Univerzitet u Beogradu

Fakultet organizacionih nauka

Laboratorija za elektronsko poslovanje



APLIKACIJA ZA POLAGANJE VOŽNJE

Seminarski rad iz predmeta Internet tehnologije

Studenti:

Nataša Jovanović 2017/1084

Stefan Jovanović 2017/0153

Beograd, 2022.

Sadržaj

[1 Korisnički zahtev 3](#_Toc108813231)

[2 Slučajevi korišćenja 4](#_Toc108813232)

[2.1 Prijava na sistem 5](#_Toc108813233)

[2.2 Upis kandidata 5](#_Toc108813234)

[2.3 Unos rasporeda ispita 5](#_Toc108813235)

[2.4 Prijava ispita 5](#_Toc108813236)

[2.5 Unos broja poena 5](#_Toc108813237)

[2.6 Pregled korisnika 6](#_Toc108813238)

[3 Arhitektura 11](#_Toc108813239)

[3.1 Dijagram klasa 11](#_Toc108813240)

[4 Specifikacija REST API-ja 12](#_Toc108813241)

[5 Opis korišćenih tehnologija 13](#_Toc108813242)

[HTML 13](#_Toc108813243)

[CSS 14](#_Toc108813244)

[PHP 15](#_Toc108813245)

[AJAX(Asynchronous JavaScript And XML) 16](#_Toc108813246)

[JSON 17](#_Toc108813247)

[JavaScript 17](#_Toc108813248)

[Flight 18](#_Toc108813249)

[6 Korisničko uputstvo 19](#_Toc108813250)

[6.1 Instruktor 19](#_Toc108813251)

[6.2 Administrator 22](#_Toc108813252)

[7 Arhitektura koda aplikacije 27](#_Toc108813253)

# Korisnički zahtev

Potrebno je projektovati i implementirati web aplikaciju poslovnog informacionog sistema škole za obuku vozača. Sistem treba da radi sa dve vrste korisnika: zaposlenima u školi obuke (instruktorima) i sa administratorima sistema. Korisnicima sistema treba omogućiti logovanje na sistem. Škola je zadužena za unos/izmenu broja poena ostvarenih na predmetima, unos kandidata u sistem, kao i dodavanje novih predmeta u skladu sa kriterijumima ministarstva unutrašnjih poslova. Administrator sistema ima pristup istim funkcionalnostima kao i instruktori u školi, a osim tih funkcionalnosti ima mogućnost izmene i mogućnost dodavanja novih instruktora.

Slučajevi korišćenja:

1. Prijava korisnika na sistem

2. Upis kandidata

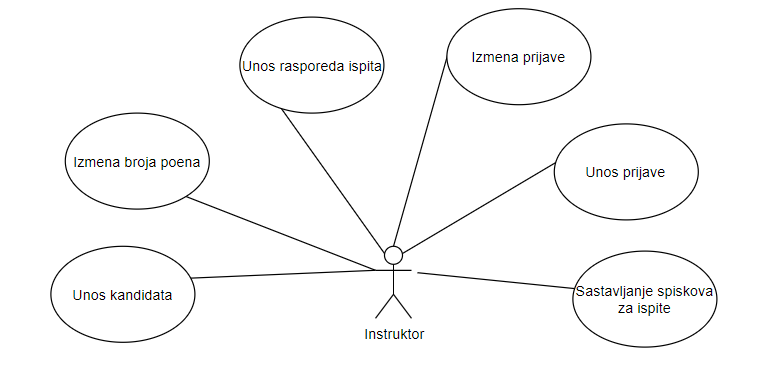
3. Unos rasporeda ispita

4. Prijava ispita

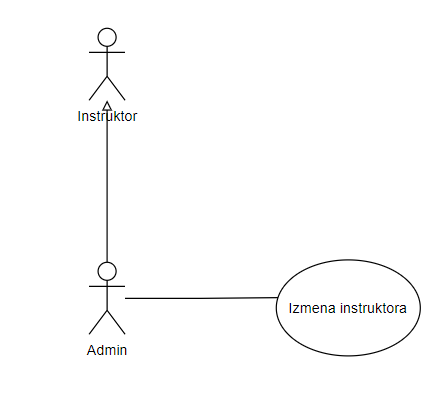
5. Unos broja ostvarenih poena

6. Pregled korisnika

# Slučajevi korišćenja



Slika - Dijagram slučajeva korišćenja za zaposlene u školi



Slika - Dijagram slučajeva korišćenja za administratore

## Prijava na sistem

Prijavu na sistem mogu izvršiti samo autorizovana lica tj. administratori sistema ili odgovarajući instruktori koji su zaposleni u školi. Administratori mogu dodati nove zaposlene koji zatim mogu da se prijave na sistem kao instruktori.

## Upis kandidata

Instruktor za svakog prijavljenog kandidata koji je izašao da polaže ispit popunjava u prijavi broj poena koje je kandidat dobio. Za ostale se evidentira da kandidat nije izašao na ispit. Rezultati ispita i prijave se potom unose u sistem, gde se sve prijave raspoređuju po odgovarajućim dosijeima prijavljenih kandidata (nevezano od toga da li je kandidat položio deo ispita ili ne, odnosno da li je uopšte izašao na ispit).

## Unos rasporeda ispita

Nekoliko dana pre nego što kandidati mogu da polažu deo ispita, instruktor unosi raspored ispita u sistem. Ukoliko je rok u periodu održavanja neke manifestacije zbog koje je saobraćaj u gradu povećan ili zabranjen, prvobitan raspored koji je određen intervalima, podlaže eventualnim alteracijama nakon čega se određuju tačni datumi kada kandidati mogu da izađu i polažu deo ispita, te instruktor ažurira raspored delova ispita.

## Prijava ispita

U vreme rokova kada je moguće polagati neke od delova ispita, kandidati popunom prijava prijavljuju one delove ispita koje žele da polažu u definisanom ispitnom roku. Sve prijave se evidentiraju u školi gde instruktor proverava da li su uneti podaci na prijavi korektni i da li kandidat ima pravo da polaže prijavljeni ispit.

Uvidom u karton kandidata, ako je broj pokušaja polaganja dela ispita veći od tri, kandidat dodatno plaća odgovarajuću nadoknadu.

Ako kandidat ne podnese prijavu na vreme, naknadni prijem prijave se takođe plaća.

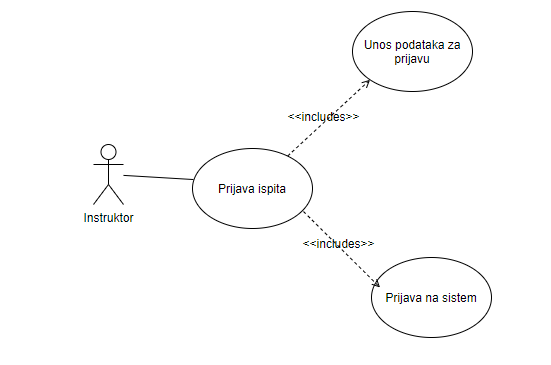
Na osnovu ispravnih prijava formiraju se ispitni spiskovi koji se kače na oglasnoj tabli škole gde se kandidati mogu blagovremeno iformisati o njihovom terminu polaganja dela ispita. Način polaganja ispita je definisan postojećim programom propisanim od strane MUP-a.

## Unos broja poena

Instruktor, nakon završetka roka unosi brojeve poena koje je kandidat ostvario na delu ispita na sistemu. Po default-u broj poena je 0, te ukoliko kandidat nije izašao na ispit broj poena ostaje nepromenjen, ako kandidat nije položio ispit instruktor je dužan da unese ostvaren broj poena kandidata. Takođe, mogući su slučajevi grešaka, te instruktor i administrator mogu da vrše ažuriranje ocene.

## Pregled korisnika

Na sistemu škole za obuku vozača omogućen je celokupan pregled korisnika, odnosno, kandidata , što znači da je moguće videti ispite koje je polagao, koje poene ima kao i lične podatke korisnika.



Slika - Slučaj korišćenja: Prijava ispita

**Slučaj korišćenja – Prijava ispita**

***Naziv SK:*** Prijava dela ispita

***Aktori SK:*** Instruktor

***Učesnici SK:*** Instruktor i sistem

***Preduslov:*** Sistem je pokrenut.

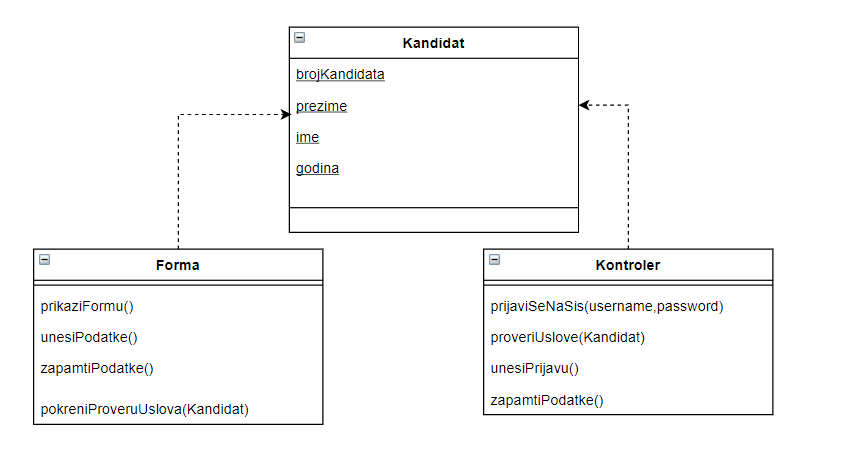
***Osnovni scenario:***

1. Instruktor se prijavljuje na sistem - slučaj korišćenja 1. Prijava na sistem
2. Aktor saopštava sistemu da želi da pokrene unos podataka o prijavi dela ispita kandidata. (APSO)
3. Sistem prikazuje formu za unos podataka o prijavi ispita. (IA)
4. Aktor saopštava sistemu da želi da pokrene proveru zadovoljenja uslova za prijavu ispita. (APSO)
5. Sistem obaveštava aktora da su uslovi zadovoljeni. (IA)
6. Aktor u službi unosi podatke u sistem o prijavi ispita
7. Aktor predaje podatke sistemu. (APSO)
8. Sistem obaveštava aktora da je prijava ispita uspešno izvršena (IA).

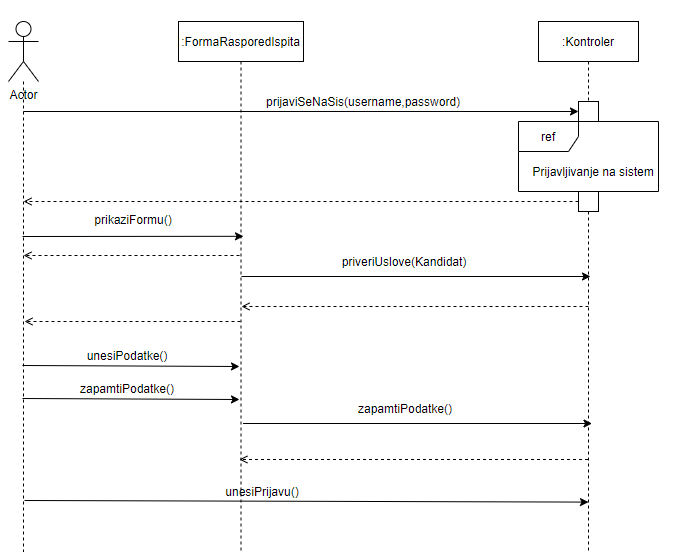
***Alternativni scenario:***

5.1 Sistem saopštava aktoru da uslov prijave ispita nije ispunjen. (IA)

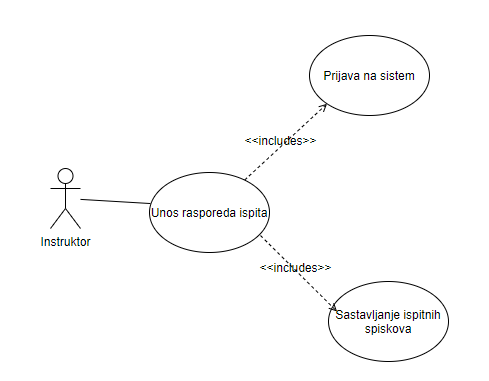
8.1 Sistem saopštava aktoru da prijava ispita nije uspešno izvršena. (IA)



