Univerzitet u Beogradu

Fakultet organizacionih nauka

Laboratorija za elektronsko poslovanje



Web aplikacija studentske službe

Projektni rad iz Internet tehnologija

Studenti:

Veljko Blagojević 0353/2019

Nikola Vujčić 0062/2019

Milan Stanković 1004/2020

U Beogradu, 2023.

Korisnički zahtev

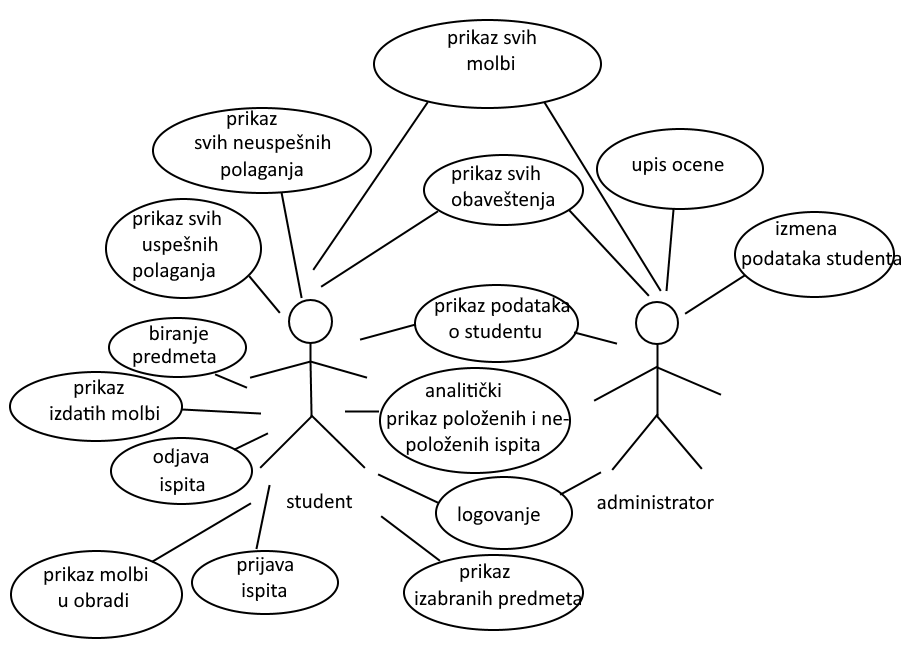
Potrebno je napraviti web aplikaciju studentske službe. Rad sa službom je moguće obaviti u ulozi studenta ili u ulozi administratora. Prilikom pokretanja web sajta, korisniku treba da se prikazuje web stranica i meni sa sledećim opcijama: Obaveštenja, Molba, Podaci o studentu, O fakultetu i Kraj Rada. Ukoliko je prijavljeni korisnik u funkciji student, njemu se prikazuju i dodatne opcije: Prijava ispita, Biranje predmeta, Uspešna polaganja, Neuspešna polaganja, Moji Predmeti i Analitika. Prijavljenom administratoru se prikazuju dodatne opcije: Upiši ocenu, Izmeni podatke o studentu i ADMIN sekcija.

U sekciji obaveštenja su izlistana sva obaveštenja datog korisnika. Prijava ispita se obavlja klikom na odgovarajuće dugme, a odjava ispita se obavlja klikom na dugme x pored prijavljenog ispita, u istoimenoj tabeli. Izbor predmeta za slušanje treba biti implementiran na sledeći način: klikom na dugme “Potvrdi” se izabrani predmeti čuvaju. Potrebno je takođe ispisati i podatke o ukupnom broju ESP bodova, kao i prosečnoj oceni studenta u sekciji “Uspešna polaganja”. Neuspešna polaganja studenta je neophodno evidentirati i prikazati tabelarno korisniku. U odeljku “Molbe” je potrebno omogućiti unos nove molbe studenta i prikazati Izdate molbe i molbe u obradi. Klikom na dugme “Potvrdi” se molba čuva u bazi podataka. Podaci o studentu koje treba prikazati su: broj indeksa, ime, prezime, username, ime roditelja, JMBG, mesto rođenja, lični email, studentski email, broj telefona, kao i slika studenta. Analitika treba da sadrži 2 dijagrama koji prikazuju odnos položenih ispita i neuspešnih polaganja, kao i raspodelu ocena po ispitima. O fakultetu treba da sadrži podatke o fakultetu kao i google map prikaz lokacije na kojoj se fakultet nalazi. Neophodno je omogućiti i upis ocene studenta administratoru. Administrator može i da izmeni podatke o studentu kojeg on izabere iz liste dostupnih studenata, i klikom na dugme “Potvrdi” se ažurirani podaci o studentu čuvaju u bazi podataka. Na kraju, neophodno je omogućiti administratoru pristup servisnim informacijama o mreži putem ADMIN sekcije.

Model slučajeva korišćenja

U ovom sistemu prepoznajemo 15 slučajeva korišćenja:

1. Logovanje
2. Prikaz svih obaveštenja
3. Unos molbe
4. Prikaz molbi u obradi
5. Prikaz Izdatih molbi
6. Upis ocene
7. Prikaz podataka studenta
8. Izmena podataka studenta
9. Prijava ispita
10. Odjava ispita
11. Biranje predmeta
12. Prikaz svih uspešnih polaganja ispita
13. Prikaz svih neuspešnih polaganja ispita
14. Prikaz izabranih predmeta
15. Analitički prikaz položenih i nepoloženih ispita



Slika 1: Model slučajeva korišćenja

* **Prikaz svih obaveštenja** – korisniku su izlistana sva dostupna obaveštenja zasebno, na jednoj strani u celini
* **Unos molbe** – student bira iz combo-box-a tip molbe, unosi tekstualno obrazloženje za datu molbu, i klikom na dugme “Potvrdi” podnosi elektronsku molbu studentskoj službi
* **Prijava ispita** – studentu su izlistani dostupni ispiti u operativnoj memoriji. Ispiti su prikazani tabelarno. Pritiskom na dugme “Prijavi ispit”, student prebacuje dati ispit u sekciju “Prijavljeni ispiti”. Klikom na X dugme pored prijavljenog ispita, dati ispit se vraća u početnu tabelu. Baza podataka se ažurira na ispravan način nakon akcija studenta
* **Izmena podataka studenta** – Administratoru su tabelarno prikazani svi studenti učitani u operativnu memoriju. Izborom jednog studenta i klikom na dugme “Izmeni” otvara se stranica na kojoj su prikazani podaci o studentu koje je moguće izmeniti. Klikom na dugme “Potvrdi” se date izmene čuvaju u bazi podataka.
* **Prikaz svih uspešnih polaganja ispita** – Studentu su tabelarno prikazana sva uspešna polaganja ispita, prosečna ocena, kao i broj ostvarenih ESP bodova.

itd.

Ponašanje softverskog sistema

Radi detaljnijeg opisa ponašanja sistema pozvaćemo se na UML dijagrame sekvenci. S obzirom na obim rada, ovde su prikazani dijagrami sekvenci prvih 5 slučajeva korišćenja. Sve ovde pokazano analogno važi I za ostale slučajeve korišćenja.

NAZIV: Logovanje

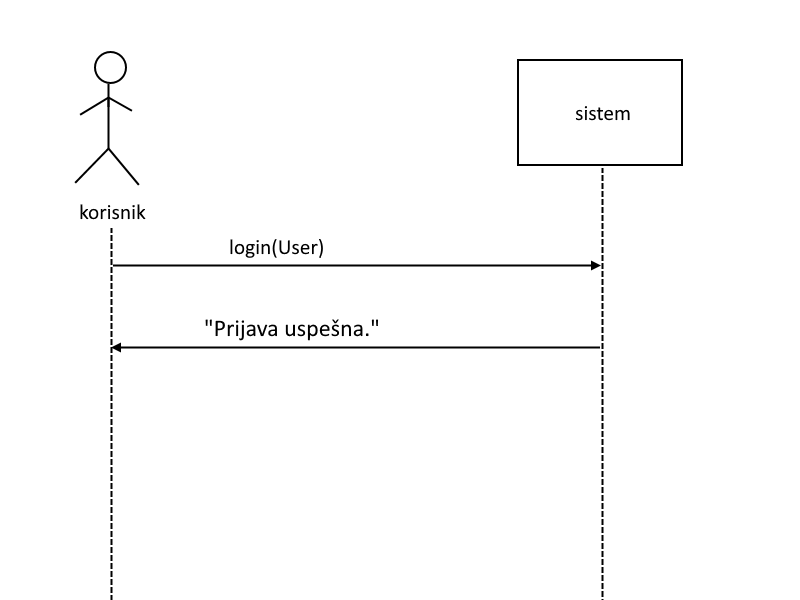
AKTERI: Korisnik (student ili admin)

UČESNICI: korisnik I sistem

PREDUSLOV: Sistem je uključen i korisniku je omogućeno prijavljivanje

SK1: Osnovni scenario:

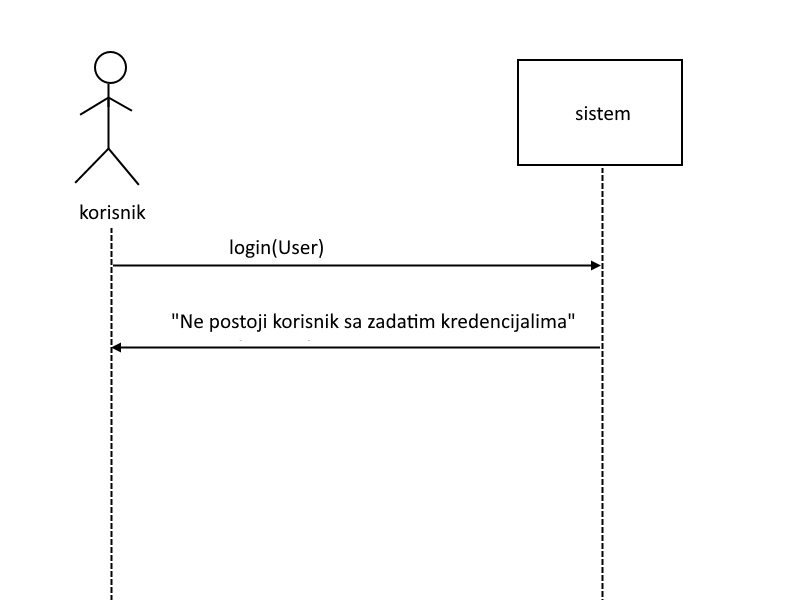
1. Korisnik pokreće formu za prijavu (APSO)
2. Sistem prikazuje korisniku formu (SO)
3. Korisnik poziva sistem da prijavi korisnika (APSO)
4. Sistem pronalazi korisnika i instancira objekat (SO)
5. Sistem prikazuje korisniku poruku “prijava uspešna”



Slika : Dijagram sekvenci glavnog scenarija SK1

Alternativni scenario:

* 1. Ukoliko sistem ne može da pronađe korisnika sa zadatim kredencijalima, ispisuje korisniku poruku “Ne postoji korisnik sa zadatim kredencijalima.” (IA)



Slika : Dijagram sekvenci prvog alternativnog scenarija SK1

NAZIV: Prikaz svih obaveštenja

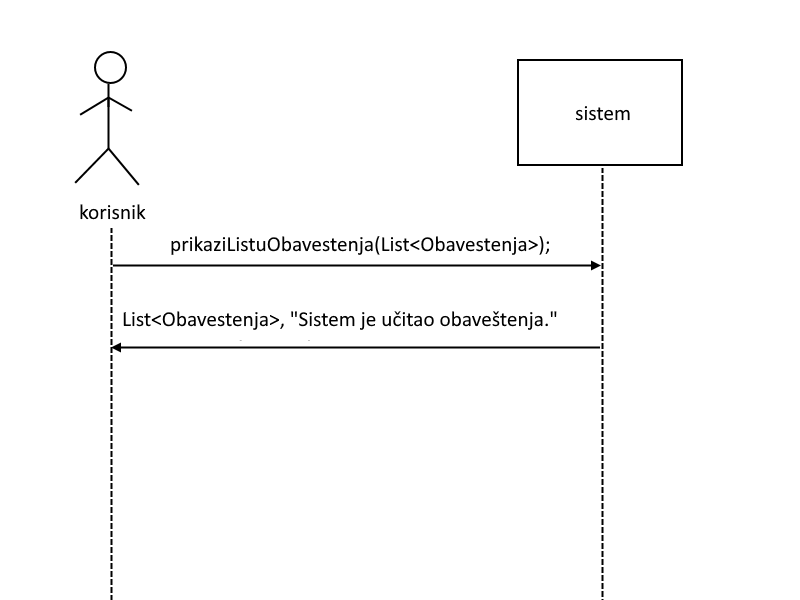
AKTERI: Korisnik (student ili admin)

UČESNICI: korisnik I sistem

PREDUSLOV: Sistem je uključen i prikazana je forma za rad sa klijentom.

SK2: Osnovni scenario:

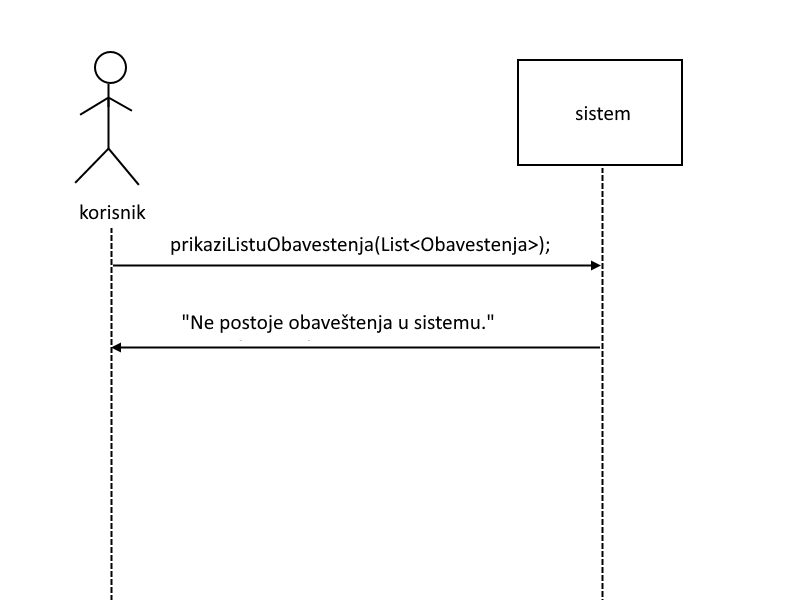
1. Korisnik poziva sistem da prikaže sva obaveštenja (APSO)
2. Sistem pronalazi sva obaveštenja (SO)
3. Sistem prikazuje korisniku obaveštenja i poruku “Sistem je učitao obaveštenja”



Slika : Dijagram sekvenci glavnog scenarija SK2

Alternativni scenario:

3.1 Ukoliko sistem ne može da pronađe obaveštenja, ispisuje korisniku poruku “Ne postoje obaveštenja u sistemu.” (IA)



Slika : Dijagram sekvenci prvog alternativnog scenarija SK2

NAZIV: Unos Molbe

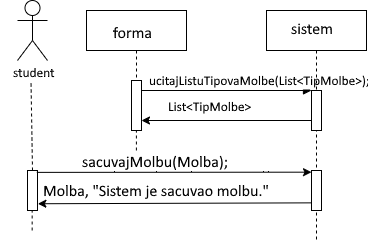
AKTERI: student

UČESNICI: student I sistem

PREDUSLOV: Sistem je uključen i prikazana je forma za rad sa studentom. Učitana je lista tipova molbe

SK3: Osnovni scenario:

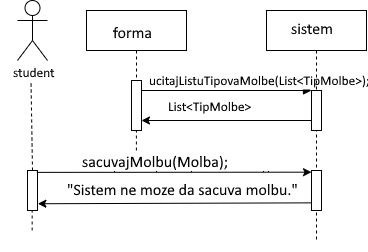
1. Korisnik poziva sistem da sačuva unete podatke o molbi (APSO)
2. Sistem čuva molbu i prikazuje korisniku poruku “Sistem je sačuvao molbu.”.



Slika : Dijagram sekvenci glavnog scenarija SK3

Alternativni scenario:

2.1 Ukoliko sistem ne može da sačuva molbu, ispisuje korisniku poruku “Sistem ne može da sačuva molbu.” (IA)



Slika : Dijagram sekvenci prvog alternativnog scenario SK3

NAZIV: Prikaz molbi u obradi

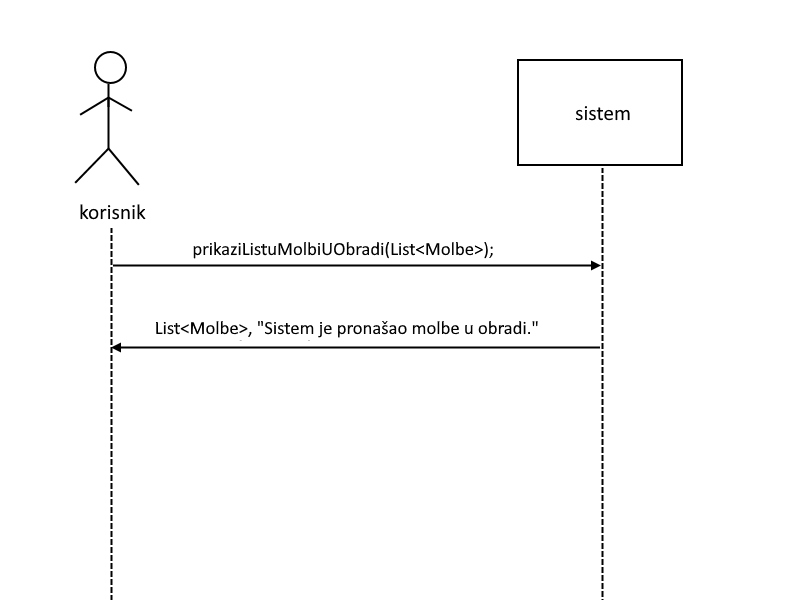
AKTERI: student

UČESNICI: student I sistem

PREDUSLOV: Sistem je uključen i prikazana je forma za rad sa studentom.

SK3: Osnovni scenario:

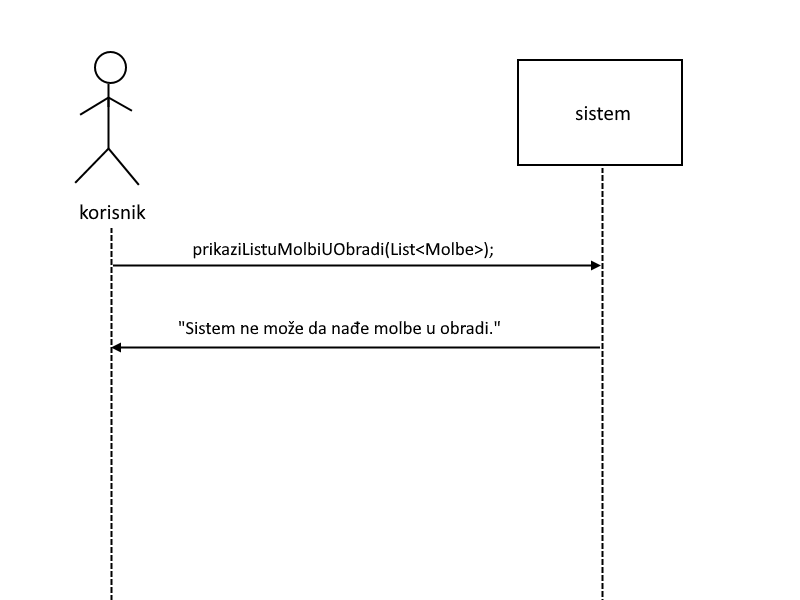
1. Korisnik poziva sistem da prikaže sve molbe u obradi (APSO)
2. Sistem nalazi sve molbe u obradi (SO)
3. Sistem prikazuje korisniku sve molbe u obradi i poruku “Sistem je pronašao molbe u obradi.”.



Slika : Dijagram sekvenci glavnog scenarija SK4

Alternativni scenario:

2.1 Ukoliko sistem ne može da nađe molbe u obradi, ispisuje korisniku poruku “Sistem ne može da nađe molbe u obradi.” (IA)



Slika : Dijagram sekvenci prvog alternativnog scenarija SK4

NAZIV: Prikaz izdatih molbi

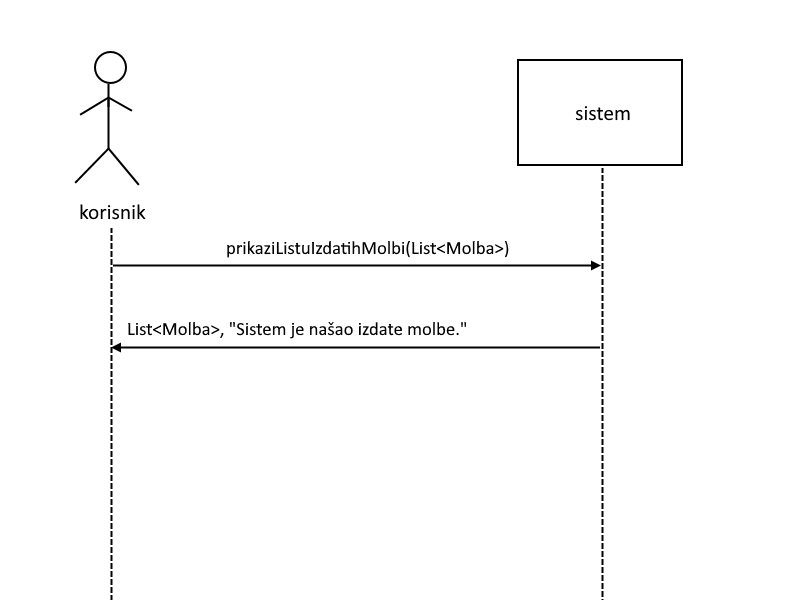
AKTERI: student

UČESNICI: student I sistem

PREDUSLOV: Sistem je uključen i prikazana je forma za rad sa studentom.

SK3: Osnovni scenario:

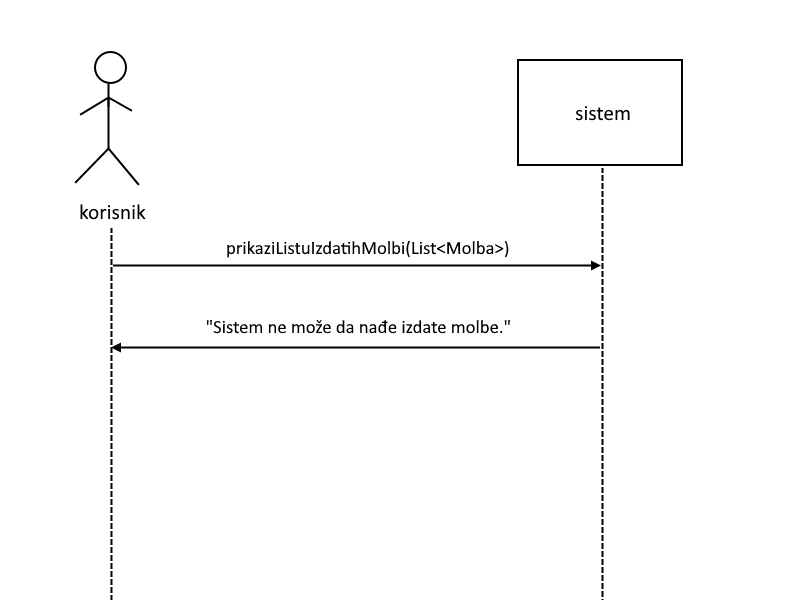
1. Korisnik poziva sistem da prikaže sve izdate molbe (APSO)
2. Sistem nalazi sve izdate molbe (SO)
3. Sistem prikazuje korisniku sve izdate molbe i poruku “Sistem je našao izdate molbe.”.



