Aplikacija za podršku interaktivnim online instrukcijama

Pužar, Erik

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:168:196671

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-23**

Repository / Repozitorij:

FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repozitory

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1619

APLIKACIJA ZA PODRŠKU INTERAKTIVNIM ONLINE INSTRUKCIJAMA

Erik Pužar

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1619

APLIKACIJA ZA PODRŠKU INTERAKTIVNIM ONLINE INSTRUKCIJAMA

Erik Pužar

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1619

Pristupnik:	**Erik Pužar (0036544239)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	doc. dr. sc. Tomislav Jagušt

Zadatak: **Aplikacija za podršku interaktivnim online instrukcijama**

Opis zadatka:

U posljednjih nekoliko godina instrukcije su iz klasičnih individualnih poduka uživo evoluirale u svijet virtualne i proširene stvarnosti te u razne sustave za internetsku ili udaljenu komunikaciju, kao što su Zoom, Discord ili Twitch. U sklopu ovog završnog rada potrebno je istražiti mogućnosti postojećih aplikacija za prijenos zvuka i slike uživo (eng. streaming) i načine na koje ih se može integrirati u druge aplikacije ili sustave. Također potrebno je istražiti aplikacije virtualne stvarnosti koje omogućavaju interakciju više korisnika i razmjenu digitalnih sadržaja. Na temelju provedenih istraživanja potrebno je osmisliti i kreirati vlastitu aplikaciju koja će učenicima i studentima omogućiti davanje i primanje instrukcija iz raznih predmeta. Sustav treba davateljima instrukcija omogućiti otvaranje korisničkog računa i kreiranje profila te objavu oglasa. Korisnicima koji su u potrazi za instrukcija sustav će omogućiti pretraživanje ponude instrukcija, odabir termina te pregled profila organizatora. U dogovorenom terminu korisnici se trebaju prijaviti u sustav i obaviti instrukcije kroz implementiranu tehnologiju za komunikaciju, i nakon toga, ako žele, ostaviti recenziju.

Rok za predaju rada: 14. lipnja 2024.

Zahvaljujem mentoru doc. dr. sc. Tomislavu Jaguštu na savjetima i vođenju prilikom izrade ovog rada

Sadržaj

Uvod ... 1
1. Slične aplikacije ... 2
1.1. Aplikacije za instrukcije ... 2
1.1.1. Instrukcije-hr ... 2
1.1.2. Lekcije ... 3
1.2. Aplikacije za komunikaciju ... 4
1.2.1. Discord ... 5
1.2.2. Teams ... 6
1.3. Oglašivačke aplikacije ... 7
1.3.1. Njuškalo ... 7
1.3.2. Facebook Marketplace ... 8
2. Ideja rada i opis dijelova sustava ... 9
3. Model podataka ... 10
3.1. Relacijski model podataka ... 10
3.2. Opis relacijskog modela podataka ... 10
3.3. Opisi entiteta ... 11
4. Korištene tehnologije i alati ... 14
4.1. React.js ... 16
4.2 MUI ... 17
4.3 Flask ... 18
4.4 SQLAlchemy i Elephant DB ... 19
4.5 PostgreSQL ... 20
4.6 Agora.io ... 20
5 Pregled rada aplikacije ... 23
6 Daljnji razvoj ... 30
Zaključak ... 31
Literatura ... 32
Sažetak ... 34
Summary ... 35

Uvod

Instrukcije su važan dio obrazovanja velikog broja učenika i studenata današnjice. Obrazovanje se sve više razvija i napreduje, a za učenike i studente se postavljaju puno veći pragovi znanja. Rastom pragova raste i potražnja za dodatnim satovima usavršavanja gradiva „instrukcijama“ .

Unatoč sve većem razvoju školstva, obrazovanja i tehnologije davatelji instrukcija se i dalje koriste „zastarjelim metodama“ (npr. lijepljenje letaka, instrukcije oči u oči itd...).

Kako bi se unaprijedilo ovo područje kreirao sam interaktivni i jednostavni sustav za podršku interaktivnim online instrukcijama, koji će instruktorima olakšati organizaciju, reklamaciju svojih usluga i omogućiti rad od kuće, a učenicima i studentima olakšati pretragu potrebne pomoći i također omogućiti učenje od kuće.

Aplikacija je jednostavna za korištenje, te specifična unatoč velikom broju sličnih aplikacija. Prilikom prijave korisnik može odabrati u koje svrhe će koristiti aplikaciju: Student (uloga za učenike i studente koji su u potrazi za instrukcijama) i Instructor (uloga za korisnike koji žele davati instrukcije). Uloga Student korisnicima omogućava pregled svih termina, filtraciju istih i prijavu na odabrane termine, korisnici mogu posjetiti profile organizatora termina i ostaviti recenziju i omogućava im ulazak u videopozive u kojima mogu pratiti obradu gradiva. Uloga Instructor korisnicima omogućava kreiranje i uređivanje termina, pregled i filtraciju svih termina, pregled vlastite ukupne recenzije i kreiranje videopoziva za odabrani termin instrukcija.

U ovome radu opisuje se izrada sustava za podršku interaktivnim online instrukcijama. Prikazat će se aplikacije sa sličnim obilježjima, korištene tehnologije i alati, arhitektura, model podataka, te sustav u cijelosti (korisničko sučelje i pozadinski servisi).

1. Slične aplikacije

Postoje brojne aplikacije namijenjene instrukcijama (detaljnije u poglavlju 1.1), a kreirana aplikacija je kombinacija dviju vrsta aplikacija: Aplikacija za komunikaciju (poglavlje 1.2) i Oglašivačkih aplikacija (poglavlje 1.3).

1.1. Aplikacije za instrukcije

Aplikacije za instrukcije koriste se za uspostavu komunikacije između instruktora i studenata, sadrže dijeljene spremnike na kojima instruktori objavljuju svoje materijale, a studenti preuzimaju iste. Unutar aplikacija omogućeni su pregledi termina i organizacija privatnog kalendara. U nastavku su navedena dva primjera aplikacija za instrukcije: Instrukcije-hr i Lekcije.

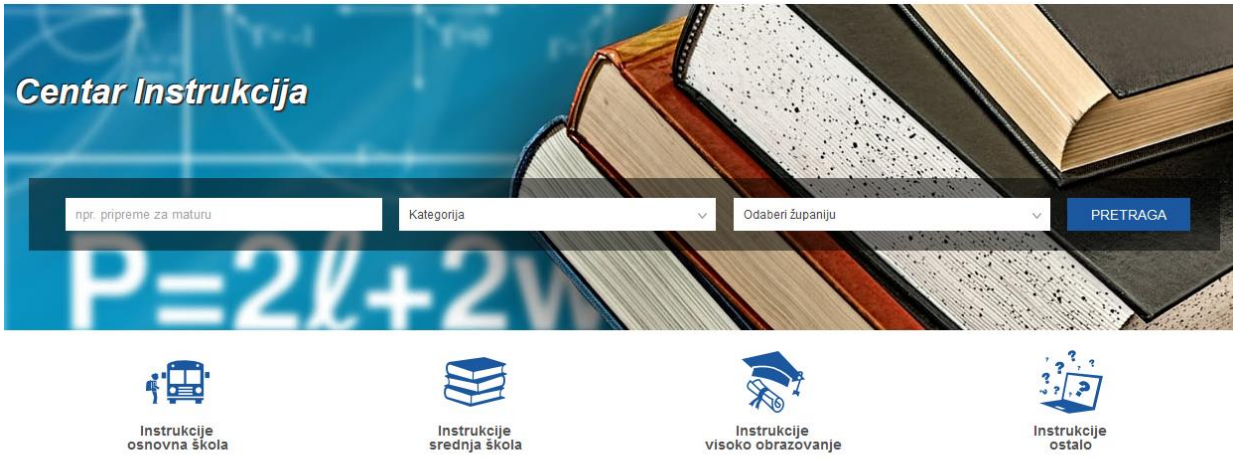
1.1.1. Instrukcije-hr

Instrukcije-hr je aplikacija za instrukcije na kojoj je moguća pretraga brojnih gradova i predmeta iz kojeg su instrukcije potrebne. Aplikacija prilikom ulaska prikazuje najnovije oglase i oglasi su prikazani kao kartice na kojima su podatci o instruktoru, predmetu, lokaciji i kratki opis o instruktoru. Omogućena je posjeta profila na kojima je moguće ostaviti recenziju. Na aplikaciji nije omogućena komunikacija, osmišljena je kao oglasnik za instrukcije i komunikacija se uspostavlja s danim podatcima u opisu (telefonski broj, e-mail).

Slika 1.1 Prikaz pretrage termina na https://www.instrukcije-hr.com/

1.1.2. Lekcije

Lekcije je aplikacija za instrukcije na kojoj je moguća pretraga termina instrukcija prema kategorijama obrazovanja, županijama i posebnim tipovima. Aplikacija prilikom ulaska prikazuje tražilicu te najnovije oglase. Oglasi su prikazani kao kartice koje sadrže poveznicu na detaljniji opis o terminu, kategoriju obrazovanja, lokaciju i cijenu. Prilikom posjete detaljnijeg opisa prikazuju se dodatne informacije o instruktoru i terminu, omogućeno je ostavljanje recenzije na termin te posjeta profila instruktora. Prilikom posjete profila instruktora prikazuju se podatci o intsruktoru, njegovi termini i omogućeno je slanje poruke putem aplikacije.



