Web-aplikacija za pronalazak i upravljanje uslugama podučavanja

Oroz, Gabriela

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:168:434835

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. 1317

WEB-APLIKACIJA ZA PRONALAZAK I UPRAVLJANJE USLUGAMA PODUČAVANJA

Gabriela Oroz

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1317

WEB-APLIKACIJA ZA PRONALAZAK I UPRAVLJANJE USLUGAMA PODUČAVANJA

Gabriela Oroz

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1317

Pristupnica:	**Gabriela Oroz (0036543637)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	prof. dr. sc. Igor Mekterović

Zadatak: **Web-aplikacija za pronalazak i upravljanje uslugama podučavanja**

Opis zadatka:

Učenici na različitim razinama, od osnovne škole do fakulteta, imaju potrebu za dodatnim podučavanjem ("instrukcijama") iz različitih predmeta. Često nije lako pronaći odgovarajućeg instruktora, te se takve usluge najčešće traže usmenim preporukama. Slično vrijedi i za instruktore, nije lako doći do šire publike, posebice u početku. Cilj rada je izrada web-platforme koja će olakšati pronalazak, dogovor i razmjenu podataka i digitalnih materijala između učenika i instruktora. Potrebno je napraviti web-aplikaciju koja će, pored administratora, imati dvije grupe registriranih korisnika: učenike i instruktore. Osim njih, inicijalni pristup platformi s osnovnim funkcijama pretraživanja treba omogućiti i anonimnim korisnicima. Učenicima treba omogućiti pretraživanje instrukcija po vrstama gradiva, geografskim i financijskim atributima, kao i pretraživanje putem upisanog slobodnog teksta. Instruktori se, nakon registracije, mogu oglašavati na web-platformi, gdje mogu staviti svoj opis djelovanja, adresu, načine podučavanja, cijene i sl. Platforma treba instruktorima i učenicima omogućiti kalendar u kojem će voditi zakazane instrukcije. Instruktori mogu, ako žele, kalendar ili dijelove kalendara učiniti javnim. Omogućiti dopisivanje i dogovor termina preko platforme. Dodatno, instruktori omogućiti da pohranjuju digitalne materijale pod svojim računom. Materijali mogu biti javni, privatni, omogućiti dijeljenje materijala s učenicima s rokom isteka. Učenicima na kraju omogućiti ocjenjivanje instruktora. Instruktorima također omogućiti da vode evidenciju o učenicima. Omogućiti komunikaciju putem platforme između svih registriranih korisnika. Razmotriti i uporabu tematskih foruma (npr. "Kemija 8. razred") koji bi bili vidljivi svima, ali bi u njima mogli pisati samo registrirani korisnici. Donijeti ocjenu ostvarenog pristupa te smjernice za budući razvoj.

Rok za predaju rada: 14. lipnja 2024.

Sadržaj

1. Uvod 4
2. Korisnički zahtjevi 5
3. Pregled mogućih rješenja 7
   3.1. Pregled sličnih projekata 7
      3.1.1. MyTutor 7
      3.1.2. Tutor Hunt 8
      3.1.3. Preply 9
4. Podatkovni model rješenja 10
5. Opis rješenja 13
   5.1. Korištene tehnologije 14
      5.1.1. React 14
      5.1.2. Next.js 15
      5.1.3. Chakra UI 16
      5.1.4. PostgreSQL 17
      5.1.5. FullCalendar 19
      5.1.6. PubNub 20
      5.1.7. Mailjet 20
      5.1.8. Supabase 21
      5.1.9. Stripe 21
      5.1.10. Typescript 22
   5.2. Arhitektura rješenja 23
   5.3. Autentifikacija i autorizacija 24
   5.4. Stranice i komponente 25
      5.4.1. Početna stranica 26
      5.4.2. Registracija i prijava 28
      5.4.3. Profil 29
      5.4.4. Početna stranica registriranog korisnika 30
      5.4.5. Instrukcije 31
      5.4.6. Kalendar 36
      5.4.7. Materijali 38
      5.4.8. Stranica za komunikaciju 39
      5.4.9. Stranica za plaćanje i ocjenjivanje 40

5.4.10. Forum 42
5.4.11. Novosti 43
Zaključak 44
Literatura 45
Sažetak 47
Summary 48

1. Uvod

Učenici i studenti na raznim razinama obrazovanja, od osnovne škole do fakulteta, ali i u dodatnim obrazovnim školama poput glazbene škole, imaju potrebu za dodatnim podučavanjem iz različitih predmeta. Međutim, pronalaženje odgovarajućeg instruktora može biti izazovno, te se takve usluge često traže usmenim preporukama. Isto tako, instruktorima može biti teško doći do šire publike, pogotovo u početku njihove karijere. Cilj je ovoga rada izraditi web aplikaciju koja će olakšati pronalazak, dogovaranje te razmjenu podataka i digitalnih materijala između studenata i instruktora.

Ovom web aplikacijom korisnicima će biti omogućeno upravljanje, pregledavanje i pretraživanje instrukcija, foruma, materijala, kalendara te će moći međusobno komunicirati. Također, studentima će biti omogućeno online plaćanje instruktora i ocjenjivanje.

2. Korisnički zahtjevi

Web aplikacija mora omogućiti registraciju korisnika kao studenta ili instruktora, ali isto tako pružiti osnovne funkcionalnosti za neregistrirane korisnike poput pregledavanja početne stranice, foruma i dostupnih instrukcija.

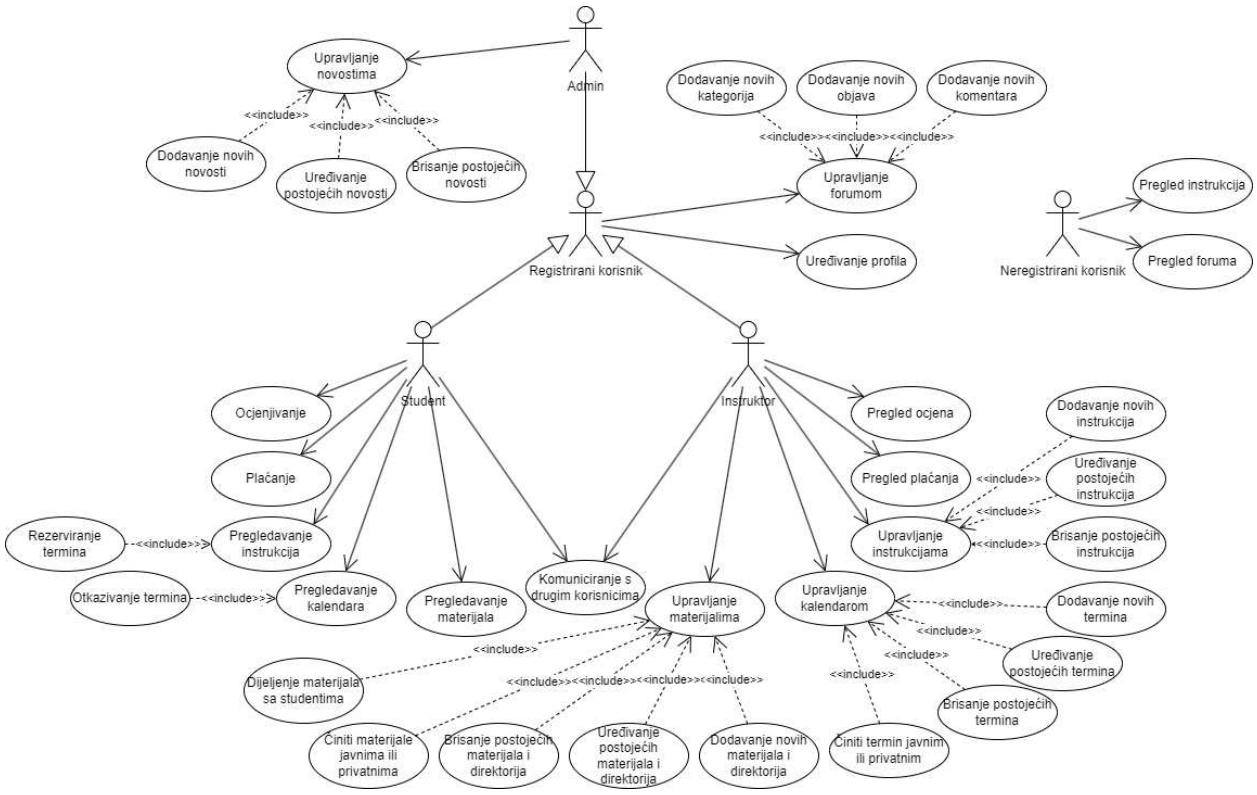
Registrirani korisnici moraju moći unijeti osobne podatke te dodavati kategorije, objavljivati i komentirati na forumu. Također bi trebali moći međusobno komunicirati te ovisno o ulozi, upravljati instrukcijama, materijalima, kalendarom, plaćanjima i ocjenjivanjem.