Slika : Dijagram sekvenci glavnog scenarija SK5

Alternativni scenario:

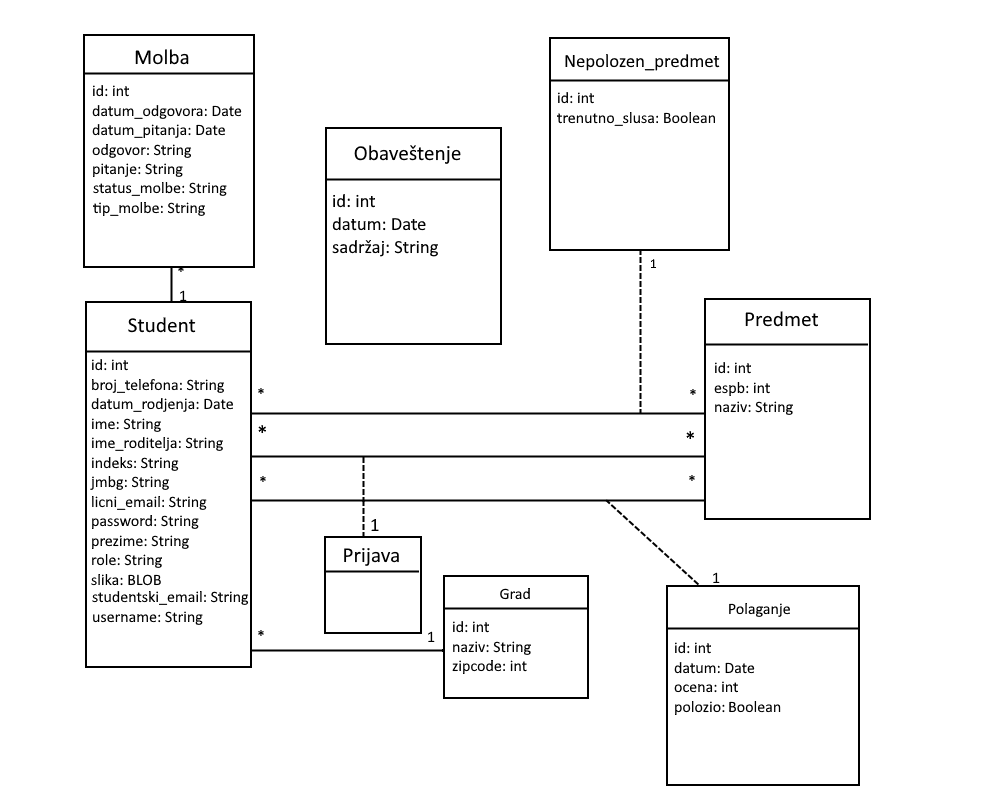
2.1 Ukoliko sistem ne može da nađe izdate molbe, ispisuje korisniku poruku “Sistem ne može da nađe izdate molbe.” (IA)



Slika : Dijagram sekvenci prvog alternativnog scenarija SK5

Struktura softverskog sistema

Da bismo detaljnije opisali strukturu softverskog sistema, koristimo sledeći dijagram klasa:



Slika 2: Dijagram klasa sistema

* Student tabela čuva relevantne podatke o studentima (i administratorima)
* Prijava je tabela koja označava ispit koji je prijavljen od strane studenta
* Predmet je tabela koja sadži relevantne podatke o predmetima
* Grad je tabela koja ima vezu sa studentom, koja sadrži relevantne podatke o rodnom mestu studenta
* Polaganje je asocijativna klasa Studenta i Predmeta (isto kao i Prijava), koja sadrži podatke o uspešnosti polaganja, oceni i datumu polaganja ispita
* Nepoložen predmet sadži podatke o tome da li student sluša predmet ili ne (asocijativna klasa studenta i predmeta)
* Obaveštenje je klasa koja čuva sve relevantne podatke koje je neophodno prikazati svim studentima

API Specifikacija

|  |  |
| --- | --- |
| Opis funkcije | Registracija |
| HTTP metoda | POST |
| URL | /register |
| URL parametri | (nema) |
| HTTP body parametri | JSON objekat sa atributima firstname[string], lastname[string], email[string], username[string], password[string]  Primer:  {  "Ime": "Pera",  "Prezime" : "Peric",  "Email" : "example@gmail.com",  "username" : "ppetar",  “password”: “123abc”  } |
| Format HTTP body parametara | application/json |
| Izlazni parametri |  |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Autentikacija |
| HTTP metoda | POST |
| URL | /authenticate |
| URL parametri | (nema) |
| HTTP body parametri | JSON objekat sa atributima username[string], password[string].  Primer:  {  "username”: “ppetar”,  “password”: “123abc  } |
| Format HTTP body parametara | application/json |
| Izlazni parametri | (nema) |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Preuzimanje grada iz baze po id-ju |
| HTTP metoda | GET |
| URL | /api/v1/gradovi |
| URL parametri | id[long] |
| HTTP body parametri |  |
| Format HTTP body parametara | application/json |
| Izlazni parametri | JSON objekat tipa grad  Primer:  {  “id”: 1,  “naziv”: “Beograd” ,  “zipcode”: 11000  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje grada sa datim id-jem |
| HTTP metoda | DELETE |
| URL | /api/v1/gradovi |
| URL parametri | id[long] |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri |  |
| Format izlaznih parametara | Application/json |

|  |  |
| --- | --- |
| Opis funkcije | Preuzimanje svih molbi iz baze |
| HTTP metoda | GET |
| URL | /api/v1/molbe |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | JSON niz objekata tipa molba  Primer:  [  {  “id” : 1,  “datum\_odgovora”: “2023-02-06”,  “datum\_pitanja”: “2023-02-04”,  “odgovor”: “Nije moguće ovo izvršiti”,  “pitanje”: “Da li možete da mi prijavite sve ispite?”,  “status\_molbe”: “RAZRESENA”,  “tip\_molbe”: “PROMENA\_PODATAKA\_O\_STUDENTU”,  “student\_id”: 1  },  {  “id” : 2,  “datum\_odgovora”: null,  “datum\_pitanja”: “2023-02-01”,  “odgovor”: null  “pitanje”: “Da li možete da mi promenite adresu stanovanja?”,  “status\_molbe”: “U\_OBRADI”,  “tip\_molbe”: “PROMENA\_PODATAKA\_O\_STUDENTU”,  “student\_id”: 2  },  {  “id” : 3,  “datum\_odgovora”: null,  “datum\_pitanja”: “2023-01-31”,  “odgovor”: null  “pitanje”: “Da li možete da promenite moju sliku?”,  “status\_molbe”: “U\_OBRADI”,  “tip\_molbe”: “PROMENA\_PODATAKA\_O\_STUDENTU”,  “student\_id”: 1  },  {  “id” : 3,  “datum\_odgovora”: 2023-02-07,  “datum\_pitanja”: “2023-01-30”,  “odgovor”: “Ispravljeno.”  “pitanje”: “Da li možete da ispravite moj broj indeksa?”,  “status\_molbe”: “RAZRESENA”,  “tip\_molbe”: “PROMENA\_PODATAKA\_O\_STUDENTU”,  “student\_id”: 2  }  ] |
| Format izlaznih parametara | Application/json |

|  |  |
| --- | --- |
| Opis funkcije | Preuzimanje molbi u obradi |
| HTTP metoda | GET |
| URL | /api/v1/molbe/uobradi |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | [  {  “id” : 2,  “datum\_odgovora”: null,  “datum\_pitanja”: “2023-02-01”,  “odgovor”: null  “pitanje”: “Da li možete da mi promenite adresu stanovanja?”,  “status\_molbe”: “U\_OBRADI”,  “tip\_molbe”: “PROMENA\_PODATAKA\_O\_STUDENTU”,  “student\_id”: 2  },  {  “id” : 3,  “datum\_odgovora”: null,  “datum\_pitanja”: “2023-01-31”,  “odgovor”: null  “pitanje”: “Da li možete da promenite moju sliku?”,  “status\_molbe”: “U\_OBRADI”,  “tip\_molbe”: “PROMENA\_PODATAKA\_O\_STUDENTU”,  “student\_id”: 1  },  ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Preuzimanje molbi koje su razrešene |
| HTTP metoda | GET |
| URL | /api/v1/molbe/uobradi |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | [  {  “id” : 1,  “datum\_odgovora”: “2023-02-06”,  “datum\_pitanja”: “2023-02-04”,  “odgovor”: “Nije moguće ovo izvršiti”,  “pitanje”: “Da li možete da mi prijavite sve ispite?”,  “status\_molbe”: “RAZRESENA”,  “tip\_molbe”: “PROMENA\_PODATAKA\_O\_STUDENTU”,  “student\_id”: 1  },  {  “id” : 3,  “datum\_odgovora”: 2023-02-07,  “datum\_pitanja”: “2023-01-30”,  “odgovor”: “Ispravljeno.”  “pitanje”: “Da li možete da ispravite moj broj indeksa?”,  “status\_molbe”: “RAZRESENA”,  “tip\_molbe”: “PROMENA\_PODATAKA\_O\_STUDENTU”,  “student\_id”: 2  }  ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Preuzimanje molbi koje su razrešene |
| HTTP metoda | GET |
| URL | /api/v1/molbe/{id} |
| URL parametri | Id:long |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | {  “id” : 1,  “datum\_odgovora”: “2023-02-06”,  “datum\_pitanja”: “2023-02-04”,  “odgovor”: “Nije moguće ovo izvršiti”,  “pitanje”: “Da li možete da mi prijavite sve ispite?”,  “status\_molbe”: “RAZRESENA”,  “tip\_molbe”: “PROMENA\_PODATAKA\_O\_STUDENTU”,  “student\_id”: 1  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Dodavanje molbe u obradi |
| HTTP metoda | POST |
| URL | /api/v1/molbe/uobradi |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | {  “id” : 5,  “datum\_odgovora”: null,  “datum\_pitanja”: “2023-02-10”,  “odgovor”: null,  “pitanje”: “Da li možete da mi odjavite ispit “Matematika”?”  “status\_molbe”: “U\_OBRADI”,  “tip\_molbe”: “PROMENA\_PODATAKA\_O\_STUDENTU”,  “student\_id”: 1  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Razresavanje molbe |
| HTTP metoda | PATCH |
| URL | /api/v1/molbe/razresi |
| URL parametri | (nema) |
| HTTP body parametri | {  “id” : 5,  “datum\_odgovora”: null,  “datum\_pitanja”: “2023-02-10”,  “odgovor”: null,  “pitanje”: “Da li možete da mi odjavite ispit “Matematika”?”  “status\_molbe”: “U\_OBRADI”,  “tip\_molbe”: “PROMENA\_PODATAKA\_O\_STUDENTU”,  “student\_id”: 1  } |
| Format HTTP body parametara | application/json |
| Izlazni parametri | {  “id” : 5,  “datum\_odgovora”: “2023-02-20”,,  “datum\_pitanja”: “2023-02-10”,  “odgovor”: null,  “pitanje”: “Da li možete da mi odjavite ispit “Matematika”?”,  “status\_molbe”: “RAZRESENA”,  “tip\_molbe”: “PROMENA\_PODATAKA\_O\_STUDENTU”,  “student\_id”: 1  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Razresavanje molbe sa zadatim id-jem |
| HTTP metoda | PATCH |
| URL | /api/v1/molbe/{id}/razresi |
| URL parametri | Id: long |
| HTTP body parametri | Id: long, odgovor: String |
| Format HTTP body parametara | application/json |
| Izlazni parametri | {  “id” : 5,  “datum\_odgovora”: “2023-02-20”,,  “datum\_pitanja”: “2023-02-10”,  “odgovor”: “Urađeno”,  “pitanje”: “Da li možete da mi odjavite ispit “Matematika”?”,  “status\_molbe”: “RAZRESENA”,  “tip\_molbe”: “PROMENA\_PODATAKA\_O\_STUDENTU”,  “student\_id”: 1  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Preuzimanje obaveštenja sa zadatim ID-jem |
| HTTP metoda | GET |
| URL | /api/v1/obavestenje/{id} |
| URL parametri | Id: long |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | {  “id” : 1,  “datum”: “2023-02-20”,  “sadrzaj”: “Obavestavamo studente da je subota, 04-03. radni dan i da se radi po rasporedu za sredu.”  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje obaveštenja sa zadatim ID-jem |
| HTTP metoda | DELETE |
| URL | /api/v1/obavestenje/{id} |
| URL parametri | Id: long |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri |  |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Dodavanje obaveštenja |
| HTTP metoda | POST |
| URL | /api/v1/obavestenje |
| URL parametri | (nema) |
| HTTP body parametri | {  “id” : 1,  “datum”: “2023-02-20”,  “sadrzaj”: “Konkurs za prakse firme X.”  } |
| Format HTTP body parametara | application/json |
| Izlazni parametri |  |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Preuzimanje svih polaganja |
| HTTP metoda | GET |
| URL | /api/v1/polaganja |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | [  {  “id”: 1,  “datum”: “2023-01-04”,  “ocena”: “10”,  “polozio”: “true”,  “predmet\_id”: 1,  “student\_id”: 2  },  {  “id”: 2,  “datum”: “2023-01-17”,  “ocena”: “5”,  “polozio”: “false”,  “predmet\_id”: 2,  “student\_id”: 2  },  {  “id”: 3,  “datum”: “2023-01-04”,  “ocena”: “8”,  “polozio”: “true”,  “predmet\_id”: 2,  “student\_id”: 1  }  ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Preuzimanje polaganja sa zadatim ID-jem |
| HTTP metoda | GET |
| URL | /api/v1/polaganja/{id} |
| URL parametri | Id: long |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | {  “id”: 1,  “datum”: “2023-01-04”,  “ocena”: “10”,  “polozio”: “true”,  “predmet\_id”: 1,  “student\_id”: 2  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Dodavanje polaganja |
| HTTP metoda | POST |
| URL | /api/v1/polaganja |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | {  “id”: 4,  “datum”: “2023-02-10”,  “ocena”: “7”,  “polozio”: “true”,  “predmet\_id”: 2,  “student\_id”: 2  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje polaganja |
| HTTP metoda | DELETE |
| URL | /api/v1/polaganja/{id} |
| URL parametri | Id:long |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri |  |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Preuzimanje uspešnih polaganja |
| HTTP metoda | GET |
| URL | /api/v1/polaganja/uspesna |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | [  {  “id”: 1,  “datum”: “2023-01-04”,  “ocena”: “10”,  “polozio”: “true”,  “predmet\_id”: 1,  “student\_id”: 2  },  {  “id”: 3,  “datum”: “2023-01-04”,  “ocena”: “8”,  “polozio”: “true”,  “predmet\_id”: 2,  “student\_id”: 1  }  ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Preuzimanje neuspešnih polaganja |
| HTTP metoda | GET |
| URL | /api/v1/polaganja/uspesna |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | [  {  “id”: 2,  “datum”: “2023-01-17”,  “ocena”: “5”,  “polozio”: “false”,  “predmet\_id”: 2,  “student\_id”: 2  }  ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Preuzimanje nepoloženih predmeta |
| HTTP metoda | GET |
| URL | /api/v1/predmeti/nepolozeni |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | [  {  “id”: 1,  “trenutno\_slusa”: false,  “predmet\_id”: 1,  “student\_id”: 1  },  {  “id”: 2,  “trenutno\_slusa”: true,  “predmet\_id”: 2,  “student\_id”: 1  },  {  “id”: 3,  “trenutno\_slusa”: true,  “predmet\_id”: 1,  “student\_id”: 2  }  ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Preuzimanje nepoloženih predmeta koji se trenutno slušaju |
| HTTP metoda | GET |
| URL | /api/v1/predmeti/slusa |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | [  {  “id”: 2,  “trenutno\_slusa”: true,  “predmet\_id”: 2,  “student\_id”: 1  },  {  “id”: 3,  “trenutno\_slusa”: true,  “predmet\_id”: 1,  “student\_id”: 2  }  ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Biranje predmeta za slušanje |
| HTTP metoda | PATCH |
| URL | /api/v1/predmeti/slusa |
| URL parametri | {  “id”: 3,  “espb: 6,  “naziv”: “Elektronsko poslovanje”  } |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri |  |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Preuzimanje svih prijava |
| HTTP metoda | GET |
| URL | /api/v1/prijave/ |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | [  {  “student\_id”: 1,  “predmet\_id”: 2  },  {  “student\_id”: 2,  “predmet\_id”: 2  },  {  “student\_id”: 1,  “predmet\_id”: 4  }  ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Preuzimanje prijave sa zadatim id-jem |
| HTTP metoda | GET |
| URL | /api/v1/prijave/{id} |
| URL parametri | Id:Long |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | [  {  “student\_id”: 1,  “predmet\_id”: 2  },  {  “student\_id”: 2,  “predmet\_id”: 2  },  {  “student\_id”: 1,  “predmet\_id”: 4  }  ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Preuzimanje prijave sa zadatim id-jem |
| HTTP metoda | GET |
| URL | /api/v1/prijave/{id} |
| URL parametri | Id:Long |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | [  {  “student\_id”: 1,  “predmet\_id”: 2  },  {  “student\_id”: 2,  “predmet\_id”: 2  },  {  “student\_id”: 1,  “predmet\_id”: 4  }  ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Dodavanje nove prijave |
| HTTP metoda | POST |
| URL | /api/v1/prijave |
| URL parametri | Id:Long |
| HTTP body parametri | {  “student\_id”: 1,  “predmet\_id”: 3  } |
| Format HTTP body parametara | application/json |
| Izlazni parametri |  |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje prijave sa zadatim id-jem |
| HTTP metoda | DELETE |
| URL | /api/v1/prijave/{id} |
| URL parametri | Id:Long |
| HTTP body parametri |  |
| Format HTTP body parametara | application/json |
| Izlazni parametri |  |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Preuzimanje svih studenata |
| HTTP metoda | GET |
| URL | /api/v1/studenti |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | [  {  “id”: 0,  “broj\_telefona”: “+38167777777”,  “datum\_rodjenja”: null,  “ime”: “Admin”,  “ime\_roditelja”: “Admin”,  “indeks”: “0000/2019”,  “jmbg”: “000000000000”,  “licni\_email”: “[admin@gmail.com](mailto:admin@gmail.com)”,  “password”: “$2a$12$Y8KKkfMj1/rxHj5Uzq.U9.Ntog8gGp8BoUla9FjIo7r5Vr7UegfXW”,  “prezime”: “Admin”,  “role”: “ADMIN”,  “slika”: “<https://balkan-hub.com/prodavnica/483-Niara_thickbox/admin.jpg>”,  “studentski\_email”: “aa00000000@student.fon.bg.ac.rs”,  “username”: “aa00000000”,  “mesto\_rodjenja\_id”: 1  },  {  “id”: 1,  “broj\_telefona”: “+381677116969”,  “datum\_rodjenja”: null,  “ime”: “Veljko”,  “ime\_roditelja”: “Borislav”,  “indeks”: “0353/2019”,  “jmbg”: “00208000860069”,  “licni\_email”: “[veljko@gmail.com](mailto:veljko@gmail.com)”,  “password”: “$2a$12$yLCUlQXTK4LQM1jGzZ0IdeAOs7QPW7B1z62g6o8c.07KbDq5K7Xka”,  “prezime”: “Blagojevic”,  “role”: “USER”,  “slika”: “https://media.licdn.com/dms/image/D4D03AQGfDciO9qccSg/profile-displayphoto-shrink\_800\_800/0/1665691382240?e=1680739200&v=beta&t=OmDSbHTkRPAUegG52SrowRDi8GiqjSDvE7c6OCQvL6k”,  “studentski\_email”: “vb20190353@student.fon.bg.ac.rs”,  “username”: “vb20190353”,  “mesto\_rodjenja\_id”: 3  },  {  “id”: 2,  “broj\_telefona”: “+381677114512”,  “datum\_rodjenja”: null,  “ime”: “Nikola”,  “ime\_roditelja”: “Nesa”,  “indeks”: “0062/2019”,  “jmbg”: “1201001860051”,  “licni\_email”: “[nikola@gmail.com](mailto:nikola@gmail.com)”,  “password”: “$2a$12$DSxUIJvdJcrIv6iBFxmpGup6.7Mf6MVwiFtCZEM3H7iqvqPDerq6W”,  “prezime”: “Vujcic”,  “role”: “USER”,  “slika”: “https://media.licdn.com/dms/image/D4E03AQERPaxNYTUzXg/profile-displayphoto-shrink\_800\_800/0/1667431634582?e=1680739200&v=beta&t=KXdvm7fvNZBwtoiPuZ1xT3gZKLlrRWMsn1dudYf-Ir0”,  “studentski\_email”: “nv20190062@student.fon.bg.ac.rs”,  “username”: “nv20190062”,  “mesto\_rodjenja\_id”: 6  }  ] |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | /api/v1/studenti/trenutni |
| HTTP metoda | GET |
| URL | /api/v1/studenti |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | application/json |
| Izlazni parametri | {  “id”: 0,  “broj\_telefona”: “+38167777777”,  “datum\_rodjenja”: null,  “ime”: “Admin”,  “ime\_roditelja”: “Admin”,  “indeks”: “0000/2019”,  “jmbg”: “000000000000”,  “licni\_email”: “[admin@gmail.com](mailto:admin@gmail.com)”,  “password”: “$2a$12$Y8KKkfMj1/rxHj5Uzq.U9.Ntog8gGp8BoUla9FjIo7r5Vr7UegfXW”,  “prezime”: “Admin”,  “role”: “ADMIN”,  “slika”: “<https://balkan-hub.com/prodavnica/483-Niara_thickbox/admin.jpg>”,  “studentski\_email”: “aa00000000@student.fon.bg.ac.rs”,  “username”: “aa00000000”,  “mesto\_rodjenja\_id”: 1  } |
| Format izlaznih parametara | application/json |

Opis arhitekture aplikacije

# Java Spring

Java Spring je open source okvir (eng. Framework) programskog jezika Java koji se neretko koristi u web development-u. Iako se najčešće koristi u svrhe razvoja web aplikacija, jezgro samog okvira može koristiti bilo koja Java aplikacija. Snaga ovog okvira leži u tome što on nije glomazan (eng. Lightweight), te obična verzija ovog okvira zauzima samo oko 2MB. Još jedna prednost je i modularnost samog okvira, te je fajlsistem (eng. Filesystem) vrlo intuitivan i jasan programeru. Ovaj okvir je kompatibilan sa MVC paternom projektovanja i kao takav idealan za pravljenje web aplikacija. Mi smo Java Spring okvir koristili za izradu backend dela ovog sajta.

# React

React je besplatna i open source frontend biblioteka programskog jezika Javascript koja služi za izradu korisničkih interfejsa zasnovanih na komponentama. Jednostavan je i lak za upotrebu jer se oslanja na osnovne tehnologije HTML-a i CSS-a (eng. Hypertext markup language i cascading style sheets). Osnovan od strane kompanije Meta (facebook), 2013. godine, React je i dalje jedan od najpopularnijih i najpoznatijih biblioteka programskog jezika Javascript. Komponente u ovoj biblioteci mogu biti klasne ili funkcionalne, a mi smo koristili oba tipa komponenata u našem radu. Ovu biblioteku smo prevashodno (čak iskljičivo) koristili za dizajn korisničkog interfejsa i frontend deo naše aplikacije.

# Axios

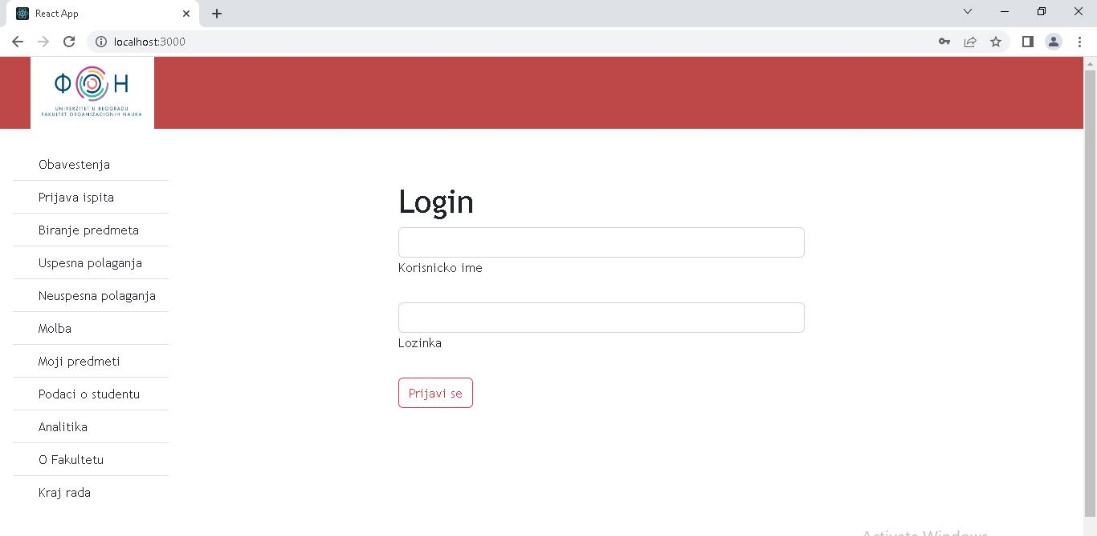
Axios je lightweight HTTP klijent zasnovan na Javascript obećanjima (eng. Promise) koji omogućava slanje HTTP zahteva sa i bez tela, kao i kreiranje zaglavlja. Ovaj moćan alat za slanje podataka smo koristili kako bismo povezali backend deo sa frontend-om, i vukli smislene podatke iz baze, kadgod ih bilo neophodno prikazati korisniku. Podatke koje je korisnik uneo i izmenio smo slali nazad na backend. S obzirom da je “promise-based”, ovaj alat podržava korišćenje kako regularne sintakse (sa “then” nakon izvršavanja), tako i async-await sintakse karakteristične za Javascript programski jezik. U ovom radu smo koristili oba tipa sintakse za različite pozive.