Slika - Dijagram klasa: Prijava dela ispita



Slika - Dijagram sekvence: Prijava ispita



Slika - Slučaj korišćenja: Sastavljanje ispitnih spiskova

**Slučaj korišćenja – Unos rasporeda ispita**

***Naziv SK:*** Sastavljanje ispitinih rokova

***Aktori SK:*** Instruktor

***Učesnici SK:*** Instruktor i sistem

***Preduslov:*** Sistem je pokrenut.

***Osnovni scenario:***

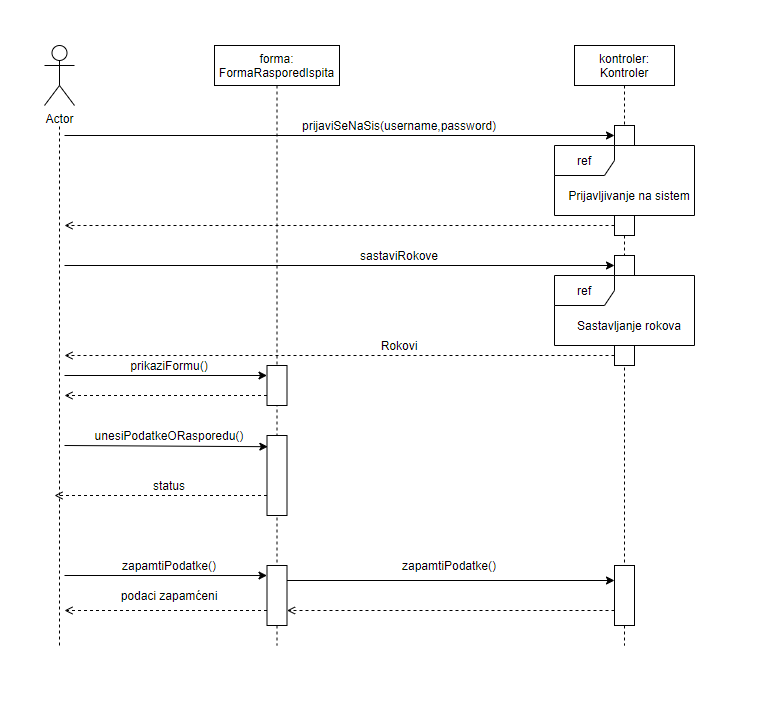
1. Aktor se prijavljuje na sistem — slučaj korišćenja 1. Prijava na sistem
2. Aktor sastavlja ispitne spiskove — slučaj korišćenja 6. Sastavljanje ispitnih spiskova
3. Aktor saopštava sistem da želi da pokrene unos rasporeda ispita. (APSO)
4. Sistem prikazuje formu za unos rasporeda ispita (IA)
5. Aktor unosi podatke o rasporedu ispita (APUSO)
6. Aktor saopštava sistemu da želi zapamti podatke o rasporedu ispita. (APSO)
7. Sistem pamti podatke o rasporedu ispita (SO)
8. Sistem saopštava aktora da je uspešno sačuvao podatke o rasporedu ispita.(IA)

***Alternativni scenario:***

8.1 Sistem saopštava aktoru da snimanje podataka o rasporedu ispita nije uspelo. (IA)



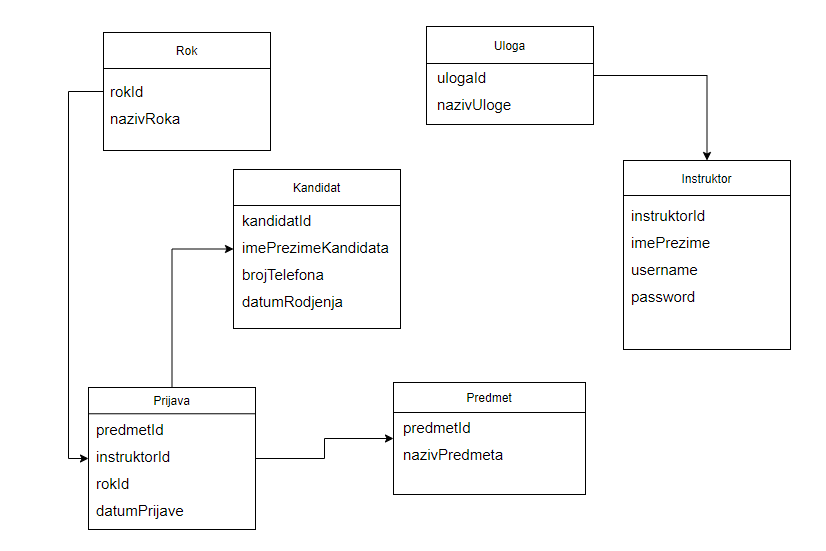
Slika - Dijagram klasa: Raspored ispita



Slika - Dijagram sekvence: Raspored ispita

# Arhitektura

## Dijagram klasa



Slika – Dijagram klasa

U radu je korišćena baza podataka : Služba.

Baza podataka se sastoji od šest tabela:

1. Predmet – sadrzi atribute: predmetId, nazivPredmeta
2. Prijava – sadrzi atribute: predmetID, brojKandidata, brojPoena, datumPrijave

i spoljne ključeve: rokID, instruktorID

1. Rok – sadrzi atribute: rokID, nazivRoka
2. Kandidat – sadrzi atribute: brojKandidata, imePrezimeKandidata, brojTelefona i datumRodjenja
3. Instruktor – sadrzi atribute: instruktorID, imePrezime, username, password

i spoljni ključ: ulogaId

1. Uloga – sadrzi atribute: ulogaID i nazivUloge

# Specifikacija REST API-ja

|  |  |
| --- | --- |
| Opis funkcije | Unesi predmet |
| HTTP metoda | POST |
| URL | /unesiPredmet |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | Niz JSON objekata. Svaki element niza ima atribute predmetID[int],nazivPredmeta[varchar]  Primer:  [  {  "predmetID":"1",  "nazivPredmeta":"Tipovanje",  },  {  "predmetID":"2",  "nazivPredmeta":" Prva pomoc",  },  {  "predmetID":"3",  "nazivPredmeta":"Voznja",  }    ] |
| Format izlaznih parametara | application/json |

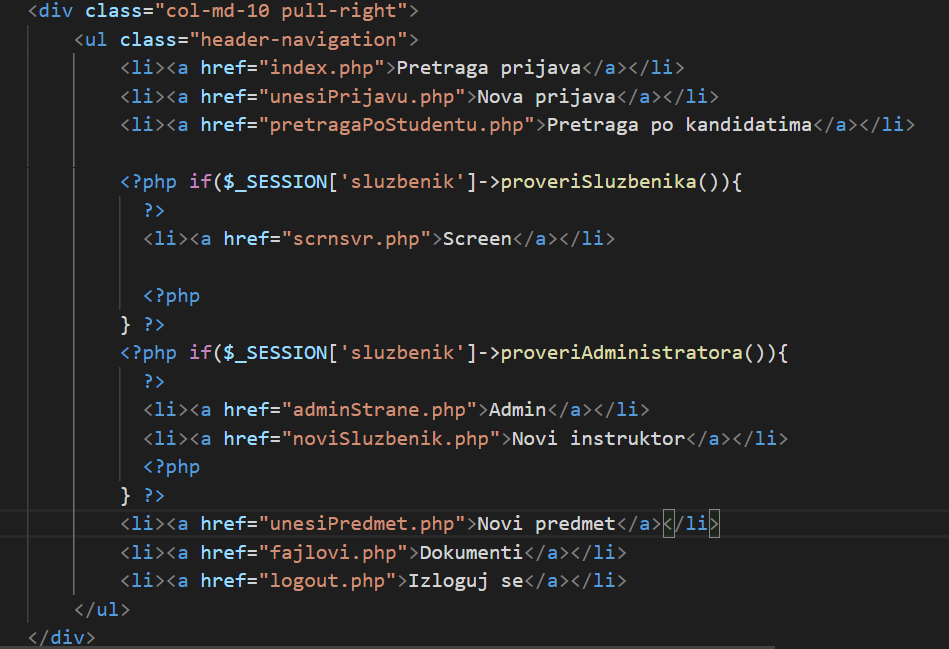
# Opis korišćenih tehnologija

U našem radu koristili smo sledeće tehnologije:

1. HTML
2. CSS
3. PHP
4. AJAX
5. JSON
6. JavaScript
7. Flight

## HTML

HTML (HyperText Markup Language) je opisni jezik specijalno namenjen opisu web stranica. U retkim slučajevima se koristi za dizajn web stranice, ali je to loša praksa pa se takav pristum izbegava. Definiše naslove, paragrafe, odvojene blokovie I drugo. Pored toga, u HTML standard su ugrađeni elementi koji detaljnije opisuju sam dokument kao što su kratak opis dokumenta, ključne reči, podaci o autoru i slično. Primer korišćenja HTML-a u našoj aplikaciji možemo videti u sledećem primeru.

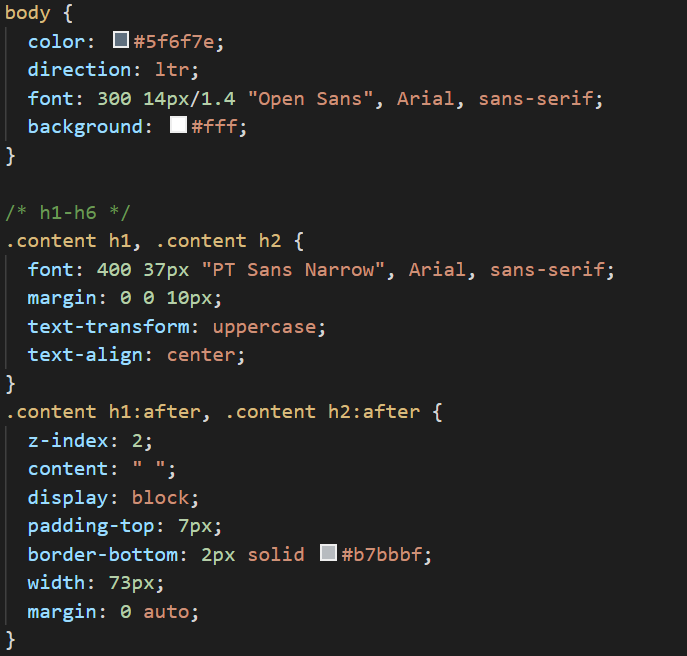


Slika - HTML kod

## CSS

CSS je takozvani dizajnerski jezik. Stranice napisane samo HTML sintaksom nisu vizuelno atraktivne. CSS fajlovi su, po strukturi, takodje obični tekstualni fajlovi koji slede jednostavnu sintaksu pisanja CSS koda. Iako se CSS kod može nalaziti unutar HTML dokumenta, ne preporučuje se kako to kod čini nabacnim I težim za čitanje I održavanje. Preporučeno je da se CSS folder piše kao samostalan dokumentsa ekstenzijom .css .

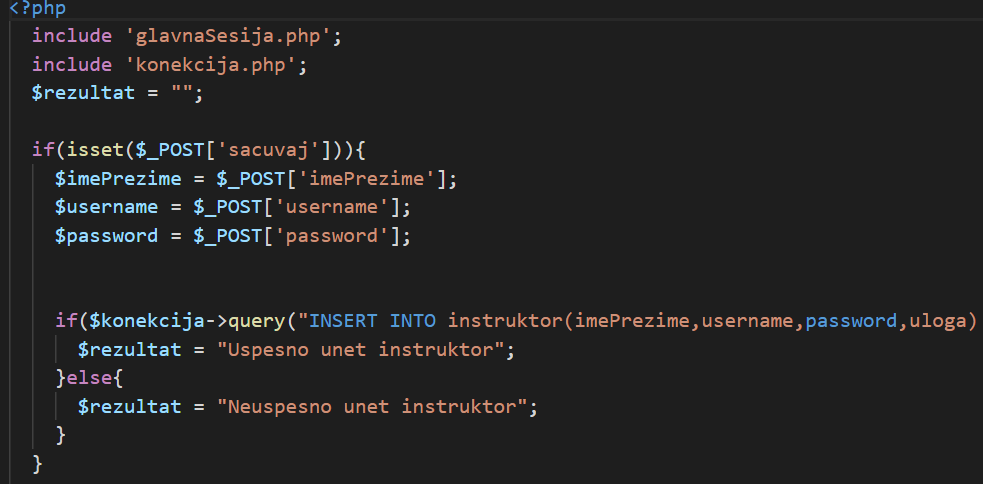
Svaki opis elementa se sastoji od definicije ciljnih elemenata, svojstva i vrednosti. Primer jednostavnog CSS koda je prikazan u primeru.



Slika - CSS kod

## PHP

PHP (Hypertext Preprocessor) je specijalizovani skriptni programski jezik koji se koristi za izradu dinamičkih web stranica. PHP je poznat po svojim tagovima ‘<?php ?>’ izmedju kojih se nalazi kod. Varijabla se pored imena označava prefiksom $. Kao razmak može se koristiti znak “\_”. Još jedna vrlo bitna stvar kod promenljivih u PHP-u je da su imena case-sensitive. PHP fajl ima ekstenziju .php. Primer PHP koda koji formira while petlju.