Slika 1.2 Prikaz pretrage termina na http://lekcije.hr/

1.2. Aplikacije za komunikaciju

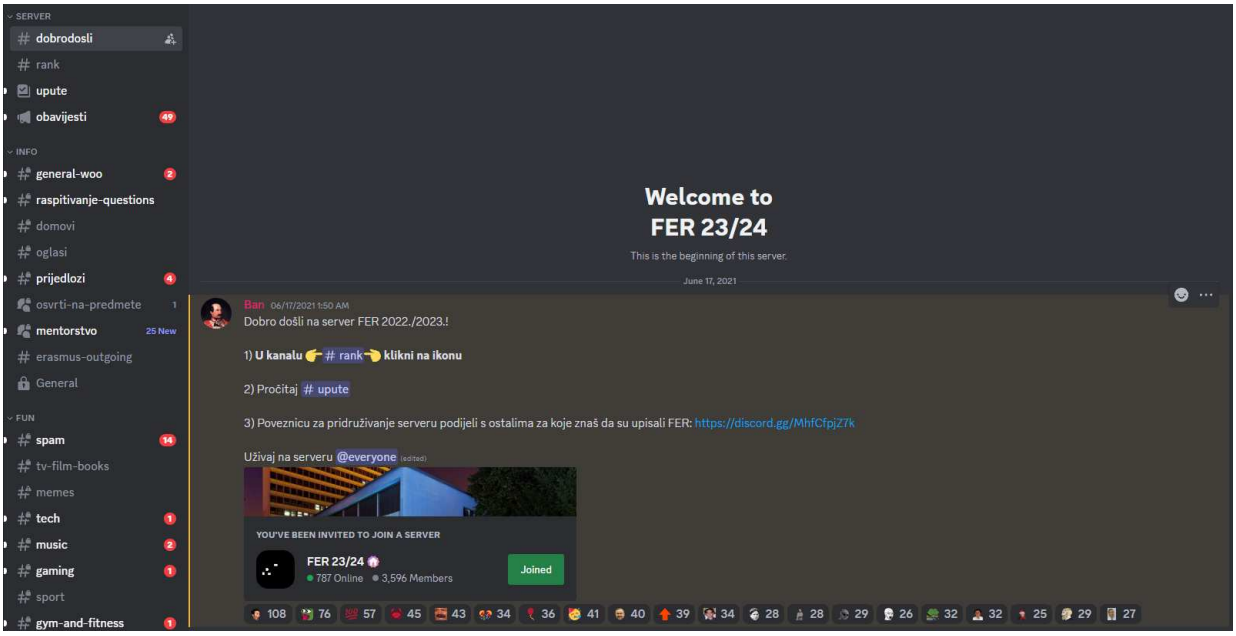
Aplikacije za komunikaciju koriste se za uspostavu poziva, audio/audio-vizualnog prijenosa signala između velikog broja korisnika, u trenutnom vremenu uz mnoštvo korisnih značajki ovisno o tome kako je sustav implementiran (npr. dijeljenje zaslona, zajedničke radne plohe itd...). U nastavku su navedena dva primjera aplikacija za komunikaciju: Discord i Teams.

Slika 1.3 Prikaz uspostave komunikacije [1]

1.2.1. Discord

Discord je sustav za komunikaciju kreiran za gaming zajednicu, a sve češće se koristi u obrazovne svrhe [2]. Korisnik nakon registracije i prijave ima mogućnost kreiranja privatnih grupa ili servera u koje mogu ulaziti i drugi korisnici, omogućena mu je i komunikacija direktno s pojedinim korisnikom ako taj korisnik prihvati njega kao siguran kontakt. Discord serveri omogućavaju uloge koje se kreiraju i daju korisnicima s povjerenjem, a mogu koristiti i botove koji se prilagođavaju raznoraznim funkcionalnostima (npr. puštanje glazbe, slanje obavijesti, sprječavanje spam-a itd...). Pozivi u aplikaciji imaju dodatne mogućnost poput: dijeljenja cijelog zaslona, dijeljenja prikaza pojedine aplikacije, prilagođavanje audio-vizualnih svojstava drugih korisnika itd...

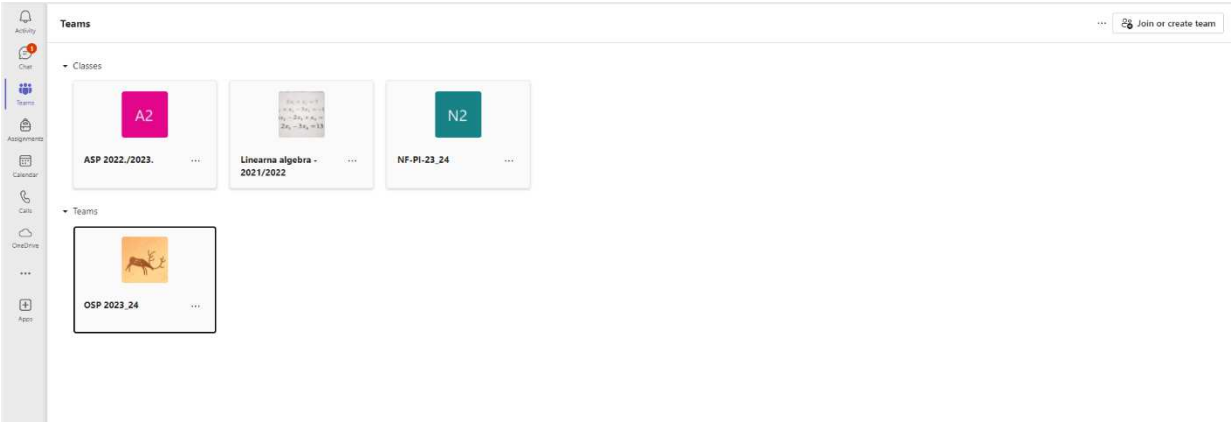
Aplikacija se sve češće koristi u obrazovne svrhe zbog odličnih performansi i prvenstveno dostupnosti i cijene (aplikacija je besplatna).



Slika 1.4 Prikaz Discord servera

1.2.2. Teams

Teams je aplikacija za timsku suradnju koju je razvio Microsoft za održavanje poslovnih videokonferencija [3]. Nakon prijave korisnik može kreirati vlastiti tim ili se s primljenom pozivnicom pridružiti postojećem timu, a može i privatno komunicirati s pojedinim korisnicima. Teams se često koristi u obrazovanju jer je aplikacija povezana sa svim ostalim Microsoftovim aplikacijama koje većina korisnika koristi (npr. word, excel, powerpoint itd...). Aplikacija sadrži kalendare za korisnike u kojima mogu vidjeti svoje dnevne i buduće obaveze i drugi korisnici ih mogu dodavati na popise za njihove konferencije. U pozivu korisnicima su omogućeni: dizanje ruke (ako korisnik želi priupitati organizatora), dijeljenje zaslona, radnu ploču po kojoj mogu pisati korisnici, dijeljenje zaslona, slanje direktne poruke pojedinom korisniku, postavljanje audio-vizualnih postavki za pojedinog korisnika ...
Uz teams postoji jako puno aplikacije koje, zbog njihovih sličnosti s Teamsom, ne opisujemo detaljnije a neke od njih su: Zoom, Skype...



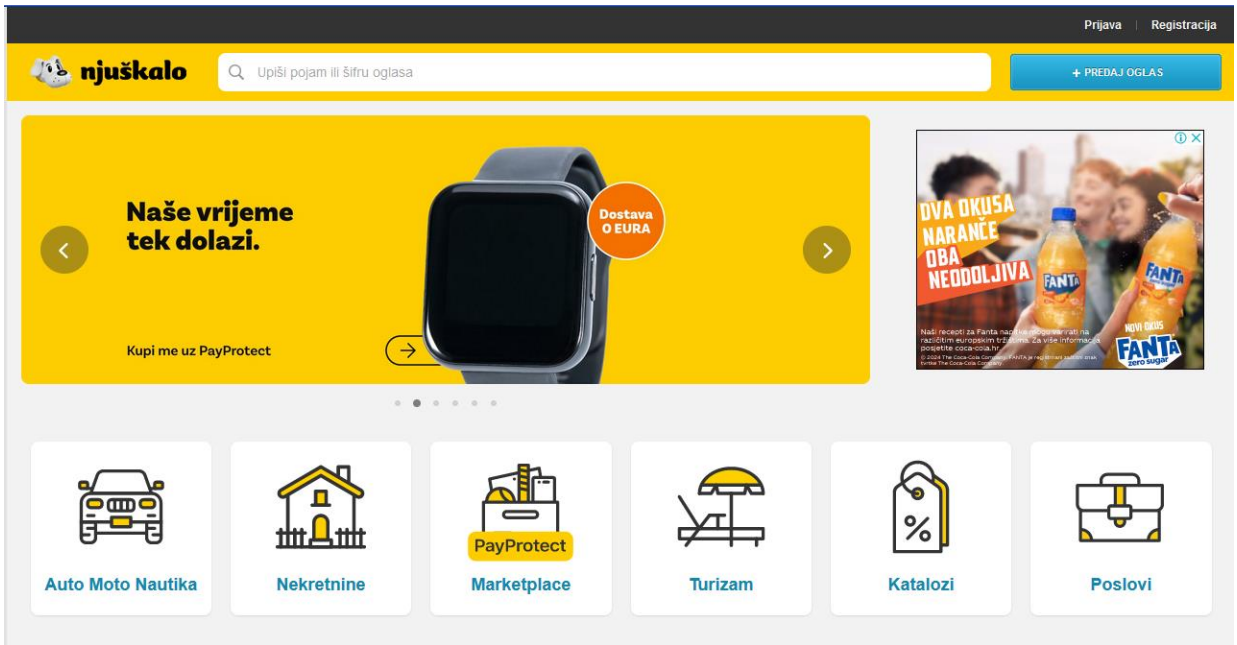
Slika 1.5 Prikaz timova u Teamsu

1.3. Oglašivačke aplikacije

Oglašivačke aplikacije temelje se na kupoprodajnim odnosima između oglašivača i kupca. U upotrebi su diljem svijeta jer korisnicima omogućavaju jednostavnu objavu oglasa s kojima mogu prodavati korištene stvari, nekretnine i svoje usluge. Korisnicima koji su u potrazi za uslugama ili predmetima omogućena je laka pretraga stvari za koje su zainteresirani uz brojne pogodnosti ovisno o aplikaciji. U nastavku su navedena dva primjera oglašivačkih aplikacija: njuškalo i Facebook Marketplace (dio aplikacije Facebook).