Instruktori kreiraju i objavljuju instrukcije, dodaju javne ili privatne materijale te ih dijele sa studentima na određeno vrijeme. Također, unose slobodne termine u kalendar te ih čine javnima ili privatnima, pregledavaju status plaćanja odrađenih termina te komentiraju studente.

Studenti pretražuju i pregledavaju instrukcije, odabiru termine, pregledavaju javne ili podijeljene materijale, vrše plaćanja odrađenih termina i ocjenjuju instruktore.

Osim ovih korisničkih uloga, postoji i administrator koji ima mogućnost kreiranje i objavljivanja novosti na početnoj stranici.

Dijagram obrasca uporabe koji opisuje ove korisničke zahtjeve prikazan je na Sl. 1.



Sl. 1. Dijagram obrasca uporabe

3. Pregled mogućih rješenja

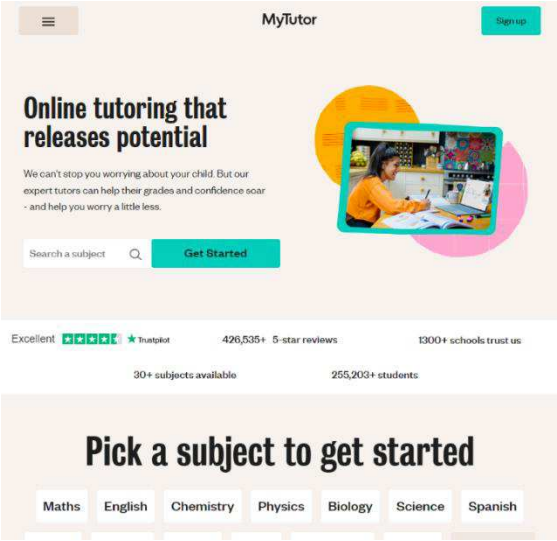
Kako bi se izgradila ova web aplikacija, potrebno je obratiti pažnju na nekoliko bitnih faktora koji će osigurati njenu funkcionalnost, sigurnost i održivost. Za sigurnost korisničkih podataka, važno je implementirati autentifikaciju i autorizaciju, a za ostvarenje funkcionalnosti aplikacije i njene održivosti, potrebno je odabrati odgovarajuće tehnologije i alate. Također treba razmisliti i o dizajnu web aplikacije.

3.1. Pregled sličnih projekata

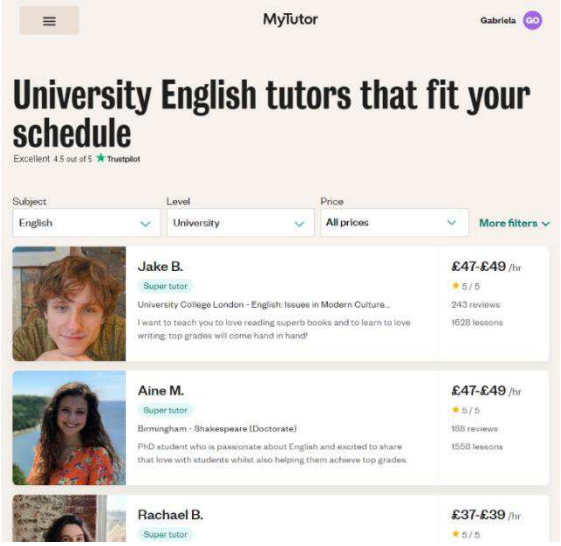
Mnoge web aplikacije su već implementirale slične funkcionalnosti. Pregledom tih aplikacija može se dobiti inspiracija za izgradnju ove web aplikacije.

3.1.1. MyTutor

MyTutor je web stranica koja povezuje studente s instruktorima za instrukcije, a osnovali su je 2013. godine Bertie Hubbard i Robert Grabiner [1]. Web stranica je razvijena korištenjem AWS (*Amazon Web Services*) usluga u oblaku i modela strojnog učenja [2]. Izgled početne stranice je prikazan na Sl. 3, a stranica za instrukcije na Sl. 2.



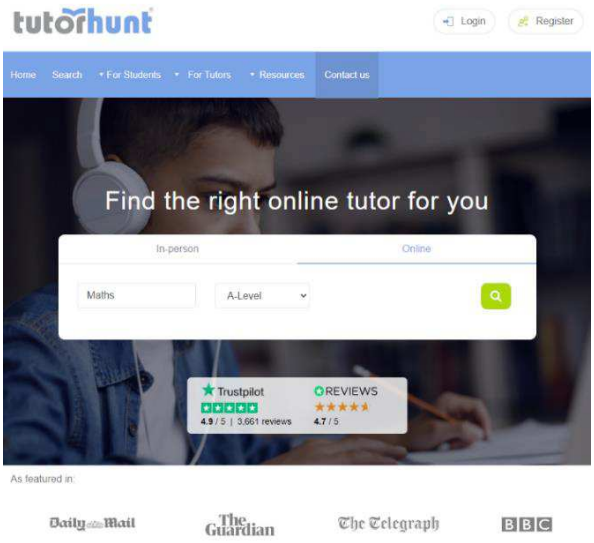
Sl. 3. MyTutor početna stranica



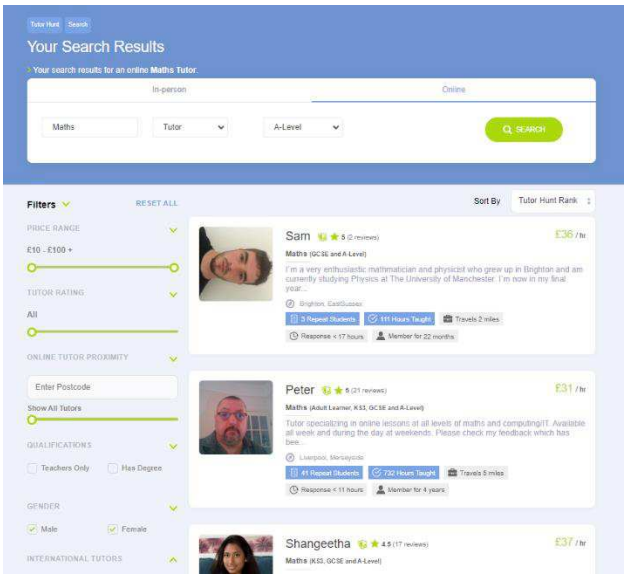
Sl. 2. MyTutor stranica za instrukcije

3.1.2. Tutor Hunt

Tutor Hunt je web stranica osmišljena za povezivanje studenata s instruktorima, koju su osnovali Mark Hughes i Andrea Ruoff 2005. godine. Stranica podržava razne alate za videopozive poput Skypea, Zooma i FaceTimea te Ziteboard za poboljšano iskustvo podučavanja [3]. Izgled početne stranice je prikazan na Sl. 4, a stranica za instrukcije na Sl. 5.