Slika - PHP kod

## AJAX(Asynchronous JavaScript And XML)

AJAX nije novi jezik programiranja, već tehnika za kreiranje boljih, bržih i interaktivnijih web aplikacija. Sa AJAX-om, vaš JavaScript može da komunicira direktno sa serverom, koristeći JavaScript XMLHttpRequest objekat. Pomoću ovog objekta, vaš JavaScript može da razmenjuje podatke sa web serverom, bez ponovnog učitavanja stranice.

AJAX koristi asinhroni transfer podataka (HTTP zahtevi) između pretraživača i web servera, mogućujući web stranicama da traže male bitove informacija od servera umesto čitavih stranica.AJAX tehnika programiranja omogućuje da Internet aplikacije budu manje, brže i prijateljskije za korisnika.To je tehnologija pretraživača koja je nezavisna od softvera web servera.

AJAX je zasnovan na sledećim web standardima:

* JavaScript
* XML
* HTML
* CSS



Slika - AJAX kod

## JSON

JSON, odnosno JavaScript Object Notation, je tekstualno baziran otvoreni standard dizajniran za razumljivu ljudima razmenu podataka. Ona je izvedena iz Javaskript jezika za predstavljanje jednostavnih struktura podataka i asocijativni niz, nazvan objektima. Uprkos vezi sa JavaSkriptom, to je jezicki nezavistan, za mnoge jezike.

Struktura može biti organizovana u vidu:

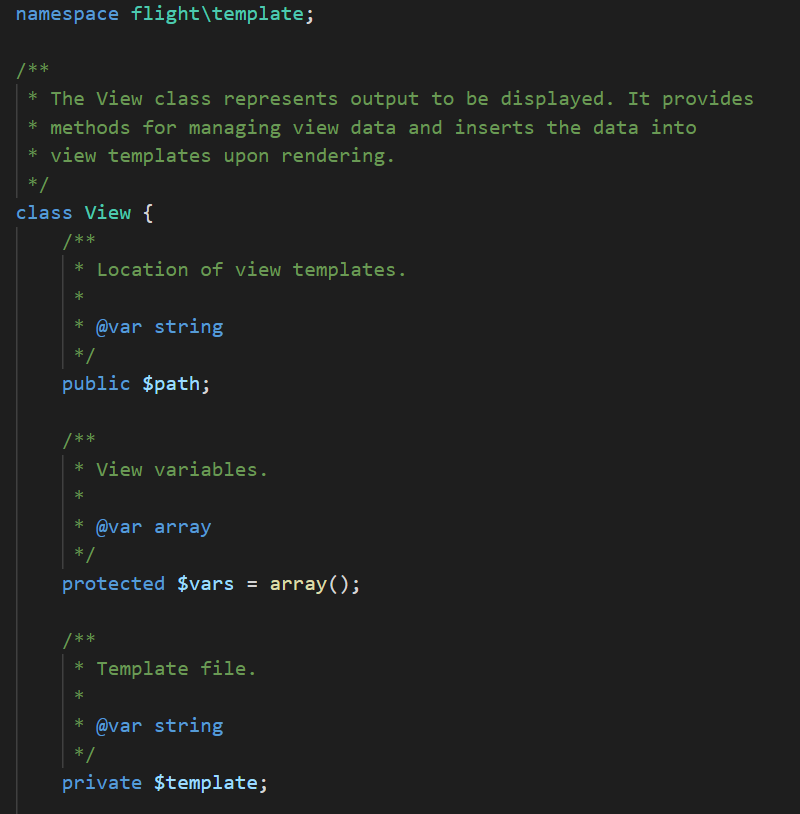
* Zbirke parova (ime / vrednost) - Na raznim jezicima, to je realizovano kao objekat, zapis, struktura, rečnik, heš tabela, lista sa ključevima ili asocijativni niz. Ova struktura se obeleažva sa { } vitičastim zagradama, a podaci u obliku ime/vrednost su odvojeni sa zarezom.
* Niza (uređena lista vrednosti) -Niz se obeležava sa [ ] uglastim zagradama, a članovi niza u odvojeni zarezom.

## JavaScript

JavaScript je objektno zasnovan skriptni jezik. Uključujemo ga u web stranicu da bi je učinili dinamičnijom. HTML (osnovni kod web stranica) se koristi samo za oblikovanje i uređivanje elemenata stranice (tekst, forme, linkove i tabele), ali nema šanse da diktiramo kako će se ti elementi ponašati. Mogućnost uključenja JavaScript skripte daje nam mnogo veću kontrolu kako se web stranica ponaša. Kombinovan sa HTML-om i CSS-om JavaScript čini DHTML (Dinamic HTML).

## Flight

Flight je specijalizovani PHP framework namenjen prvenstveno razvoju REST servisa. Osnovna funkcionalnost ovog framework-a je rutiranje, odnosno jednostavno kreiranje funkcija koje odgovaraju na HTTP zahteve upućene korišćenjem neke standardne HTTP metode (GET, POST, PUT, DELETE). Flight je objektno-orijentisan framework i može se koristiti bilo pozivanjem statičkih metoda koje su definisane u klasama Flight-a, bilo instanciranjem novog objekta. U ovoj skripti će biti korišćene statičke metode. Moguće je definisanje sopstvenih globalnih promenljivih u Flight-u, segmentno rutiranje, korišćenje Flight-a za automatizaciju dodavanja HTML templejta i dr.



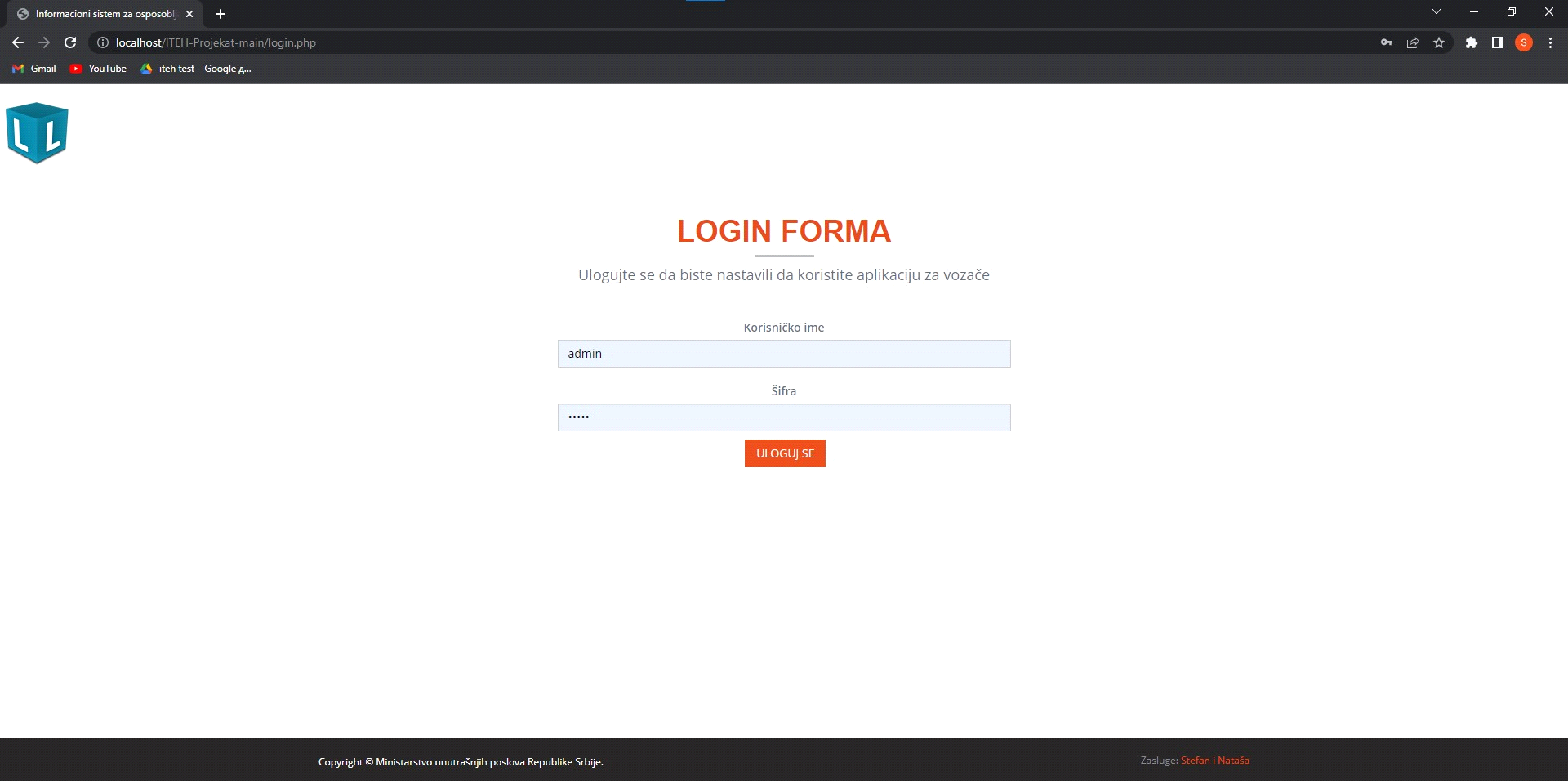
Slika - Flight genericki fajl

# Korisničko uputstvo

Aplikacija služi za rad u autoškoli kao pomoć instruktorima I vlasnicima autoškole, kao I administraciji.

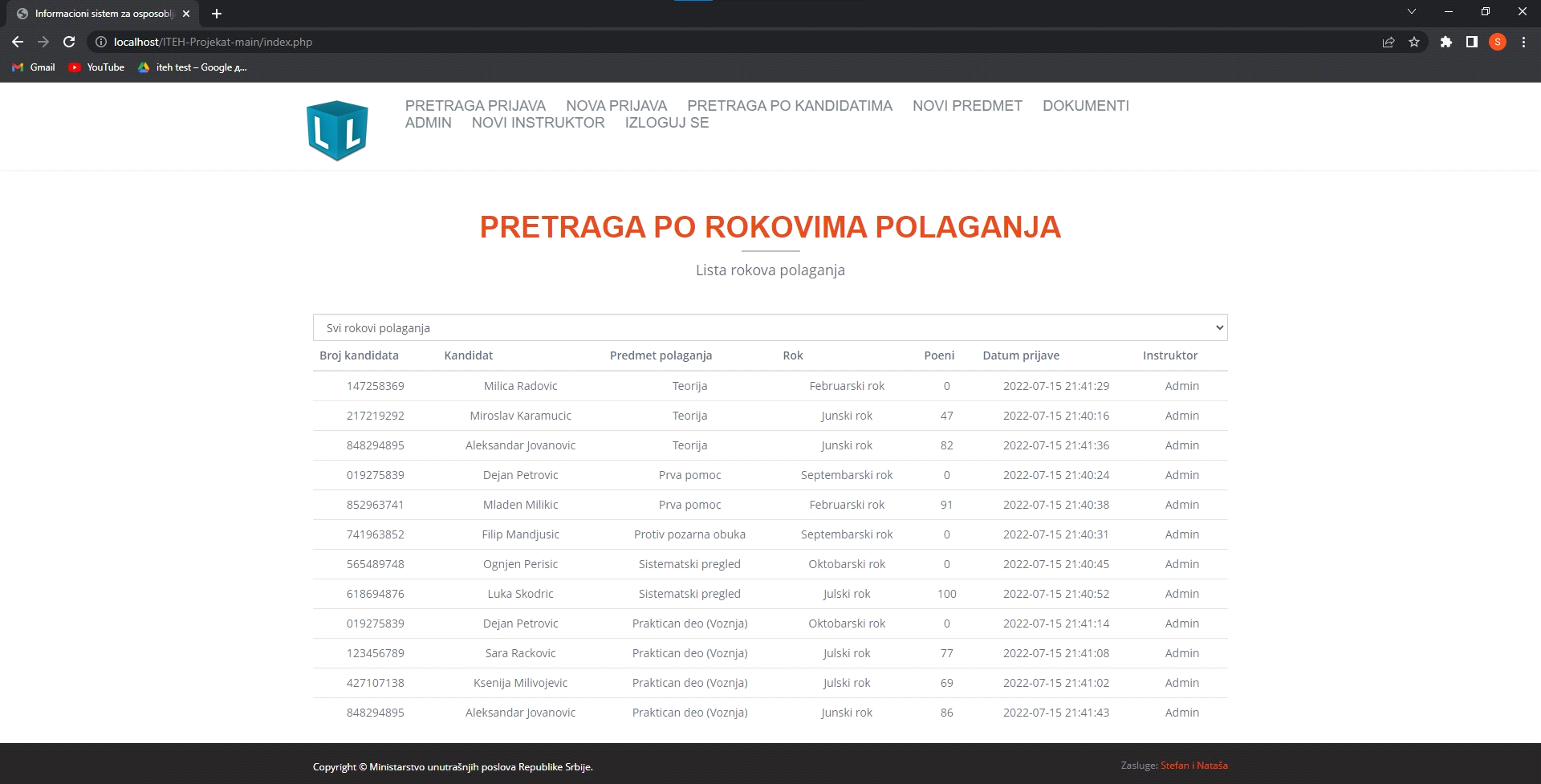
## Instruktor

Da bi se prikačio na aplikaciju, instruktor mora prvo da se loguje putem forme.



