andrej Zupančič, 1b
tehnik računalništva

Anton Kovač, 1b
tehnik računalništva

Avtorja: Nik Jukič, Gašper Podbregar
Mentor: Žiga Podplatnik

Slika 41: Razred profesorja

Pretekle dejavnosti 1b:

Nogomet

Mentor: Franc Koren

Opis: Igrali bomo nogomet 5 na 5.

Datum začetka: 2021-04-21

Datum konca: 2021-04-21

PDF

Avtorja: Nik Jukič, Gašper Podbregar
Mentor: Žiga Podplatnik

Slika 42: Pretekle dejavnosti profesor

To je primer izvoza prisotnosti na dejavnostih.


Seznam prijavljenih na dejavnost

Dejavnost	Ime	Priimek	Datum	Prisotnost
Osnove strojne opreme	Anton	Kovač	2021-04-21	ne
Osnove strojne opreme	Anton	Kovač	2021-04-22	ne
Osnove strojne opreme	Anton	Kovač	2021-04-23	ne
Osnove strojne opreme	Jožef	Mlakar	2021-04-21	da
Osnove strojne opreme	Jožef	Mlakar	2021-04-22	ne
Osnove strojne opreme	Jožef	Mlakar	2021-04-23	ne
Osnove strojne opreme	Ivan	Potočnik	2021-04-21	da
Osnove strojne opreme	Ivan	Potočnik	2021-04-22	ne
Osnove strojne opreme	Ivan	Potočnik	2021-04-23	ne
Osnove strojne opreme	Andrej	Zupančič	2021-04-21	ne
Osnove strojne opreme	Andrej	Zupančič	2021-04-22	ne
Osnove strojne opreme	Andrej	Zupančič	2021-04-23	ne

Slika 43: Izvoz prisotnosti profesor

5.3.4 Ustvari dejavnost profesor

Spodaj je prikazan obrazec s katerim vnesemo novo dejavnost.



Naziv Dejavnosti

Možna mesta

Opis dejavnosti:

Malica:

☐ Da

☐ Ne

Datum začetka dejavnosti:

Datum konca dejavnosti:

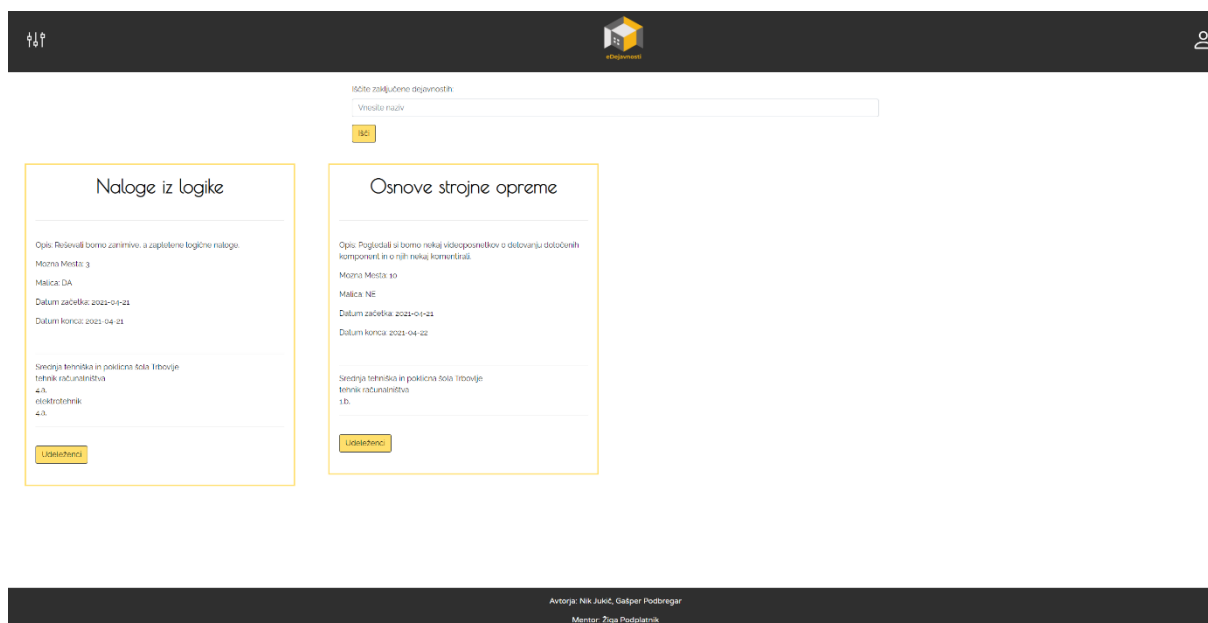
Avtorje: Nik Jukič, Gašper Podbregar

Mentor: Žiga Podplatnik

Slika 44: Nova dejavnost profesor

5.3.5 Zaključene dejavnosti profesor

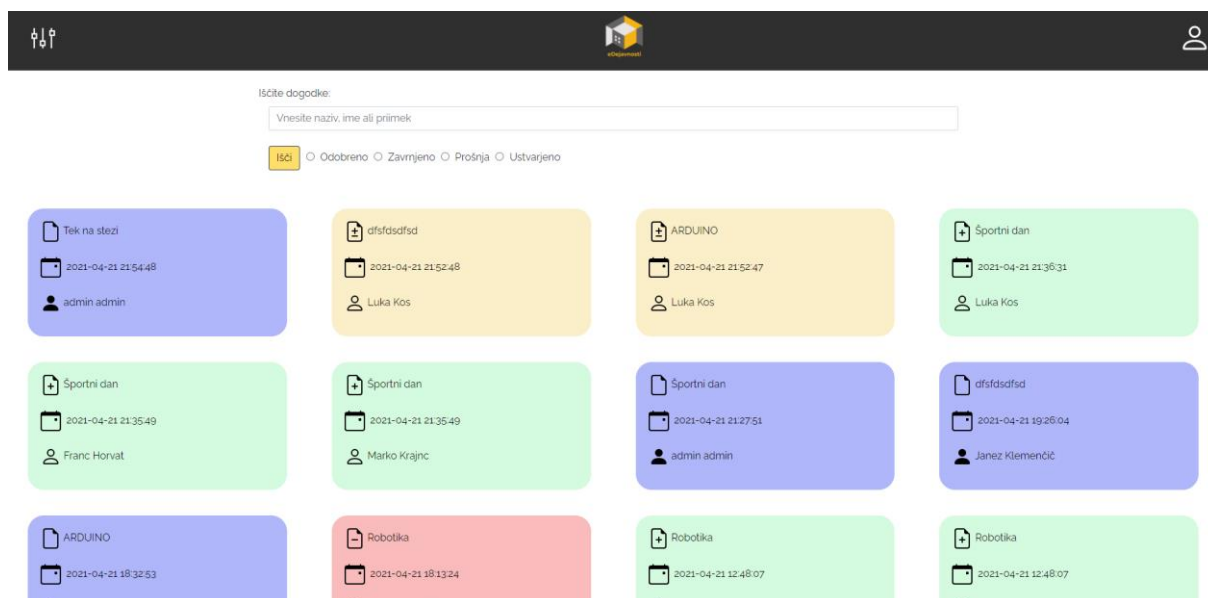
Profesor ima tudi možnost pregledati pretekle dejavnosti in pogledati njihovo udeleženoost.



Slika 45: Zaključene dejavnosti profesor

5.4 Administrator

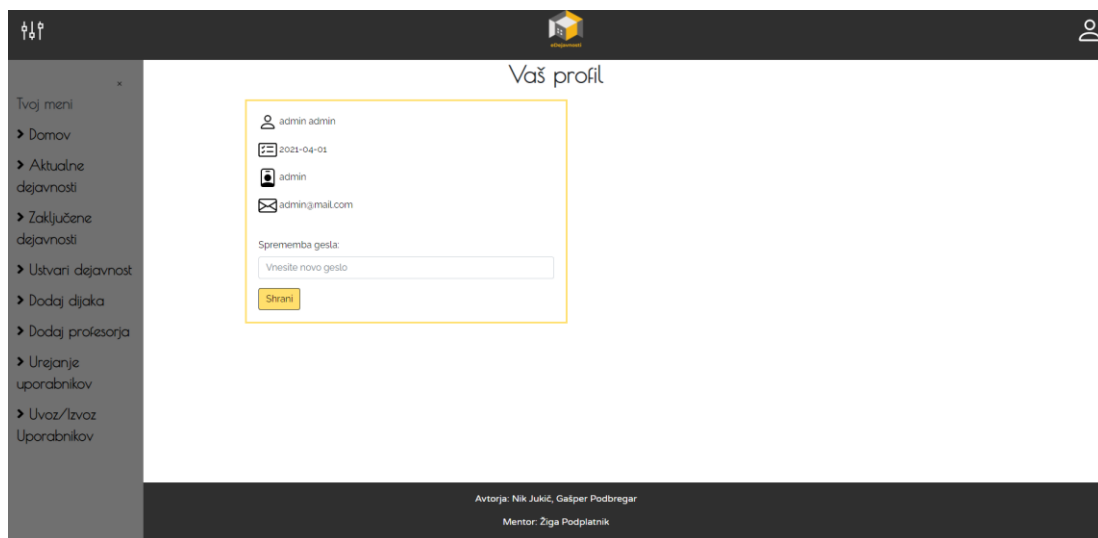
Domača stran administratorja je podobna kot profesorjeva le da ima ta še več sporočil.