1.3.1. Njuškalo

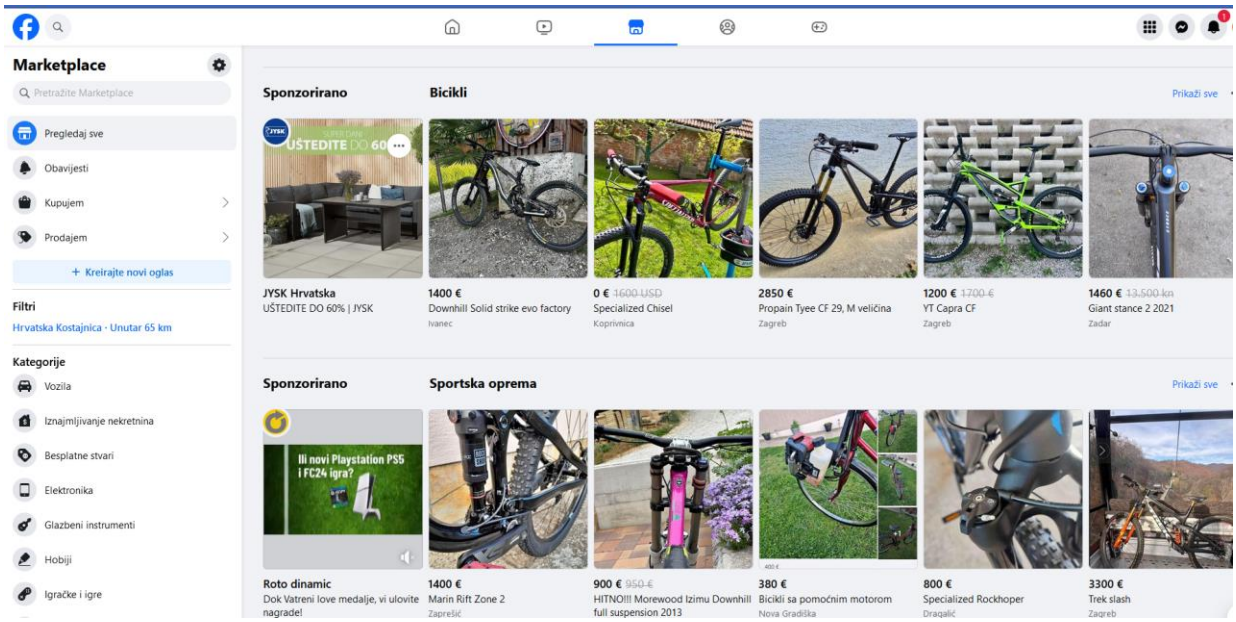
Njuškalo je kupoprodajna aplikacija koja je u najvećoj upotrebi u Republici Hrvatskoj. Aplikaciju možemo koristiti u dvije svrhe: možemo objaviti oglas i možemo potražiti predmete u oglasniku. Oglasi su u aplikaciji organizirani u mnoštvo kategorija i korisnicima su omogućeni brojni filteri s pomoću kojih mogu zadati specifičnosti predmeta koje traže. Aplikacija korisnicima omogućava pregled profila trgovca, direktno kontaktiranje trgovca putem aplikacije i puno podataka o trgovcu. Trgovcima aplikacija omogućava objavu oglasa kojima je moguće dodati proizvoljno mnogo opisa, i ovisno o kategoriji predmeta, upis specifičnih karakteristika predmeta kojeg prodaju. U aplikaciji postoji opcija plaćenih oglasa koja prikazuje oglas pri vrhu prilikom potrage.



Slika 1.6 Početna stranica njuškala

1.3.2. Facebook Marketplace

Facebook je društvena mreža kreirana u fakultetske svrhe, međutim s vremenom je postala najpopularnije mjesto za objavu slika i uspostavljanje kontakata [4]. Svojima razvojem Facebook dodaje nove značajke a jedna od njih je i Marketplace. Marketplace je dio Facebook aplikacije kreiran za objavu oglasa i reklama, te pregled istih ovisno o ciljevima upotrebe. Prednost integracije takvog kupoprodajnog sustava u aplikaciju tolike popularnosti, poput Facebooka, je mnoštvo oglasa i jednostavnost uspostavljanja kontakta između kupca i trgovca. Marketplace svojom jednostavnošću i popularnošću ozbiljno konkurira drugim aplikacijama iako je ugrađen unutar druge aplikacije. Svaki korisnik jednostavno može dodati oglas i opise vezane uz oglas, a kupac može pregledati ostale oglase korisnika, njegov društveni profil te direktno kontaktirati trgovca. Sustav kao i svi ostali, sličnih karakteristika, nudi brojne filtere i svakodnevno prikazuje nove oglase, temeljene na našim proteklim pretragama, koji se nalaze u našoj blizini.



Slika 1.7 Početna stranica facebookovog marketplacea

2. Ideja rada i opis dijelova sustava

Ideja ovog rada je razviti sustav kao kombinaciju dvaju tipova navedenih aplikacija: aplikacija za komunikaciju i oglašavačkih aplikacija, koji bi omogućio online instrukcije, studentima uvelike pomogao pri pretraživanju termina instrukcija prema zadanim kriterijima, a instruktorima omogućio jednostavnu objavu oglasa.

Prvi korak prilikom korištenja navedenog sustava je registracija novog korisnika. Tijekom registracije korisnik unosi osobne podatke i može odabrati u koju svrhu će upotrebljavati aplikaciju: kao instruktor ili kao korisnik koji je u potrazi za terminima.

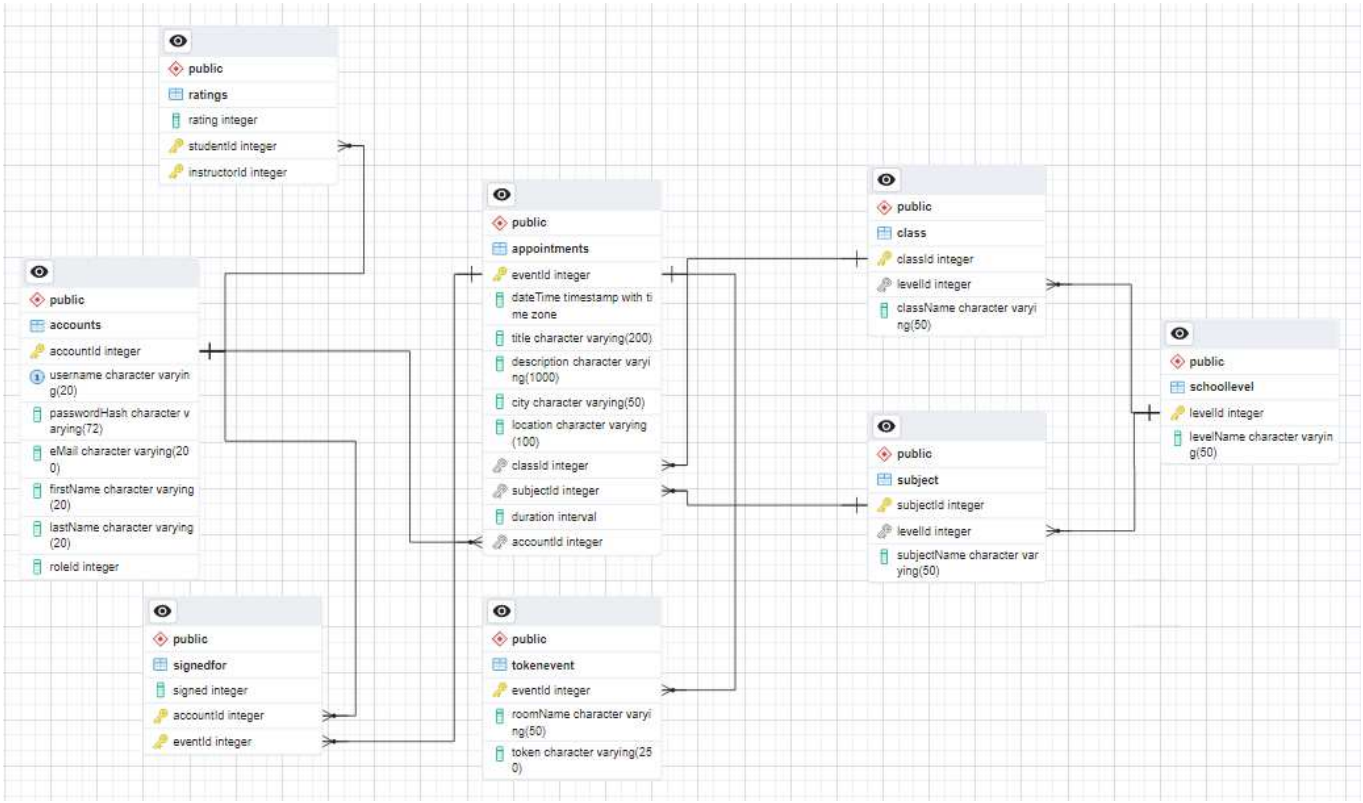
Nakon registracije korisnici koji su prijavljeni u sustav kao studenti imaju dostupne sve termine instrukcija koji su spremljeni u bazu podataka. Radi lakšeg i bržeg odabira korisnicima su omogućeni filteri. Kada korisnik pronađe termin instrukcija koji zadovoljava njegove kriterije može rezervirati svoju prisutnost u zadanom terminu te ako promjeni odluku otkazati rezervaciju. Korisniku je na glavnoj stranici sa svim terminima omogućen prikaz njegovih termina za koje je uspješno prijavljen, radi lakšeg pregleda njegovih obaveza. Za svaki termin korisnik može posjetiti profil instruktora te ostaviti recenziju. Ako je za termin kreirana soba za razgovor u opisu termina korisniku će se prikazati ime sobe te će se moći pridružiti online sastanku.

Nakon registracije korisnici koji su prijavljeni u sustav kao instruktori imaju dostupne sve termine instrukcija koji su spremljeni u bazu podataka te jednake filtere kao i studenti. Takvim korisnicima onemogućeno je posjećivanje profila instruktora i prijave za termin, a na prikazu njegovih termina nalaze se termini koje je on kreirao. Korisnicima je omogućeno kreiranje termina putem forme za unos podataka te uređivanje i brisanje istih. Za svaki svoj termin takvi korisnici mogu kreirati sobu za sastanak čije će vrijeme trajanja biti sat vremena.

Neovisno o kategoriji korisnika svaki korisnik ima pristup svom profilu te uređivanje istoga. Korisnici prijavljeni kao instruktori na svom profilu dodatno će moći postaviti kratki opis o sebi. Na profilu su također prikazani rezervirani termini za studene te kreirani termini za instruktore.