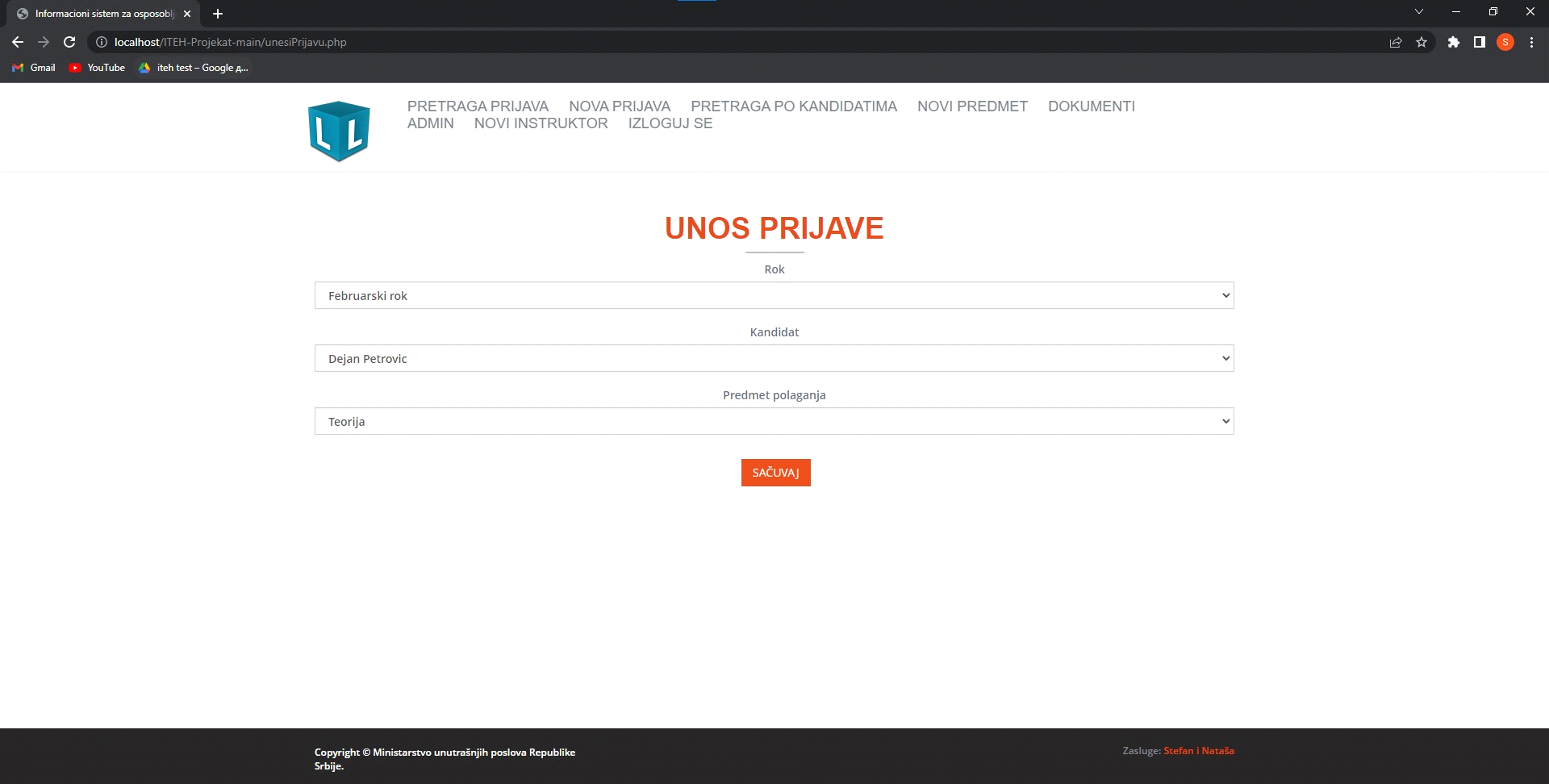
Slika - Forma za logovanje

Nakon što se uloguje na početnoj strani po default-u se pojavljuje pretraga po rokovima, sa prikazom svih rokova. Nudi se opcija pretrage po rokovima iz liste.

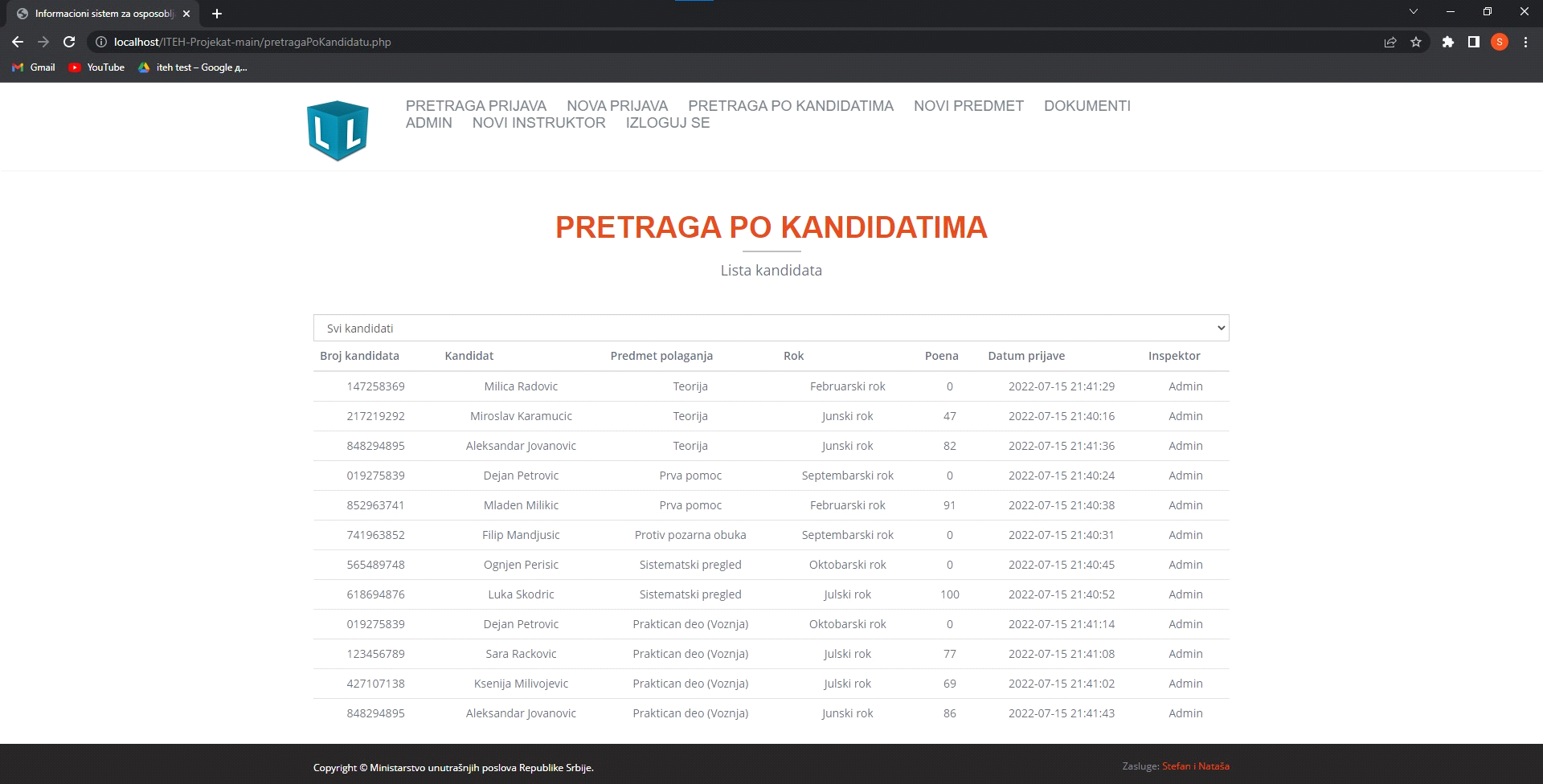


Slika - Prikaz pretrage po rokovima

Zatim je omogućena i pretraga po kandidatima koja nam daje sve relevantne informacije o zadatom kandidatu. Sledeća stavka jeste nova prijava kojom se omogućuje prijava ispita u određenom roku za datog studenta.

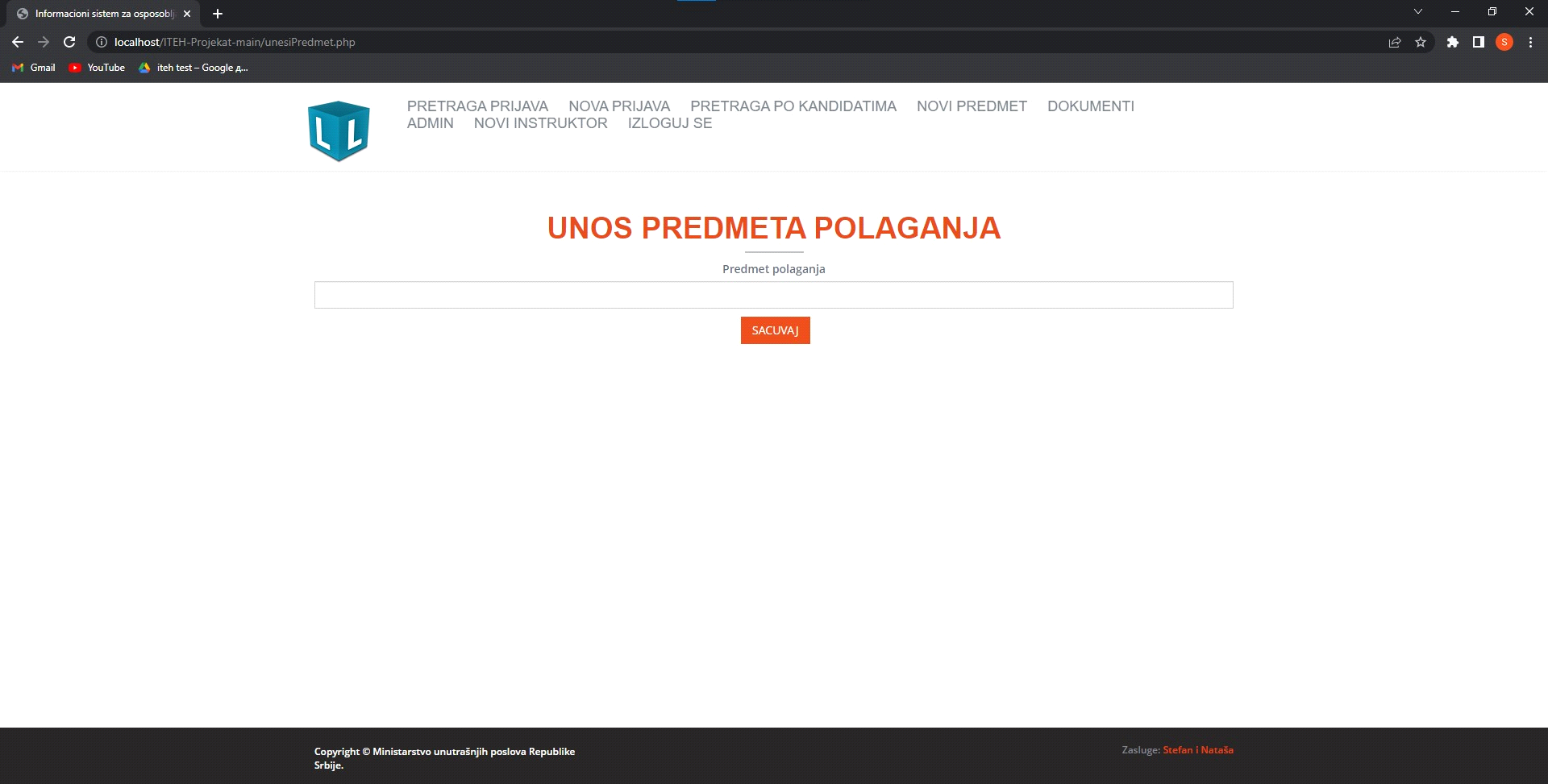


Slika - Unos prijave kandidata



Slika - Pretraga kandidata

Kao opcija se javlja i unos novog predmeta. Nakon unosa predmeta prilikom sledećeg unosa nove prijave dodaje se i novi predmet kao jedna od opcija izbora. Kao predmeti se mogu unetiTeorija, Tipovanje, Prva pomoc itd..