# MySQL baza podataka

Za MySQL bazu podataka nije potreban uvod i pojašnjenje, jer ovaj tip relacione baze podataka se ustalio kao industrijski standard u sferi baza podataka. Alat koji je korišćen za rad sa tabelama u MySQL bazi podataka je besplatni program pod nazivom SQLYog.

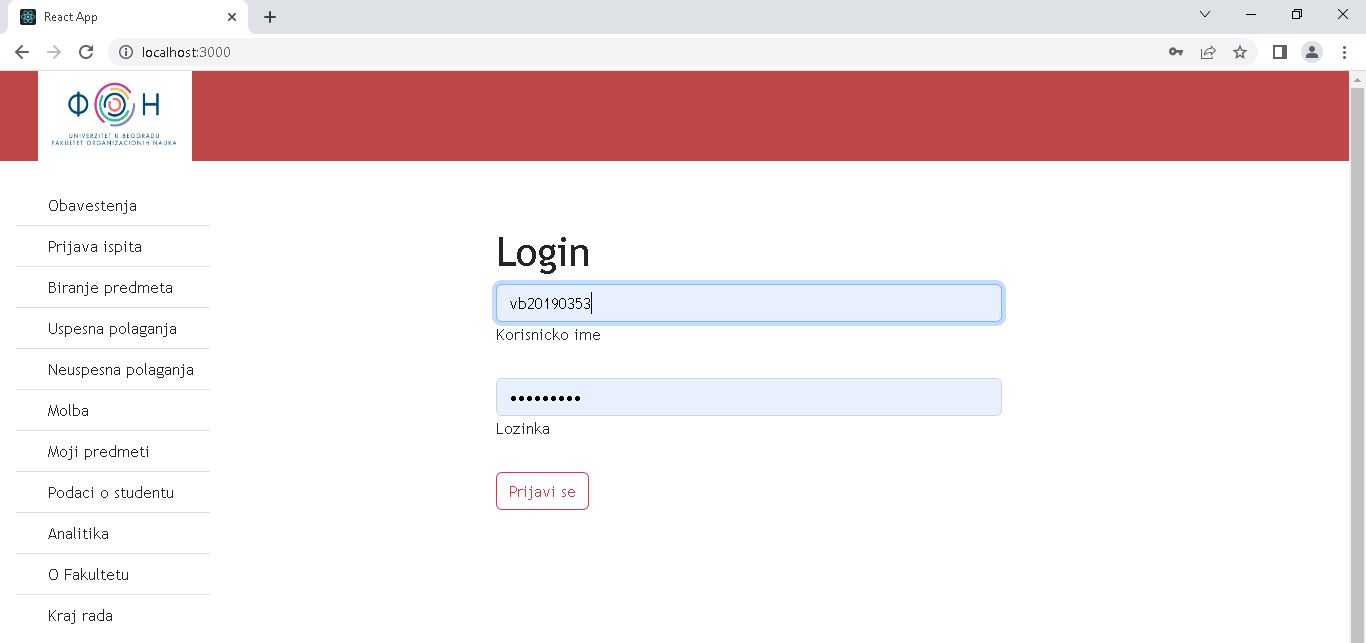
Korisničko uputstvo

Kada korisnik pokrene sajt, njemu se otvara sledeća stranica:



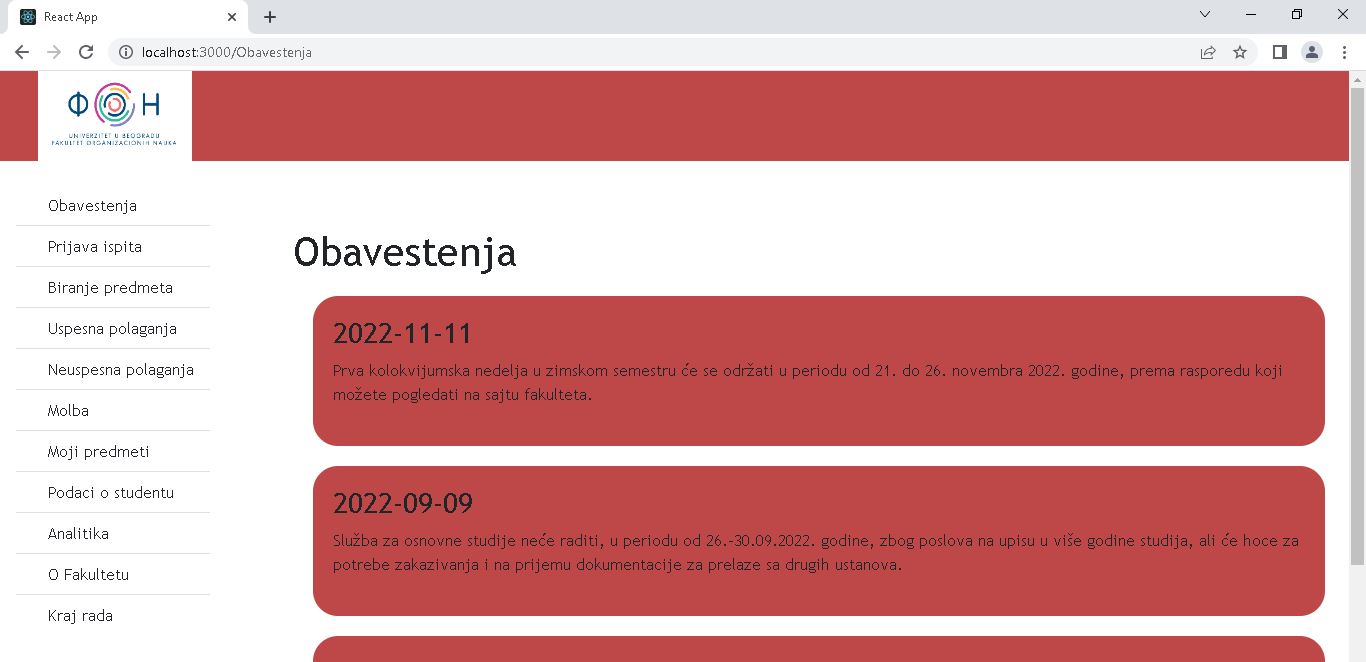
Slika: Početna stranica

Sa leve strane sa nalazi navigacioni meni korisnika. Na samoj stranici je forma za prijavu gde korisnik unosi relevantne podatke i pritiska dugme “Prijavi se” da bi nastavio rad na sajtu.



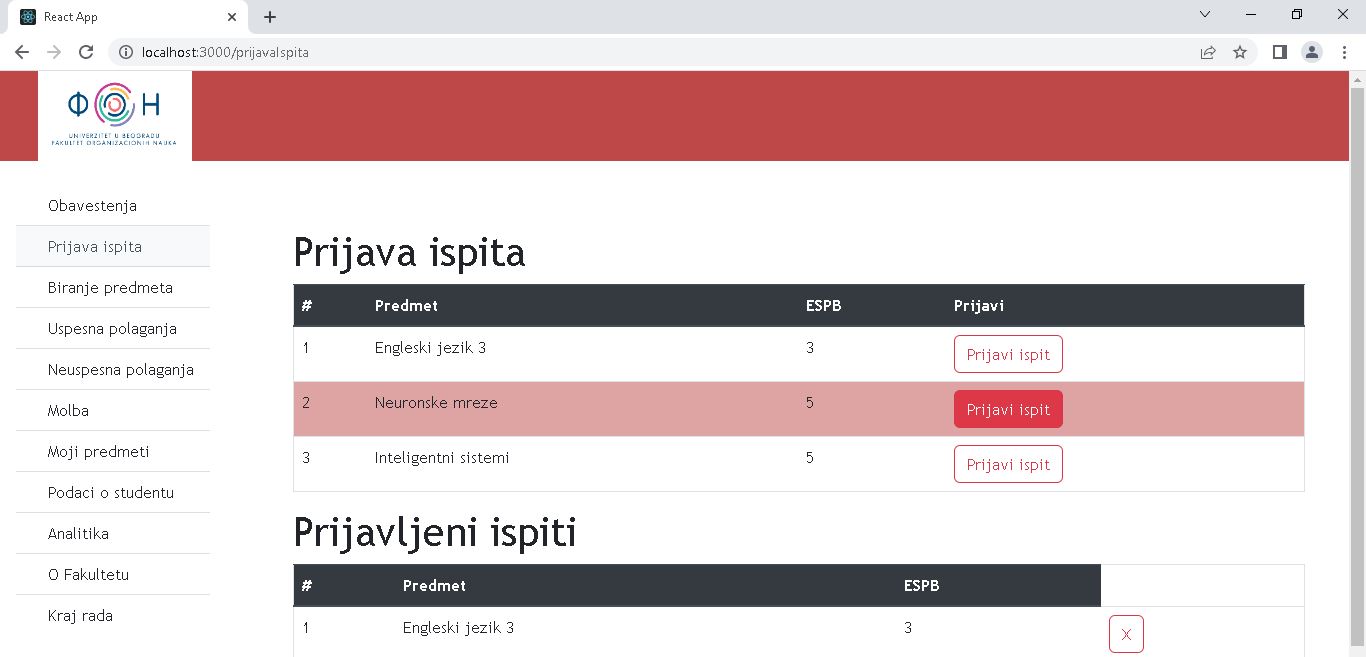
Slika: Korisnik je uneo podatke. U pitanju je student

Nakon unosa podataka (vidimo da je korisnik običan student, te nema admin privilegije), on biva preusmeren na stranicu sa Obaveštenjima.



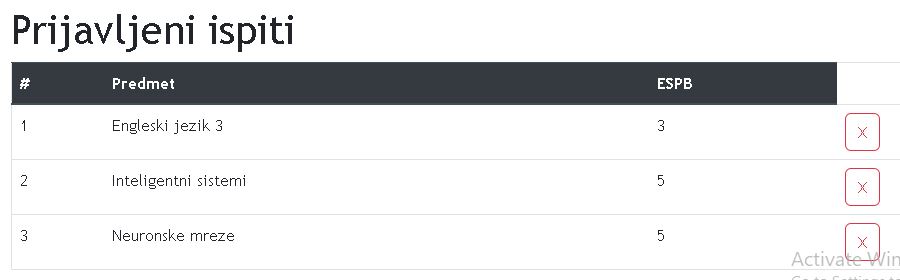
Slika : Stranica sa obaveštenjima

Nakon što je student pročitao obaveštenja, on može preko navigacionog menija da ode na drugu lokaciju, što mu otvara nove stranice. Prva od tih lokacija je sekcija za prijavu ispita. Onda se otvara sledeća stranica:



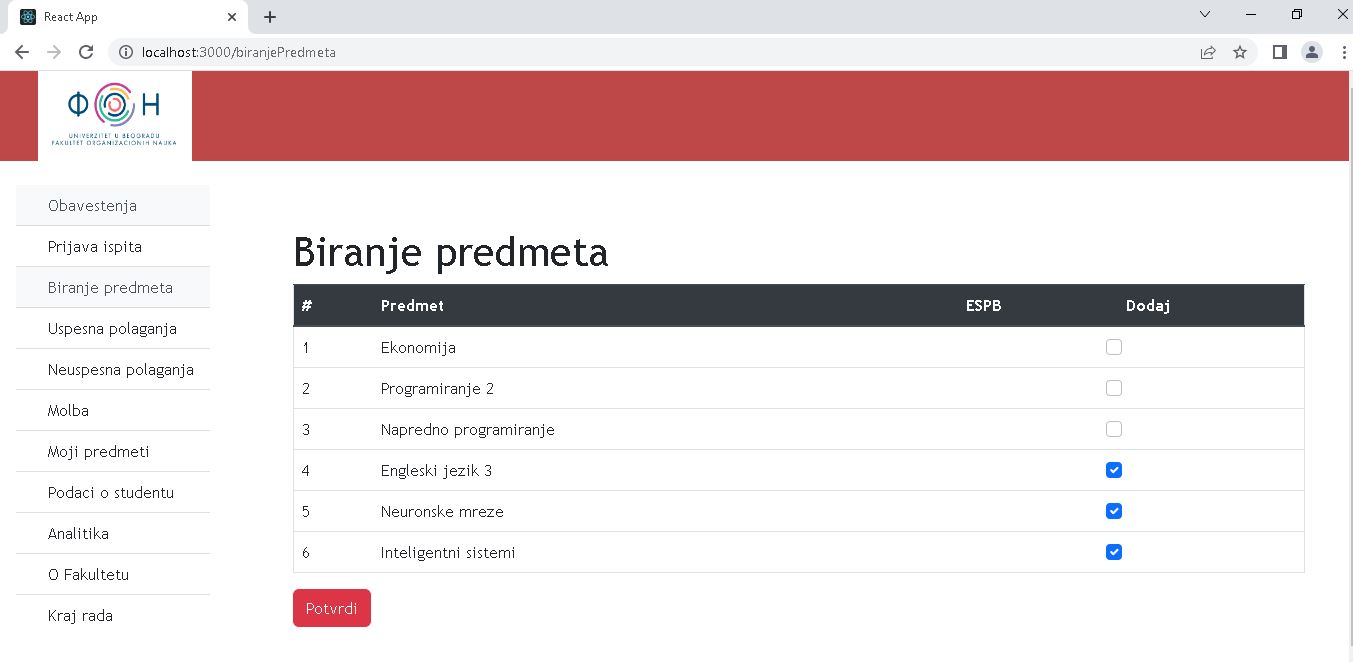
Slika : Prijava ispita

Klikom na dugme “Prijavi ispit” se dati ispit briše iz sekcije “Prijava ispita” i pojavljuje u sekciji “Prijavljeni ispiti”. Klikom na dugme X pored prijavljenog ispita, se isti odjavljuje.



Slika : Prijavljeni ispiti

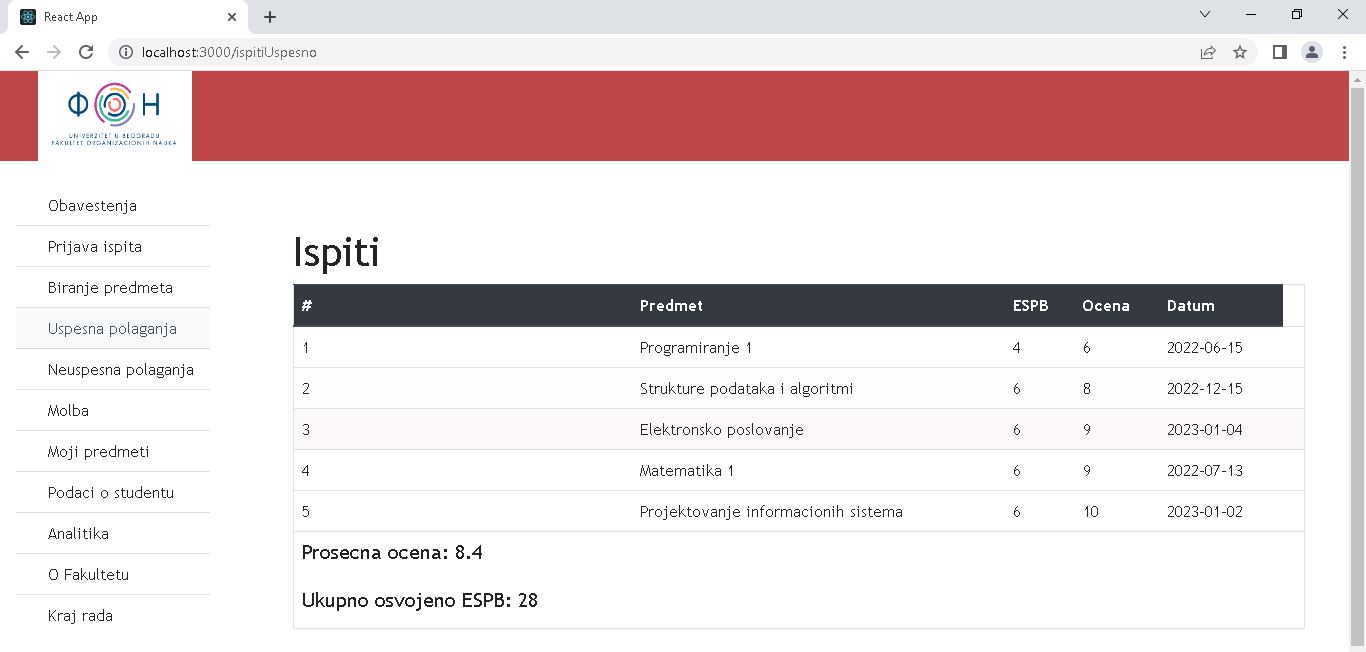
Sledeća adresa na koju može da se uputi student je sekcija za biranje predmeta. Nakon otvaranja stranice, student vidi sledeće:



Slika : Opcija- izbor predmeta za slušanje

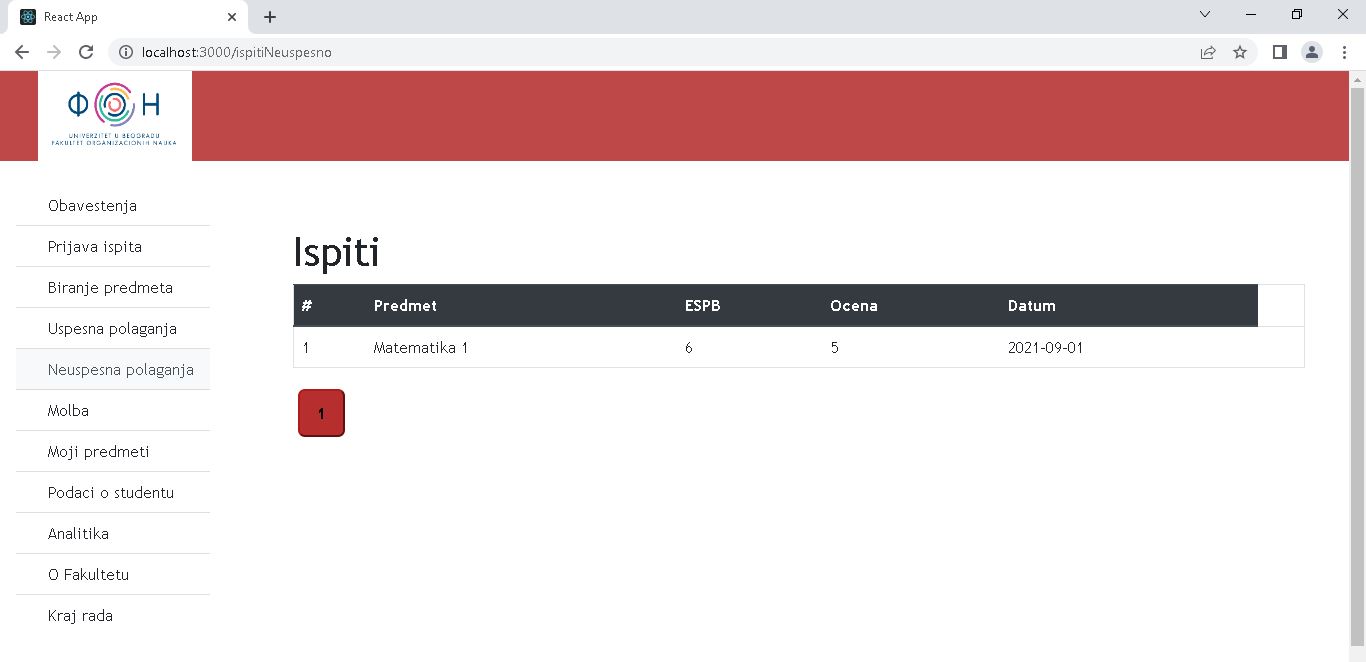
Čekiranjem checkbox-a pored odgovarajućeg predmeta se isti obeležava za izbor. Da bi se finalizovao izbor predmeta za slušanje, neophodno je pritisnuti dugme “Potvrdi” čime se izbor čuva.

U sekciji „Uspešna polaganja” se studentu prikazuje spisak uspešno položenih ispita, broj ESP bodova i podaci o prosečnoj oceni.



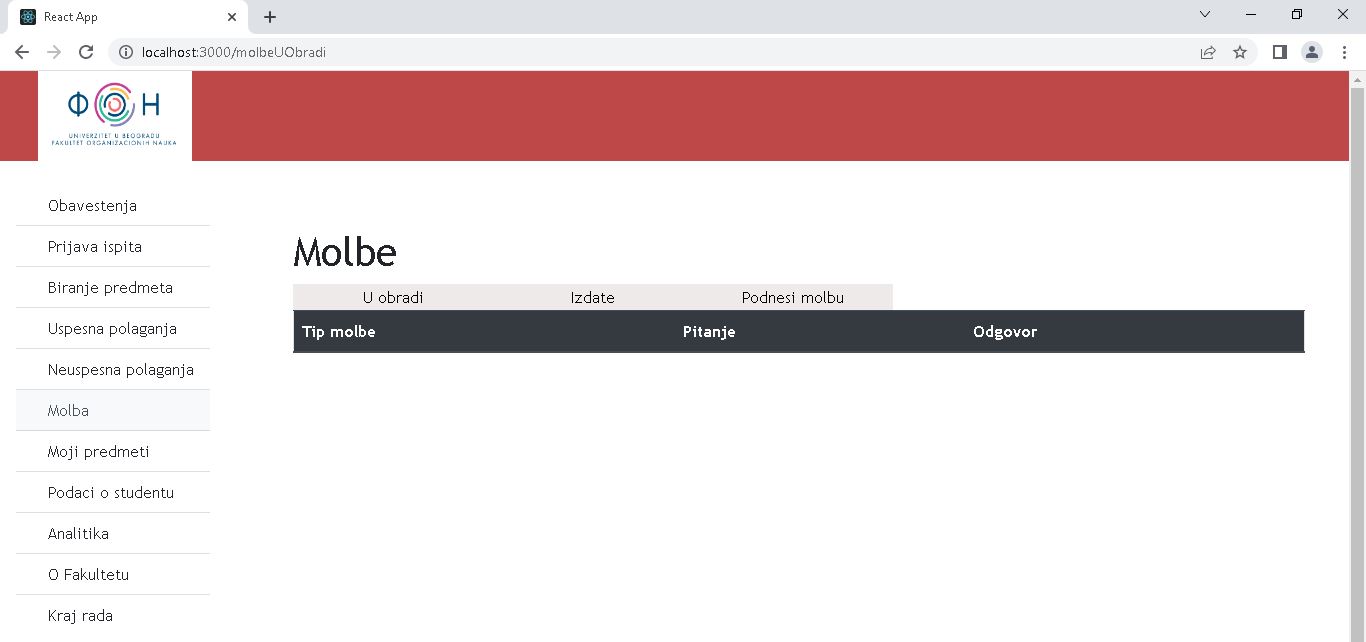
Slika : Sekcija Uspešna polaganja

Neuspešna polaganja funkcionišu na sličan način, mada je prikaz malo drugačiji. Korišćena je paginacija, tako da se studentu na početku prikazuje svega par ispita, a on klikom na odgovarajuće dugme prelazi na adekvatnu stranicu. S obzirom da postoji samo jedna stranica u ovom slučaju, njemu će se samo pojaviti dugme sa brojem 1.