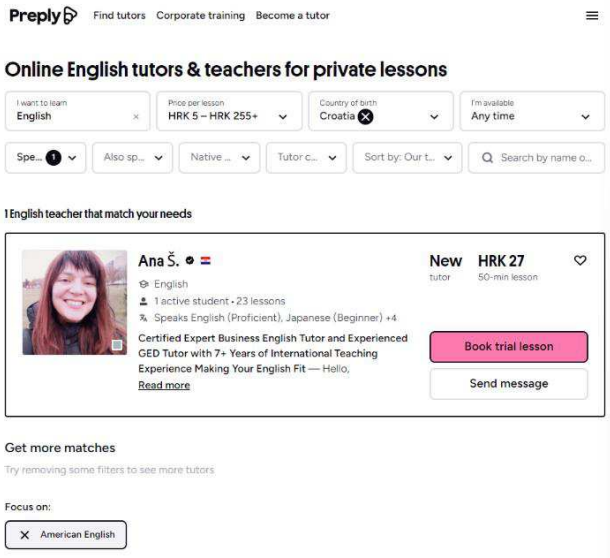
Sl. 4. Tutor Hunt početna stranica



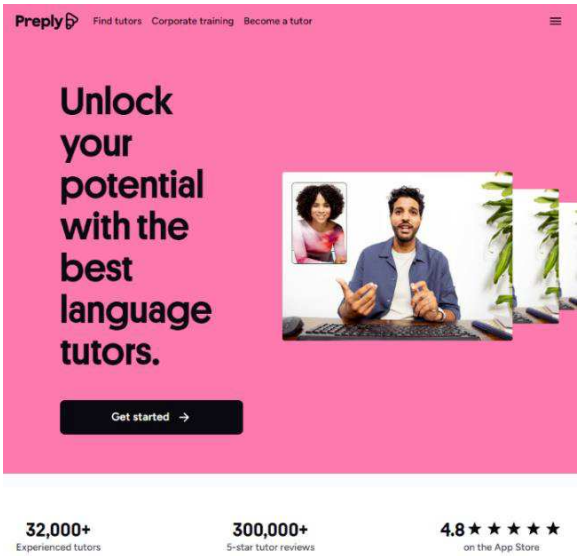
Sl. 5. Tutor Hunt stranica za instrukcije

3.1.3. Preply

Preply je web stranica za učenje jezika koja povezuje studente i profesore, a osnovali su je Kirill Bigai, Dmytro Voloshyn i Serge Lukyanov 2012. godine [4]. Izgled početne stranice prikazan je na Sl. 6, a stranice za instrukcije na Sl. 7.



Sl. 7. Preply stranica za instrukcije



Sl. 6. Preply početna stranica

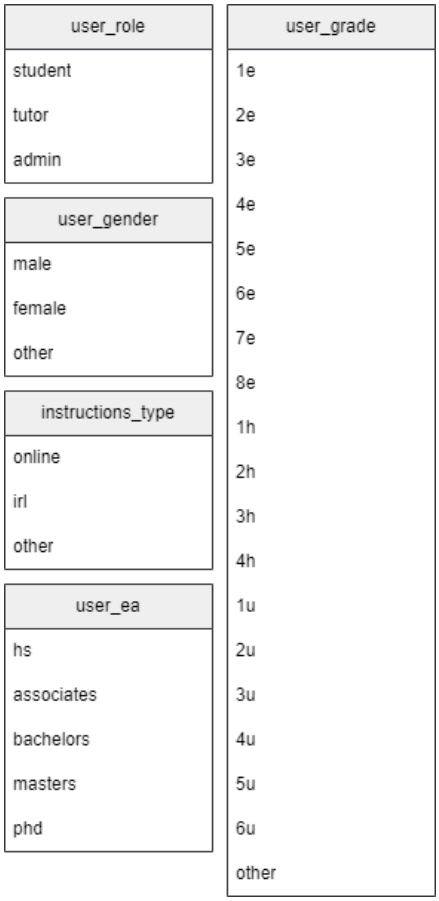
4. Podatkovni model rješenja

Podaci se za ovu web aplikaciju pohranjuju u relacijsku bazu podataka koristeći PostgreSQL. Baza podataka se sastoji od 18 tablica, od kojih su 2 kataloga i 5 enumeracija. Enumeracije su prikazane na Sl. 8, a baza podataka sa svim tablicama na Sl. 9.

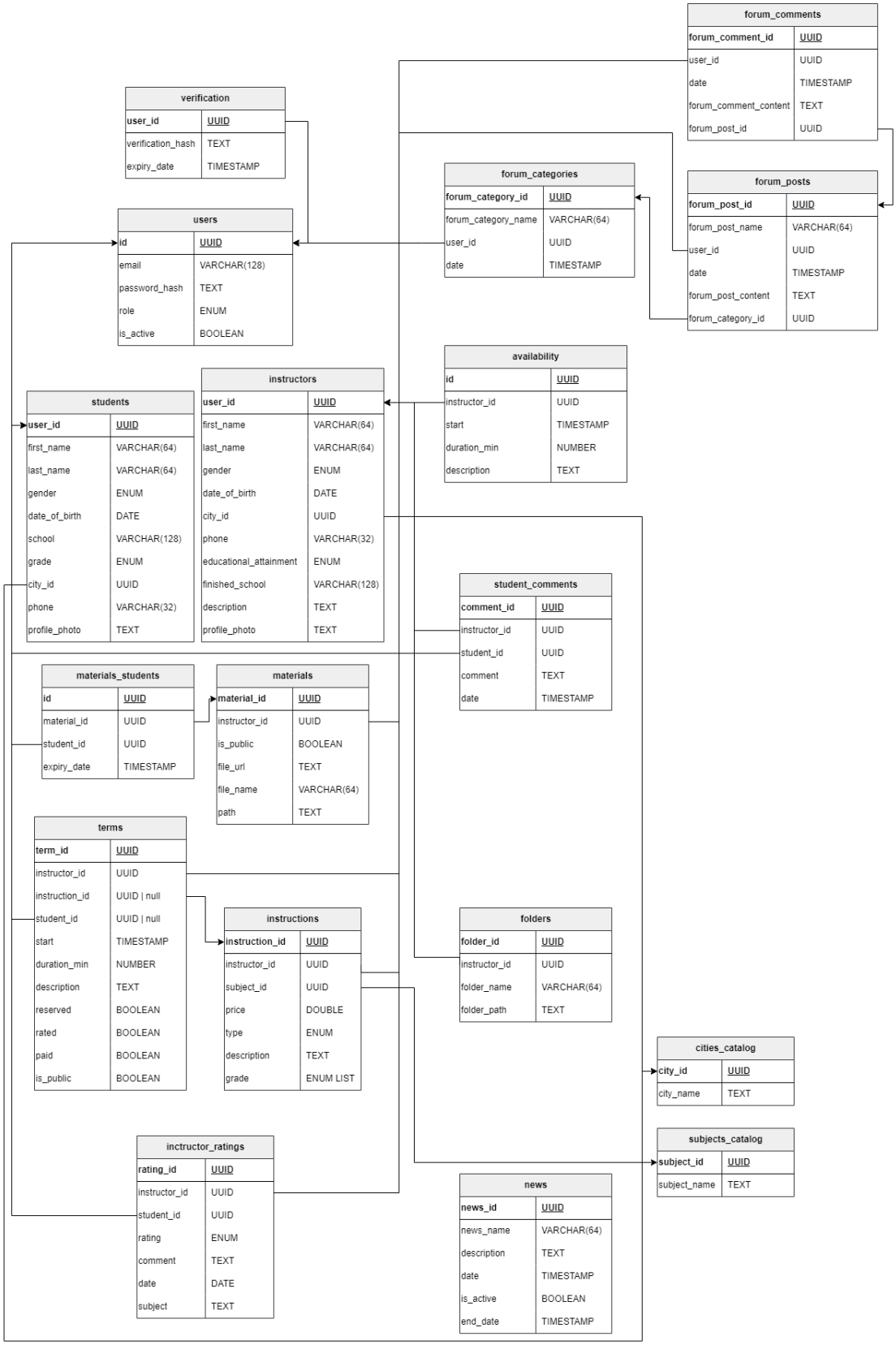
Tablice u bazi podataka međusobno su povezane primarnim ključevima. Primarni ključevi jedinstveno identificiraju svaki redak unutar tablice, dok se strani ključevi koriste za uspostavu veza između tablica. Za primarne ključeve koristi se automatsko generiranje UUID (*Universally Unique Identifier*) vrijednosti kako bi se osigurala jedinstvenost identifikatora bez potrebe za ručnim upravljanjem identifikatorima. Međutim, ovo se može ostvariti i korištenjem serijskog tipa podatka koji je u mnogim bazama podataka, poput PostgreSQLa, uobičajen za automatsko generiranje jedinstvenih primarnih ključeva.