3. Model podataka

3.1. Relacijski model podataka



Slika 3.1 Relacijski model podataka

3.2. Opis relacijskog modela podataka

Glavna tablica baze podataka je accounts koji predstavlja korisnički račun. Korisnici se u sustav prijavljuju s korisničkim imenom i lozinkom iz koje se kriptografskom funkcijom sažetka generira niz znakova. Ostali podatci za korisnika su: ime, prezime, e-mail i roleId koji se upisuje u bazu ovisno o odabiru uloge.

Objekte iz tablice appointments kreiraju korisnici prijavljeni u sustav kao instruktori putem forme za unos podataka, a podatci su: naslov, opis, grad, lokacija (ulica...), razred, predmet, početak termina i kraj termina.

Objekte iz tablice signedfor kreiraju korisnici prijavljeni u sustav kao učenici/studenti rezervacijom termina i takav zapis sastoji se od signedfor (element koji označava da je korisnik odabrao termin), ID-a eventa i ID-a korisnika.

Objekti iz tablice ratings označavaju recenziju učenika/studenta na profil instruktora i sastoje se od: ocjene, ID-a učenika/studenta i ID-a instruktora.
Objekti iz tablice schoollevel označavaju razinu obrazovanja i sastoje se od: ID-a razine i imena razine obrazovanja (tablica i tablice koje su povezane s njom koriste se za unos kategorija u područjima: razine obrazovanja, razreda i predmeta, administrator sustava ih unosi ručno).
Objekti iz tablice class povezani su s objektima tablice schoollevel i označavaju razrede za pojedinu razinu obrazovanja. Sastoje se od: ID-a razreda, ID-a razine obrazovanja i imena razreda.
Objekti iz tablice subject označavaju školske predmete i povezani su s objektima tablice schoollevel. Sastoje se od: ID-a predmeta. ID-a razine obrazovanja i imena predmeta.
Objekti iz tablice tokens označavaju specijalnu oznaku potrebnu prilikom ulaska u video poziv i povezani su s objektima tablice termins jer se za svaki termin može kreirati token. Sastoje se od: ID-a termina, naziva sobe i niza znakova koji predstavlja sam token.

3.3. Opisi entiteta

Razina obrazovanja

- **levelId** – šifra razine obrazovanja (PK)
- levelName – ime razine obrazovanja

Razred

- **classId** – šifra razreda (PK)
- levelId – šifra razine obrazovanja (FK)
- className – naziv razreda

Predmet

- **subjectId** – šifra predmeta(PK)
- levelId – šifra razine obrazovanja (FK)
- subjectName – ime predmeta

Korisnički račun

- **accountId** – šifra korisničkog računa (PK)
- username – korisničko ime korisnika (U)
- eMail - email za kontakt (U)
- firstName – ime korisnika
- lastName – prezime korisnika
- roleId – šifra uloge
- passwordHash – lozinka prikazana kriptograskom funkcijom sažetka

Recenzija

- **instructorId** – šifra korisničkog računa instruktora (PK)
- **studentId** – šifra korisničkog računa učenika/studenta (PK)
- rating – recenzija

Rezervacija

- **accountId** – šifra korisničkog računa (PK)
- **eventId** – šifra termina instrukcija (PK)
- signed – rezervacija

Termin

- **eventId** – šifra termina instrukcija (PK)
- title – naziv termina instrukcija
- description – opis termina instrukcija
- city – grad u kojem korisnik živi
- location – lokacija na kojoj korisnik živi
- classId – šifra razreda (FK)
- subjectId – šifra predmeta iz kojeg se održavaju instukcije (FK)
- dateTime – datum i vrijeme početka termina

- duration – trajanje termina
- accountId – šifra kreatora termina (FK)

Token

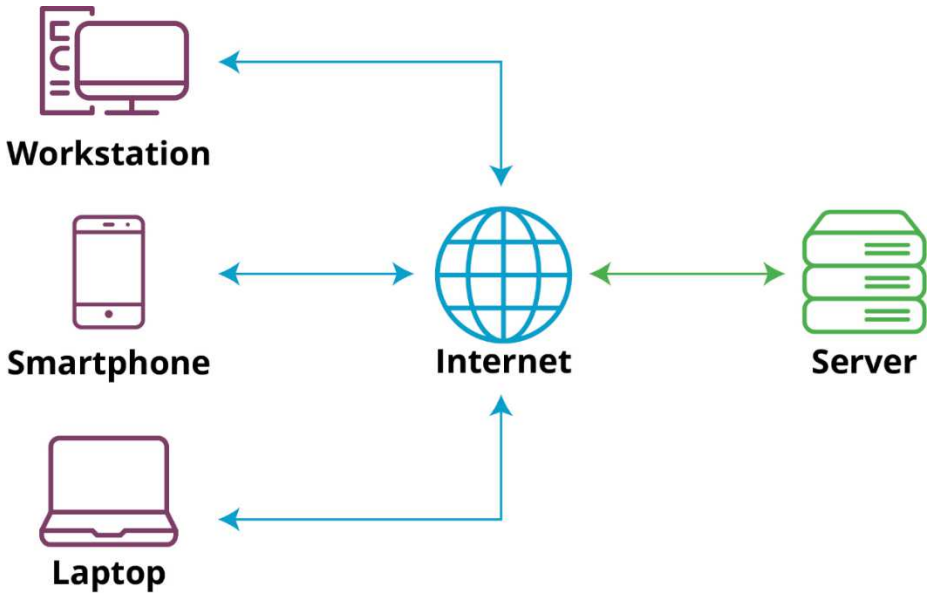
- **eventId** – šifra termina instrukcija (PK)
- roomName – ime sobe u kojoj se održava videopoziv
- token – generirana šifra za pristup sobi

4. Korištene tehnologije i alati

Razvijeni sustav se sastoji od tri dijela:

- Klijent – korisnički dio sustava koji prikazuje sve korisniku i omogućuje mu slanje zahtjeva na poslužitelj od kojeg dobiva informacije
- Poslužitelj – serverski dio sustava koji služi za obradu zahtjeva i informacija poslanih od strane klijenta, vraća klijentu obrađene informacije koje klijent zatim prikazuje
- Baza podataka – dio sustava koji služi za pohranu entiteta, njihovih atributa i dodatnih informacija kojima se koriste klijent i poslužitelj

Korišteni model arhitekture sustava je klijent-poslužitelj, a označava razmjenu podataka između dvaju ili više uređaja povezanih u mrežu [5]. Klijentski dio ove arhitekture obavlja web-preglednik, koji upotrebom grafičkog sučelja korisniku prikazuje sadržaj stranice i korisnicima omogućava slanje podataka poslužitelju i primanje obrađenih podataka. Komunikaciju uspostavljaju HTTP protokolom. Poslužiteljski dio obavlja web radni okvir. Poslužitelj obrađuje podatake primljene od klijenta, te šalje iste nakon obrade.



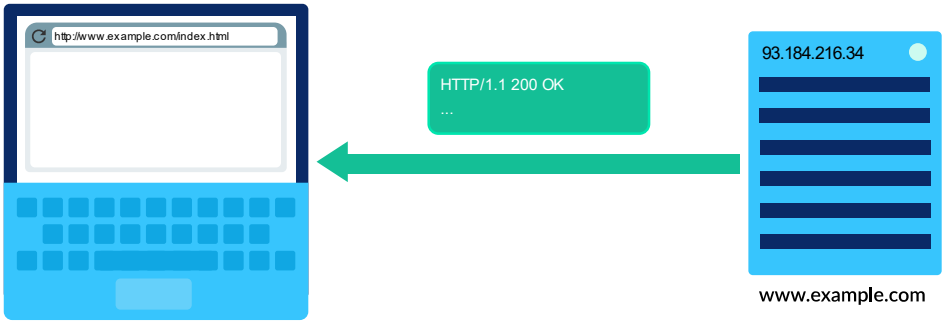
Slika 4.1 Arhitektura klijent-poslužitelj [6]

HTTP je zahtjev/odgovor protokol koji služi za komunikaciju između klijenta i poslužitelja. Klijent uspostavlja TCP vezu sa udaljenim poslužiteljom na određenome portu. Poslužitelj konstantno osluškuje komunikacijski kanal na zadanom portu te

dolaskom novog zahtjeva pokušava uspostaviti vezu. Klijentski zahtjevi, kreirani su prema REST načinu rada, najčešće sadrže jednu od tipa poruka: GET (dohvati sadržaj), POST (objavi sadržaj) ili DELETE (obriši sadržaj), a uz tip poruke klijent može dodatno poslati strukturu podataka koja će poslužitelju trebati za obradu. Nakon primitka poruke poslužitelj izvršava obradu podataka te šalje povratnu informaciju, odgovor, o uspješnosti obrade dobivenog zahtjeva te ako je obrada uspješna najčešće šalje strukturu obrađenih podataka klijentu.

Implementiran je DataController.js koji omogućava klijentsko slanje podataka na poslužitelj (implementacija GET, POST, DELETE i PUT zahtjeva te autorizacijskih tokena).

Slika 4.2 HTTP zahtjev [7]



Slika 4.3 HTTP odgovor [7]

4.1. React.js

Za prikaz sadržaja korisniku i omogućen unos podataka u forme koristi se JavaScriptova biblioteka otvorenog koda React. Utemeljili su ga vlasnici Facebooka, a održava ga tvrtka Meta (prije Facebook). React se koristi za razvoj jednostranih, mobilnih ili drugih aplikacija i bavi se samo korisničkim sučeljem i komponenata u DOM-u. React aplikacije se stoga oslanjaju na biblioteke za usmjeravanje i komunikaciju s poslužiteljem [8]. Jedna od ključnih prednosti reacta je što rendera samo promijenjene dijelove sadržaja, i time izbjegava nepotrebno ponovno renderiranje nepromijenjenih DOM elemenata, čime je znatno efikasniji.