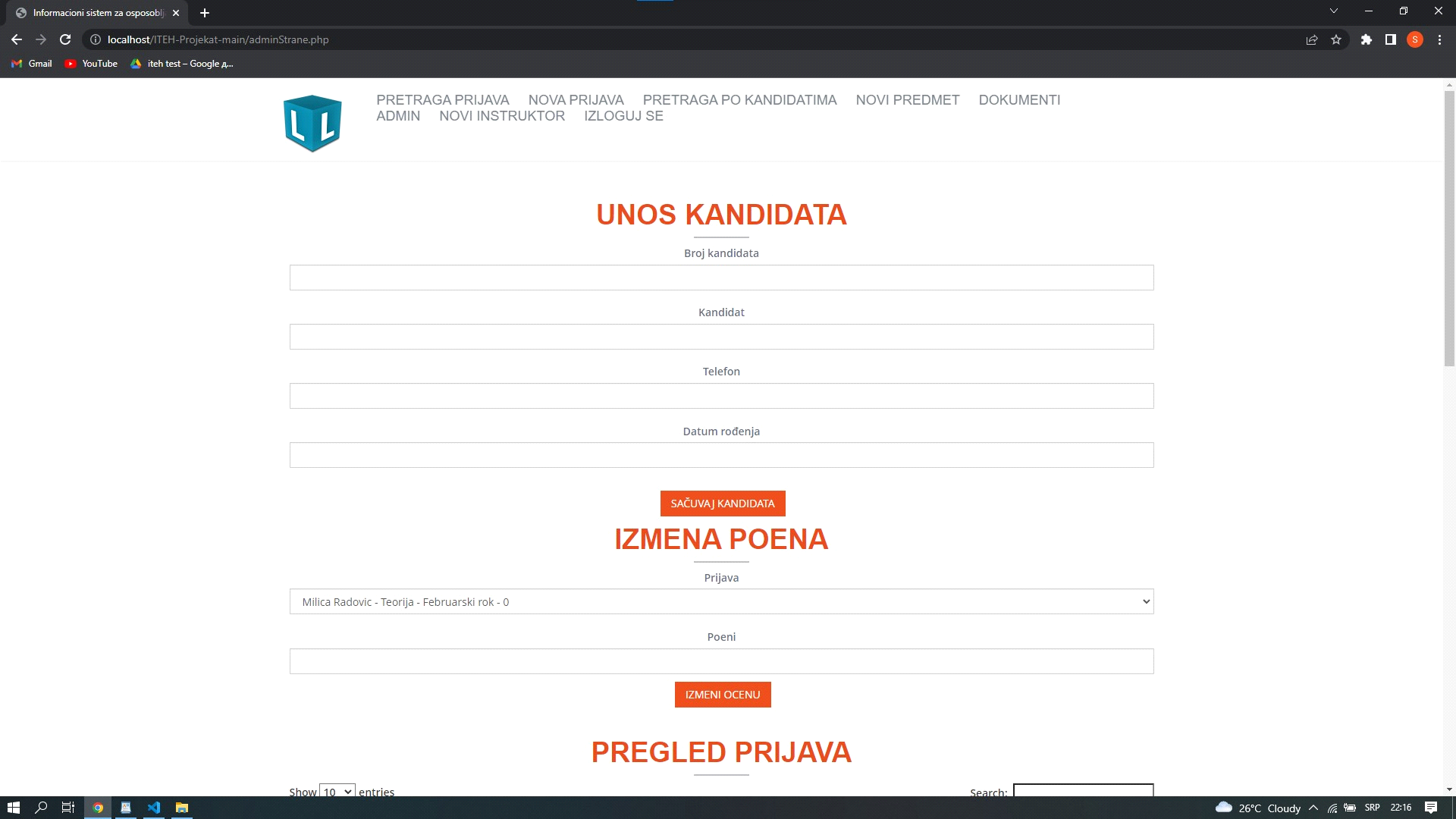


Slika - Unos novog predmeta

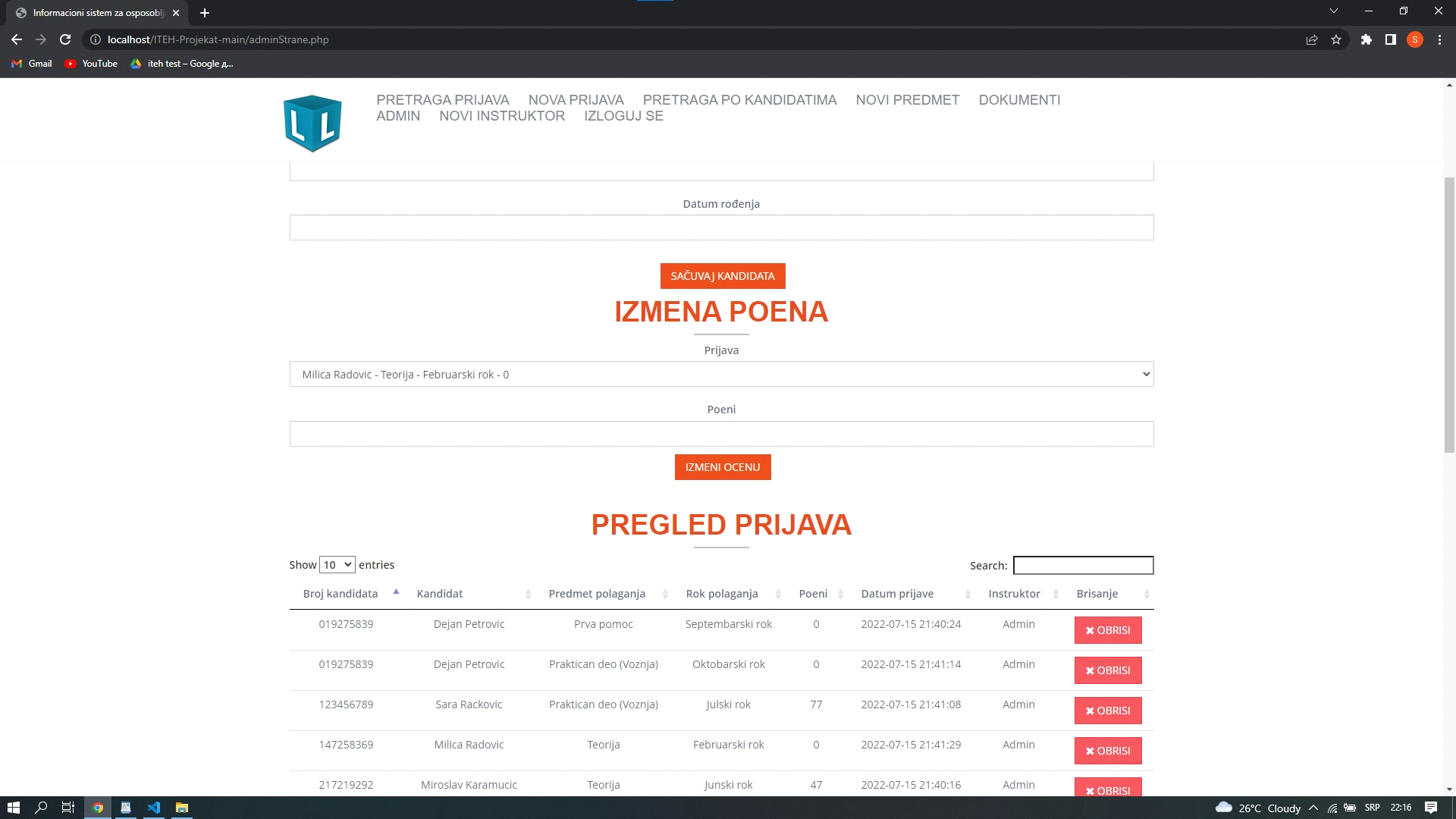
## Administrator

Ukoliko je instruktor ulogovan kao administrator,njemu su omogućene dodatne opcije.

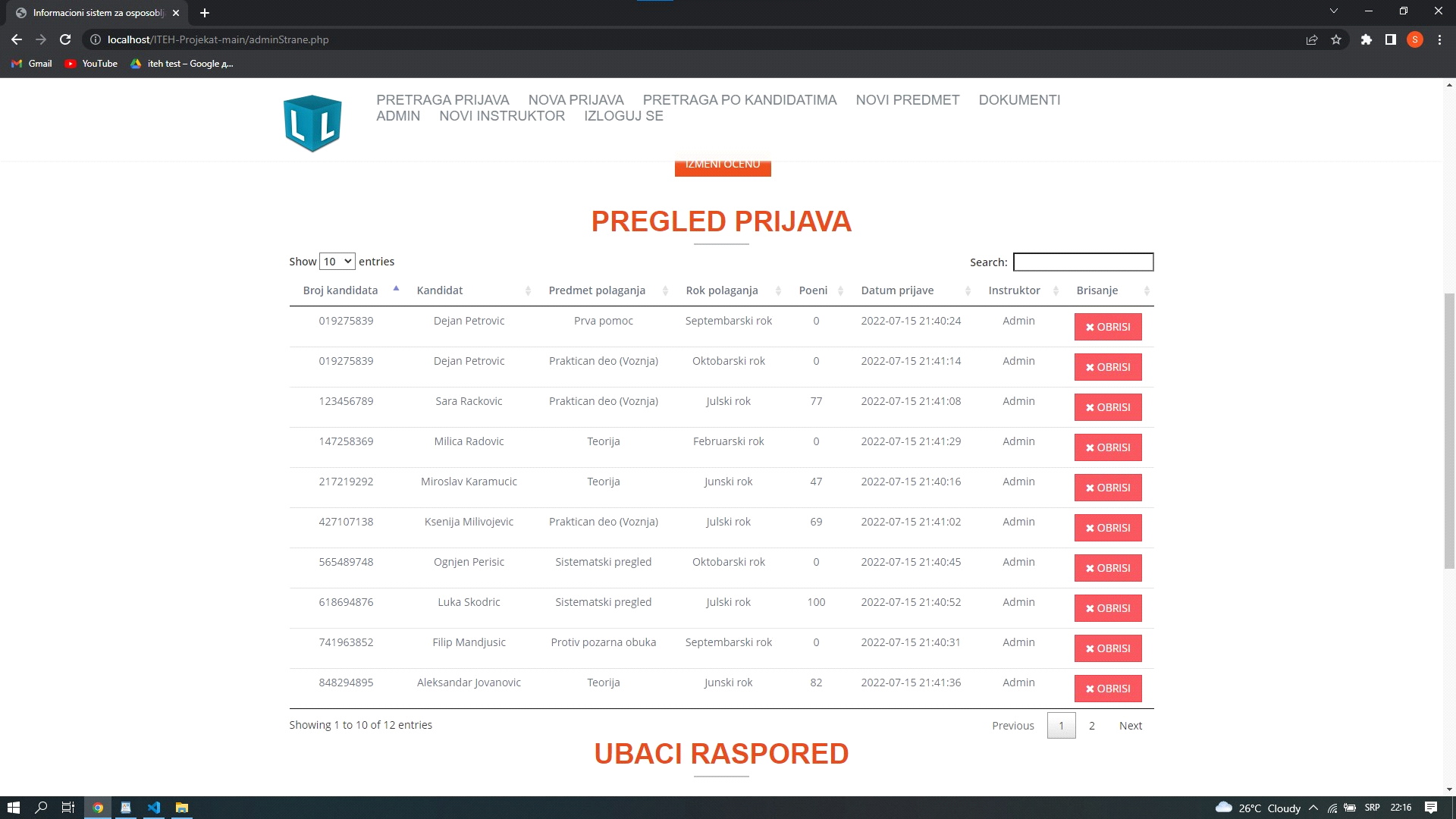
Poput unosa podataka o novom kandidatu u sistem. Zatim, mogućnost izmene ocena,kao i pregled prijava gde je omogućena pretraga,kao i brisanje istih.