U nekim tablicama su postavljene DEFAULT vrijednosti za određene atribute, poput is_active u tablici users koji je inicijalno postavljen na vrijednost FALSE. Također, u bazi podataka se koriste i ograničenja poput UNIQUE i NOT NULL.

Tablice kataloga sadrže skupinu podataka i osiguravaju da se referenciranjem koriste isti podaci kroz cijelu web aplikaciju. Enumeracije sadrže fiksne vrijednosti što osigurava konzistentnost podataka i olakšava njihovu obradu. Na primjer, ENUM user_role omogućuje definiranje korisničkih uloga što pojednostavljuje upravljanje privilegijama korisnika.



Sl. 8. Enumeracije za ulogu korisnika, spol korisnika, tip instrukcije, obrazovno postignuće (engl. educational attainment) korisnika i razred u školi ili godinu fakulteta



Sl. 9. Prikaz relacijske baze podataka

5. Opis rješenja

Za izradu web aplikacije, potrebna nam je klijentska strana (*frontend*), serverska strana (*backend*) te pohrana podataka (relacijska i binarna).

Klijent je dio aplikacije s kojim korisnici izravno komuniciraju putem web preglednika. Prikazuje podatke, omogućuje korisnicima interakcije s aplikacijom te dohvaća i šalje podatke uz pomoć servera. Tehnologije koje to ostvaruju u ovoj web aplikaciji su: Next.js, React, Chakra UI i TypeScript.

Server je dio aplikacije koji obrađuje zahtjeve klijenata i komunicira s bazom podataka. Ova web aplikacija koristi Next.js za izradu API pristupnih točaka (engl. *endpoint*) koje obrađuju zahtjeve klijenata i komuniciraju s bazom podataka.

Relacijska pohrana podataka služi za pohranu strukturiranih podataka u tablice, dok se binarna pohrana podataka koristi za nestrukturirane podatke poput slika, videozapisa i drugih datoteka. Iako se nestrukturirani podaci mogu pohraniti i relacijski, binarna pohrana je optimizirana za datoteke koje su memorijski zahtjevne te omogućuje bolje performanse i skalabilnost. Za ostvarenje relacijske pohrane korišten je PostgreSQL, a za binarnu pohranu Supabase.

Osim ovoga, potrebna nam je i platforma Stripe za obradu online plaćanja, PubNub za komunikaciju između korisnika u stvarnom vremenu, Mailjet za slanje e-pošte prilikom registracije, rezervacije i otkazivanja termina te FullCalendar za prikaz kalendara i interakcije s terminima.

Web aplikacija može biti izrađena kao MPA (*Multiple Page Application*), SPA (*Single Page Application*) ili hibridni tip aplikacije. MPA se sastoji od više stranica, a svaka ima posebno definiran sadržaj, obično u obliku više HTML (*HyperText Markup Language*) datoteka. SPA ima samo jedan HTML kod u kojem se sadržaj dinamički mijenja uz pomoć JavaScripta. U ovoj web aplikaciji koristimo Next.js, koji koristi hibridni pristup. Kada korisnik prvi put pristupi web aplikaciji, server generira zatraženu stranicu sa svim njenim sadržajem i dostavlja ju klijentu u obliku HTML koda. Prelaskom na neku drugu stranicu web aplikacije, klijent od servera zatraži sadržaj te stranice, koji se zatim dinamički prikazuje JavaScriptom.

5.1. Korištene tehnologije

5.1.1. React

React je JavaScript biblioteka za izgradnju korisničkih sučelja temeljenih na komponentama, a njegova ključna prednost je mogućnost iscrtavanja samo onih dijelova stranice koji su se promijenili. Komponente Reacta su ponovno upotrebljive i mogu se kombinirati. Vrijednosti se između komponenti prenose svojstvima (engl. *properties*), a komponente se automatski ažuriraju i iscrtavaju kada se podaci promijene. React nudi nekoliko kuka (engl. *hooks*) koje omogućuju upravljanje stanjima, efektima i drugim aspektima komponenti. Tablica 1. prikazuje neke od najčešće korištenih kuka i njihova objašnjenja. Postoje i prilagođene kuke (engl. *custom hooks*) koje omogućuju ponovnu upotrebu logike stanja i efekata između različitih komponenti. Također, React koristi virtualni DOM (*Document Object Model*) stvarajući virtualnu verziju stvarnih DOM elemenata. Kada se stanje ili svojstva komponenata promijene, React ažurira virtualni DOM. Ove promjene zatim uspoređuje sa stvarnim DOM-om te nakon što pronađe razlike, ažurira samo one dijelove stvarnog DOM-a koje su se promijenile. Ovaj postupak omogućuje brže i učinkovitije iscrtavanje. Uz to, React koristi JSX (*JavaScript XML*) sintaksu koja omogućuje pisanje HTML-a unutar JavaScript koda, a na kraju se JSX prevodi u JavaScript kako bi ga preglednik mogao izvršiti [5].

Tablica 1. Najčešće korištene React kuke (engl. hooks) i njihova objašnjenja

Kuka	Objašnjenje
useState	Kuka koja omogućuje definiranje varijable stanja i funkcije za ažuriranje tog stanja (Ako se koristi *useState* za npr. definiranje varijable koja prati broj klikova na gumb i funkcije koja povećava broj klikova svaki puta kada korisnik pritisne gumb, povećanje broja klikova će se automatski prikazati bez potrebe za osvježavanjem stranice.).

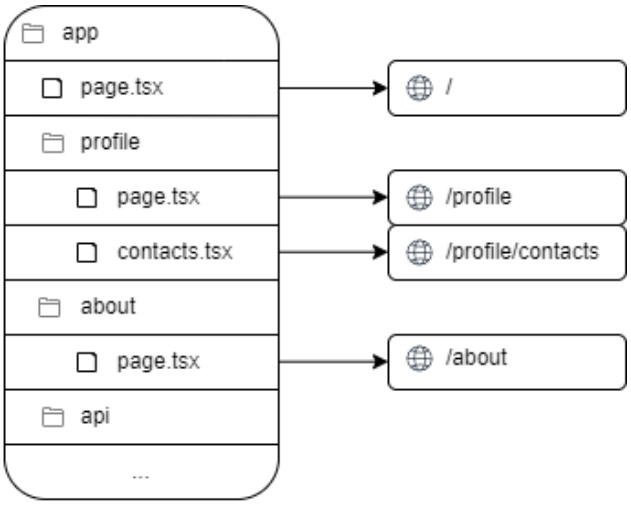
useEffect	Kuka koja omogućuje izvršavanje efekata u komponentama (npr. dohvaćanje podataka s API pristupnih točki (engl. *endpoint*) svaki puta kada se podaci promjene).
useContext	Kuka koja omogućuje komponentama da pristupe vrijednostima iz konteksta koji je definiran u višoj razini komponente (npr. kada se želi definirati boja pozadine koja će se koristiti u više komponenti, umjesto da se ta boja prosljeđuje kroz sve komponente, može se koristiti *useContext*).