Kreiranjem React projekta dobivamo index.html stranicu u čije html elemente se dinamički umeće sadržaj za prikaz. Dinamički sadržaj najčešće prikazuje jedan JavaScript element koji koristi React Router za prikaz React komponenti ovisno o promjeni putanje. U ovome projektu taj JavaScript element je App.js.

```
export default function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<Register />} />
        <Route path="/videocall" element={<VideaApp />} />
        <Route path="/videoroom" element={<VideoRoom />} />
        <Route path="/login" element={<Login />} />
        <Route path="/profile" element={<Profile />} />
        <Route path="/createTermin" element={<CreateTermin />} />
        <Route path="/allTermins" element={<AllTermins />} />
          <Route path="/instructorProfile/:id" element={<InstructorProfile
/>} />
        <Route path="/edit/:eventId" element={<EditTermin />} />
      </Routes>
    </BrowserRouter>
  );
}
```

Kod 4.1 – App.js komponenta

Komponente u Reactu značajno olakšavaju korištenje i pisanje web-stranica jer jednom napisanu komponentu možemo iskoristiti na više različitih mjesta s različitim

parametrima. To pospješuje organizaciju koda i čitljivost što uvelike olakšava programerima koji rade na većim projektima.

```
<EventCard card={card} />
```

Kod 4.2 – Poziv komponente EventCard s parametorm card

```
const { card } = props;
```

Kod 4.3 – Spremanje podataka poziva u varijablu card

React komponente mogu imati nastavak .js ili .jsx a glavna razlika je ta da .jsx sintaksa omogućava pisanje HTML elemenata unutar .js komponente. U Reactu možemo pisati i funkcije kojima manipuliramo elementima čiju aktivnost pratimo.

```
Function editTermin() {
  navigate(„/edit/“ + card.id);
}
```

Kod 4.4 – Primjer funkcije koja preusmjerava putanju

Uz sve navedeno u Reactu možemo koristiti mnoštvo biblioteka koje sadrže unaprijed izgrađene komponente čime povećavamo brzinu i efikasnost (npr. @mui/material)

4.2 MUI

MUI je biblioteka React komponenti koja omogućava preuzimanje gotovih, unaprijed dizajniranih i funkcionalnih dijelova sustava. Najčešće komponente koje biblioteka sadrži su: gumbi, forme, kartice, različiti prikazi teksta... [9] Korištenje MUI-a uvelike je olakšalo dizajniranje stranica jer je dostupan pregled komponenti na njihovoj stranici. Komponente su jednostavne za korištenje, omogućeno je upravljanje CSS svojstvima komponenti, i implementaciju unutar .jsx elemenata. Komponentu je potrebno instalirati zatim importati i ona je spremna za korištenje.

```
import Button from "@mui/material/Button";
```

Kod 4.5 – import komponente Button

```
<Button size="small" onClick={editTermin} style={{ color: '#007bff' }}>
        Edit
</Button>
```

Kod 4.6 – upotreba komponente Button

4.3 Flask

Backend dio sustava kreiran je s radim okvirom Flask. Flask je mikro web okvir jezika python, smatra se mikrookvirom jer za njegov rad nisu potrebne nikakve biblioteke ili posebni alati. Obrada zahtjeva u Flasku je jednostavna brza i postoje brojna proširenja koja olakšavaju rad u Flasku [10]. Rad u Flasku temelji se na usmjeravanju (routingu) koji dolaskom zahtjeva s određenom putanjom aktivira određenu funkciju. Za komunikaciju Flaska s bazom podataka koristili smo SQLAlchemy alate.

```
self.app.add_url_rule("/getSubjects",        view_func=self.getSubjects,
methods=["GET"])
```

Kod 4.5 – Definiranje pravila koje će flask prepoznati

```
const fetchSubjects = async () => {
   dc.GetData("http://localhost:5000" + "/getSubjects", accessToken)
     .then((resp) => setSubjects(resp.data.data))
     .catch((resp) => {
      console.log(resp);
     });
  };
```

Kod 4.6 – primjer poziva pravila

```
def getSubjects(self):
      subjects = self.db.session.query(Subject).all()
      subjects = list(map(lambda subject: subject.__dict__, subjects))
      for item in subjects:
         del item['_sa_instance_state']
      return {"success":True, "data": subjects}
```

Kod 4.7 – Primjer funkcije koju poziva flask

4.4 SQLAlchemy i Elephant DB

Elephant DB je besplatni servis PostgreSQL s pomoću kojeg se kreira baza podataka u oblaku. SQLAlchemy su alati Flaska kojima upravlja podatcima u bazi podataka. Prednost SQLAlchemya je jednostavno pisanje upita na bazu podataka te izravno kreiranje tablica iz koda u Flasku. Datoteka u kojoj su kreirane tablice je Models.py.

```
class SchoolLevel(db.Model):
   __tablename__ = "schoollevel"
   levelId = db.Column(db.Integer, primary_key = True, nullable = False)
   levelName = db.Column(db.String(50), nullable = False)
```

Kod 4.8 – Kreiranje modela za tablicu schoollevel

Kako bismo koristili SQLAlchemy potrebno je postavljanje početnih postavki.

```
DB_URL=postgresql://zfgwzpnp:TXMWN-
FXWSR3JftP48BLcQrPqTx23QFd@flora.db.elephantsql.com/zfgwzpnp
```

Kod 4.9 – URL baze podataka kreirane s Elephant DB

Prilikom upotrebe potrebno je konfigurirati bazu podataka.

```
App.config[„SQLALCHEMY_DATABASE_URI“] = os.environ.get(„DB_URL“)
db = SQLAlchemy(app)
```

Kod 4.10 – Konfiguracija baze podataka

Nakon toga možemo pisati upite na bazu.

```
data = self.db.session.query(SignedFor).join(Account, Account.accountId
== SignedFor.accountId).join(Termin, Termin.eventId ==
SignedFor.eventId).filter(Account.username ==
get_jwt_identity()).filter(Termin.eventId == eventId).first()
myUser =
self.db.session.query(Account).filter_by(username=get_jwt_identity())
.first()
```

Kod 4.11 – Primjer upita na bazu

4.5 PostgreSQL

PostgreSQL besplatan je sustav za upravljanje relacijskim bazama podataka. Otvorenog je koda i usklađen je sa SQL-om. Transakcije u PostgreSQL-u poštuju svojstva i pravila koja definiraju integritet i ispravnost podataka. Postgre sustav prvenstveno se koristi za relacijske tablice ali postoji mnoštvo funkcionalnosti kojima se rad s bazama podataka podiže na višu razinu. Stoga PostgreSQL svoju uporabu proširuje i iz obrazovnih u profesionalne svrhe [11].

4.6 Agora.io

Agora je jedna od vodećih platformi za komunikacijske API-e pomoću koje programeri imaju mogućnost integracije visokokvalitetne glasovne i video prijenose uživo u svoje sustave [12]. Prednost odabira Agore je besplatan sadržaj do 10000 minuta. Agora na svojim stranicama sadrži dodatne sadržaje koji su mogu integrirati u sustave a neki od njih su: dijeljenje ekrana, zajednička radna ploha, podešavanje audio-vizualnih postavki za pojedinog korisnika...

Za ulazak u videopoziv korisnik mora odobriti svoj audio-vizualni sadržaj i mora poznavati javni token sobe koji kreiraju dva python programa dostupna na Agorinoj stranici (AccessToken.py i RtcTokenBuilder.py). Podatci potrebni za kreiranje tokena su: appId, appCertificate (developer ih može kopirati na svojem profilu), channelName i expireTime.

```
token=          RtcTokenBuilder.buildTokenWithUid(appID,appCertificate,
channelName, userAccount, Role_Attendee, privilegeExpiredTs)
```

Kod 4.12 – Kreiranje tokena

Nakon kreiranja tokena kreira se klijent i podešavaju se audio-vizualni izvori.

```
let client = AgoraRTC.createClient({ mode: 'rtc', codec: 'vp8' });
client
    .join(APP_ID, CHANNEL, TOKEN, null)
    .then((uid) => {
      setCurrentUser(uid);
      return Promise.all([
        AgoraRTC.createMicrophoneAndCameraTracks(),
        uid,
```

```
      ]);
    })
    .then(([tracks, uid]) => {
      const [audioTrack, videoTrack] = tracks;
      setLocalTracks(tracks);
      setUsers((previousUsers) => {
        const userExists = previousUsers.some((user) => user.uid ===
uid);
        if (!userExists) {
          return [
            ...previousUsers,
            {
              uid,
              videoTrack,
              audioTrack,
            },
          ];
        } else {
          return previousUsers;
        }
      });
      client.publish(tracks);
    })
    .catch((error) => {
      console.error('Failed to join channel:', error);
    });
```

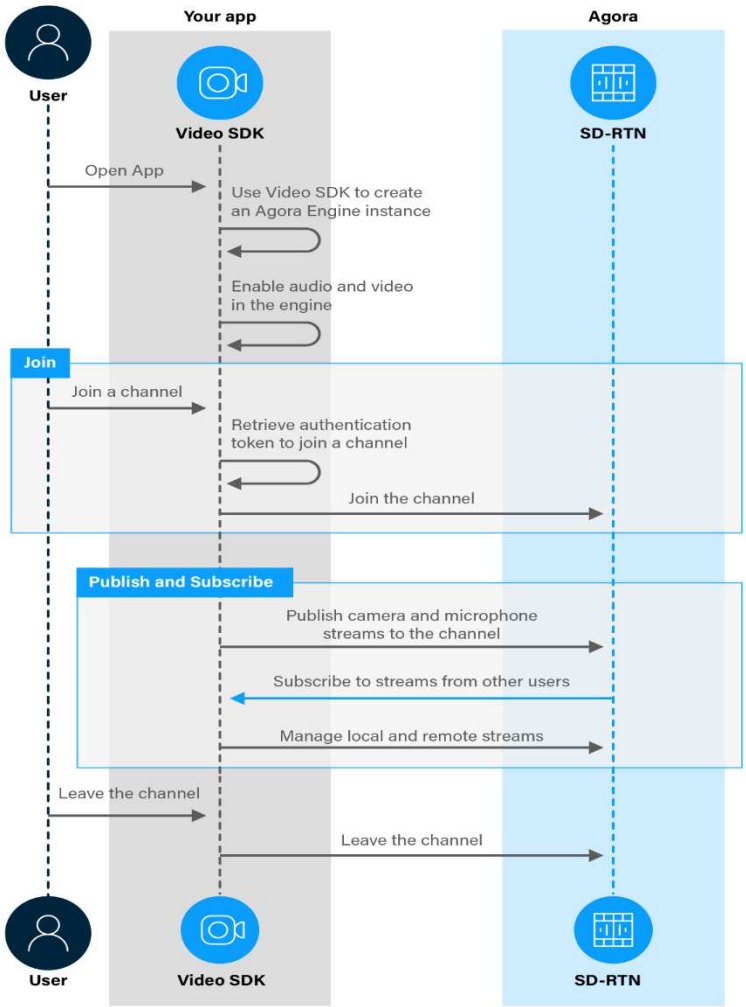
Kod 4.13 – Kreiranje klijenta i ulazak u poziv