Slika - Unos kandidata

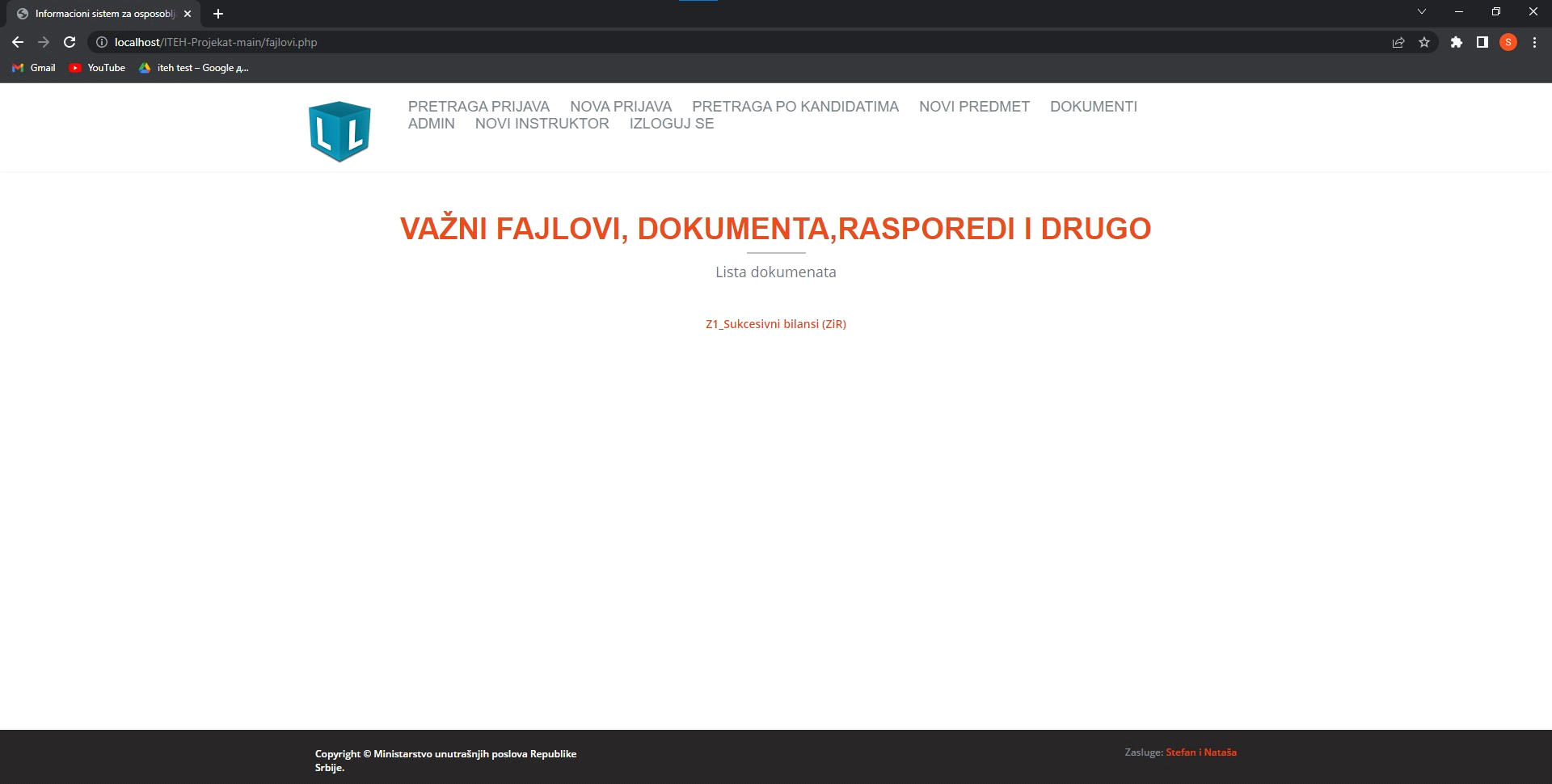


Slika - Izmena poena kandidata

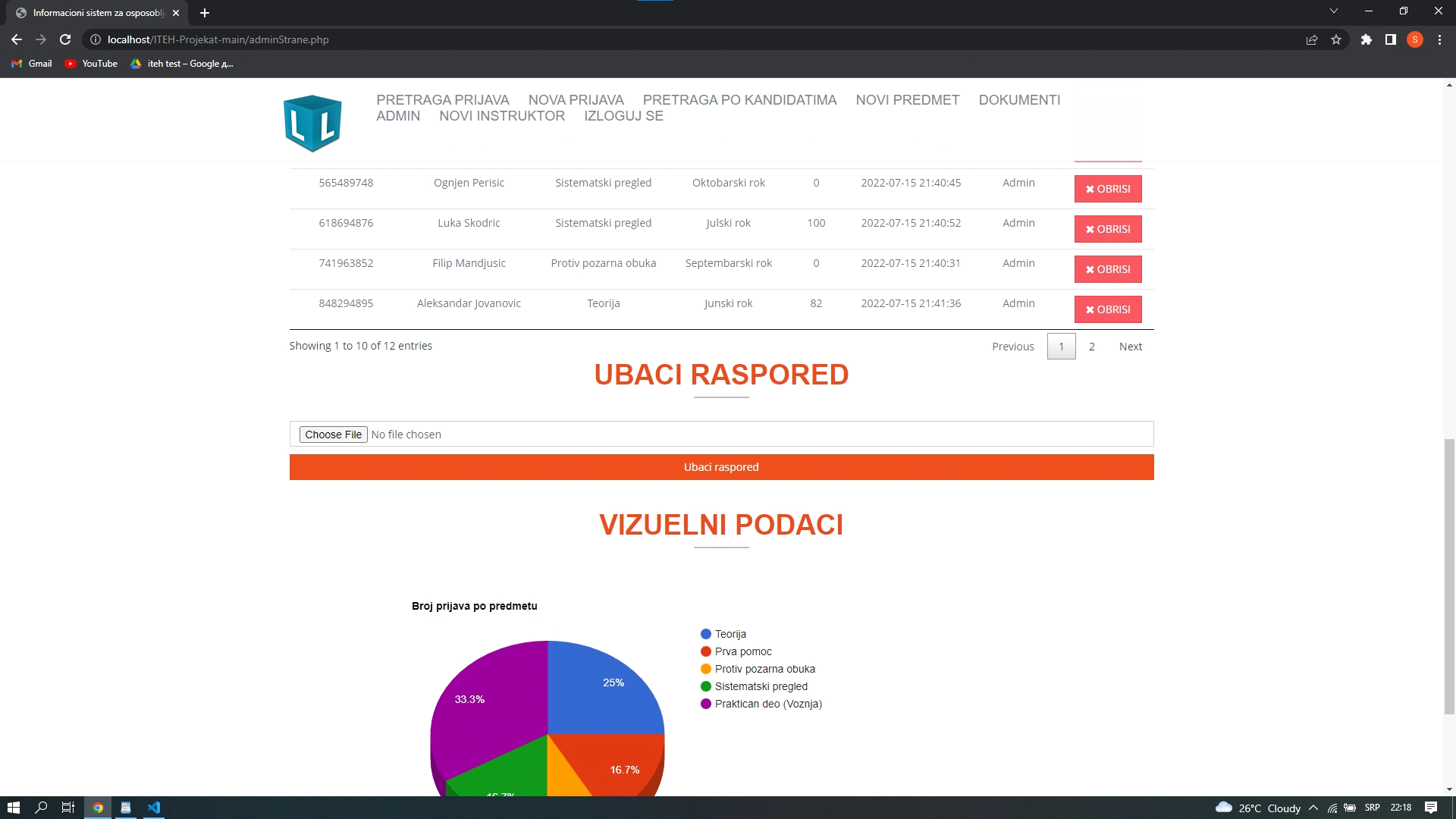


Slika - Pregled prijava

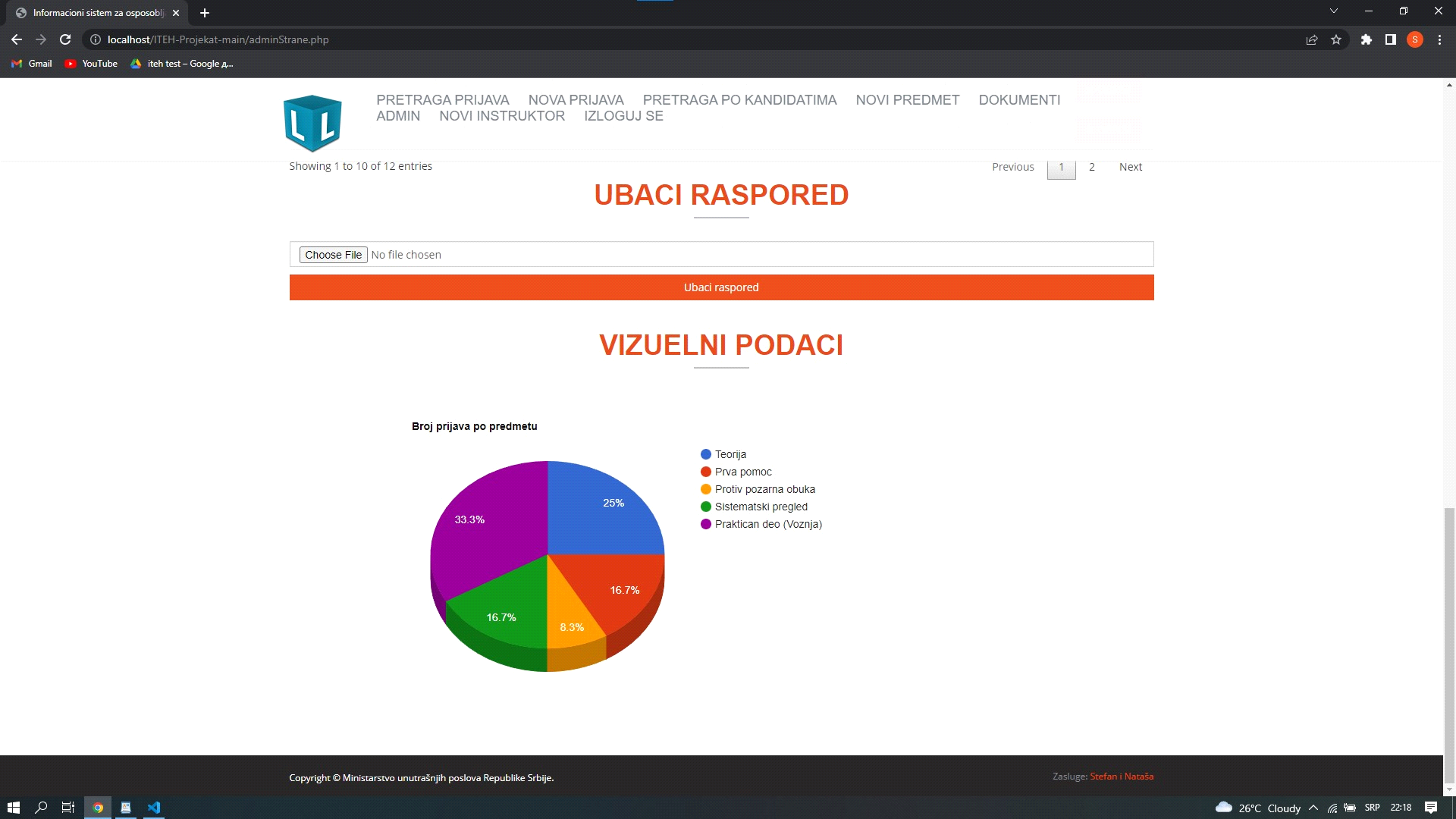
Administrator takođe može i da upload-uje raspored ispita,kojem se posle pristupa u okviru fajlova. Vizuelni podaci pružaju grafički prikaz broja prijava po predmetu na pie dijagramu. Pored toga, administrator može da kreira novog instruktora koji će imati pristup aplikaciji.