5.1.2. Next.js

Next.js je radni okvir za *frontend* i *backend*, temeljen na Reactu i razvijen od strane Vercela. Integriranje *frontenda* i *backenda* unutar jednog projekta postiže se iscrtavanjem na strani poslužitelja (SSR), iscrtavanjem s klijentske strane (CSR), generiranjem statičkih stranica (SSG) te definiranjem API pristupnih točki (engl. *endpoint*) koje komuniciraju s *backendom*. SSR radi tako da se svaka zatražena stranica generira na poslužitelju prije nego što se pošalje u preglednik. U Next.jsu, SSR koristi metodu nazvanu *getServerSideProps* koja se izvršava na serveru svaki put kada se pošalje zahtjev za stranicom, a omogućuje dohvaćanje podataka na serveru i prosljeđivanje na stranicu prije nego se prikaže korisniku. Ovo ubrzava vrijeme učitavanja stranice. CSR radi tako da se početna HTML (*HyperText Markup Language*) stranica učitava iz servera, a njen sadržaj i interakcije se iscrtavaju u pregledniku. Budući da preglednik iscrtava sadržaj, početno učitavanje stranice može biti sporo, ali sva sljedeća učitavanja su brza jer preglednik ponovo iscrtava samo promijenjene dijelove stranice. Kod SSG-a stranice se generiraju samo jednom tijekom procesa izgradnje web aplikacije, što smanjuje opterećenje servera i ubrzava prikaz sadržaja. Osim toga, nudi ugrađenu metodu nazvanu *getStaticProps* koja omogućuje dohvaćanje i prosljeđivanje podataka stranicama tijekom izgradnje. To znači da se podaci već nalaze na stranici kada se prvi put učita, odnosno nema potrebe za čekanjem na odgovor servera. Ovime se podaci na stranici brže prikazuju [6].

Next.js koristi sustav usmjeravanja (engl. *routing*) koji se temelji na konvenciji imenovanja datoteka. Prije verzije 13, usmjeravanje se obavljalo u direktoriju „pages“, dok je u verziji 13 i novijima uvedena upotreba direktorija „app“. API rute se dodaju u direktorij „pages/api/“ ili „app/api/“ te se obrađujuju kao API pristupne točke (engl. *endpoint*), omogućujući integraciju backend funkcionalnosti unutar istog projekta [7]. Primjer sustava usmjeravanja Next.jsa prikazan je na Sl. 10, a inspiracija za sliku je preuzeta s [8].

Također, Next.js optimizira web stranice automatskim poboljšavanjem i organiziranjem koda. Sve JavaScript ili TypeScript datoteke grupira i minimizira kako bi smanjio veličinu datoteka i ubrzao učitavanje stranica, kod dijeli na manje dijelove što omogućuje učitavanje samo onog dijela koda koji je potreban za prikaz trenutne stranice, unaprijed preuzima resurse (npr. slike) koje će stranica trebati te optimizira slike tako što ih komprimira bez da izgube kvalitetu [9].

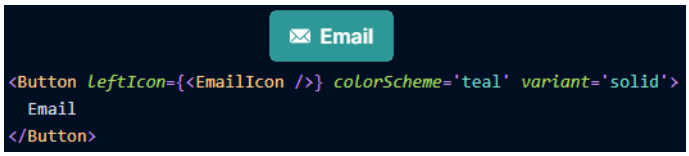


Sl. 10. Sustav usmjeravanja Next.js u verzijama 13 i novijima

5.1.3. Chakra UI

Chakra UI je biblioteka React komponenti koju je razvio Segun Adebayo. S ciljem što bolje optimizacije, ova biblioteka eliminira „mrtvi kod“ tijekom procesa izgradnje, što

pomaže u smanjenju veličine aplikacije i poboljšava performanse. Može se integrirati s Reactom, ali i okvirima koji se temelje na Reactu, poput Next.jsa. No, najbitnije kod Chakre UI su komponente. Komponente su skupovi unaprijed izgrađenih oblika ili objekata koje su responzivne, potpuno prilagodljive i jednostavno se stiliziraju putem parametara [10]. Neki od najčešće korištenih komponenti su Box koji je sličan div elementu, Button, Input, Flex, Text i Heading. Primjer stiliziranja Buttona putem parametara prikazan je na Sl. 11 [11].



Sl. 11. Primjer stiliziranja Button komponente parametrima

5.1.4. PostgreSQL

PostgreSQL je sustav za pohranu i upravljanje relacijskim bazama podataka koji naglašava proširivost i usklađenost sa SQL standardima, razvijen na Sveučilištu u Berkeleyju, Kalifornija. Poznat je po svojoj pouzdanosti, integritetu podataka i pridržavanju standardu ACID (atomnost, konzistentnost, izolacija i trajnost). Podržava različite tipove podataka, uključujući polja, geometrijske podatke, JSON i XML te je optimiran za brzinu i učinkovitost [12] [13]. Tablica 2. prikazuje sve tipove koje PostgreSQL podržava te njihova objašnjenja. Također, PostgreSQL podržava složene upite, transakcije i opsežne operacije za pohranu podataka.

Tablica 2. Tipovi podataka koje PostgreSQL podržava i njihova objašnjenja

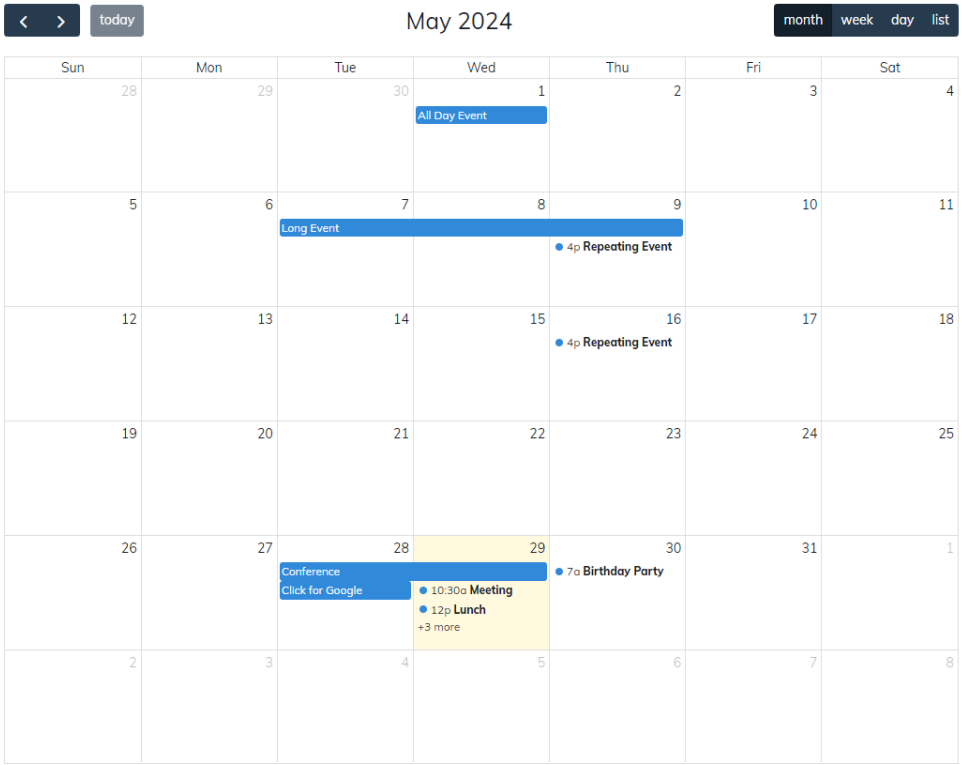
Tip podatka	Objašnjenje
Broj (engl. *numeric*)	Koristi se za pohranu brojeva različitih preciznosti i veličina (cijeli brojevi, decimalni brojevi, brojevi s pomičnim zarezom).