Slika 46: Domača stran profesor

5.4.1 Profil Administratorja

Administrator ima enake funkcije kot profesor z nekaj dodatki. Lahko dodaja uporabnike, jih ureja ter uvaža oziroma izvaža CSV datoteko uporabnikov. Lahko tudi avtomatsko prijavi dijake na dejavnosti.



Slika 47: Profil administratorja

5.4.2 Dodajanje dijakov administrator

Ime:

Priimek:

Geslo:

Datum Rojstva:

Spol:

☐ M

☐ Ž

Izberite šolo

Izberite program

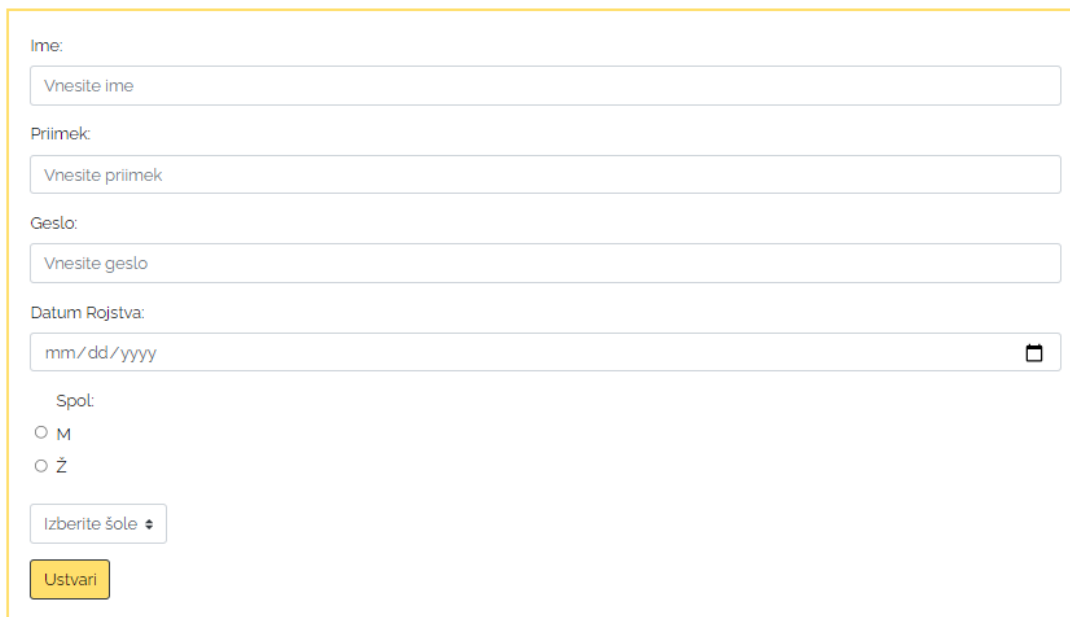
Izberite letnik

Izberite oddelek

Submit

Slika 48: Obrazec dodajanja dijakov

5.4.3 Dodajanje profesorjev administrator



Ime:

Priimek:

Geslo:

Datum Rojstva:

Spol:

☐ M

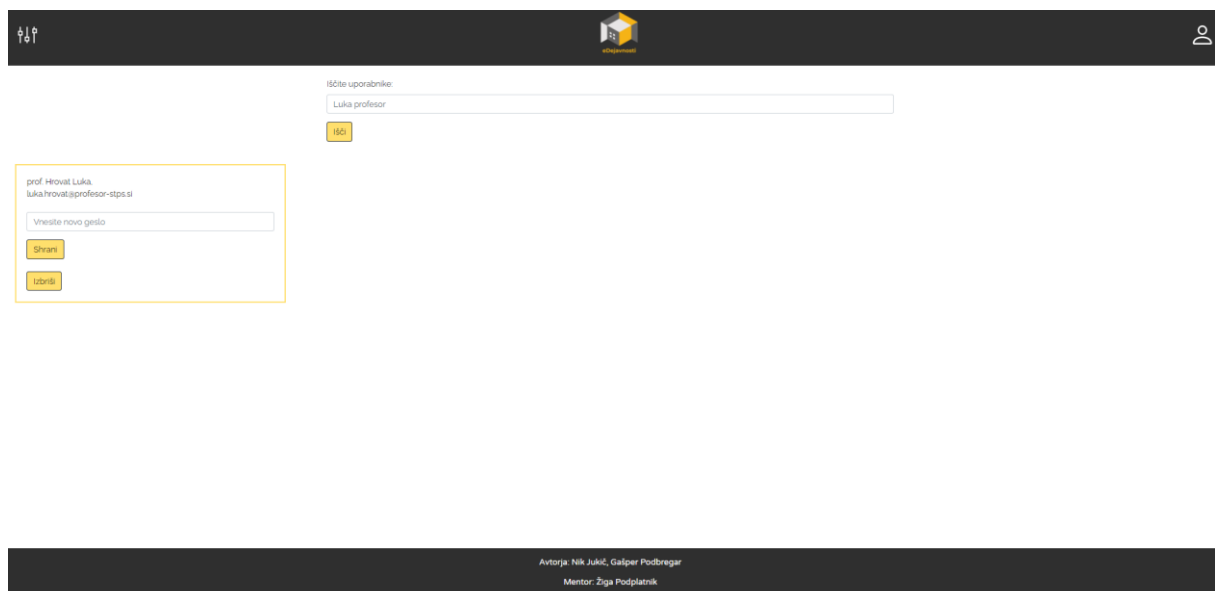
☐ Ž


Izberite šole ▾

Slika 49: Obrazec za dodajanje profesorjev

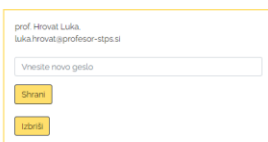
5.4.4 Urejanje uporabnikov administrator

Tukaj lahko administrator ureja gesla in izbriše uporabnika.



👤🔑  Do

Išči uporabnike:



prof. Hrovat Luka,
luka.hrovat@profesor-stps.si

Slika 50: Urejanje uporabnikov administrator

5.4.5 Izvažanje in uvažanje uporabnikov

Administrator tukaj uvozi oziroma izvozi uporabnike glede na izbrane atribute.