Prilikom napuštanja sobe trebamo ugasiti sve audio-vizualne izvore korisnika.

```
for (let localTrack of localTracks) {
      localTrack.stop();
      localTrack.close();
    }
    if (client) {
      client.leave();
    }
```

Kod 4.14 – Izlazak iz sobe

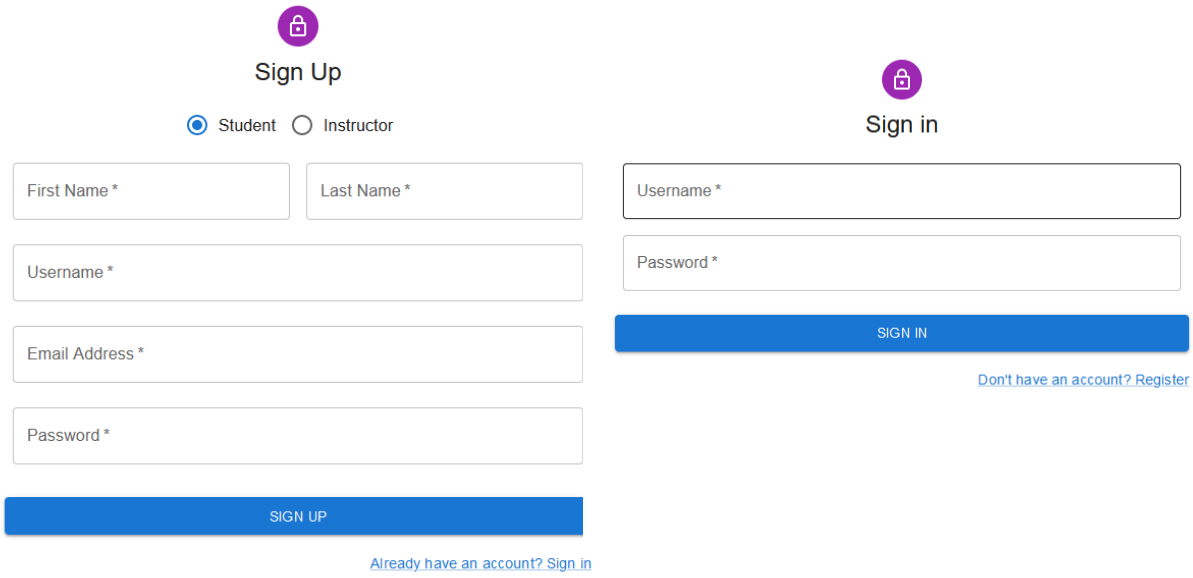
Agora na stranicama ima puno dijagrama koji prikazuju tijek komunikacije što je idealno za programere koji po prvi puta implementiraju video pozive.



Slika 4.4 – Sekvencijski dijagram Agorinog videopoziva preuzet sa Agorine stranice [12]

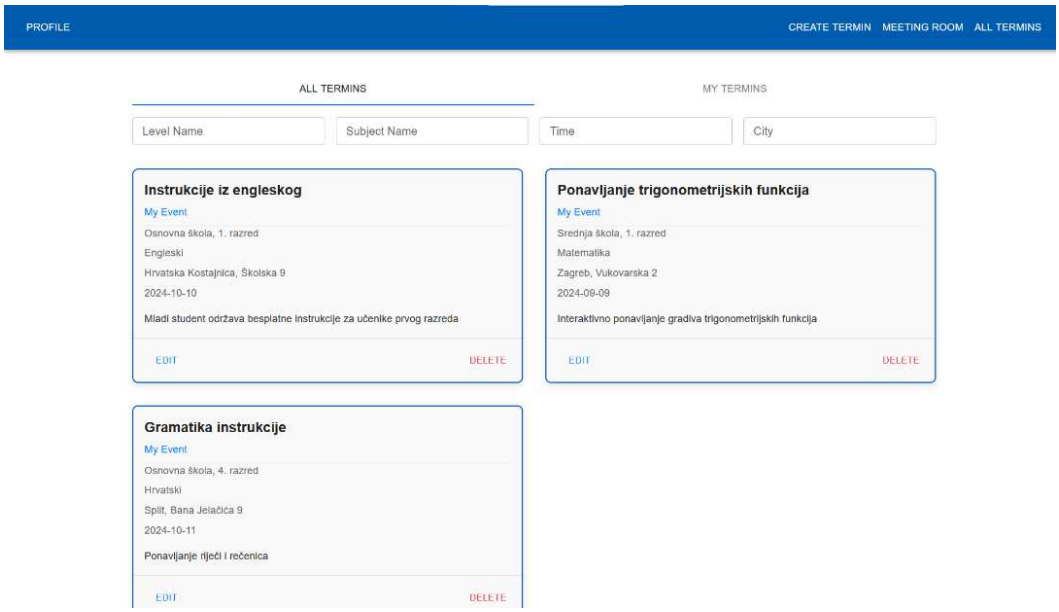
5 Opis rada aplikacije

Korisnici najprije moraju kreirati korisnički račun popunjavanjem forme za registraciju (slika 5.1). Njihovi podatci se zatim dodaju u bazu podataka iz koje će se podatci uzimati prilikom prijave u sustav. Za registraciju su potrebni osobni podatci korisnika: ime, prezime, e-mail, korisničko ime, šifra koju baza podataka automatski zapisuje s pomoću kriptografske funkcije sažetka kako ju nitko ne bi mogao saznati. Potrebno je odabrati u koje će se svrhe aplikacija koristiti, a korisnici mogu birati između uloge student i uloge instruktor. Uloga instruktor se bitno razlikuje u funkcionalnostima naspram uloge student te se za korisnike s ulogom instruktor smatra da će oni biti provoditelji instrukcija, a korisnici s ulogom student prisustvovati i pretraživati kreirane termine. E-mail se ne koristi u aplikaciji ali je potreban za buduće nadogradnje sustava. Prilikom prijave u sustav korisnik treba upisati svoje korisničko ime i lozinku (slika 5.1).



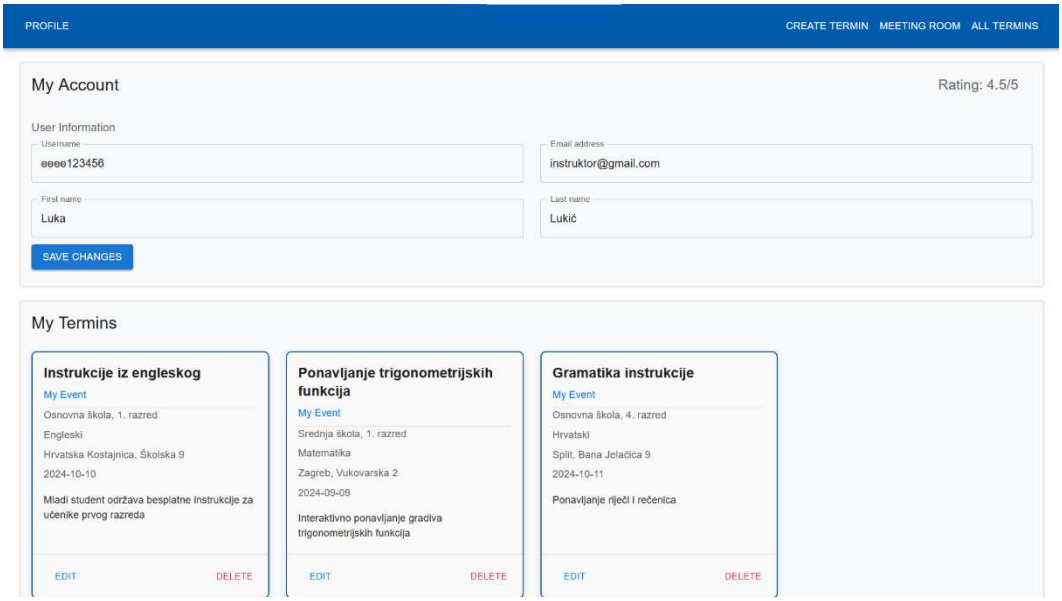
Slika 5.1 – registracija korisnika (lijevo), prijava korisnika (desno)

U sustavu se moraju razlikovati već navedena dva tipa korisničkih računa, međutim neke od funkcionalnosti su jednake za obje uloge. Korisnicima se nakon prijave prikazuje stranica na kojoj su prikazani svi termini instrukcija (slika 5.2). Na toj stranici korisnicima je omogućena filtracija termina ovisno o parametrima koji odgovaraju njihovoj pretrazi. Parametri prema kojima korisnici mogu filtrirati svoju pretragu su: tip obrazovanja, razred, predmet, vrijeme održavanja i grad.



Slika 5.2 – Početna stranica za sve korisnike

Korisnici se s početne stranice, na kojoj su prikazani svi termini, mogu prebaciti na stranicu svojeg profila klikom na profile u zaglavlju stranice na kojem mogu pregledati svoje osobne informacije i prema potrebi promijeniti iste (slika 5.3).