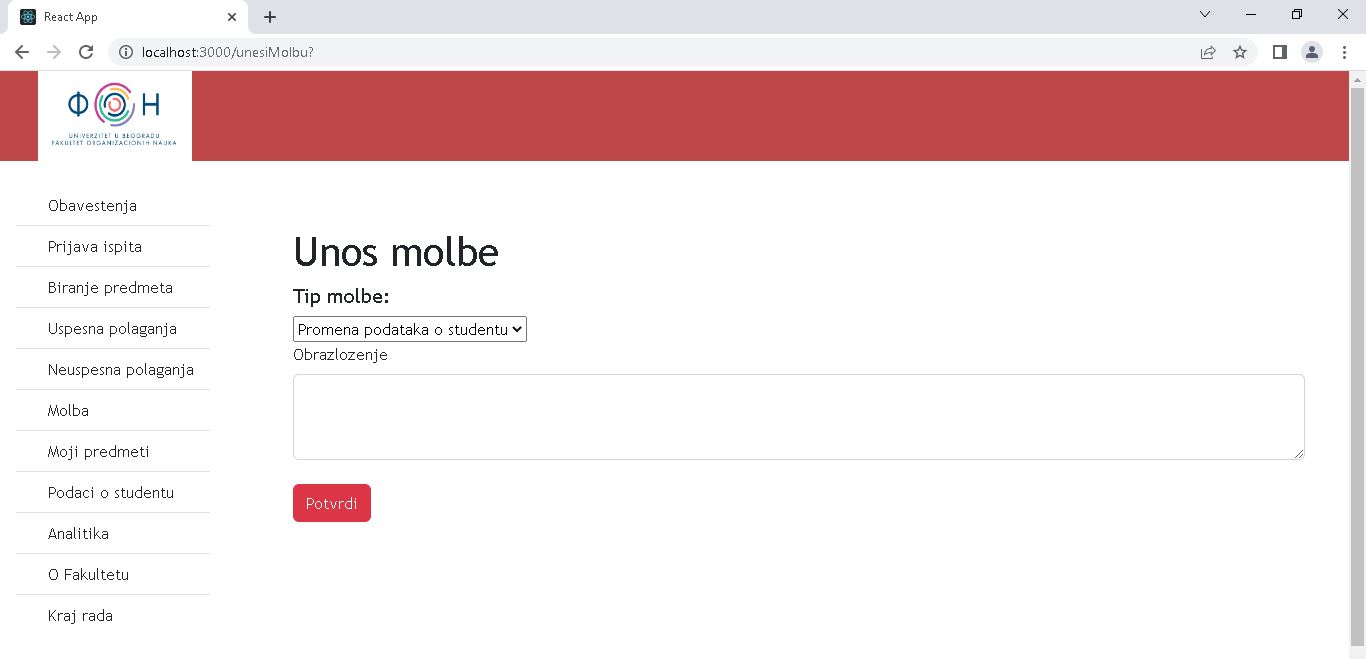
Slika : Neuspešna polaganja

Molbe se mogu podeliti u 2 kategorije. Molbe u obradi i razrešene molbe. Klikom na link “Molbe”, studentu je prikazan sledeći meni:



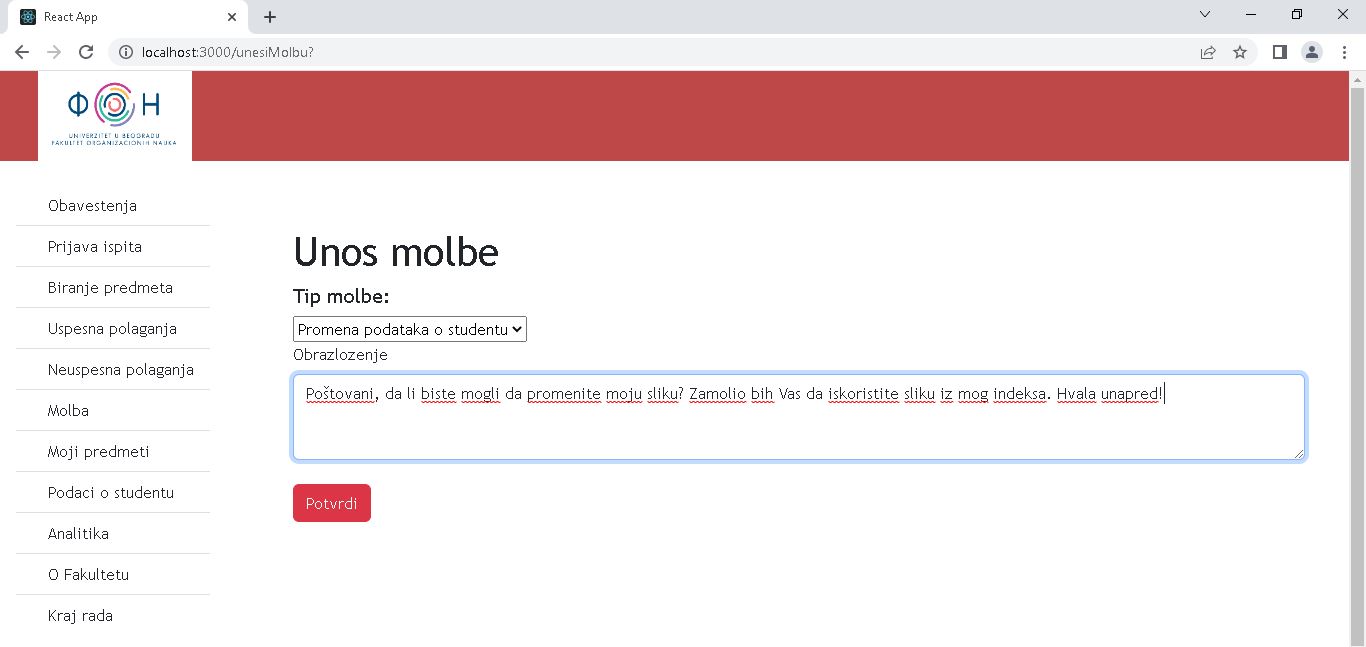
Slika : Meni molbe

Klikom na dugmiće “U obradi” I “Izdate” se korisniku prikazuju molbe u obradi i izdate molbe. S obzirom da ovaj korisnik nema molbe, prelazimo na sekciju “Podnesi molbu”.



Slika : Stranica “Podnesi molbu”

Ovde student iz menija bira tip molbe, i ručno unosi obrazloženje. Primer unosa molbe:



Slika : Unos molbe

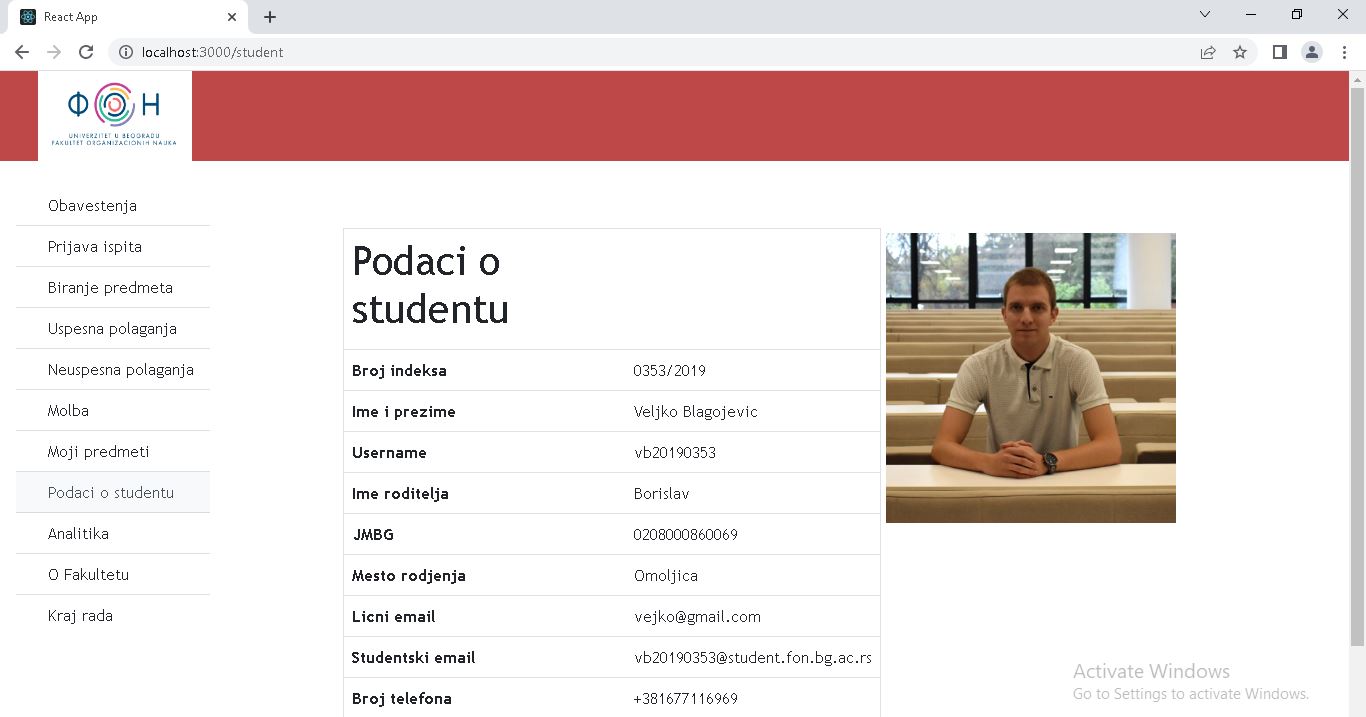
Pritiskom na dugme “Potvrdi” se molba čuva u sistemu.

Kada student ode na stranicu “Moji predmeti”, njemu je predstavljen tabelarni prikaz svih predmeta prijavljenih za slušanje. Korišćena je paginacija, kao što je objašnjeno u sekciji “Neuspešna polaganja”.



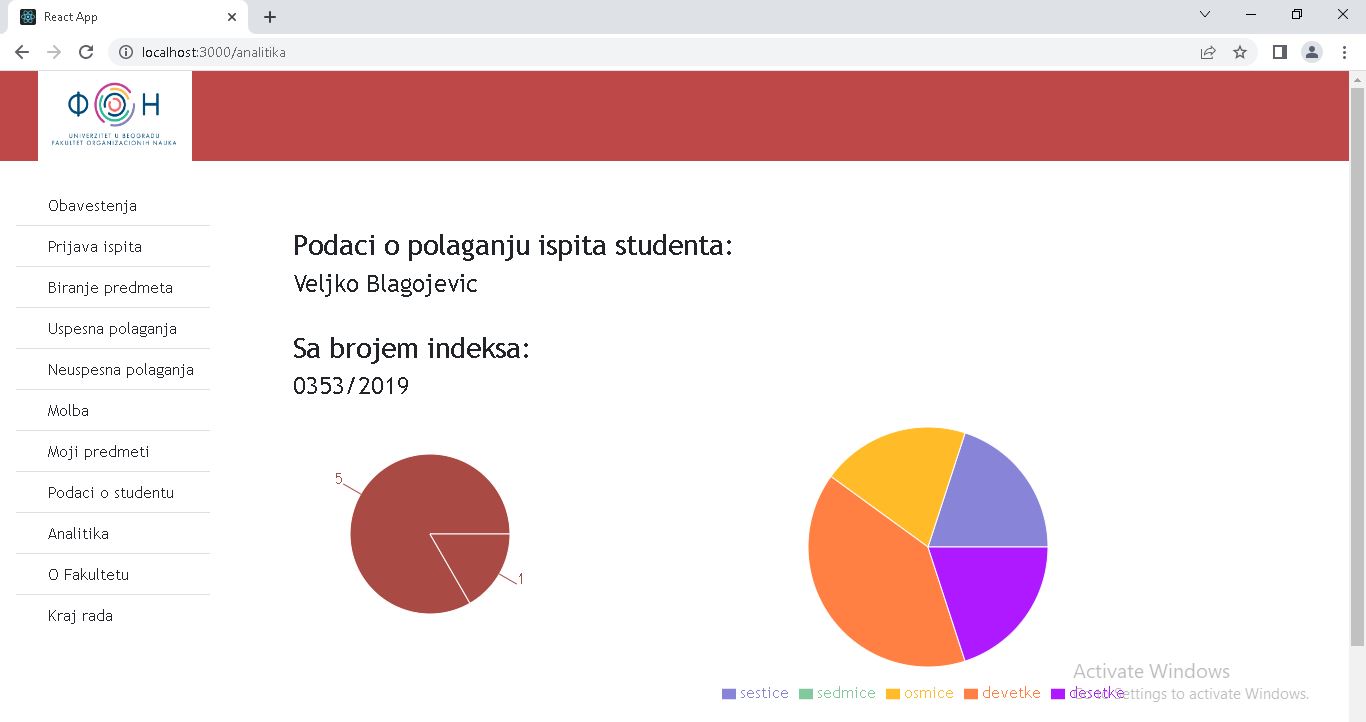
Slika : Stranica “Moji predmeti”

Stranica “Podaci o studentu” na korisniku jasan način prikazuje sve relevantne podatke prijavljenog studenta.



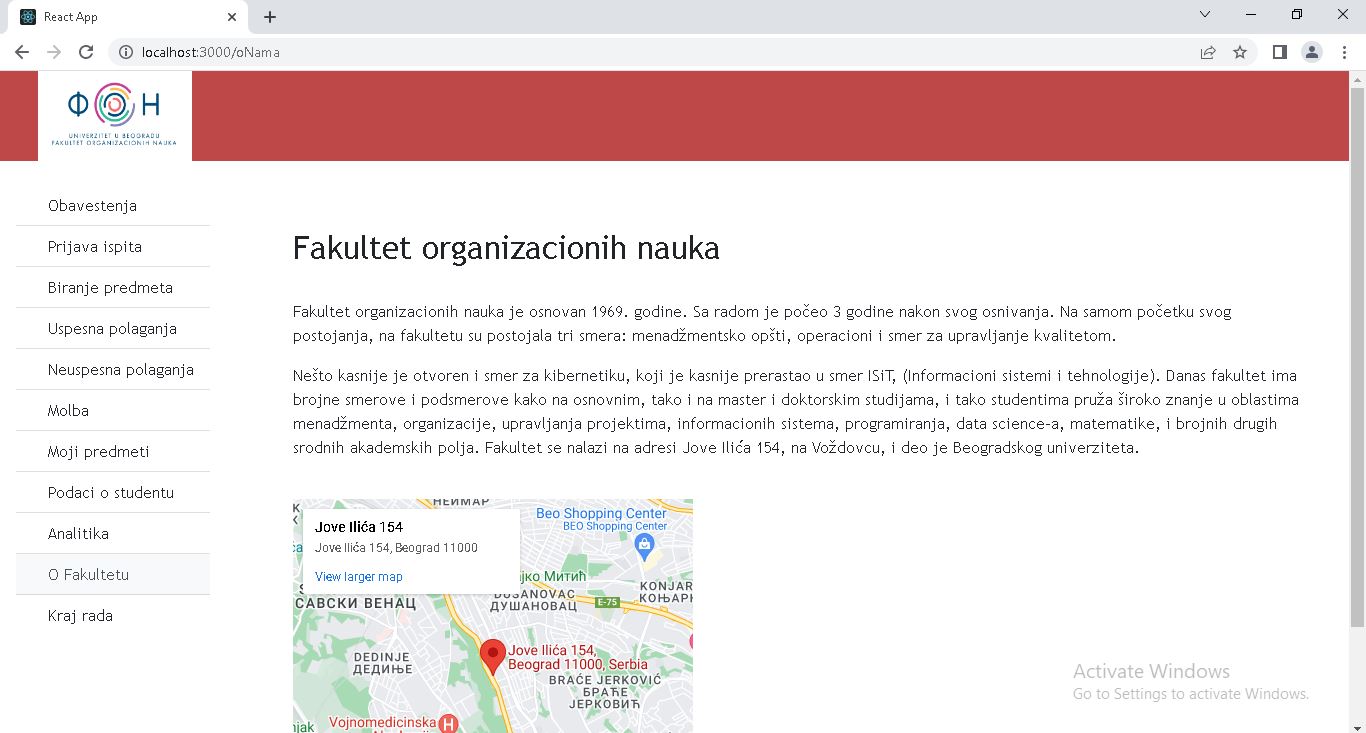
Slika : Podaci o student

Analitika je stranica na kojoj se korišćenjem kružnih dijagrama predstavlja odnos položenih i nepoloženih ispita, kao i procentualni udeo svake od ocena položenih ispita prijavljenog studenta. To izgleda ovako:



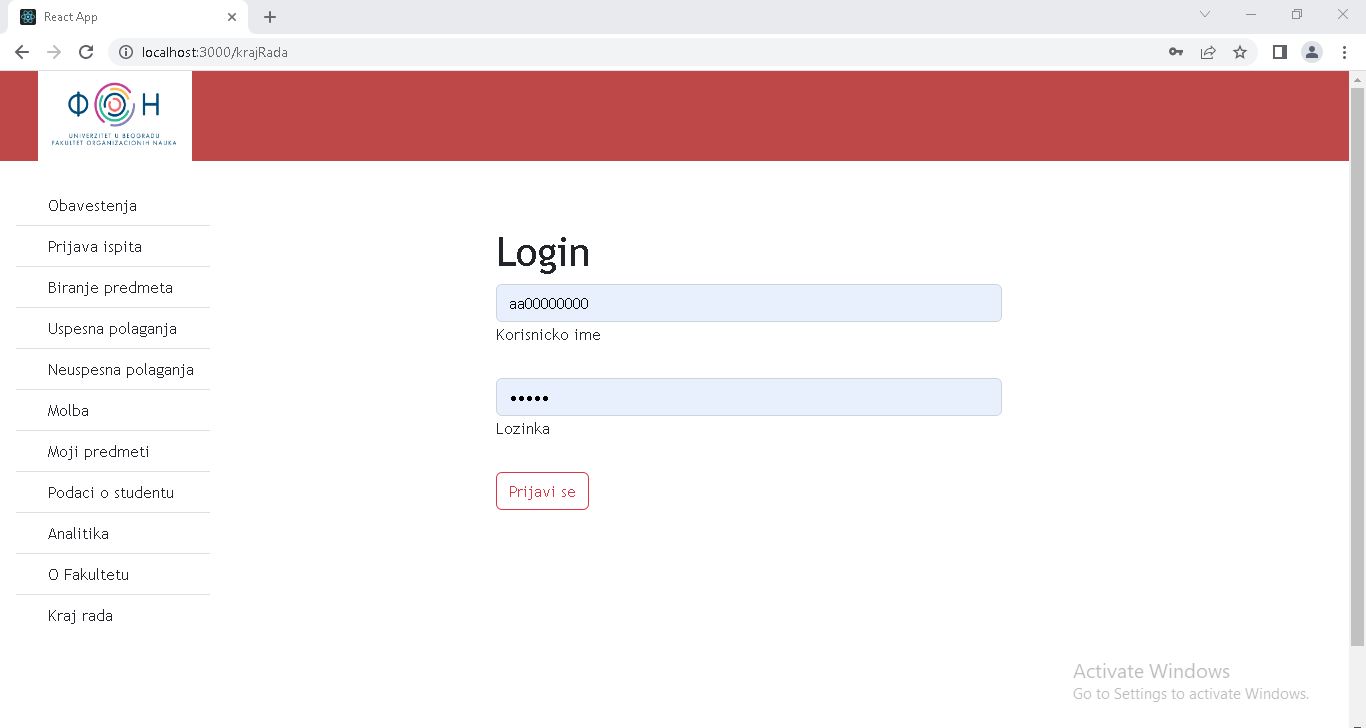
Slika : Analitika studenta

Stranica “O fakultetu” sadrži relevantne podatke o fakultetu Organizacionih nauka, kao i gugl mapu sa lokacijom Fakulteta organizacionih nauka.



Slika : Stranica “O fakultetu”

Ukoliko student klikne na link “Kraj Rada” biće preusmeren na početnu stranicu, ali i izlogovan. U nastavku sledi uputsvo za rad administratora. Prvo je neophodno da se administrator prijavi svojim kredencijalima:



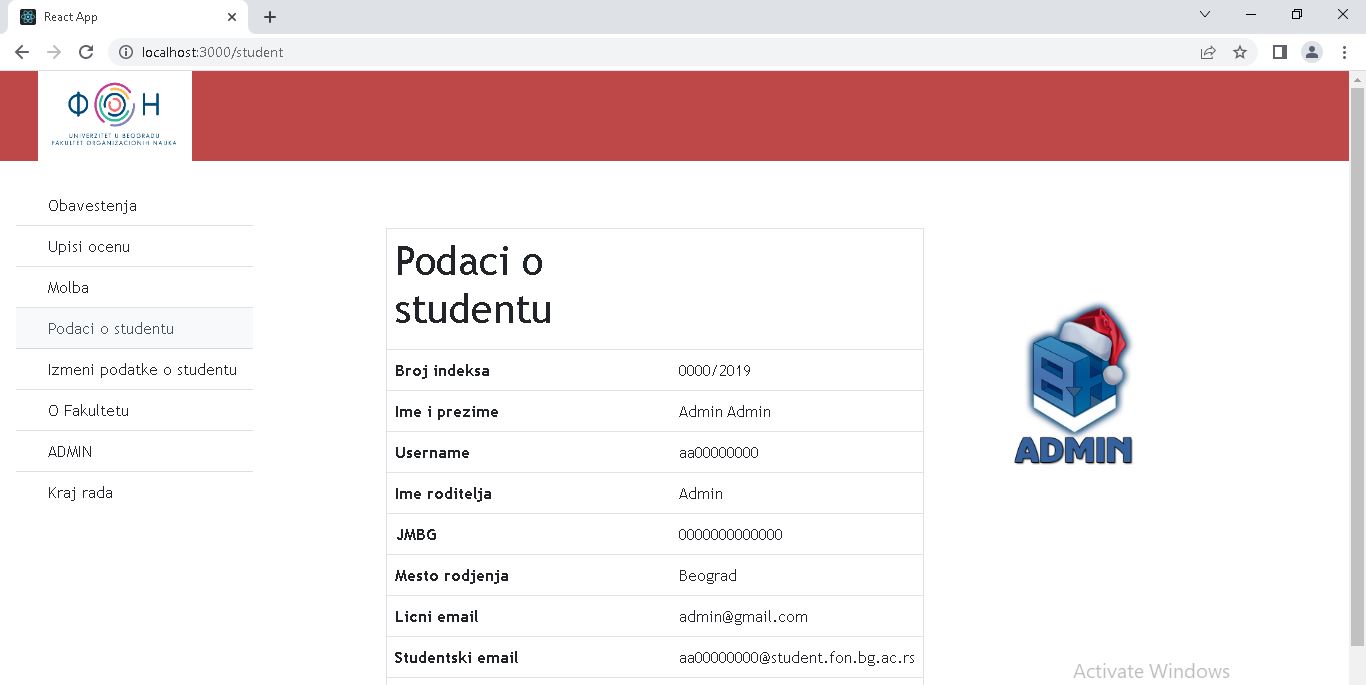
Slika : Kraj rada. Prijava administrator

Nakon prijave, i administratoru se, kao i studentu, prikazuje strana sa Obaveštenjima.