Tukaj lahko uvažate uporabnike in generirate podatke za prijavo

Uvažanje dijakov

Izberite datoteko

Choose File No file chosen

Uvozi dijake

Prenesi CSV datoteko za uvoz dijakov

Izberite šolo

Izberite program

Izberite letnik

Izberite oddelek

Generiraj podatke za prijavo

Uvažanje profesorjev

Izberite datoteko

Choose File No file chosen

Uvozi profesorje

Prenesi CSV datoteko za uvoz profesorjev

Izberite šolo

Generiraj podatke za prijavo

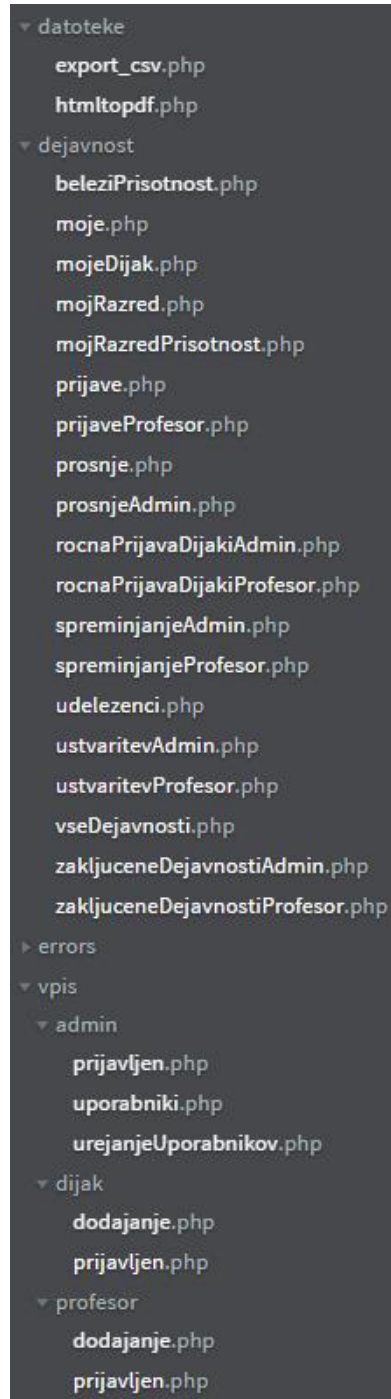
Avtorja: Nik Jukič, Gašper Podbregar

Mentor: Žiga Podplatnik

Slika 51: Izvoz in uvoz administrator

6 Pogledi

Pogledi so razdeljeni na prijavo, dejavnost ter izvoz datoteke. Sestavljeni so iz noge, menija ter glave, ki so v svojih datotekah.



Slika 52: Vsi pogledi

6.1 Knjižnice

Spodnje slike prikazujejo vse uporabljene zunanje knjižnice v aplikaciji.

Slogi pisav:

```
<!-- Vsi fonti so spodaj -->
<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?family=Raleway&display=swap" rel="stylesheet">
<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?family=Nanum+Myeongjo&display=swap" rel="stylesheet">
<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap" rel="stylesheet">
<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap" rel="stylesheet">
<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?family=Poiret+One&family=Varela+Round&display=swap" rel="stylesheet">
```

Slika 53: Zunanji slogi pisav

Knjižnice

```
<script src="<?php echo base_url() ?>assets/js/jquery-3.5.1.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>

<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-multiselect/0.9.16/js/bootstrap-multiselect.js"></script>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-multiselect/0.9.16/css/bootstrap-multiselect.css" type="text/css"/>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/3pcUUGGOn+Y7RswNrtH/TE3MoK7ZeZDys" crossorigin="anonymous"></script>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TXBt27ECRE3e/1hU7zmqVincDaySuIK24rEkgtIXeRed4M0j\1FIDPvg6uqKI2xXr2" crossorigin="anonymous"></link>
```

Slika 54: Knjižnice 1

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/malihu-custom-scrollbar-plugin/3.1.5/jquery.mCustomScrollbar.min.css">
<script defer src="https://use.fontawesome.com/releases/v5.0.13/js/solid.js" integrity="sha384-tzzSw1/Vo+8NSUHGTP3bvaMPq+uvzCMfrN1fEFe+xBmv1C/AtVxSKBuZtmChIFZ" crossorigin="anonymous"></script>
<script defer src="https://use.fontawesome.com/releases/v5.0.13/js/fontawesome.js" integrity="sha384-60Irr52G8BnPOF5Zdxxz1xdN5ndLD4vdcf/q2myIUV00VsqGh3B0RaBE01VTOY" crossorigin="anonymous"></script>
<link rel="stylesheet" href="/igniter/assets/css/style.css">
<script src="assets/js/myscripts.js"></script>
```

Slika 55: Knjižnice 2

Spodaj so prikazane tudi knjižnice za izvoz uporabnikov v PDF.

```
<?php defined('BASEPATH') OR exit('No direct script access allowed');

require_once 'dompdf/autoload.inc.php';

use Dompdf\Dompdf;
class Pdf extends DOMPDF
{
    protected function ci()
    {
        return get_instance();
    }

    public function load_view($view, $data = array())
    {
        $dompdf = new Dompdf();
        $html = $this->ci()->load->view($view, $data, TRUE);

        $dompdf->loadHtml($html);

        // (Optional) Setup the paper size and orientation
        $dompdf->setPaper('A4', 'landscape');

        // Render the HTML as PDF
        $dompdf->render();
        $time = time();