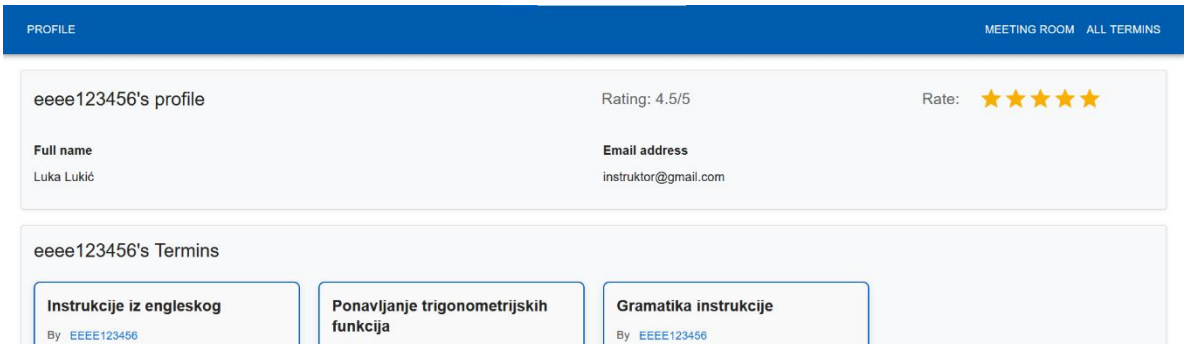
Slika 5.3 – Profil korisnika

Pored nekih jednakih funkcionalnosti, većina funkcionalnosti je različita za korisnike s različitim ulogama.

Korisnicima koji su u sustav prijavljeni s ulogom student na stranici sa svim terminima omogućen je prikaz termina za koje su rezervirali prisustvo u terminu klikom na MY TERMINS, a takvi termini bit će vidljivi i prilikom pregleda njihovog profila (ispod osobnih podataka). Kartice termina (slika 5.4) sadrže sve potrebne informacije o terminu instrukcija te omogućavaju korisnicima ovakvog tipa posjetu profila instruktora klikom na korisničko ime instruktora. Ovakvom tipu korisnika također je omogućena rezervacija za zadane termine. Rezerviranje prisustva omogućeno je klikom na reserve gumb kartice termina, a ako korisnik želi otkazati rezervaciju klikom na cancle to može napraviti za termine na kojima je već rezervirao svoju prisutnost.

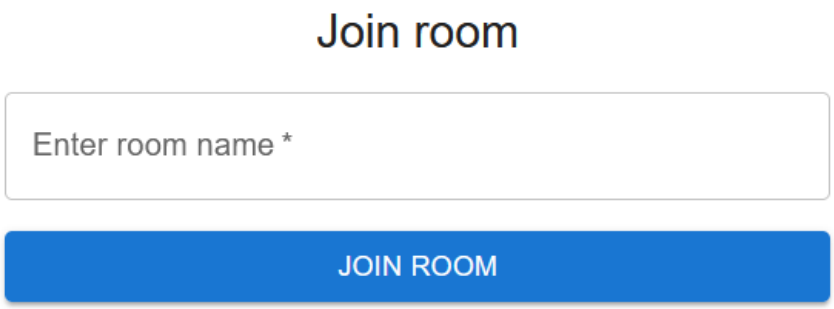
Slika 5.4 – kartice termina za učenike/studente nerezervirano (lijevo) i rezervirano (desno)

Prilikom posjete profila instruktora (slika 5.5) korisniku vidi podatke o instruktoru i može ostaviti recenziju sustavom zvjezdica sličnom Googleovim recenzijama.



Slika 5.5 – Prikaz posjećenog profila instruktora

Korisnici klikom na MEETING ROOM u zaglavlju mogu pristupiti formi za pridruživanje online videopozivu (slika 5.6) na kojem će instruktori održavati instrukcije. Od korisnika se traži ime sobe kojoj će pristupiti, a ta informacija korisnicima će se prikazati na kartici termina kada instruktor kreira sobu (slika 5.7).

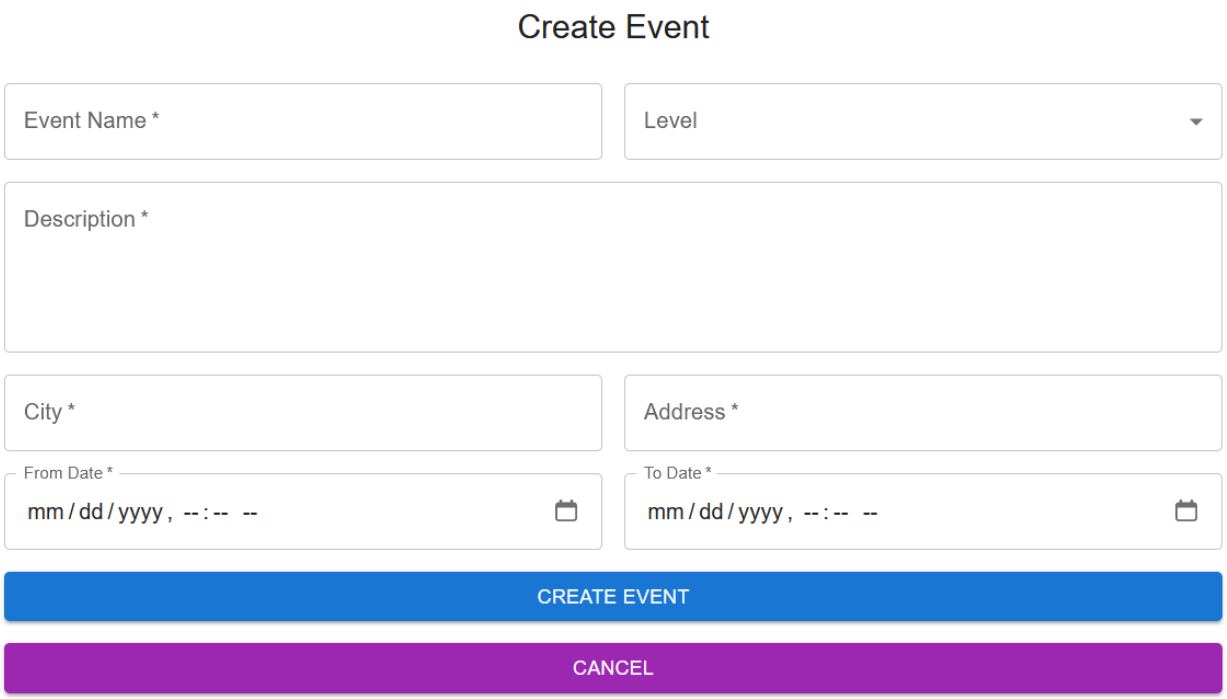


Slika 5.6 – forma za pridruživanje videopozivu

Slika 5.7 – kartica termina sa prikazom sobe online videopoziva

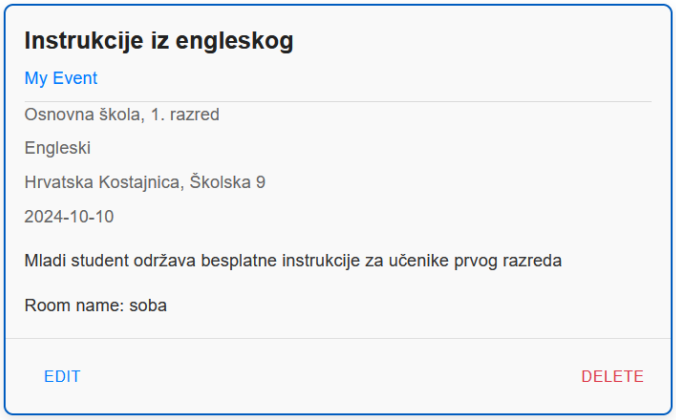
Korisnicima koji su u sustav prijavljeni s ulogom instruktor na stranici sa svim terminima omogućen je prikaz termina, koje su kreirali, klikom na my-events, a takvi termini bit će vidljivi i prilikom pregleda njihovog profila (ispod osobnih podataka).

Klikom na CREATE TERMIN u zaglavlju aplikacije instruktori mogu kreirati termine, a prikazat će im se forma za kreiranje termina (slika 5.8). Forma se sastoji od: imena termina, opisa termina, tipa obrazovanja, razreda, školskog predmeta, lokacije, adrese te početka i kraja termina, a ispravnom formom smatrat će se ona forma koja ima sva polja podataka ispunjena, a vremena početka i kraja termina nisu u prošlosti. U slučaju da se instruktor odluči prekinuti kreiranje termina može kliknuti cancle gumb, koji će ga vratiti na stranicu sa svim terminima.



Slika 5.8 – forma za kreiranje termina

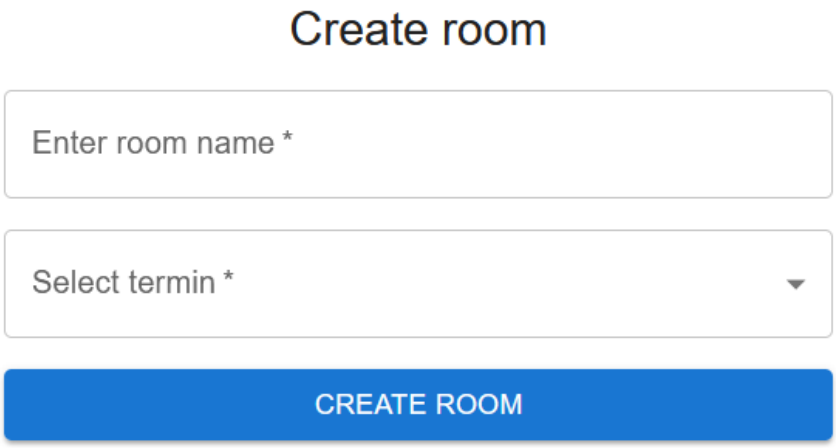
Prikaz kartice termina za instruktore (slika 5.9) nešto je drugačiji od prikaza za studente a glavne razlike su da je onemogućena posjeta profila kreatora termina, onemogućeno je rezerviranje termina i otkazivanje rezervacije, dok su omogućeni uređivanje i brisanje termina za termine koje su oni kreirali. Ako korisnik odabere uređivanje termina prikazuje mu se forma koja je identična formi za kreiranje termina međutim forma je unaprijed popunjena sa starim podatcima koje mogu promijeniti. Kada je korisnik zadovoljan promjenama može kliknuti edit termin i podatci o terminu će se ažurirati u bazi, no ako se korisnik predomislio može kliknuti cancle koji će ga vratiti na stranicu sa svim terminima, a podatci o terminu neće biti promijenjeni.