Znak (engl. *character*)	Koristi se za pohranu niza znakova (slova, brojevi, posebni znakovi). Postoje različiti podtipovi poput CHAR, VARCHAR i TEXT koji se razlikuju po duljini i upotrebi memorije.
Datum/vrijeme (engl. *date/time*)	Koristi se za pohranu datuma i vremena.
Novčani (engl. *monetary*)	Koristi se za pohranu novčanih vrijednosti (obično u lokalnoj valuti).
Binarni (engl. *binary*)	Koristi se za pohranu binarnih podataka (slike, zvukovi, dokumenti).
Logički (engl. *boolean*)	Koristi se za pohranu vrijednosti „istinito“ (engl. *true*) ili „lažno“ (engl. *false*).
Enumeracija (engl. *enumerated*)	Omogućuje definiranje ograničenog skupa mogućih vrijednosti.
Geometrijski (engl. *geometric*)	Koristi se za pohranu geometrijskih oblika (točka, linija, poligon).
Pretraživanje teksta (engl. *text search*)	Koristi se za indeksiranje i pretraživanje teksta.
UUID (Universally Unique Identifier)	Koristi se za pohranu univerzalno jedinstvenih identifikatora.
Tipovi mrežnih adresa (engl. *network address types*)	Koristi se za pohranu mrežnih adresa.
JSON (JavaScript Object Notation)	Koristi se za pohranu struktura podataka u JSON formatu.
Bitni niz (engl. *bit string*)	Koristi se za pohranu niza bitova.
XML (eXtensible Markup Language)	Koristi se za pohranu XML datoteka.
Raspon (engl. *range*)	Koristi se za pohranu raspona vrijednosti, poput raspona datuma ili brojeva.
Polja (engl. *arrays*)	Koristi se za pohranu više vrijednosti u jednom polju.
Složeni (engl. *composite*)	Omogućuje kombiniranje više vrijednosti u jedan tip.
Identifikatori objekata (engl. *object identifiers*)	Koristi se za pohranu identifikatora objekata.

Pseudo	Koristi se za posebne operacije ili za prilagodbu funkcionalnosti.
pg-lsn (PostgreSQL Logical Sequence Number)	Koristi se za pohranu LSN (brojevi logičkog slijeda) koji omogućuju praćenje i upravljanje promjenama u bazi podataka.

5.1.5. FullCalendar

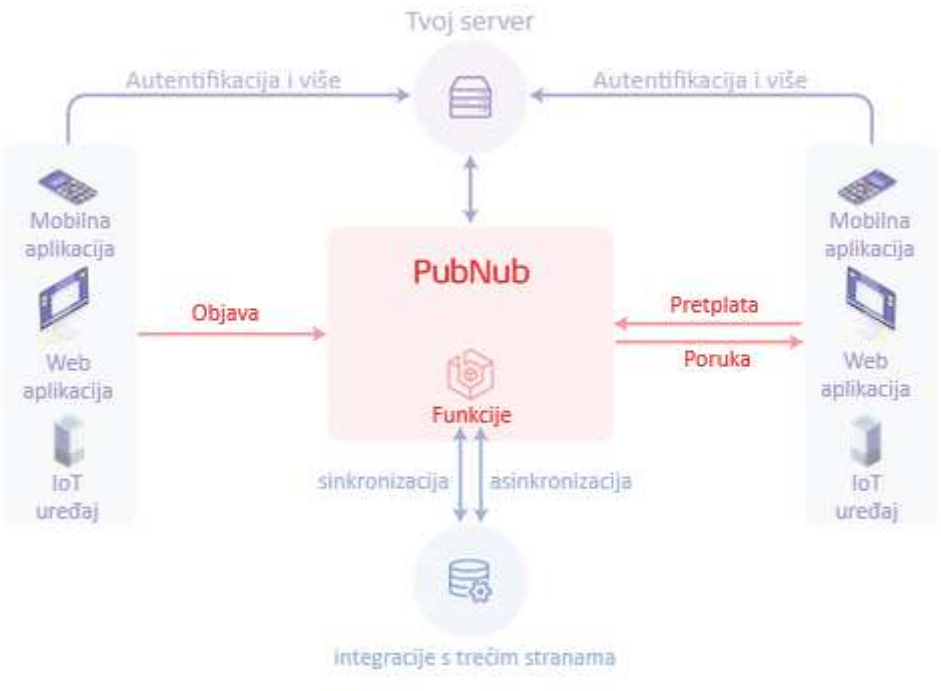
FullCalendar je JavaScript biblioteka za prikazivanje i upravljanje kalendarima i događajima unutar web aplikacije, koju je razvio Adam Shaw. Ova biblioteka nudi prilagodljive prikaze poput mjeseca, tjedna, dana i dnevnog reda te omogućuje prilagodbu izgleda kalendara i događaja kako bi se uskladio s dizajnom aplikacije. Također, korisnici mogu jednostavno kreirati, klikati, povlačiti i ispuštati događaje na odgovarajuće vremenske odsječke ili datume. Primjer pregleda FullCalendara na razini mjeseca nalazi se na Sl. 12 [15].



Sl. 12. Primjer pregleda FullCalendara na razini mjeseca, svibanj 2024. godine

5.1.6. PubNub

PubNub je platforma za razmjenu poruka u stvarnom vremenu temeljena na oblaku koja omogućuje brzu i pouzdanu komunikaciju između korisnika, a razvili su ju Stephen Blum i Todd Greene. Platforma je razvijena s fokusom na skalabilnost, pouzdanost i brzinu. PubNub arhitektura prikazana je na Sl. 13 [16].



Sl. 13. Arhitektura PubNuba za mobilne aplikacije, web aplikacije i IoT uređaje

5.1.7. Mailjet

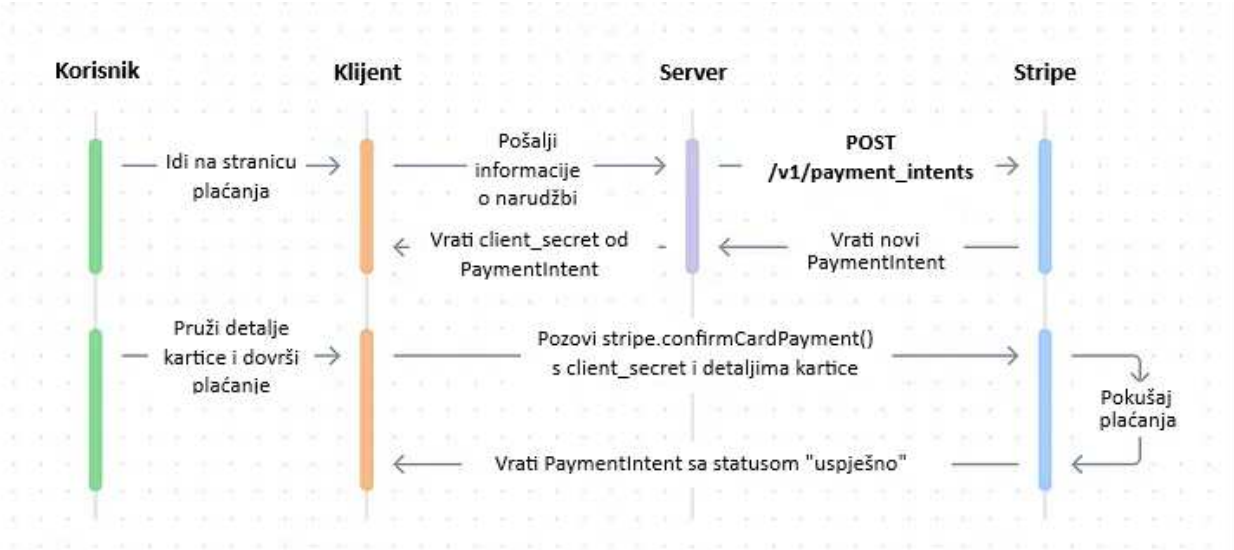
Mailjet je platforma za slanje i upravljanje e-poštama i SMS-om. Nudi sučelje koje omogućuje korisnicima da brzo stvore i prilagode predloške e-pošte, ali i integraciju s raznim programskim jezicima i platformama [17].

5.1.8. Supabase

Supabase je platforma koja olakšava izgradnju skalabilnih i sigurnih aplikacija. Ova platforma dolazi s ugrađenim značajkama autentifikacije i autorizacije putem adrese e-pošte, telefona ili društvenih mreža. Koristi PostgreSQL bazu podataka i omogućuje pregledavanje tablica, stvaranje veza između tablica i pokretanje SQL upita putem grafičkog sučelja. Supabase pruža i pohranu i posluživanje datoteka poput slika, videozapisa i dokumenata te ažuriranja podataka u stvarnom vremenu [18].