        // Output the generated PDF to Browser
        $dompdf->stream("welcome-". $time);
    }
}
```

Slika 56: Izvoz pdf

7 Izgled

7.1 Logotip



Pri izdelavi logotipa sva uporabila odtenke spodaj navedene barvne sheme. Za navdih pri poimenovanju nama je pomagala znana aplikacija eAsistent, ki deluje na podoben princip z isto ciljno množico.

Slika 57: Logotip

7.2 Barvna shema

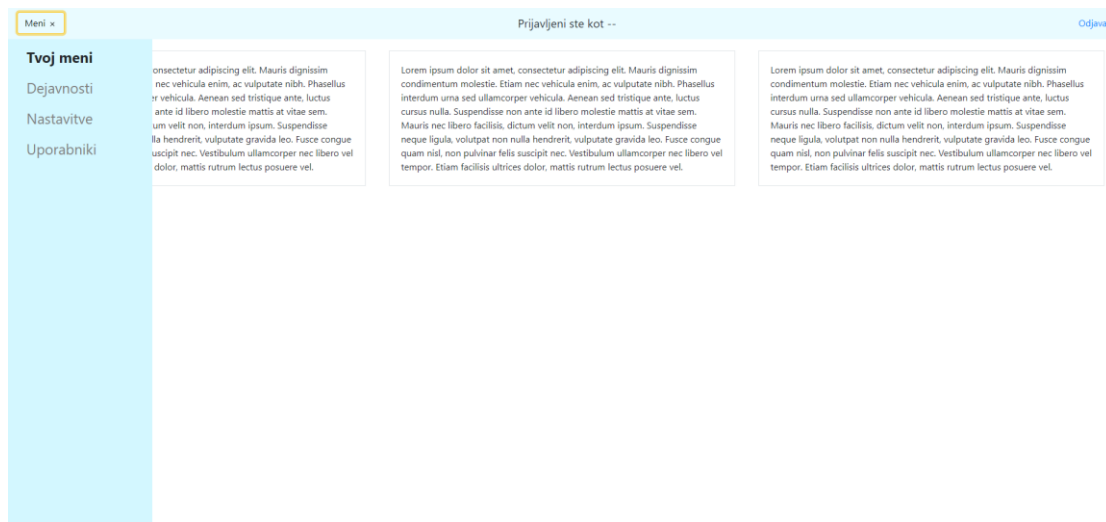
Spodnja slika predstavlja barvno shemo, uporabljeno v aplikaciji.



Slika 58: Barvna shema

7.3 Prvotni izgled

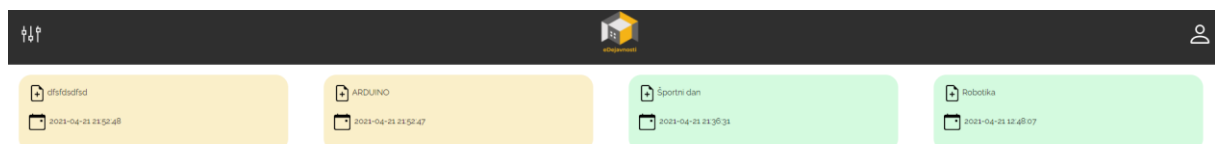
Prvotna ideja spletne aplikacije je bila zelo preprosta in svetla. Skozi samo izdelavo je bilo veliko različic z drugimi barvnimi shemami. Na koncu sva se odločila za temnejšo shemo z dodatkom ene živahnije barve.



Slika 59: Prvoten izgled

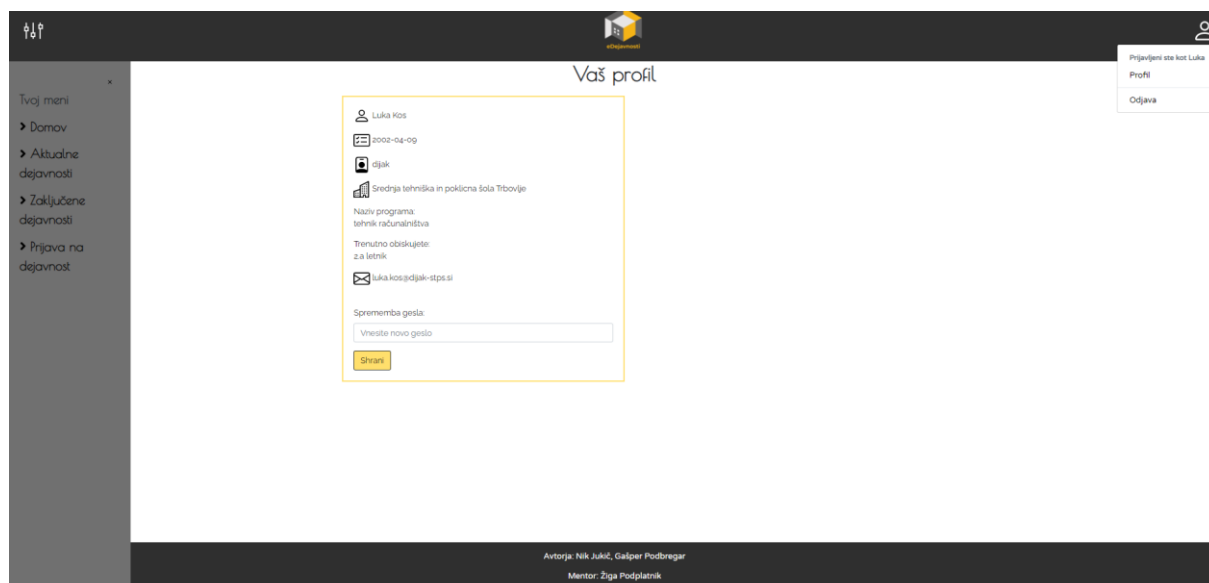
7.4 Končni izgled

Prikazana je domača stran dijaka, kjer lahko na oglasni deski vidi novosti o dejavnostih, katere se ga tičejo. Vidi svoje prošnje, potrjene prijave ter prijave, ki jih je višji uporabnik zavrnil.



Slika 60: Domača stran dijaka

Vsak uporabnik ima svoj meni in pregled uporabniškega računa.



Slika 61: Pregled računa dijak

Celotna aplikacija uporablja odzivno ogrodje Bootstrap. Zgrajena je s tehnologijo 12 stolpcev, ki se povečujejo oziroma zmanjšujejo, glede na velikost uporabnikove naprave.

7.5 Pisava

Uporabila sva naslednje pisave, ki so objavljene na spletni strani <https://fonts.google.com/>, in so brezplačne.

— Poiret One

A	B	C	Č	Ć	D	Đ	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	Š	T	U	V	W	X	Y	Z	Ž	a	b
c	č	ć	d	đ	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	š	t	u	v	w	x	y	z	ž	А	Б	В	Г
Г	Д	Ђ	Е	Ѓ	Є	Ж	З	Ѕ	И	І	Ї	Й	Ј	К	Л	Љ	М	Н	Њ	О	П	Р	С	Т	Ћ	У	Ў	Ф	Х	Ц	Ч	Џ
Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	а	б	в	г	ѓ	д	ђ	е	ё	є	ж	з	ѕ	и	і	ї	й	ј	к	л	љ	м	н	њ	о
п	р	с	т	ћ	у	ў	ф	х	ц	ч	џ	ш	щ	ъ	ы	ь	э	ю	я	Ă	Â	Ê	Ô	Õ	Ů	Ǻ	â	ê	ô	ơ	ư	1
2	3	4	5	6	7	8	9	0	'	?	'	"	!	"	(%)	[#]	[@]	/	&	\	<	-	+	÷	×	=
>	®	©	\$	€	£	¥	¢	:	:	.	.	*																				

Slika 62: Pisava Poiret One

— Varela Round

A	B	C	Č	Ć	D	Đ	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	Š	T	U	V	W	X	Y	Z	Ž	a	b
c	č	ć	d	đ	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	š	t	u	v	w	x	y	z	ž	а	б	в	г
г	д	ђ	е	ѓ	є	ж	з	ѕ	и	і	ї	й	ј	к	л	љ	м	н	њ	о	п	р	с	т	ћ	у	ў	ф	х	ц	ч	џ
ш	щ	ъ	ы	ь	э	ю	я	а	б	в	г	ѓ	д	ђ	е	ё	є	ж	з	ѕ	и	і	ї	й	ј	к	л	љ	м	н	њ	о
п	р	с	т	ћ	у	ў	ф	х	ц	ч	џ	ш	щ	ъ	ы	ь	э	ю	я	Ă	Â	Ê	Ô	Õ	Ů	Ǻ	â	ê	ô	ơ	ư	1
2	3	4	5	6	7	8	9	0	'	?	'	"	!	"	(%)	[#]	[@]	/	&	\	<	-	+	÷	×	=
>	®	©	\$	€	£	¥	¢	:	:	.	.	*																				

Slika 63: Pisava Varela Round

– Raleway

A	B	C	Č	Ć	D	Đ	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	Š	T	U	V	W	X	Y	Z	Ž	a	b
с	č	ć	d	đ	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	š	t	u	v	w	x	y	z	ž	А	Б	В	Г
Г	Д	Ђ	Е	Ё	Є	Ж	З	Ѕ	И	І	Ї	Ј	К	Л	Љ	М	Н	Њ	О	П	Р	С	Т	Ћ	У	Ў	Ф	Х	Ц	Ч	Џ	
Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	а	б	в	г	ђ	е	ё	є	ж	з	ѕ	и	і	ї	ј	к	л	љ	м	н	њ	о			
п	р	с	т	ћ	у	џ	ф	х	ц	ч	џ	ш	щ	ъ	ы	ь	э	ю	я	Ă	Â	Ê	Ô	Ö	Ů	ă	â	ê	ô	ơ	ı	1
2	3	4	5	6	7	8	9	0	'	?	'	"	!	"	(%)	[#]	[@]	/	&	\	<	-	+	÷	x	=
>	®	©	\$	€	£	¥	¢	:	;	,	.	*																				

Slika 64: Pisava Raleway

8 Razprava in zaključek

Aplikacija vsebuje vse načrtovane funkcije in opravlja vse zadane naloge. Vsebuje tri poglede – dijak, profesor in administrator – pri vsakem pogledu vsebuje zadane funkcije in se prilagaja glede na dogajanje pri drugih pogledih. Njena največja pomanjkljivost je optimizacija. Lahko bi bila tudi bolj pregledna. Če bi še enkrat izdelovala isto aplikacijo, bi se mnogo stvari lotila drugače, tako bi si olajšala delo in izboljšala delovanje aplikacije.