Slika 5.9 – kartica termina za korisnike s ulogom instruktor

Instruktori prilikom posjete svojem profilu mogu vidjeti njihovu ukupnu recenziju koja će se prikazati kao decimalni broj od 1-5 (slika 5.3).

Instruktorima se klikom na meeting room prikazuje forma za kreiranje sobe online poziva (slika 5.10) . Pri kreiranju sobe instruktori moraju odabrati termin za koji žele kreirati sobu te upisati ime sobe. Klikom na create room kreira se soba za online poziv čije će trajanje iznositi jedan sat. (napisati dio kako će kontaktirati studente) Nakon kreiranja sobe u formu za ulazaka u sobu (slika 5.6) korisnik upisuje ime sobe i klikom na join meeting ulazi u online poziv svojeg termina.

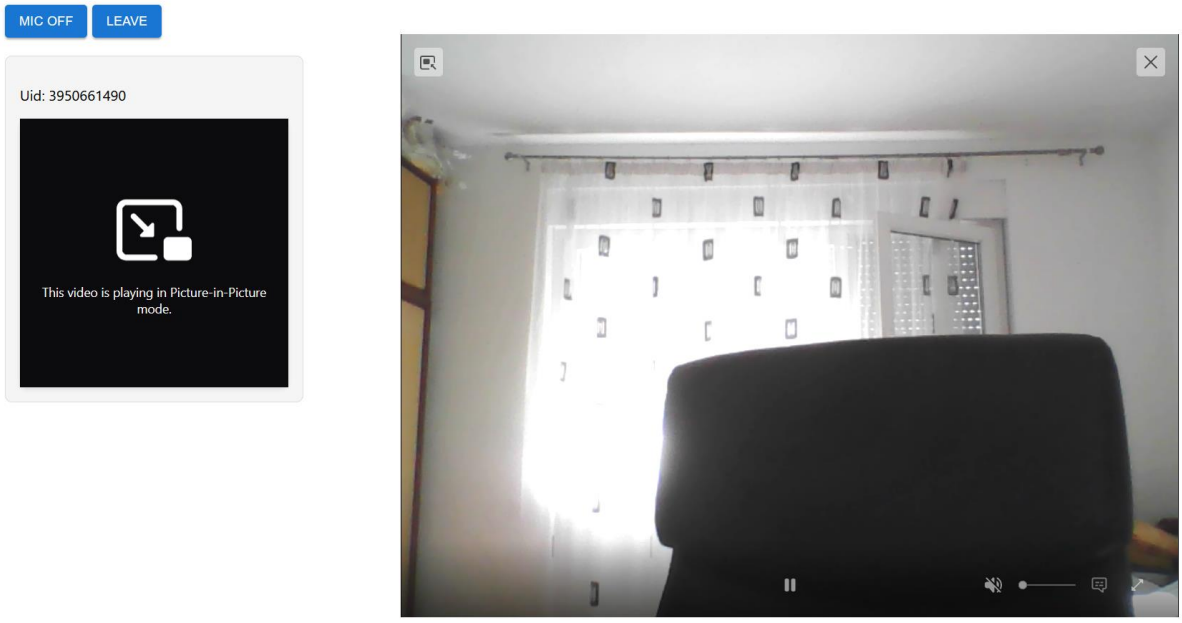


Slika 5.10 – forma za kreiranje sobe

Unutar poziva (slika 5.11) korisnici mogu gasiti i paliti svoj audio unos (klikom na mute) i mogu napustiti poziv (klikom na leave).

Slika 5.11 – soba online poziva

Korisnici također mogu proširiti karticu korisnika u pozivu (slika 5.12), veličina proširene kartice je proizvoljna.



Slika 5.12 – proširenje kartice odabranog korisnika

6 Daljnji razvoj

Zbog ograničenog vremena pisanja rada sustav sadrži brojne mogućnosti proširenja koje su ostavljene za buduće verzije. Trenutno aplikacija obavlja sve predviđene funkcionalnosti no brojne funkcionalnosti se još mogu dodati. U idućim verzijama plan je kreirati sustav komunikacije porukama između studenata i instruktora. Još jedno proširenje je dodavanje funkcionalnog kalendara koji će biti povezan s obavezama korisnika i omogućiti im lakšu preglednost i prijavu ili kreiranje termina koji nisu u konfliktu s njihovim obavezama (konflikt označava istovremeno izvođenje). U sljedećim verzijama cilj je dodati sustav obavijesti koji će korisnike prisjetiti njihovih obaveza. U trenutnoj verziji kreiran je videopoziv međutim postoji još mnogo proširenja funkcionalnosti tog videopoziva. Neka od proširenja funkcionalnosti poziva su: opcija dijeljenja ekrana za instruktore (omogućava prikaz sadržaja ostalim sudionicima poziva te instruktori lakše mogu prikazati zadatke i rješenja studentima), zajednička radna ploča (eng. „whiteboard“, po kojoj svi pripadnici poziva mogu unositi sadržaj), opcija zabrane pristupa pozivima za instruktore (neaktivnim i nekolegijalnim učenicima/studentima instruktori mogu zabraniti pristup svojim instrukcijama)... Još jedno proširenje je dodavanje pohrane u oblaku na kojoj bi instruktori mogli objaviti svoj sadržaj i materijale, a studenti ih lako preuzeti i koristiti. Aplikacija je trenutno implementirana kao web stranica, a jedan od budućih ciljeva je prebacivanje aplikacije u sustav android. U trenutnoj verziji aplikacije, termin instrukcija je javan, dostupan svima, a u budućim verzijama cilj je kreirati i privatne instrukcije koje bi instruktor prema svojem odabiru mogao naplaćivati. Također jedan od ciljeva je dodati još kategorija za termine i omogućiti formu putem koje korisnici mogu zatražiti upisivanje kategorije u bazu podataka.

Zaključak

Napretkom tehnologije mijenja se stil i kvaliteta života, a mijenjaju se i navike. Težimo prema potpunoj digitalizaciji sadržaja kako bismo naše živote učinili lakšima i pristupačnijima. Instrukcije su dio svakodnevnog života velikog broja učenika i studenata, a slabo su digitalizirane i koriste se „starinskim“ metodama poput: lijepljenje plakata, održavanje instrukcija oči u oči...

Kao rješenje tih problema kreiran je sustav za podršku interaktivnim online instrukcijama, koji će unaprijediti ovo područje i modernizirati navedene metode.

Cilj rada bio je implementirati sustav koji omogućava objavu i pretragu termina instrukcija te uspostavu komunikacije i održavanje termina unutar videopoziva. Sve značajke i elementi aplikacije, zamišljeni na početku razvoja sustava, uspješno su kreirani i mogu se koristiti, a sustav je spreman za brojna nova proširenja.

Literatura

[1] Cnectdots Video Conferencing system arhitecture, 2024, Cnectdots
Poveznica: http://www.cnectdots.com/tin-tuc/giai-phap-hoi-nghi-truyen-hinh-video-conference.html
pristupljeno 28. svibnja 2024

[2] Discord – Wikipedia
Poveznica: https://en.wikipedia.org/wiki/Discord
pristupljeno 28. svibnja 2024

[3] Teams – Wikipedia
Poveznica: https://en.wikipedia.org/wiki/Microsoft_Teams
pristupljeno 28. svibnja 2024

[4] Facebook – Wikipedia
Poveznica: https://en.wikipedia.org/wiki/Facebook
pristupljeno 28. svibnja 2024

[5] Predavanja s kolegija Programsko inžinjerstvo – Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu
Poveznica: https://www.fer.unizg.hr/
pristupljeno 28. svibnja 2024

[6] Client-server-network arhitecture, 2024, Cloudinary
Poveznica: https://res.cloudinary.com/lwgatsby/f_auto/www/uploads/2023/05/client-server-network.jpg
pristupljeno 28. svibnja 2024

[7] Khan Academys Hypertext-transfer-protocol, 2024, Khan Academy
Poveznica: https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:the-internet/xcae6f4a7ff015e7d:web-protocols/a/hypertext-transfer-protocol-http
pristupljeno 28. svibnja 2024

[8] React.js dokumentacija
Poveznica: https://react.dev/
pristupljeno 30. svibnja 2024

[9] MUI dokumentacija
Poveznica: https://mui.com/material-ui/
pristupljeno 30. svibnja 2024

[10] Flask Wikipedia

Poveznica: https://en.wikipedia.org/wiki/Flask_(web_framework)
pristupljeno 30. svibnja 2024

[11] PostgreSQL Wikipedia
Poveznica: https://en.wikipedia.org/wiki/PostgreSQL
pristupljeno 30. svibnja 2024

[12] Agora.io dokumentacija
Poveznica: https://www.agora.io/en/
pristupljeno 30. svibnja 2024

Sažetak

Naslov:

Aplikacija za podršku interaktivnim online instrukcijama

Sažetak:

Razvojem tehnologija digitaliziraju se sadržaji koji su temeljni dio naših života. Instrukcije, koje su slabije digitalizirane, temeljni su dio života velikog broja učenika i studenata. U tu svrhu kreiran je sustav za podršku interaktivnim online instrukcijama. Cilj kreiranja aplikacije bio je iskoristiti prednosti pristupačnosti i povezanosti modernih sustava u odnosu na „staromodne“ metode koje se koriste prilikom održavanje i objave instrukcija. Aplikacija je kreirana kao web stranica, korištenjem poslužitelja temeljenim na tehnologijama *React.js* i *Flaska.*

Ključne riječi:

Instrukcije, Videopoziv, Agora, React, Flask, PostgreSQL, Termini

Summary

Title:

Application for Supporting Interactive Online Instructions

Abstract:

With the development of technologies, contents that are a fundamental part of our lives are digitized. Instructions, which are less digitized, are a fundamental part of the lives of a large number of pupils and students. For this purpose, a system was created to support interactive online instructions. The goal of creating the application was to take advantage of the accessibility and connectivity of modern systems compared to the "old-fashioned" methods used when maintaining and publishing instructions. The application was created as a website, using servers based on *React.js* and *Flask* technologies.

Key words:

Instructions, Video call, Agora, React, Flask, PostgreSQL, Terms