5.1.9. Stripe

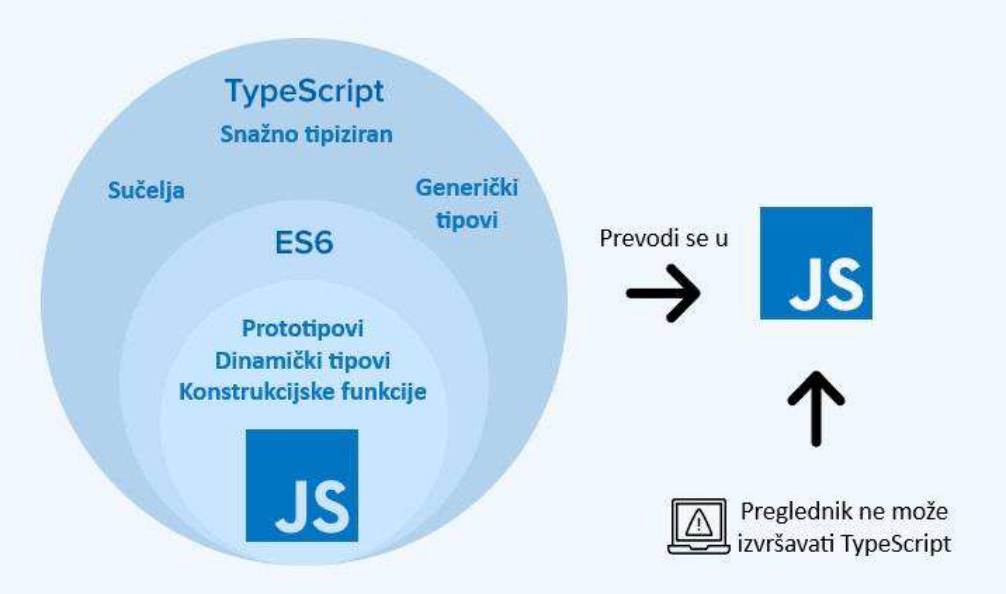
Stripe je platforma za obradu online plaćanja. Podržava razne načine plaćanja, uključujući kreditne i debitne kartice, ACH (*Automated Clearing House*) prijenose i digitalne novčanike poput Apple Pay i Google Pay. Također, omogućuje obradu plaćanja u više valuta [19]. Stripe koristi sigurnosne mjere kao što su tokenizacija, enkripcija podataka i otkrivanje prijevara temeljeno na strojnom učenju. Proces plaćanja sa Stripeom prikazan je na Sl. 14 [20].



Sl. 14. Proces plaćanja sa Stripeom

5.1.10. Typescript

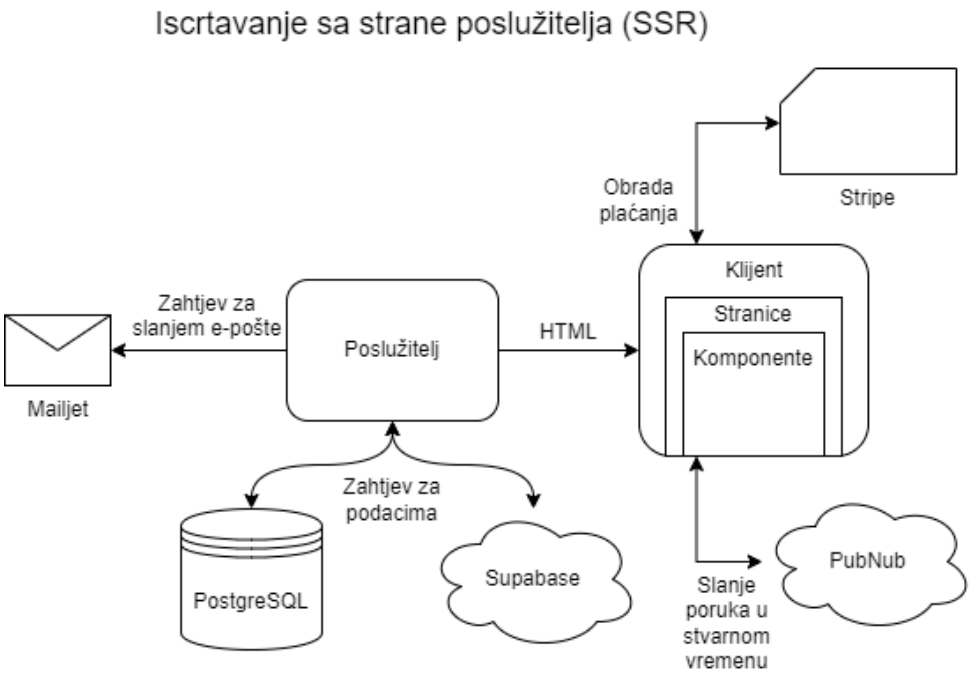
Typescript je strogo tipizirani programski jezik temeljen na JavaScriptu, razvijen od Microsofta. Typescript uvodi statičke tipove u JavaScript, odnosno varijablama, funkcijskim parametrima i povratnim vrijednostima se dodjeljuju određeni tipovi, što omogućuje rano otkrivanje pogrešaka. Također, TypeScript može sam zaključiti kojeg je tipa neka varijabla na temelju konteksta koda. Prije nego što se kod pokrene u pregledniku ili Node.js okruženju, treba ga prevesti u JavaScript pomoću TypeScript prevoditelja. Pregled rada TypeScripta prikazan je na Sl. 15 [21].



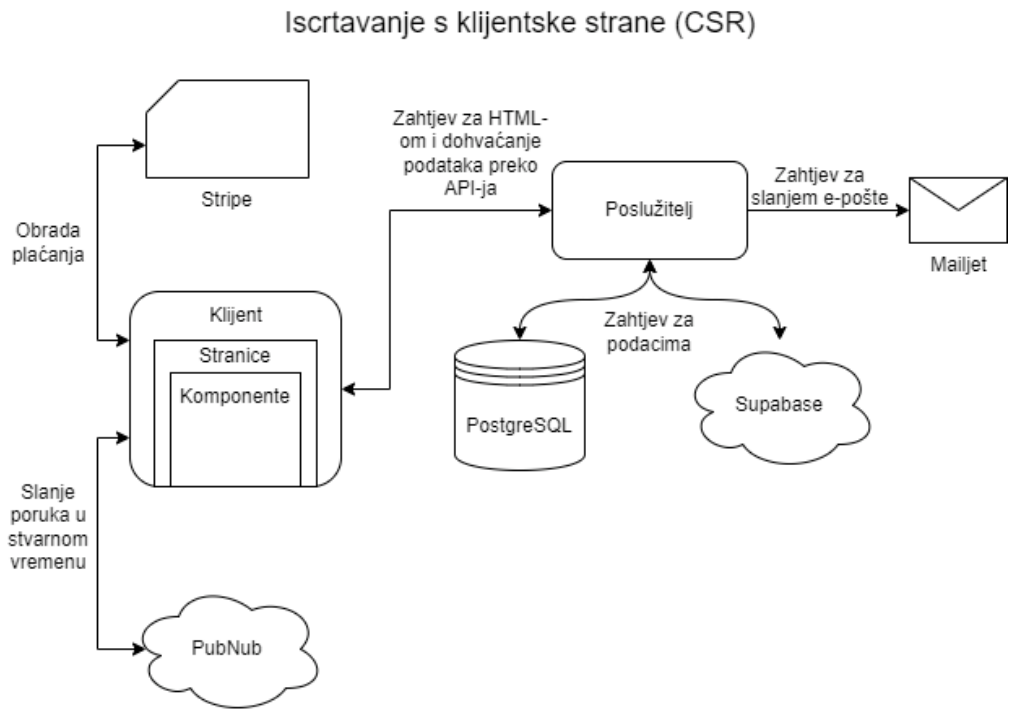
Sl. 15. Pregled rada TypeScripta koji se prevodi u JavaScript kako bi se mogao prikazati na pregledniku

5.2. Arhitektura rješenja

Za izradu ove web aplikacije korišten je Next.js kao sveobuhvatni (engl. *full-stack*) radni okvir koji omogućuje integriranje *frontenda* i *backenda* u jednom projektu te kombiniranje iscrtavanja sa strane poslužitelja (SSR) i iscrtavanje s klijentske strane (CSR). Kod SSR, poslužitelj dohvaća sve potrebne podatke te generira stranicu s tim podacima, a zatim ju iscrtava na klijentu. Kod CSR, klijent učitava stranicu, a nakon toga preko poslužitelja dohvaća potrebne podatke i automatski ažurira sadržaj stranice. Korisniku se prikazuju stranice sastavljene od React komponenti (npr. zaglavlje, podnožje). Podaci se dohvaćaju i spremaju korištenjem baze podataka PostgreSQL, a materijali za instrukcije korištenjem Supabase. Plaćanje je ostvareno uz pomoć Stripea, slanje e-pošte korisnicima uz pomoć Mailjeta, a za komunikaciju u stvarnom vremenu je korištena PubNub platforma. Prikaz arhitekture web aplikacije kada se koristi SSR nalazi se na Sl. 16, a prikaz kada se koristi CSR na Sl. 17. Inspiracija za slike preuzeta je s [22].