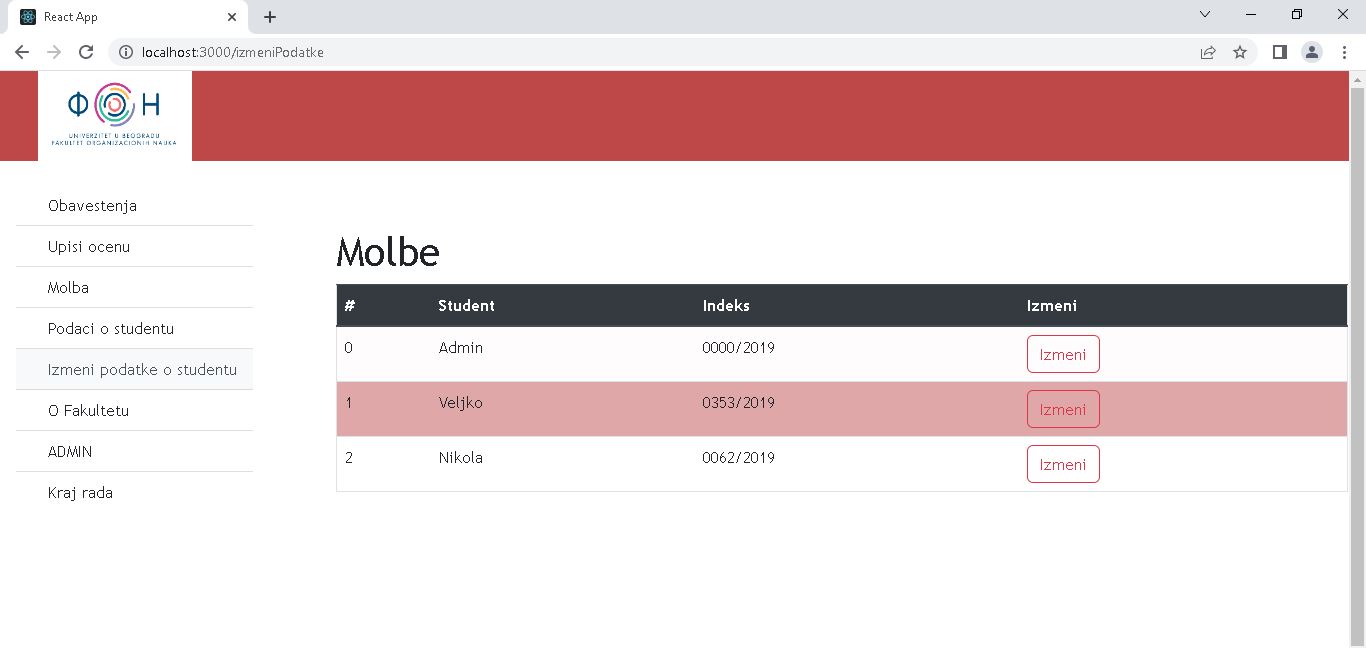
Slika : Početna strana, obaveštenja

Podaci o studentu su jedinstveni jer prikazuju administratora, a ne studenta u pravom smislu te reči.



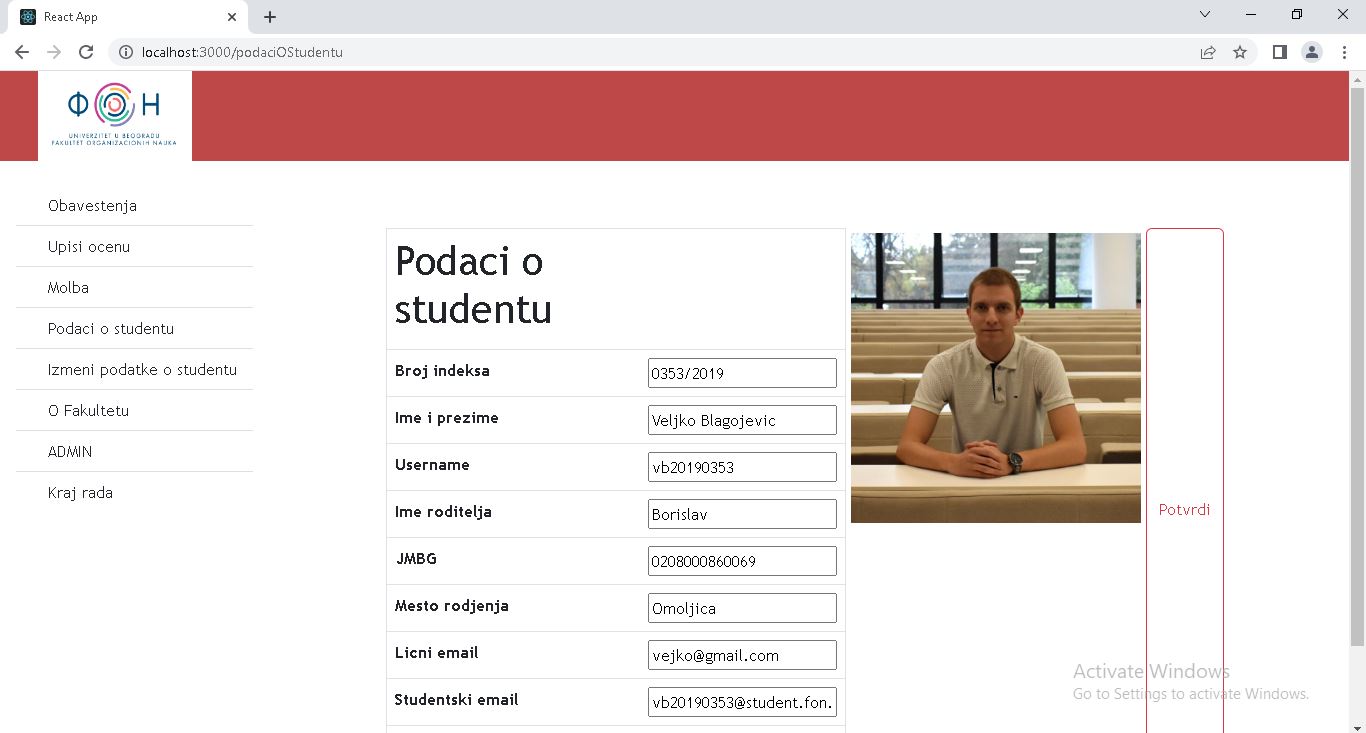
Slika : Podaci o administrator “studentu”

Ukoliko administrator želi da izmeni podatke nekog od studenata, on to vrši u sekciji „Izmeni podatke o studentu“.



Slika : Izmena podataka o studentu

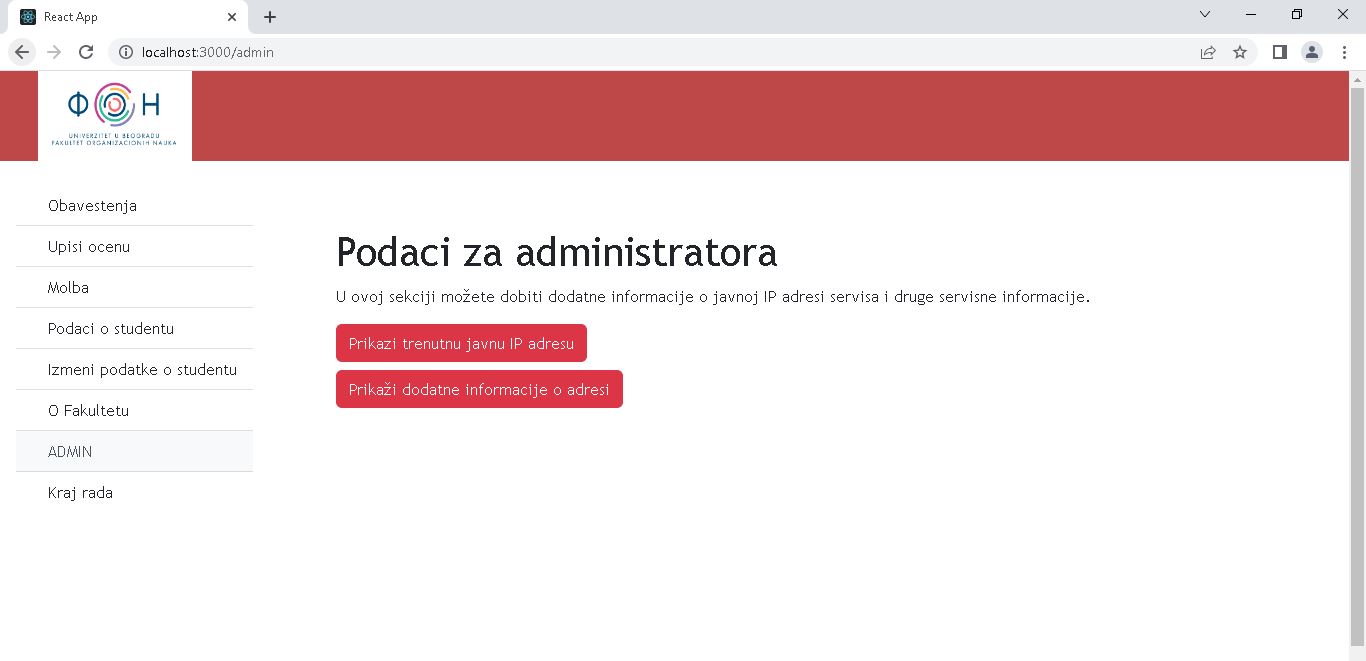
Kada se otvori stranica za izmenu podataka o studentima, administrator može izabrati studenta za izmenu iz tabele. Pritisak na dugme „Izmeni“ ga onda preusmerava na odgovarajuću stranicu.



Slika : Promena podataka studenta

Klikom na dugme „Potvrdi“ se promene o studentu čuvaju u sistemu.

Jedan od najbitnijih delova sajta, iz perspektive administratora, je i stranica “ADMIN”. Dostupna samo administratorima, ova stranica daje detaljne informacije o IP adresi mreže na kojoj je administrator prijavljen. Ovo se odvija u 2 faze.

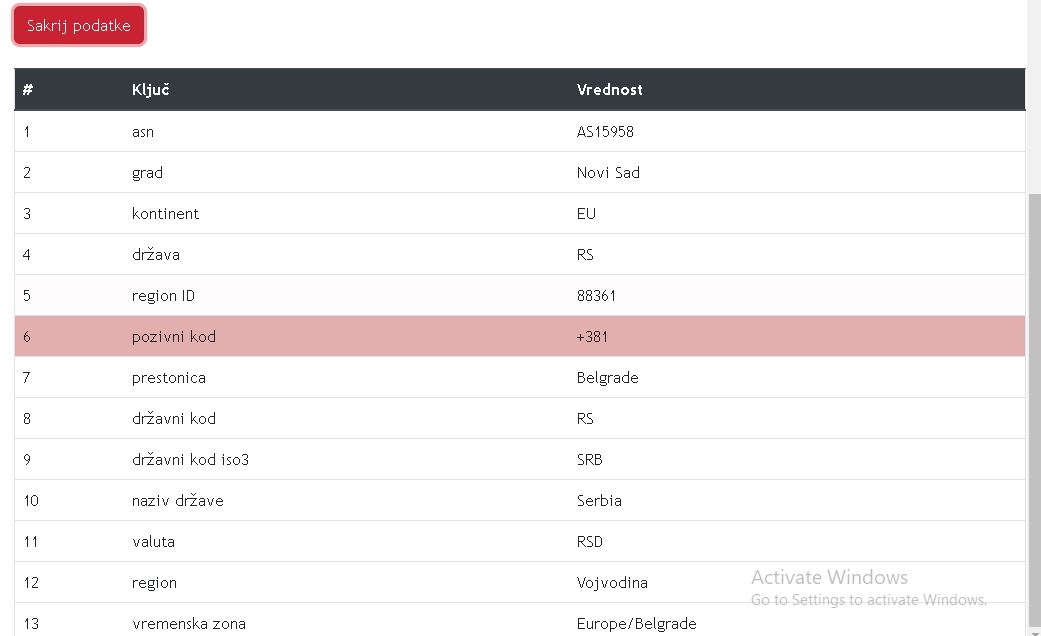


Slika : Podaci za administrator

Prvo dugme na ovoj stranici će samo prikazati javnu IP adresu računara, dok drugo dugme daje dalje relevantne informacije

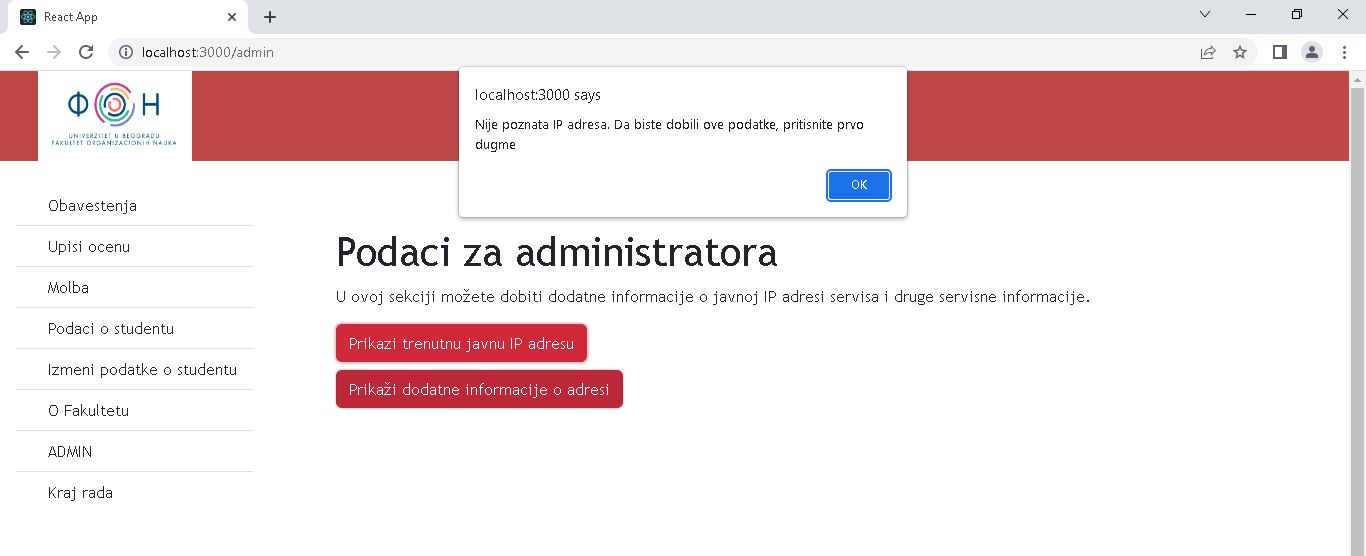


Slika : Prikazana IP adresa administratorskog sistema



Slika : Detaljnije informacije o IP adresi.

Potrebno je da administrator prvo pritisne prvo dugme, a zatim drugo. Ukoliko on to ne učini, i pokuša da prikaže detaljne informacije nepoznate IP adrese, njemu se prikazuje sledeće obaveštenje:



Slika : Greška pri radu administratora

Prikaz reprezentativnih delova koda

Ovaj softverski sistem je napravljen po uzoru na troslojnu arhitekturu, i kao takav sadrži 3 dela. To su frontend, backend i baza podataka. U ovom delu će detaljnije biti opisana prva 2 dela softverskog sistema.

# Backend

Backend deo je implementiran u programskom jeziku Java, u okviru Spring, korišćenjem Intellij razvojnog okruženja.



Slika : Main klasa backend projekta

Izvršenje kao i uvek počinje od main klase. Ovde vidimo da se prilikom mapiranja dozvolja dostavljanje podataka sa porta 3000 frontend aplikacije.

U nastavku slede screenshot-ovi domenskih klasa projekta.



Slika : Klasa Grad



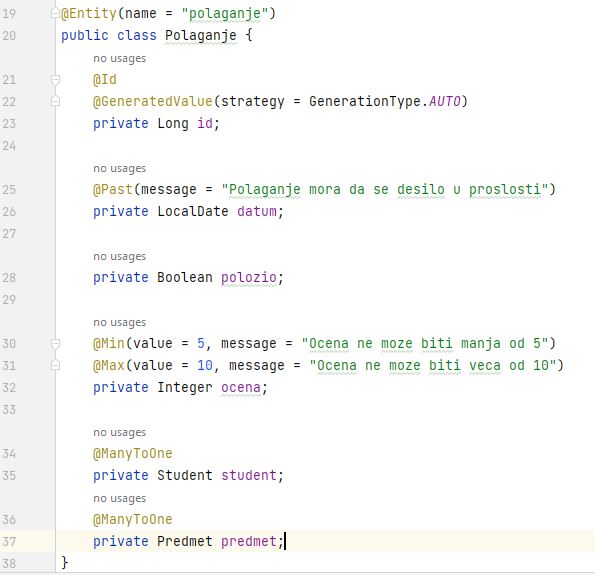
Slika : Klasa Molba



Slika : Klasa NepolozenPredmet



Slika : Klasa Obavestenje



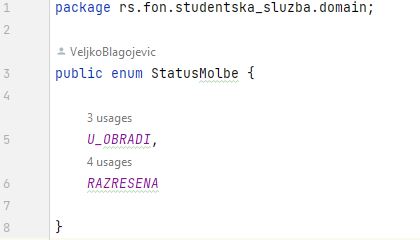
Slika : Klasa Polaganje



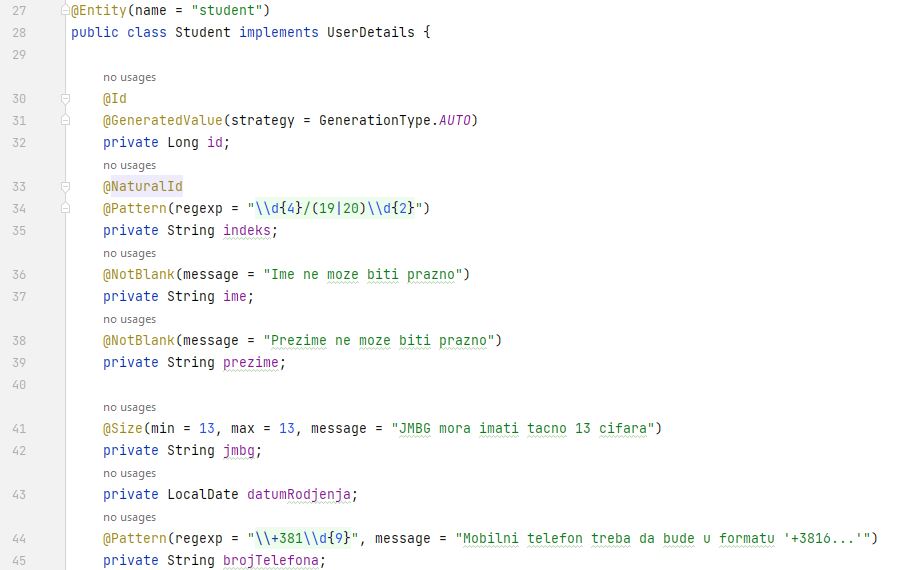
Slika : Klasa Predmet



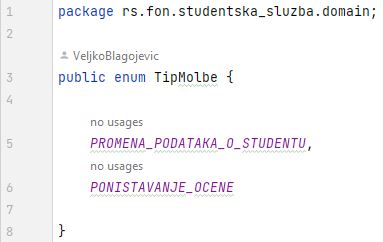
Slika : Enum Role



Slika : Enum StatusMolbe

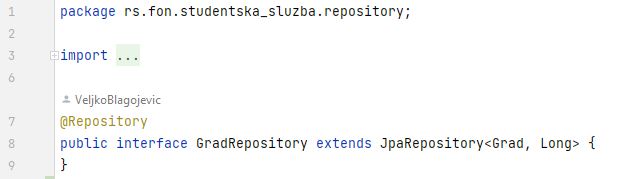


Slika : Klasa Student



Slika : Enum TipMolbe

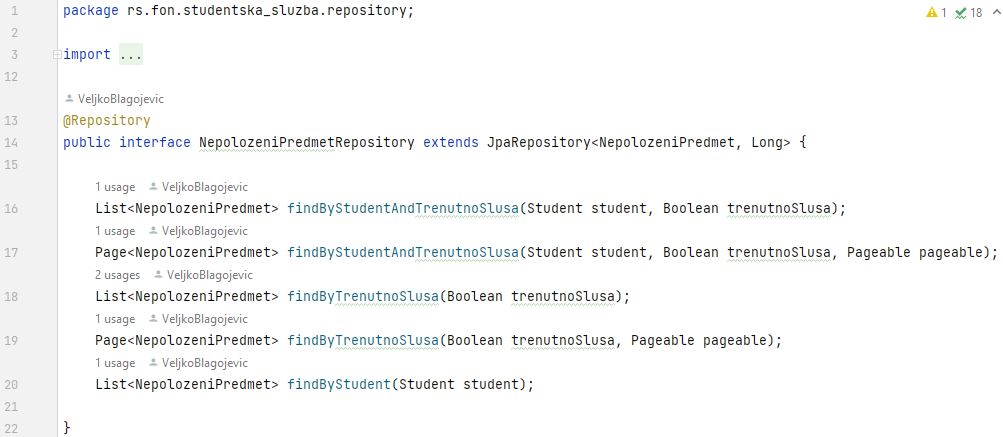
Lista konkretnih repozitorijuma korišćenih u projektu se nalazi u nastavku:



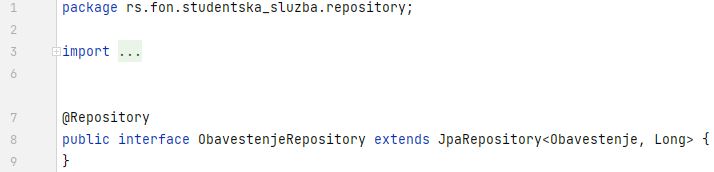
Slika : GradRepository



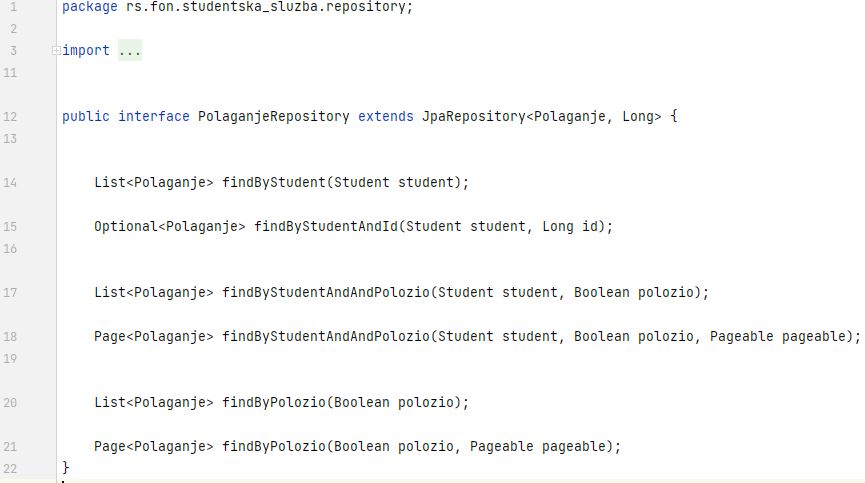
Slika : MolbaRepository



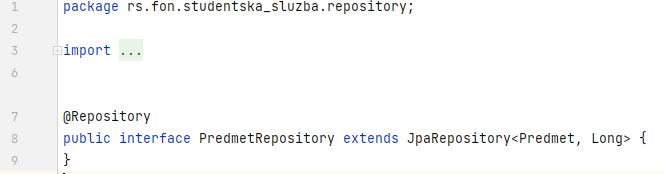
Slika : NepoloženiPredmetRepository



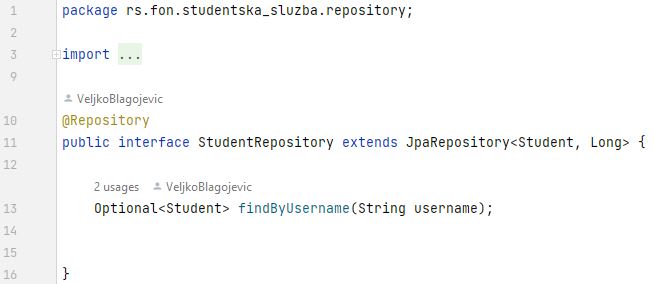
Slika : ObavestenjeRepository



Slika : Polaganje Repository

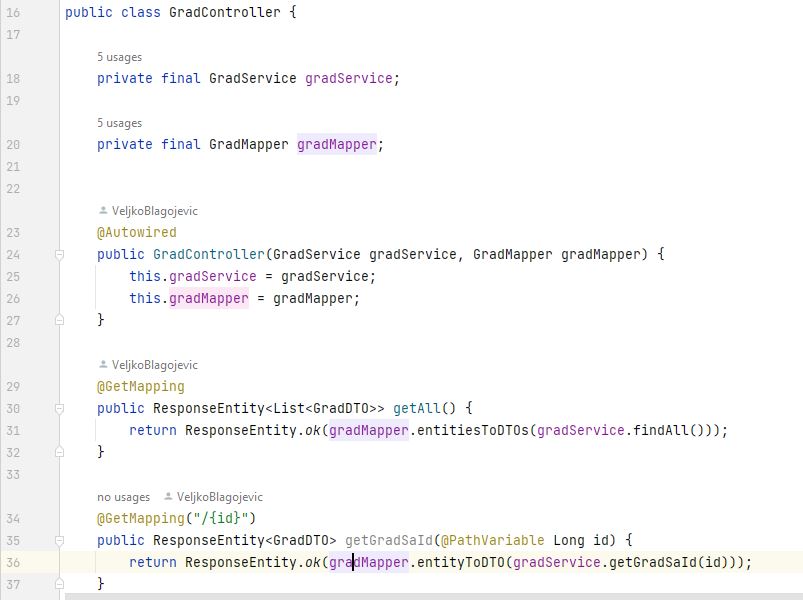


Slika : PredmetRepository



Slika : StudentRepository

Lista konkretnih kontrolera korišćenih u projektu se nalazi u nastavku:



Slika : Klasa GradController



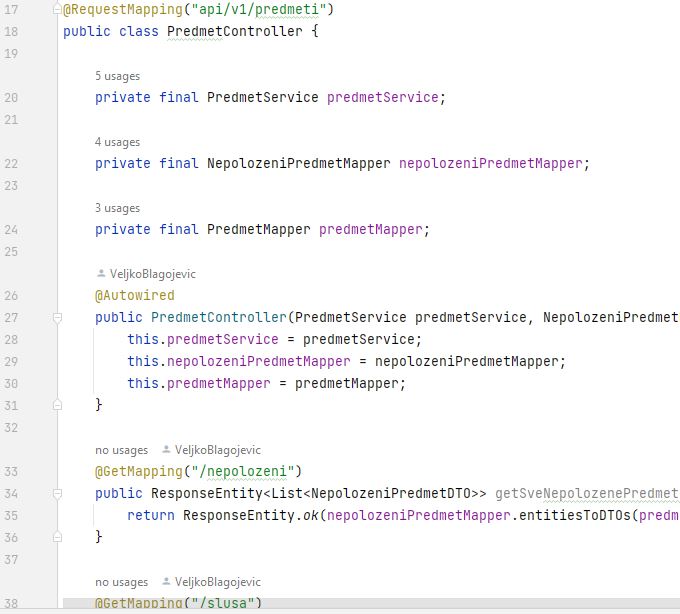
Slika : MolbaController



Slika : ObavestenjeController



Slika : PolaganjeController



Slika : PredmetController



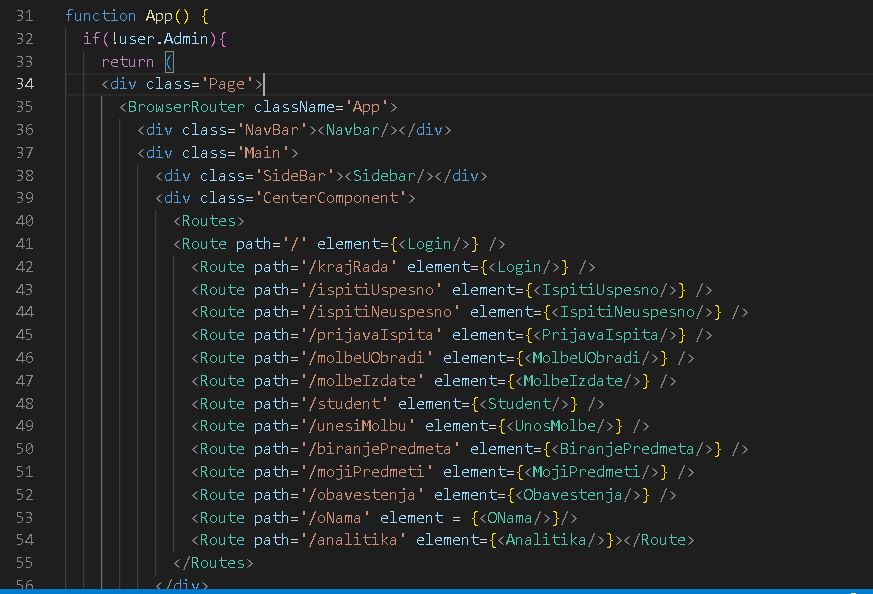
Slika : PrijavaController

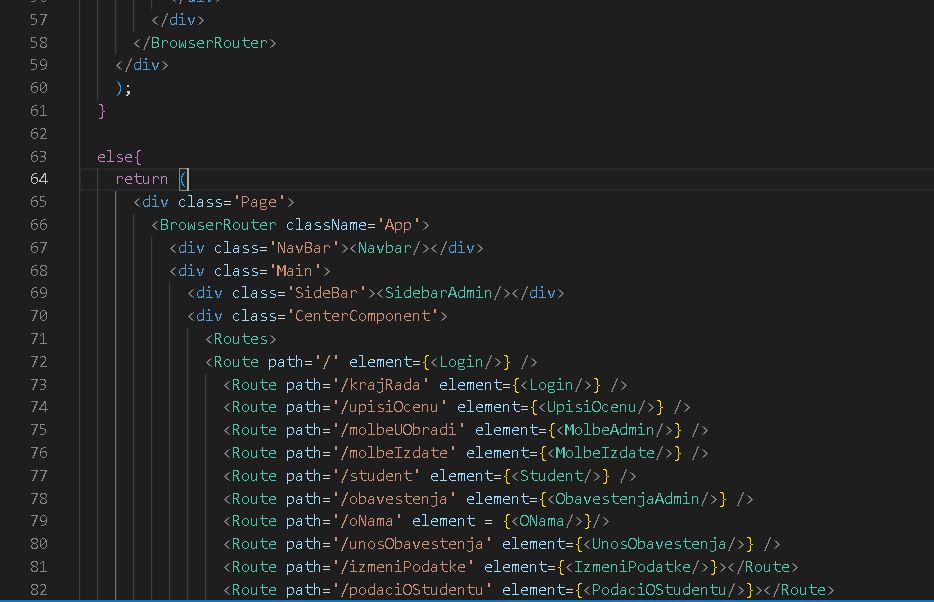


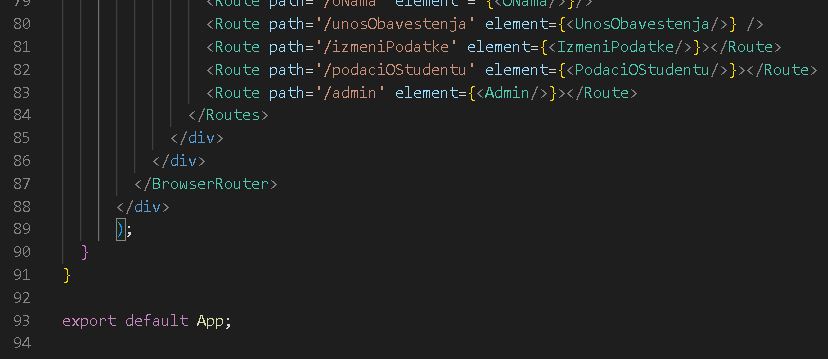
Slika : StudentController

# Frontend

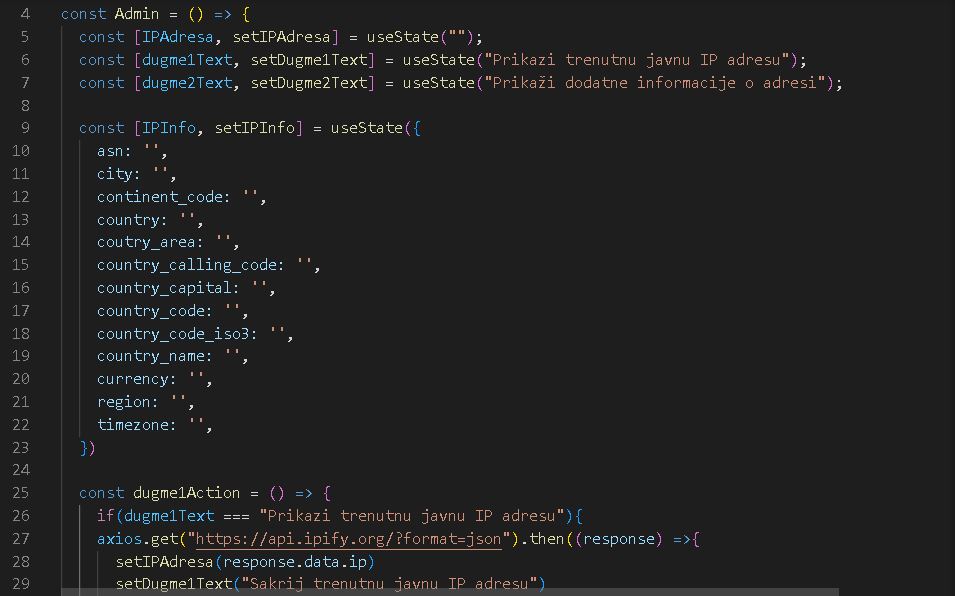
Frontend deo je implementiran u programskom jeziku Javascript, preciznije u biblioteci React ovog programskog jezika, korišćenjem VSCode razvojnog okruženja.



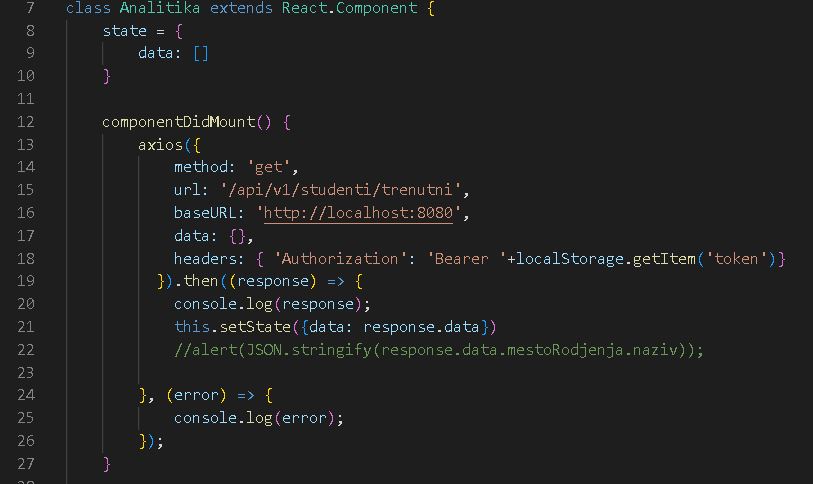




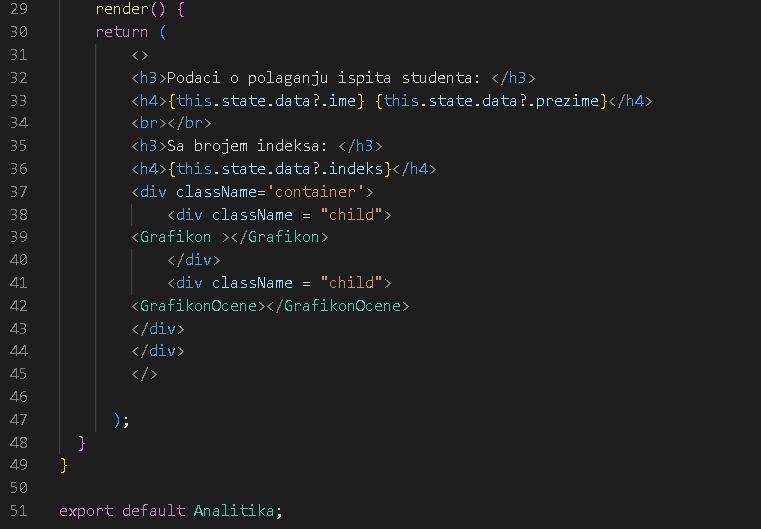
Slika : App.js fajl



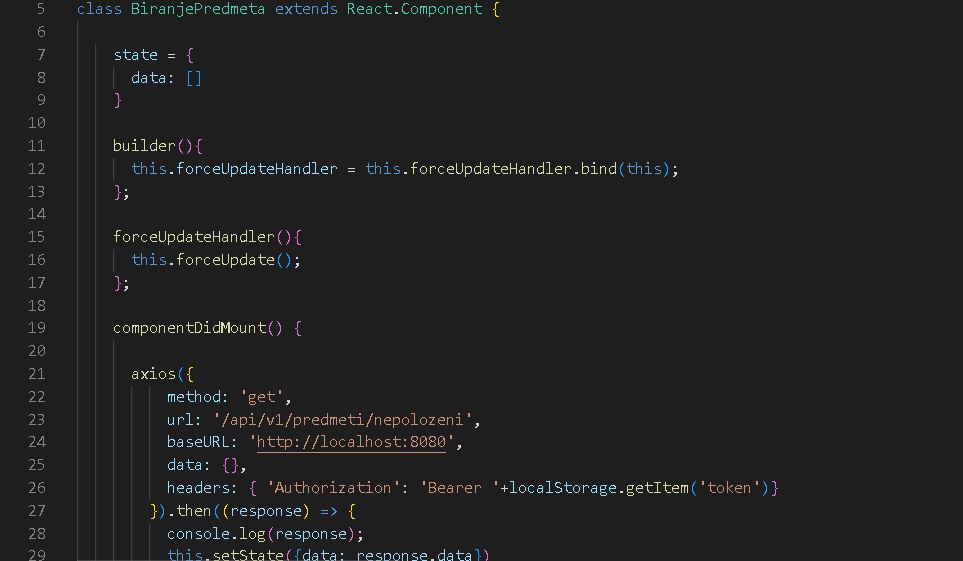
Slika : Admin React komponenta (deo)



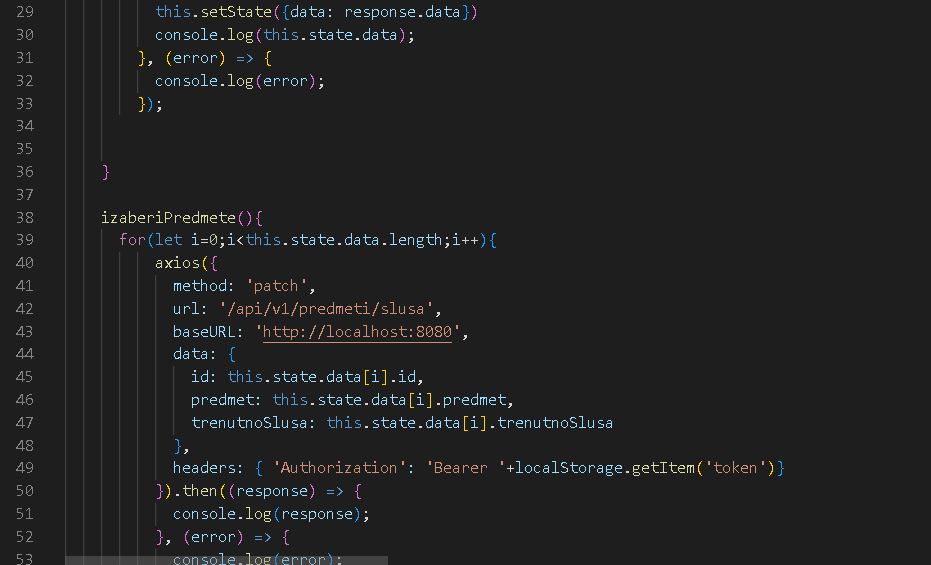
Slika : Komponenta Analitika



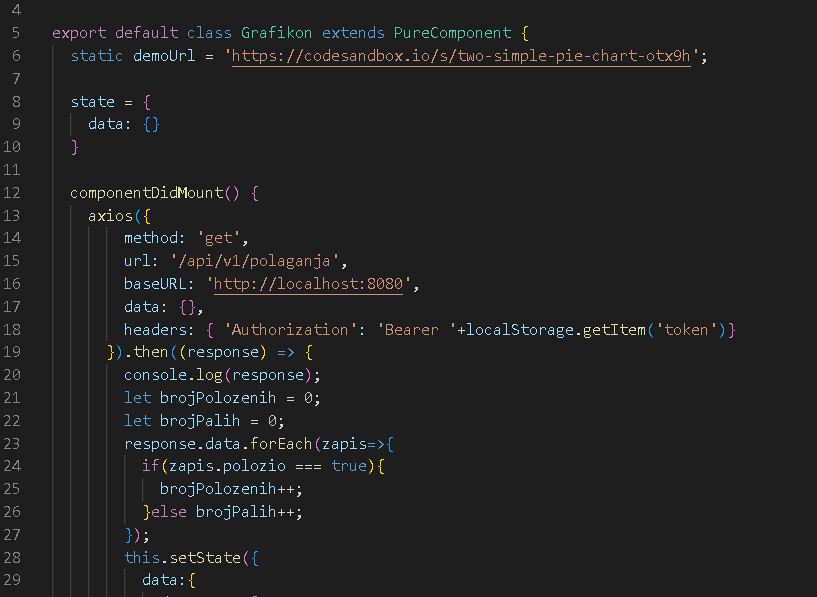
Slika : Komponenta Analitka (nastavak)



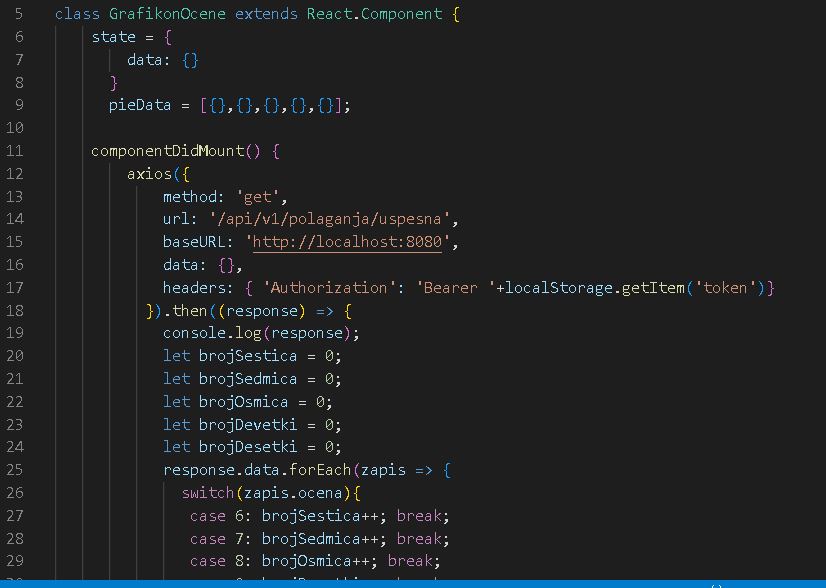
Slika : Komponenta BiranjePredmeta



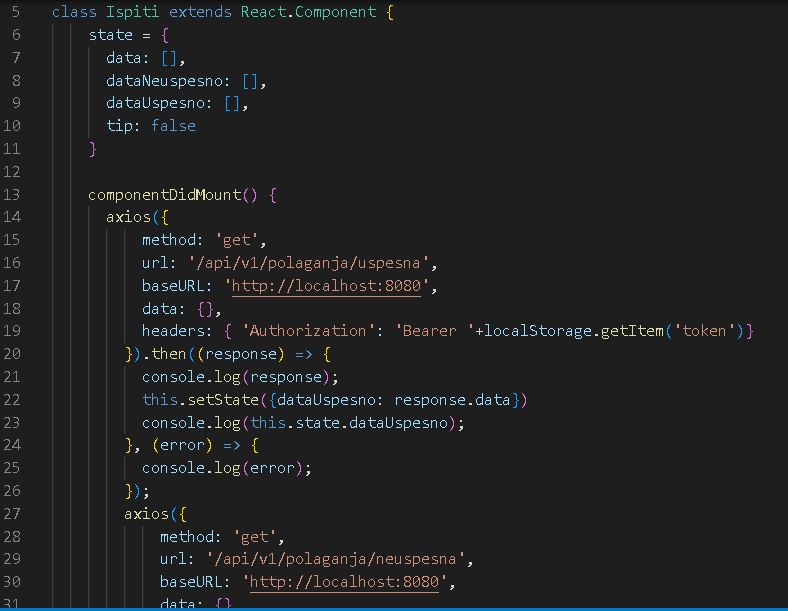
Slika : Komponenta Biranje Predmeta – Nastavak



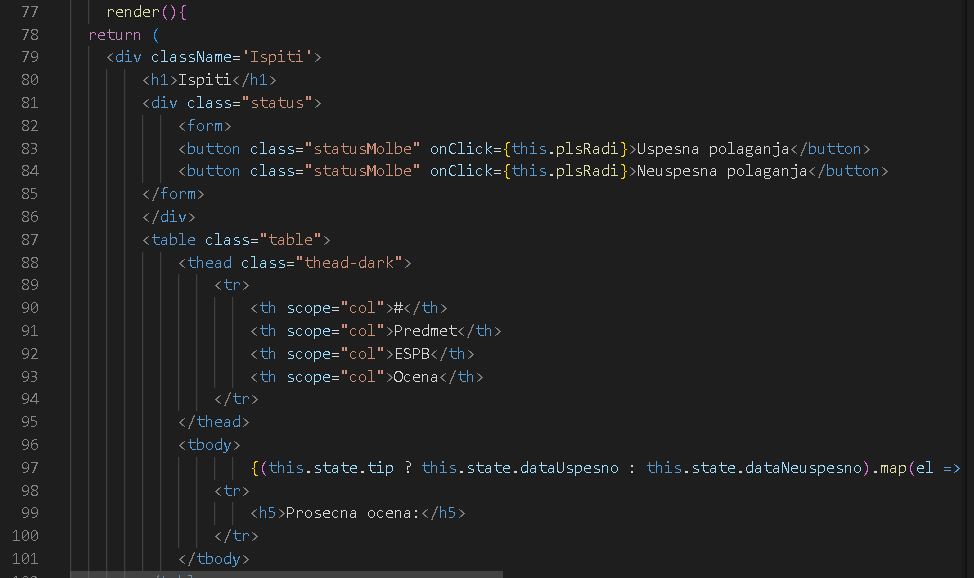
Slika : Komponenta Grafikon (deo)



Slika : Komponenta GrafikonOcene (deo)



Slika : React komponenta Ispiti



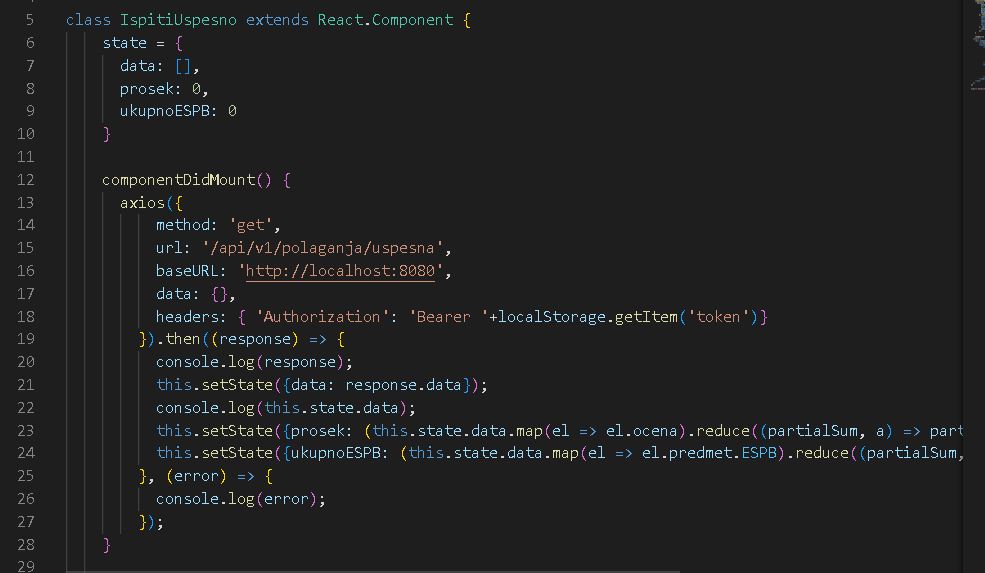
Slika : React komponenta Ispiti, render metoda