Aplikacija bi lahko bila testirana v realnem primeru, kjer bi test vseboval profesorja, ki ustvari aktivnost, ter dijake, ki se nanjo prijavijo. Administrator bi potem pregledal potek in videl dejansko delovanje aplikacije.

Največ težav sva imela pri podatkovni bazi, ki se je med izdelavo kar nekajkrat spremenila.

9 Spletni viri

- 1 Google fonts. Ogledano 5. 11. 2020.
<<https://fonts.google.com/specimen/Poiret+One?query=poiret#standard-styles>>
- 2 Google fonts. Ogledano 5. 11. 2020.
<<https://fonts.google.com/specimen/Varela+Round?query=varela#glyphs>>
- 4 Google fonts. Ogledano 5. 11. 2020.
<<https://fonts.google.com/specimen/Raleway?query=ralewa>>
- 5 CodeIgniter. Wikipedia. Ogledano 30. 11. 2020.
<<https://en.wikipedia.org/wiki/CodeIgniter>>
- 6 HTML. Wikipedia. Ogledano Ogledano 16. 4. 2021.
<<https://en.wikipedia.org/wiki/HTML>>
- 7 CSS. Wikipedia. Ogledano 16. 4. 2021. <<https://en.wikipedia.org/wiki/CSS>>
- 8 PHP. Wikipedia. Ogledano 16. 4. 2021. <<https://en.wikipedia.org/wiki/PHP>>
- 9 JavaScript. Wikipedia. Ogledano 16. 4. 2021. <<https://en.wikipedia.org/wiki/JavaScript>>
- 10 Learn CodeIgniter (v3.1.10) Framework Beginners Tutorial. Youtube. Ogledano 30. 10. 2020.
<<https://www.youtube.com/playlist?list=PLT9miexWCpPVhzIeuMKAAanfOvCbMEHH29>>
- 11 Bootstrap. Ogledano 31. 10. 2020.
<<https://getbootstrap.com/docs/4.0/getting-started/introduction/>>
- 12 GitHub. Ogledano 30. 10. 2020. <<https://github.com/>>
- 13 Dompdf. Ogledano 3. 4. 2021. <<https://github.com/dompdf/dompdf>>
- 14 Yara1990 csvimport. Ogledano 3. 4. 2021. <<https://github.com/Yara1990/csvimport>>