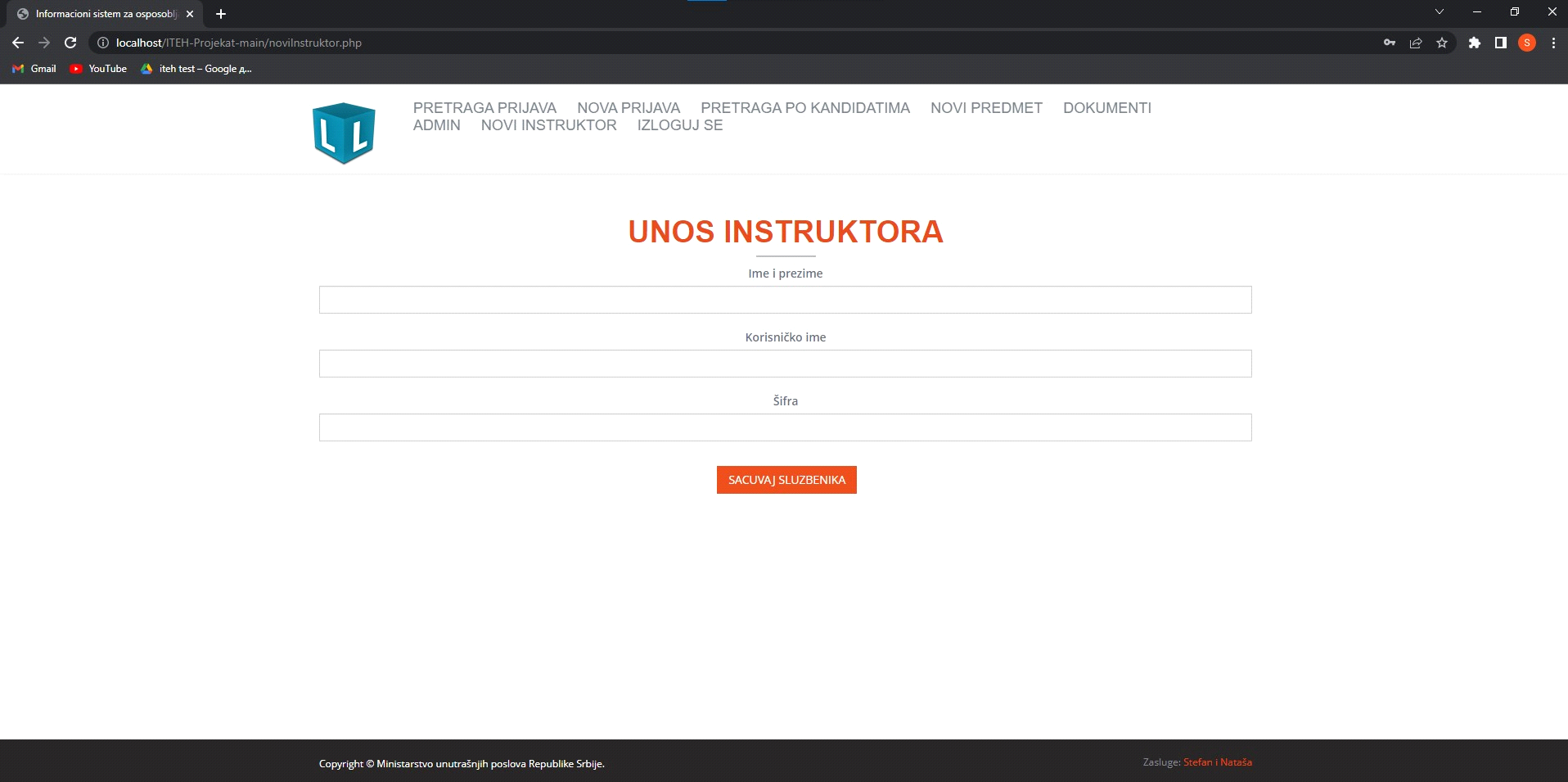
Slika - Dodavanje dokumenata, rasporeda itd..



Slika - Upload rasporeda

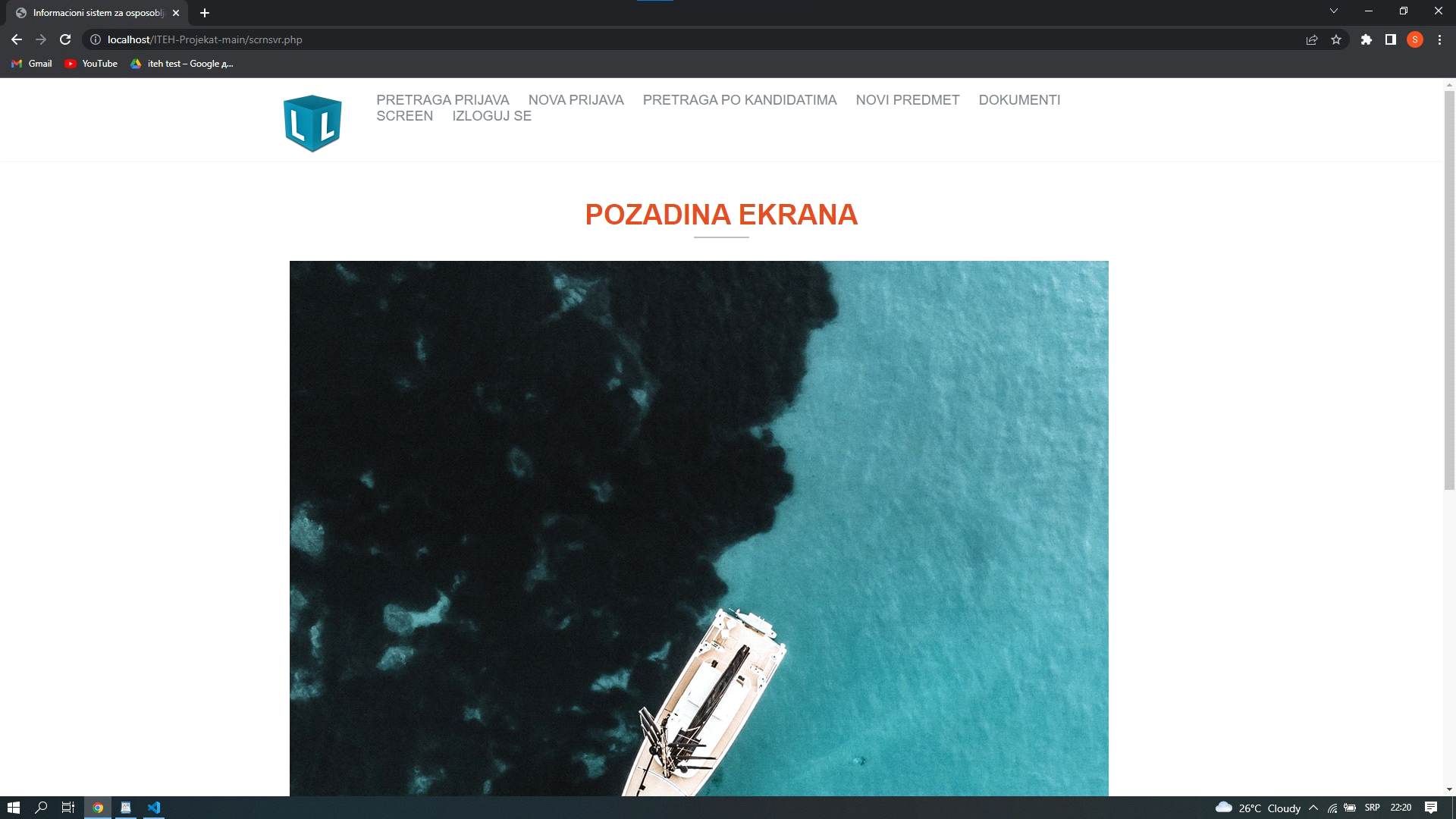


Slika - Pie chart predmeta



Slika - Kreiranje novog instruktora

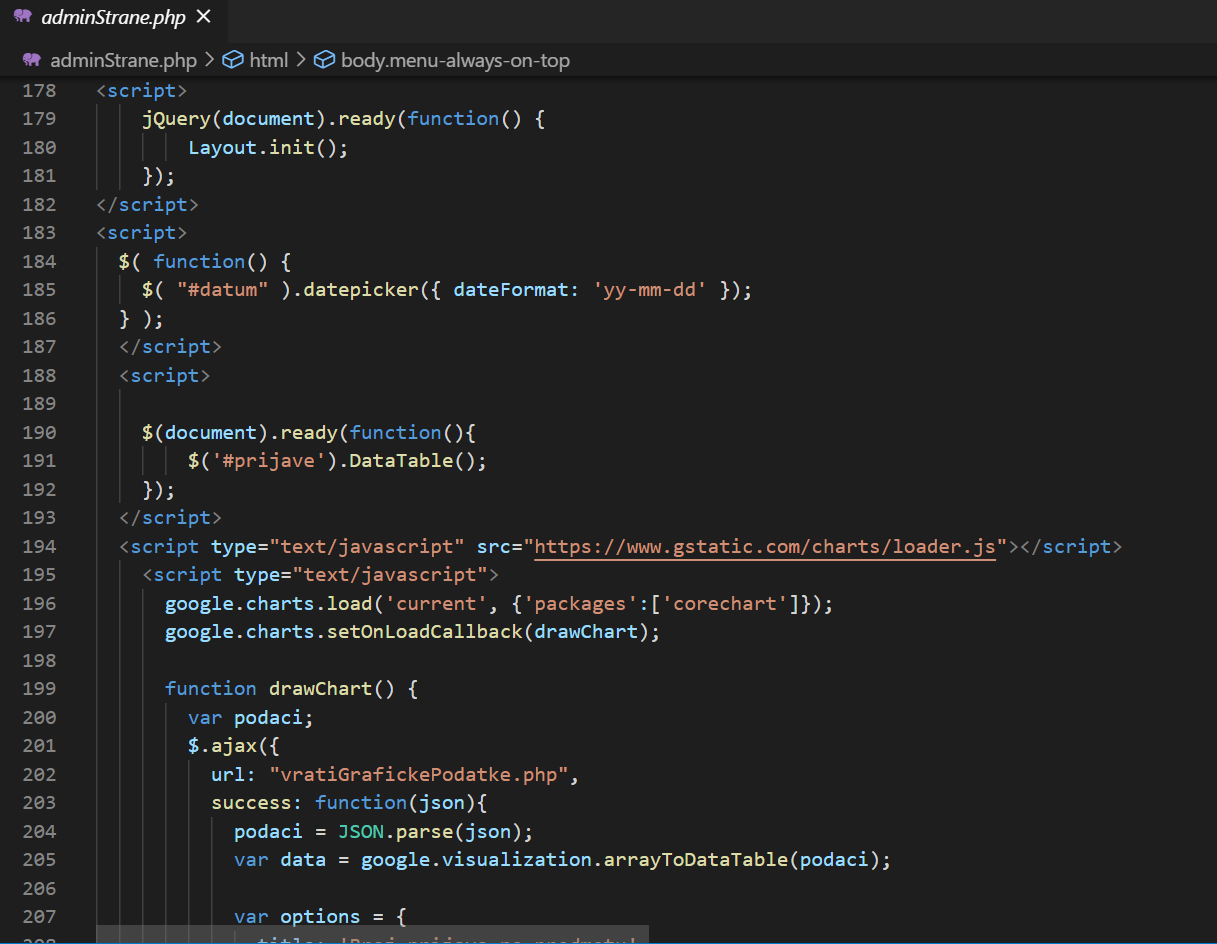
Kako bi seminarski zadovoljio sve zadate zahteve, aplikacija ima neke dodatne pogodnosti poput pregleda pozadine ekrana.