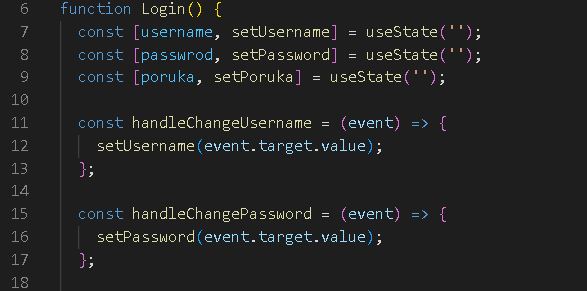




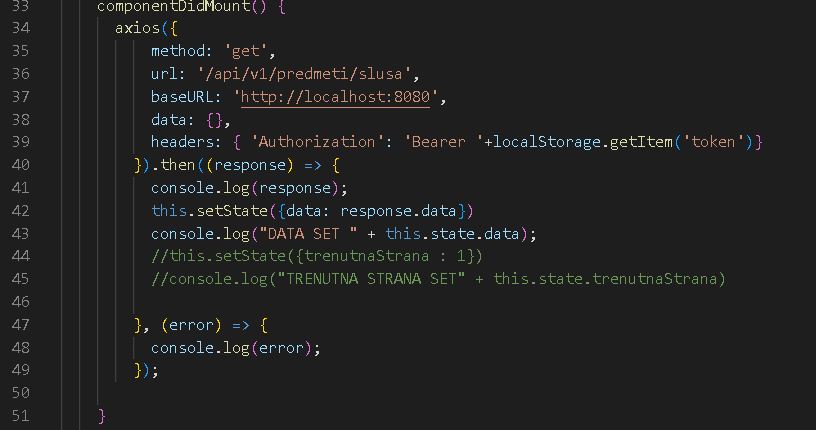
Slika : React komponenta IspitiNeuspesno



Slika : komponenta IspitiUspesno



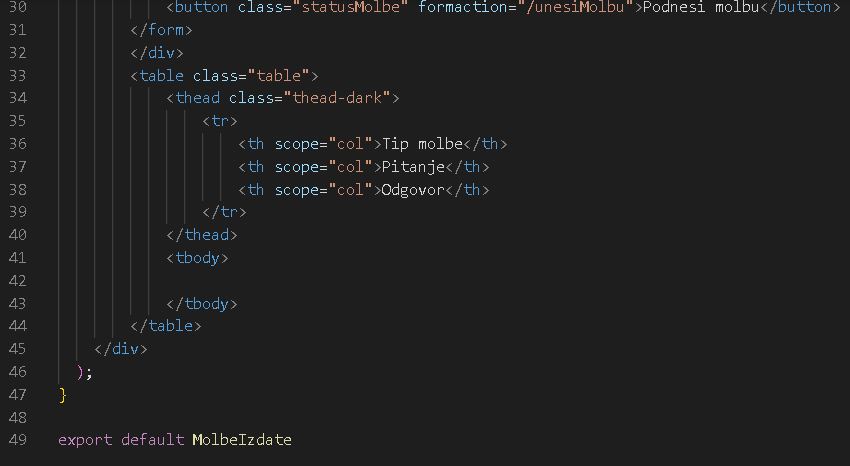
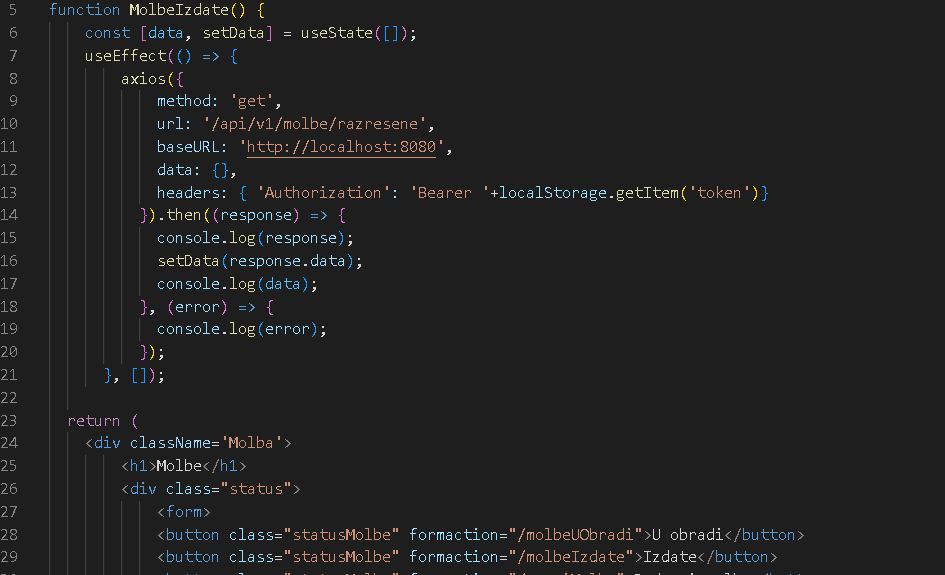
Slika : Funkcija Login



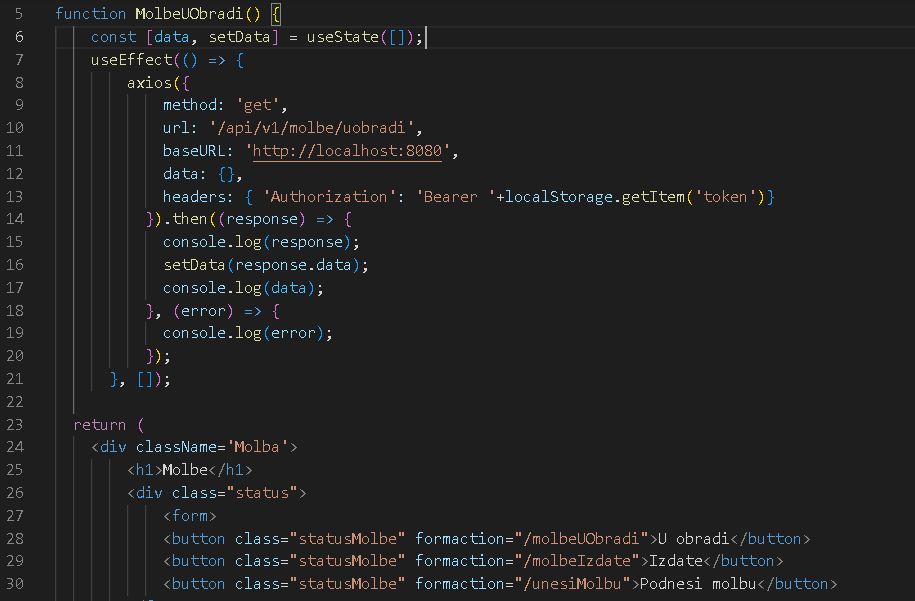
Slika : Komponenta MojiPredmeti (samo axios poziv)



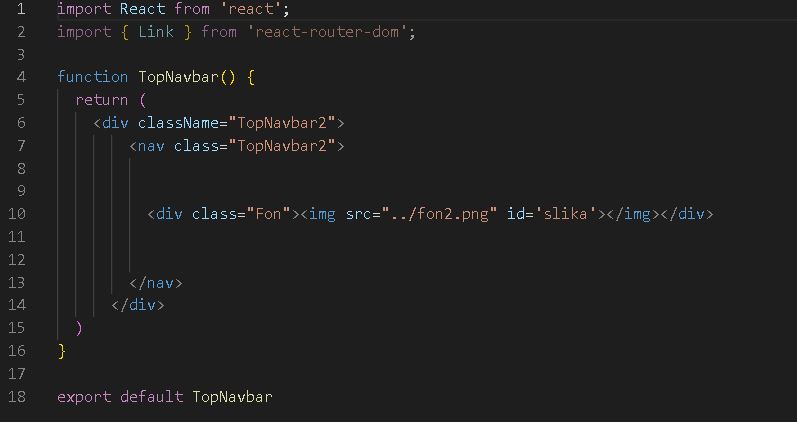
Slika : Komponenta Molba



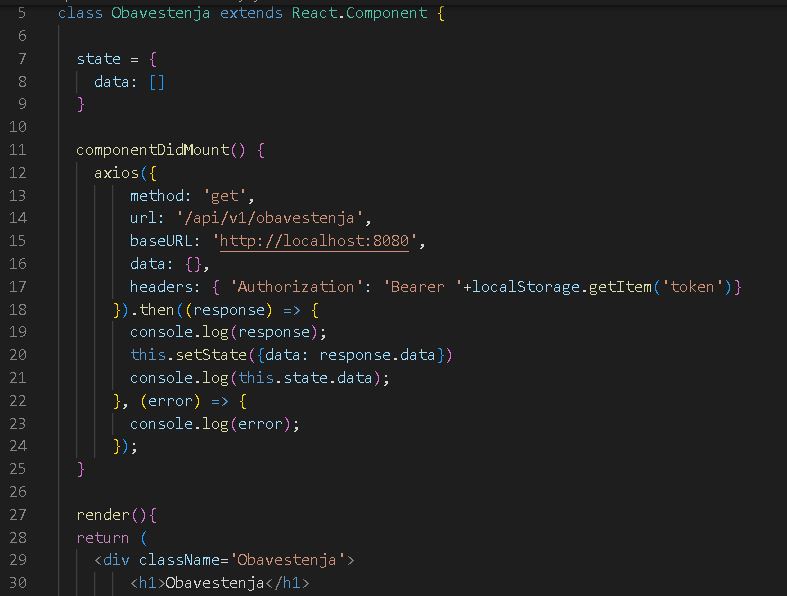
Slika : Komponenta MolbeIzdate



Slika: Komponenta MolbeUObradi



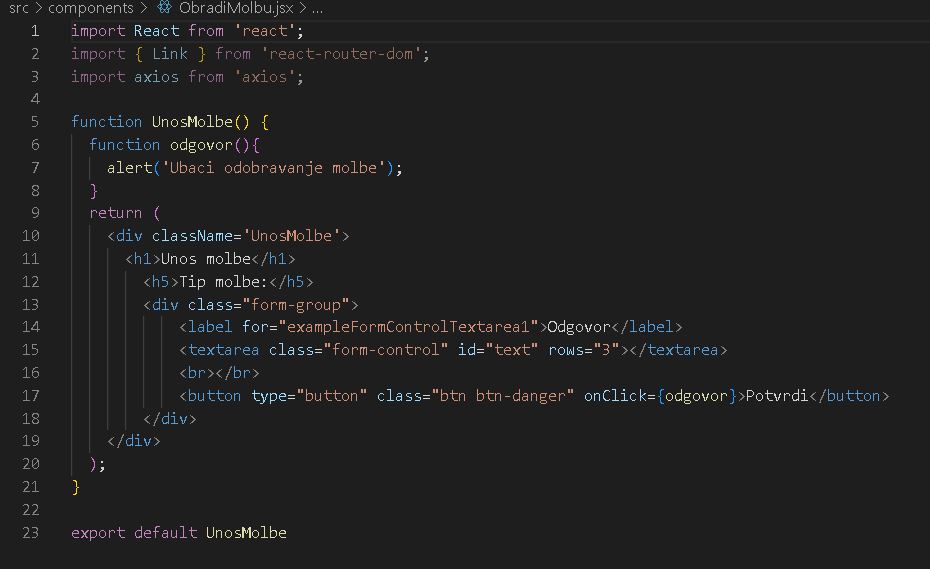
Slika : Komponenta TopNavbar



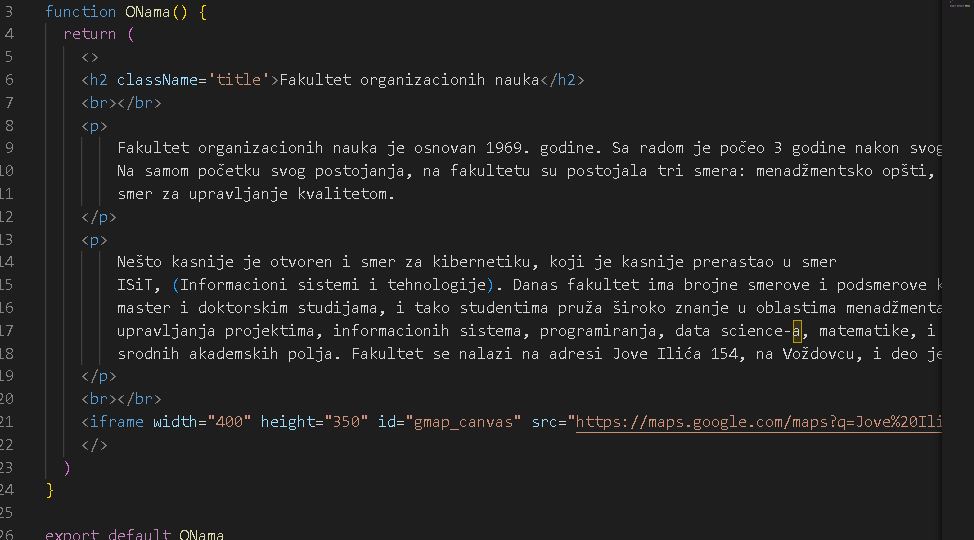
Slika : komponenta Obavestenja



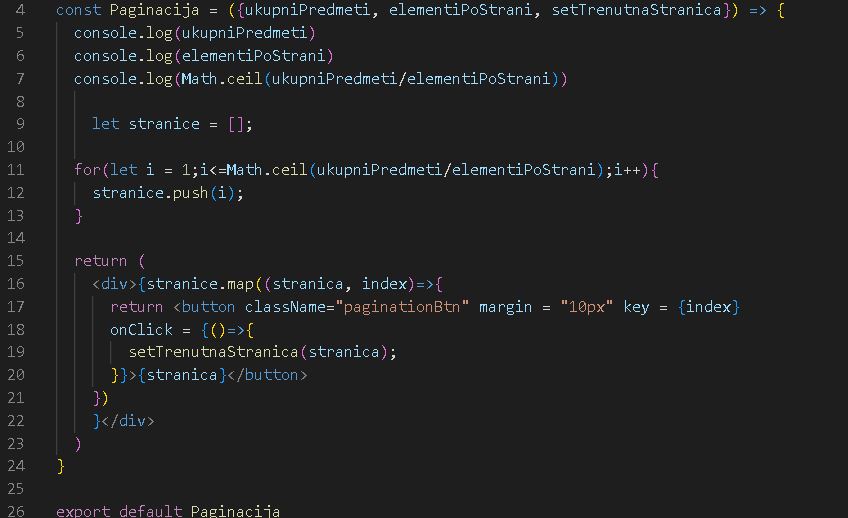
Slika: render metoda komponente Obavestenja



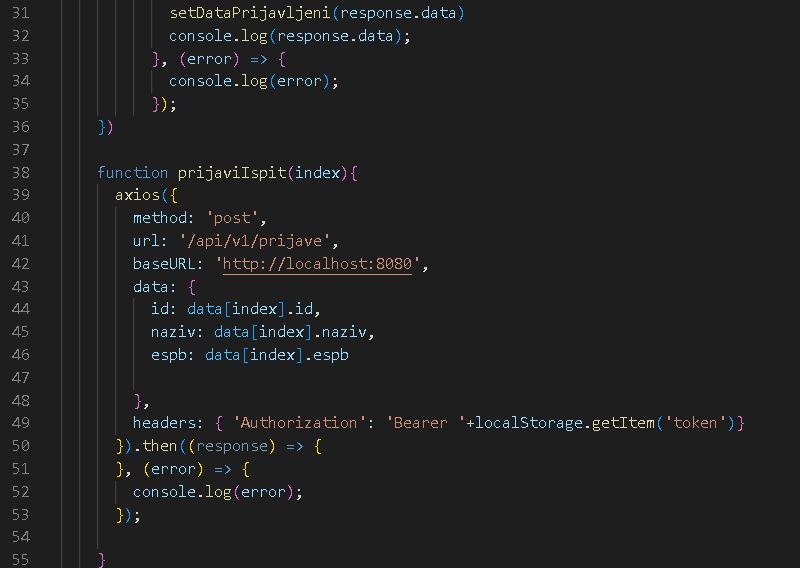
Slika: KomponentaUnosMolbe



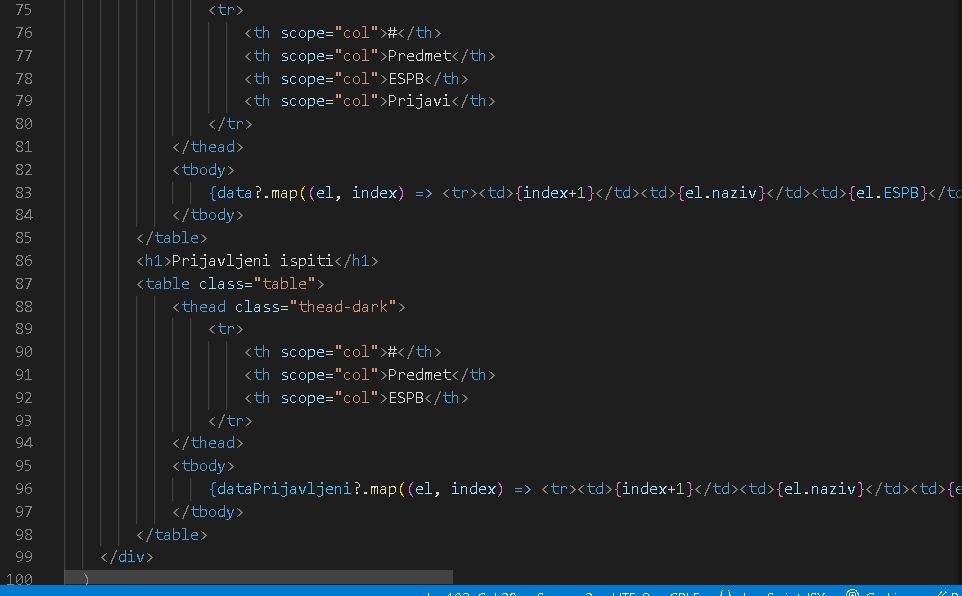
Slika: komponenta ONama



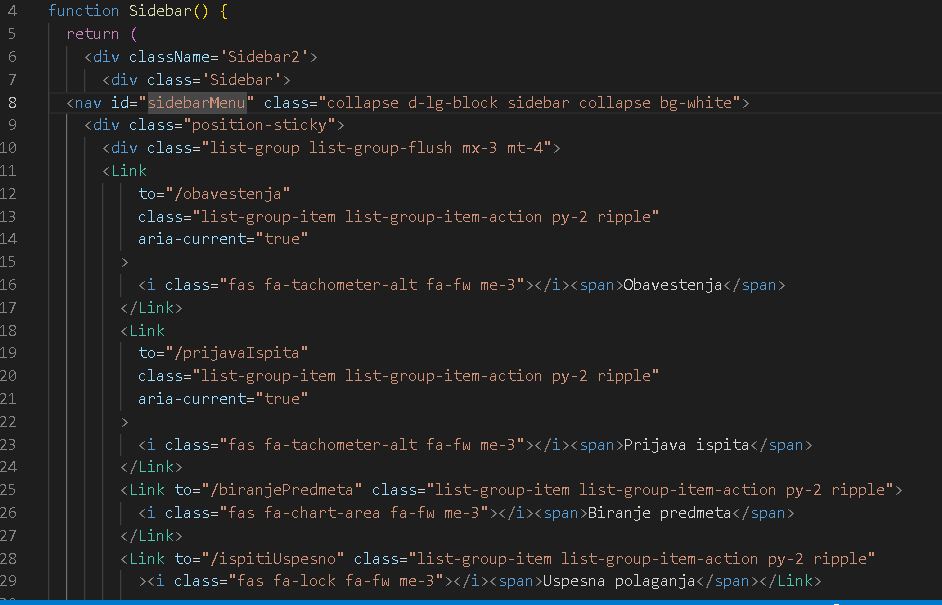
Slika : Paginacija



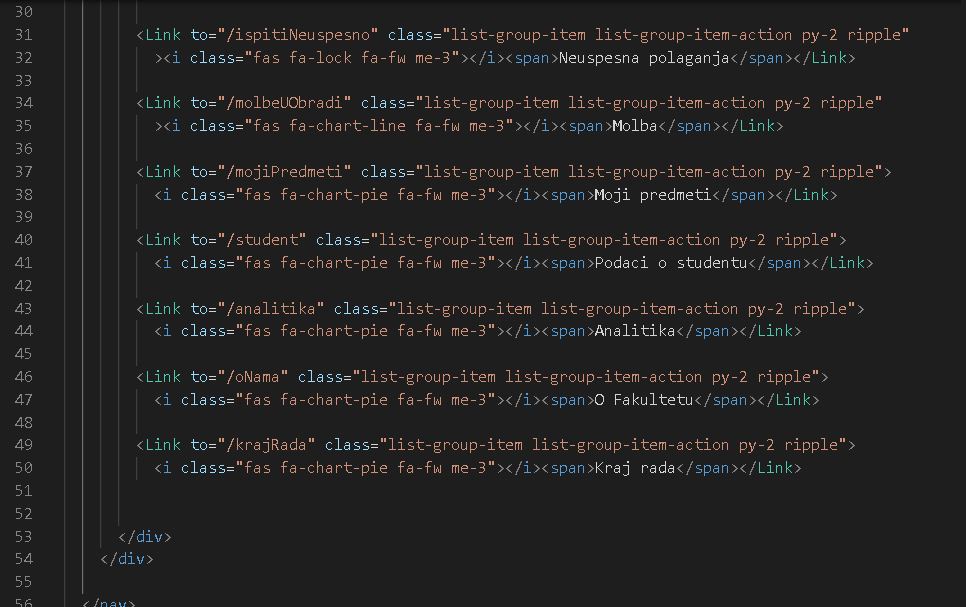
Slika : Komponenta PrijavaIspita (1)



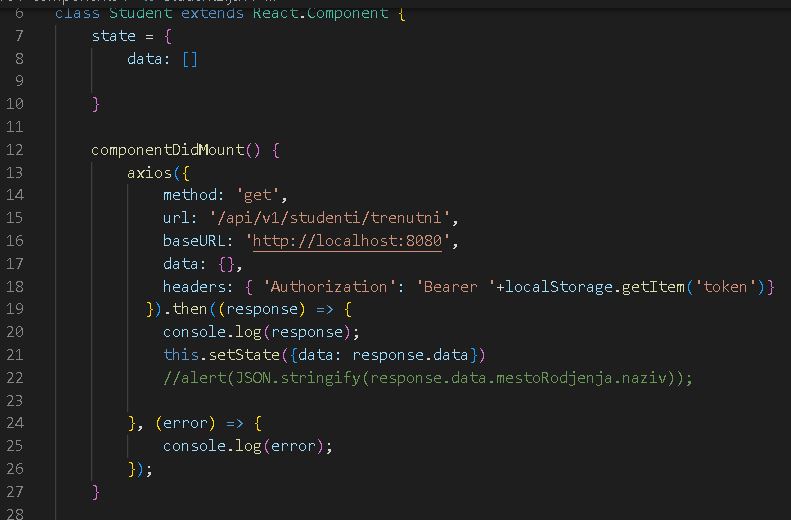
Slika : Komponenta PrijavaIspita (2)



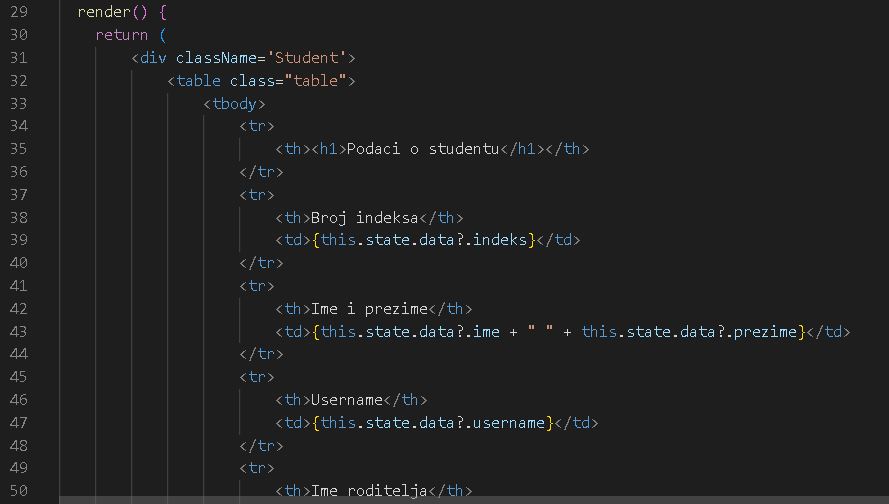
Slika : Komponenta Sidebar (1)



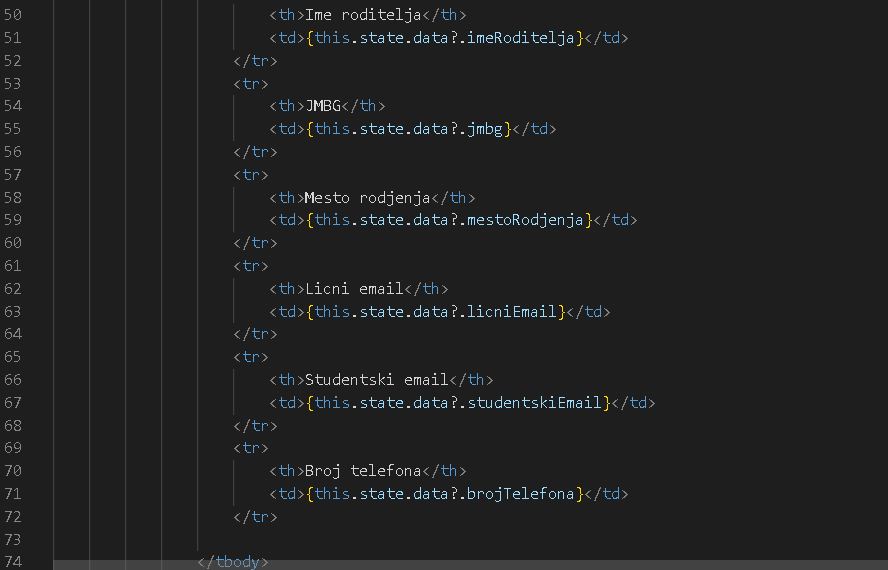
Slika : Komponenta Sidebar (2)



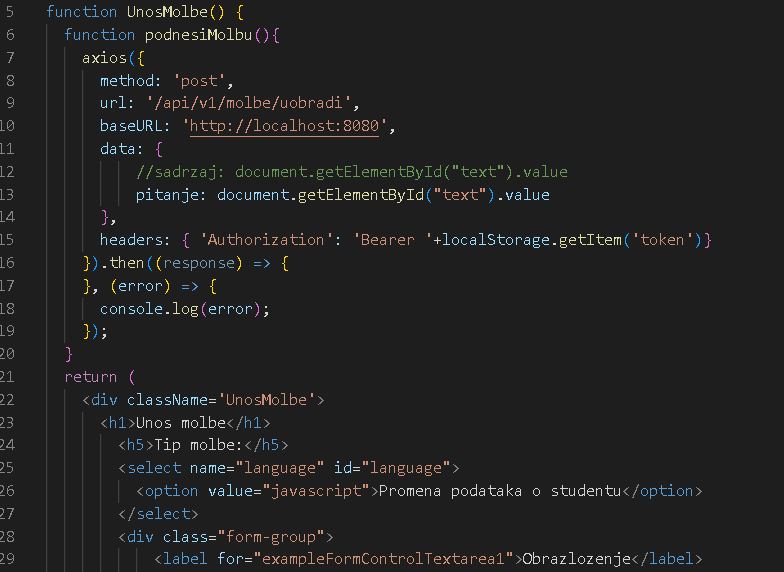
Slika : Komponenta Student (1)



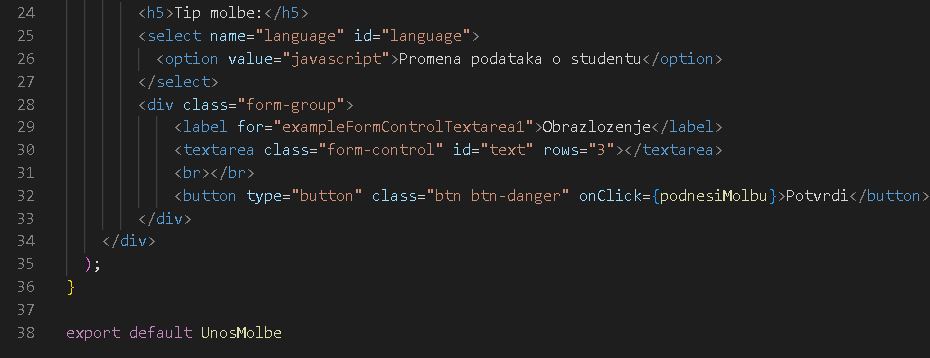
Slika : Komponenta Student (2)



Slika : Komponenta Student (3)



Slika : komponenta UnosMolbe (1)



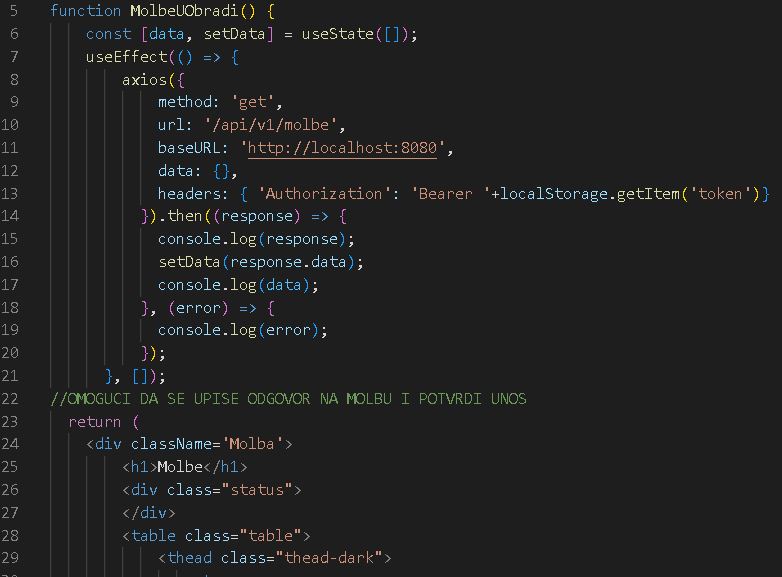
Slika : komponenta UnosMolbe (2)



Slika : Komponenta IzmeniPodatke (1)



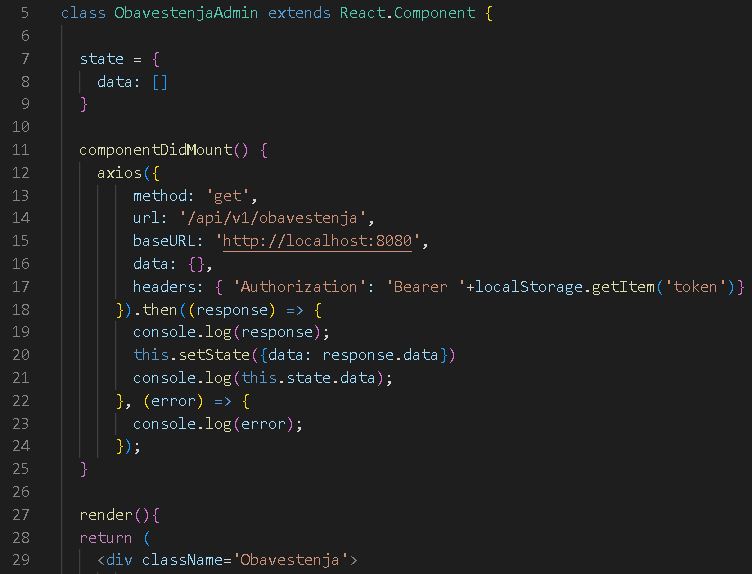
Slika: komponenta IzmeniPodatke



Slika : koponentaMolbeUObradi (Admin, 1)



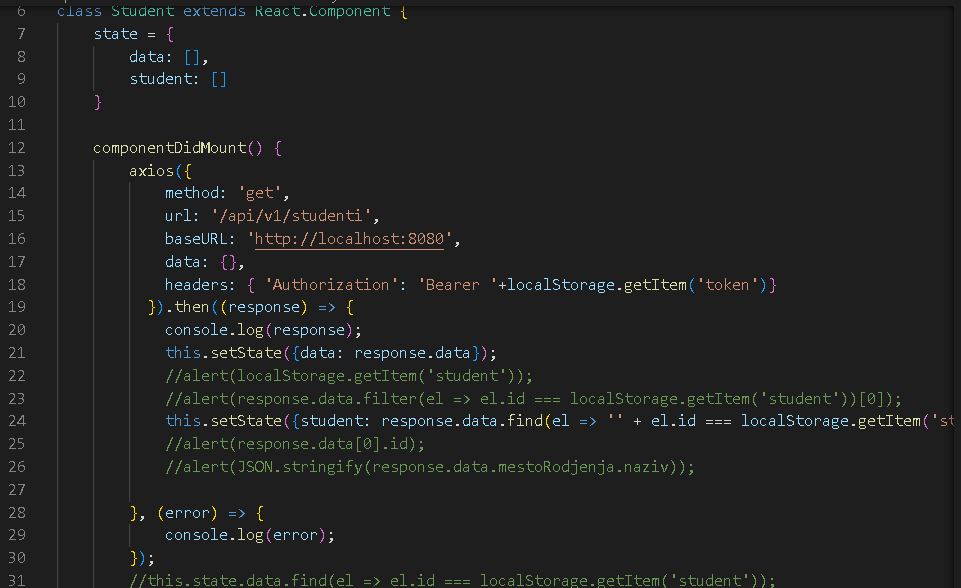
Slika : koponentaMolbeUObradi (Admin, 2)



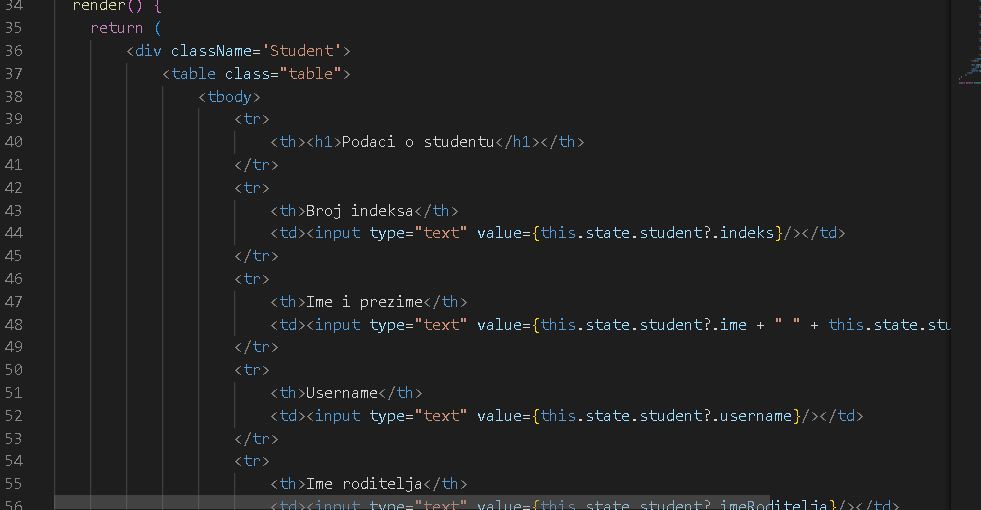
Slika: komponenta Obavestenja (Admin, 1)



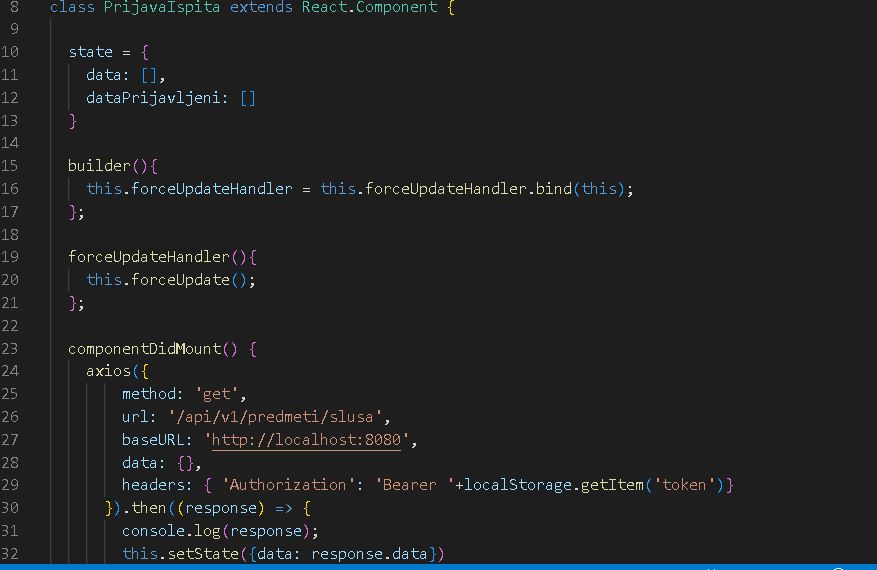
Slika : komponenta Obavestenja (Admin, 2, render metoda)



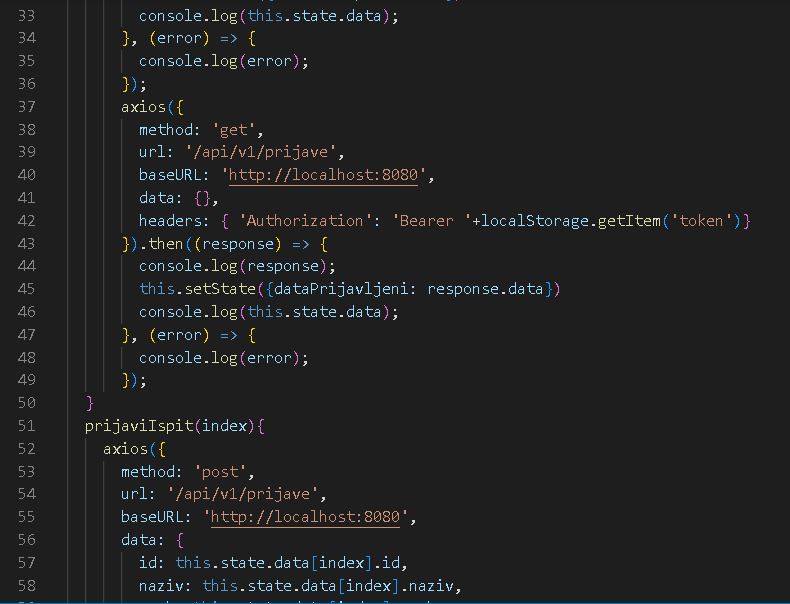
Slika : komponenta Student (1, Admin)



Slika : komponenta Student (2, Admin, deo render metode)



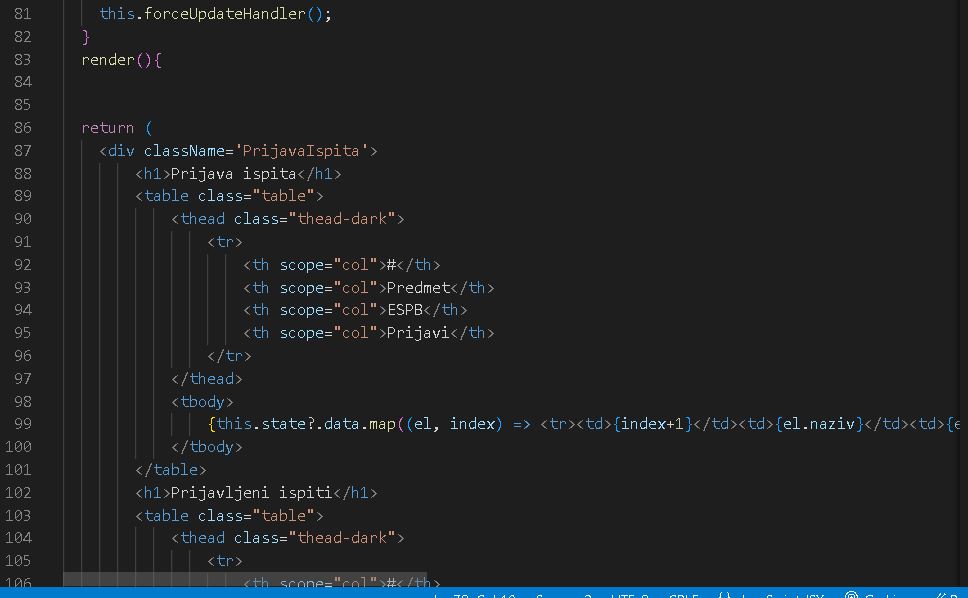
Slika : komponenta PrijavaIspita (1)



Slika : komponenta PrijavaIspita (2)



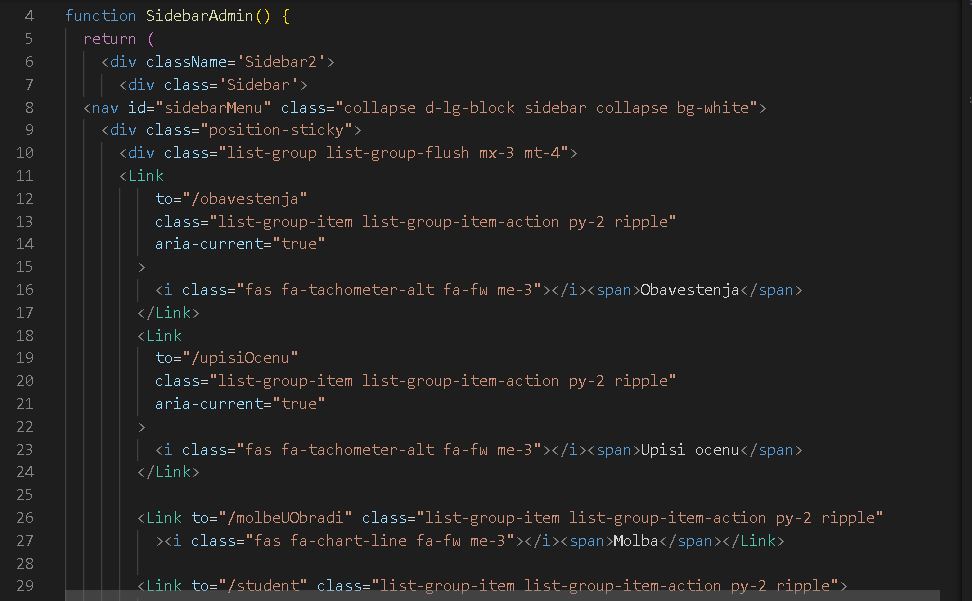
Slika : komponentea PrijavaIspita (3)



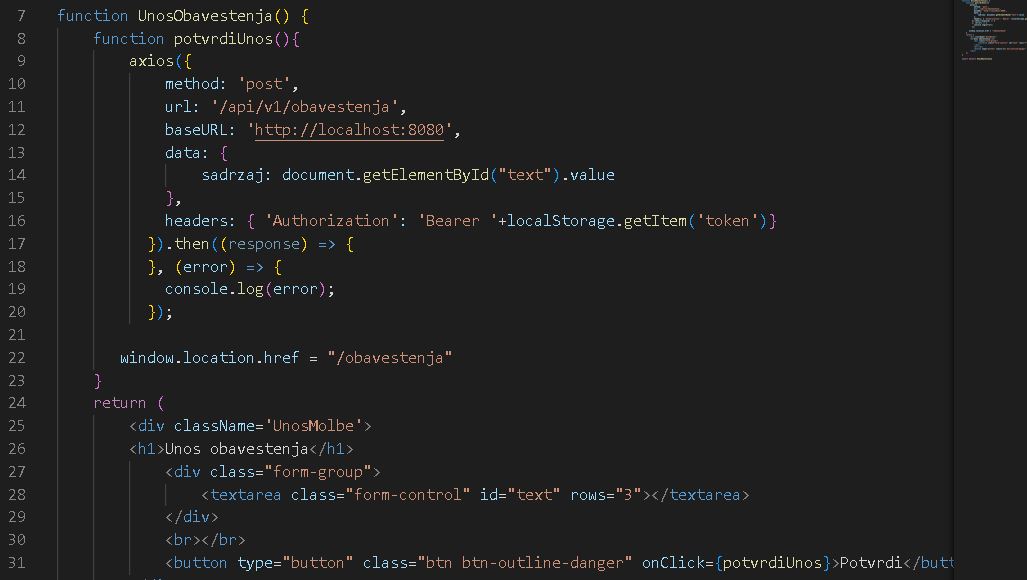
Slika: komponenta PrijavaIspita (4)



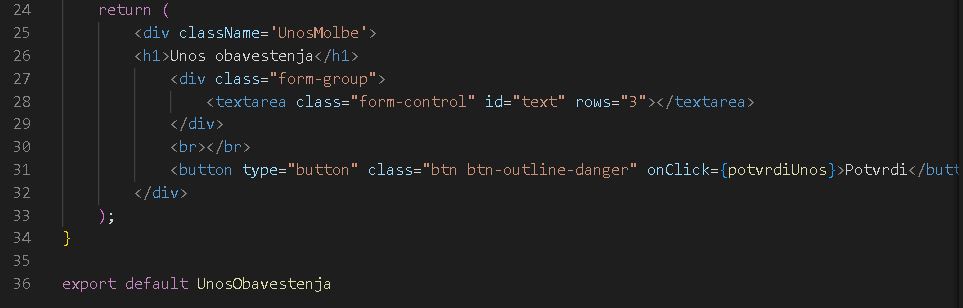
Slika: komponenta PrijavaIspita (5)



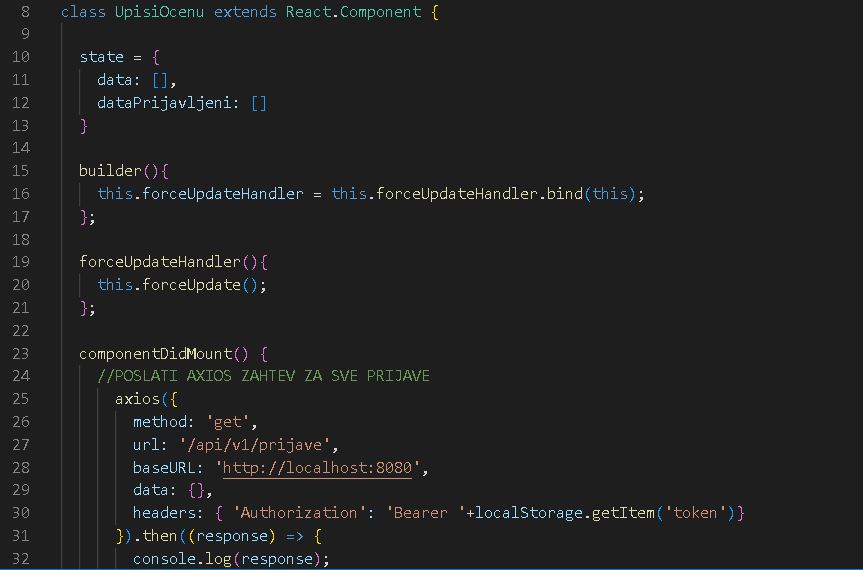
Slika : komponenta SidebarAdmin (deo)



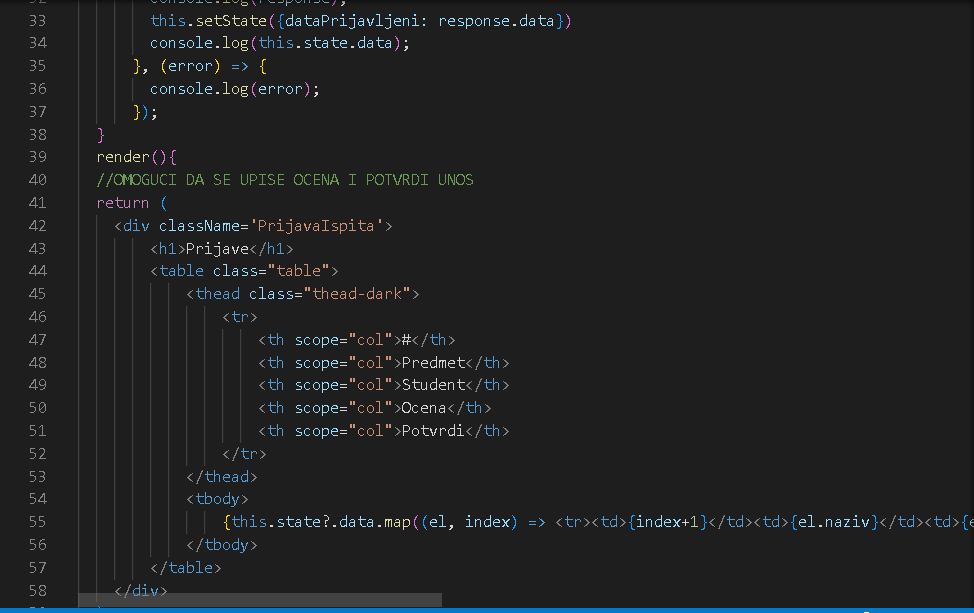
Slika : komponenta UnosObavestenja (1)



Slika: komponenta UnosObavestenja (2)



Slika : Komponenta UpisiOcenu (1)



Slika: Komponenta UpisiOcenu (2)