Slika - Pozadina ekrana

# Arhitektura koda aplikacije

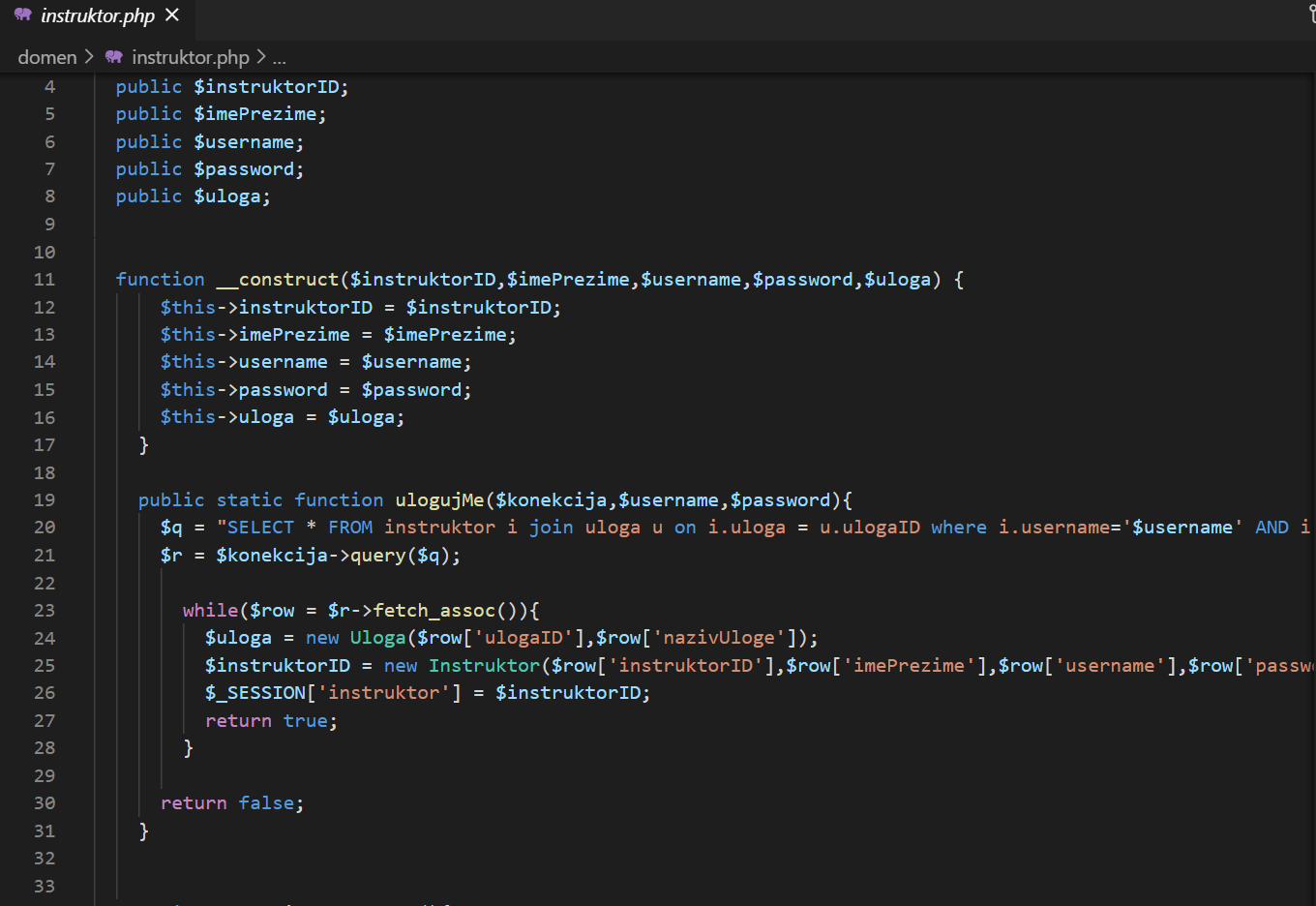
Aplikacija se sastoji iz više desetina php fajlova , sql baze I dva foldera. Prikazaćemo važnije fajlove.



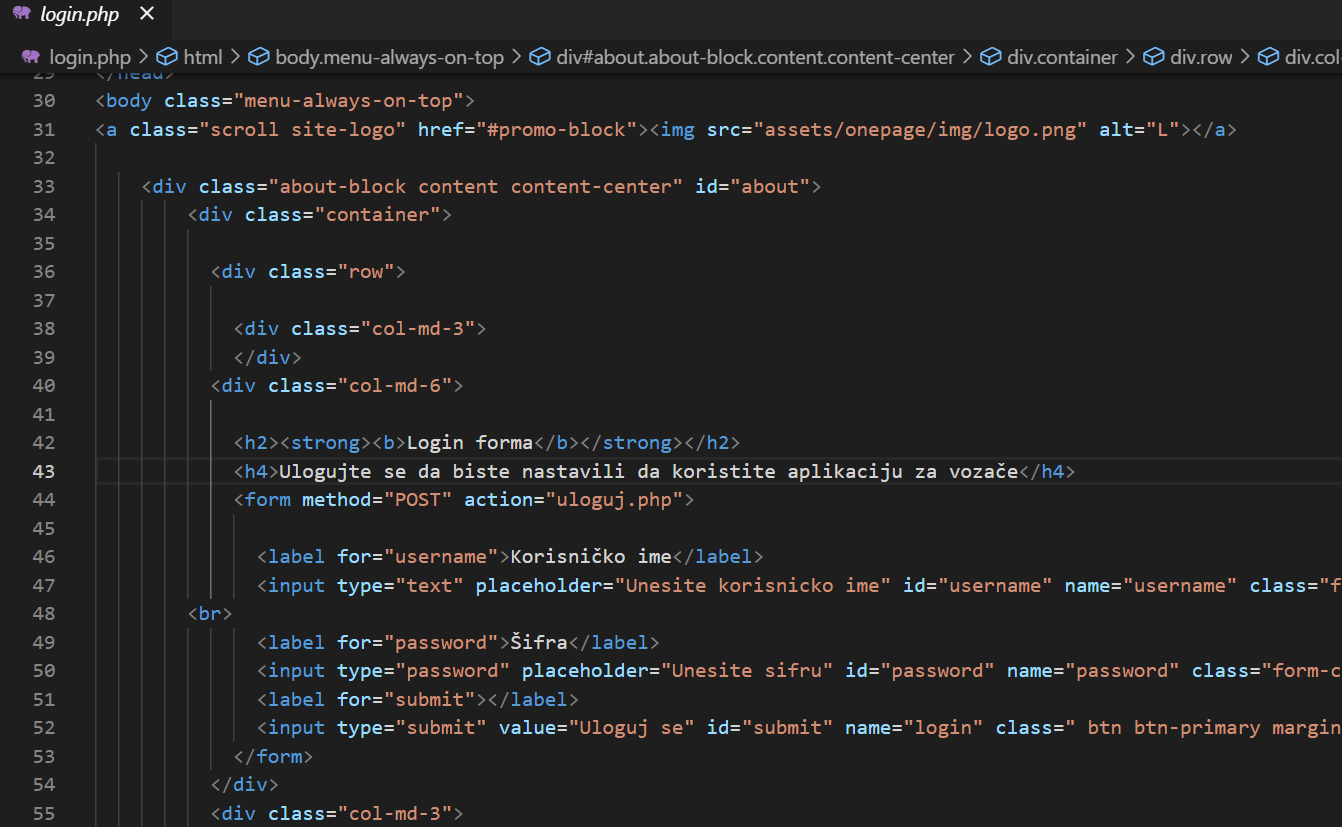
Slika - Admin strane



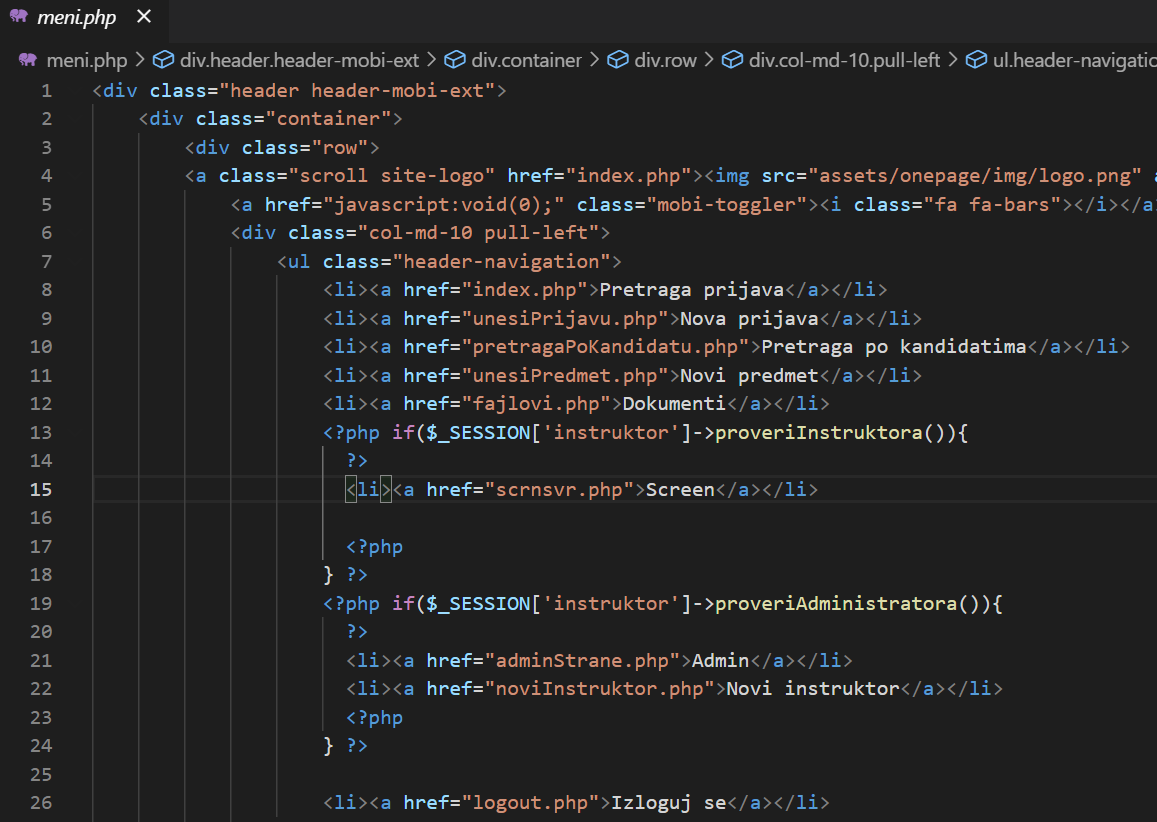
Slika - Pretraga po rokovima AJAX



Slika - Instruktor php stranica



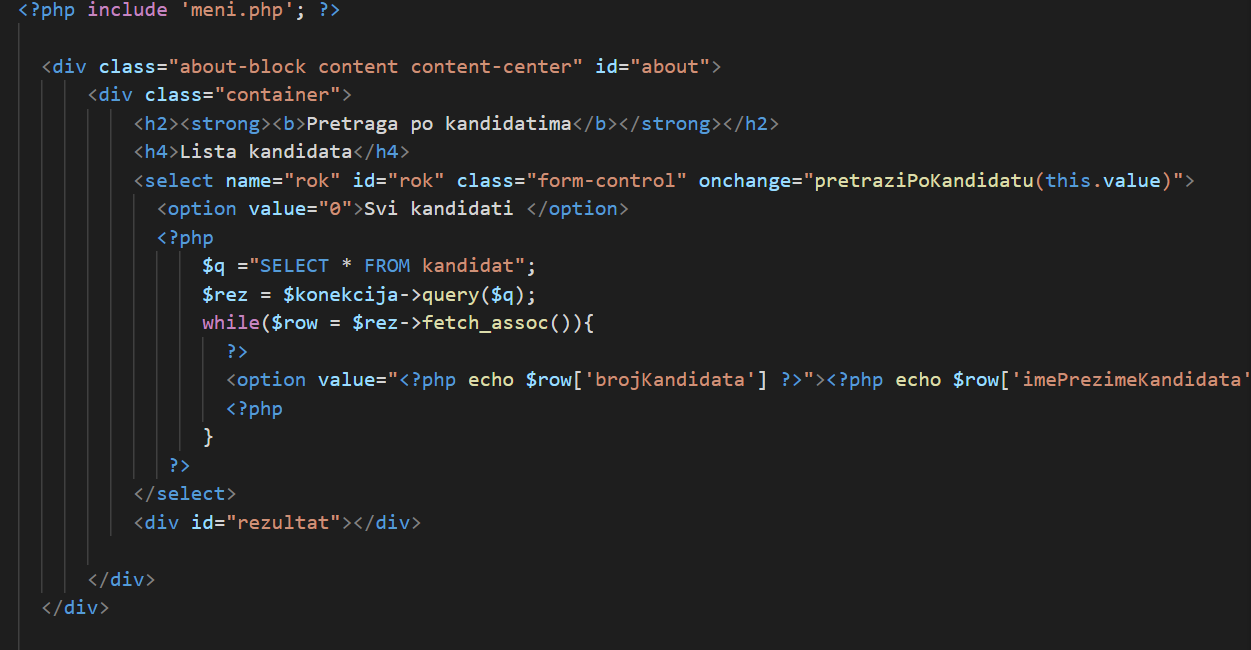
Slika - Login forma



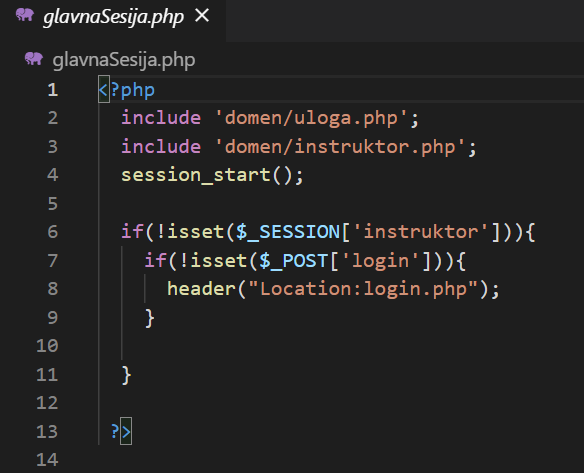
Slika - Meni bar



Slika - Unos novog instruktora



Slika - Pretraga po kandidatima



Slika - Rad sa sesijama