Sl. 16. Prikaz arhitekture web aplikacije kada se koristi SSR



Sl. 17. Prikaz arhitekture web aplikacije kada se koristi CSR

5.3. Autentifikacija i autorizacija

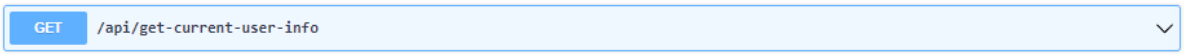
Novi se korisnik može registrirati u web aplikaciju putem registracijskog obrasca u koji unosi adresu e-pošte i lozinku. Lozinka se šifrira uz pomoć algoritma *bcrypt* prije nego što se pohrani u bazu podataka. Budući da je *bcrypt* algoritam za šifriranje dizajniran da bude spor i zahtjeva puno resursa, otporan je na napade uzastopnim pokušavanjem (engl. *brute force*). Također, svaka lozinka je šifrirana uz pomoć soli (engl. *salt*), odnosno nasumično generiranog niza znakova, koji se dodaje lozinki prije šifriranja i tako osigurava da čak i ako su dvije lozinke jednake, njihovi šifrati će biti različiti. Prilikom prijave, korisnik ponovo unosi adresu e-pošte i lozinku, koja se zatim provjerava. Provjera se obavlja uz pomoć funkcije *compare* iz algoritma *bcrypt* koja uspoređuje upisanu lozinku s prethodno pohranjenom šifriranom lozinkom. Ako je prijava bila uspješna, generira se sjednički token (engl. *session token*) koji sadrži korisnički ID, ulogu i adresu e-pošte. Token se šifrira JWT-om (*JSON Web Token*) i pretvara u kolačić (engl. *cookie*) koji se sprema u preglednik korisnika. Za dohvaćanje podataka o trenutnom korisniku, web aplikacija koristi funkciju *decrypt* koja dešifrira JWT

kolačić kako bi dobila informacije. Ovisno o ulozi, korisniku su dostupne određene stranice i određene funkcionalnosti.

Rute za prijavu, odjavu i registraciju prikazane su na Sl. 18.

Sl. 18. API rute za prijavu, odjavu i registraciju

Ruta za dohvaćanje podataka o trenutnom korisniku prikazana je na Sl. 19.



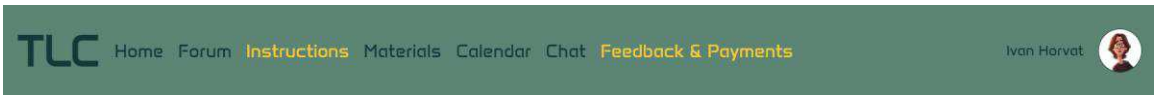
Sl. 19. API ruta za dohvaćanje podataka o trenutnom korisniku

5.4. Stranice i komponente

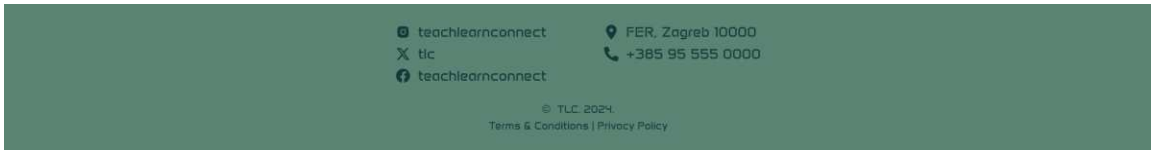
Web aplikacija se sastoji od 12 stranica i 17 komponenti. Osnovne komponente su zaglavlje (engl. *Header*) i podnožje (engl. *Footer*) na početnoj stranici te zaglavlje za registrirane korisnike na svim ostalim stranicama.

Zaglavlje početne stranice sadrži opcije za registraciju, prijavu ili odjavu, omogućuje pristup forumu, instrukcijama, kontaktima, informacijama o web aplikaciji i vijestima te je prikazano na Sl. 20. Podnožje je prikazano na slici Sl. 21, a zaglavlje registriranih korisnika, odnosno instruktora, na slici Sl. 22.

Sl. 20. Zaglavlje početne stranice



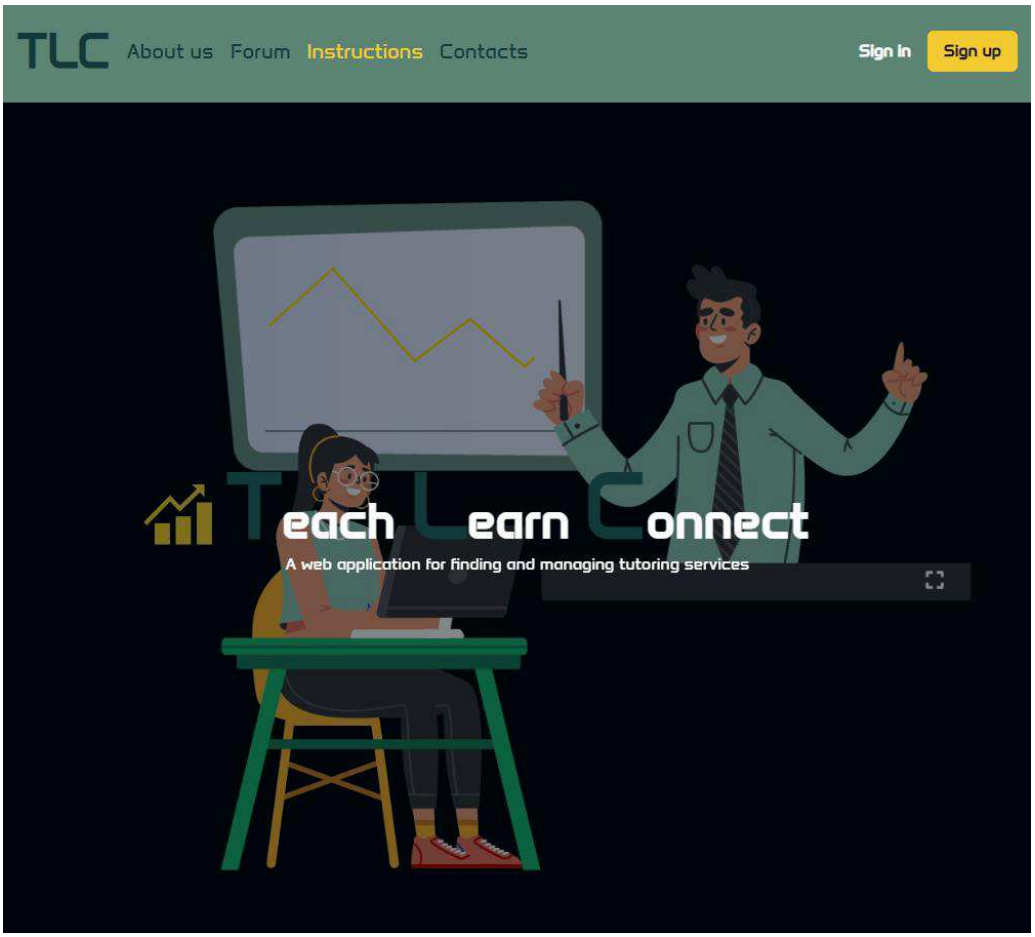
Sl. 21. Zaglavlje za instruktora



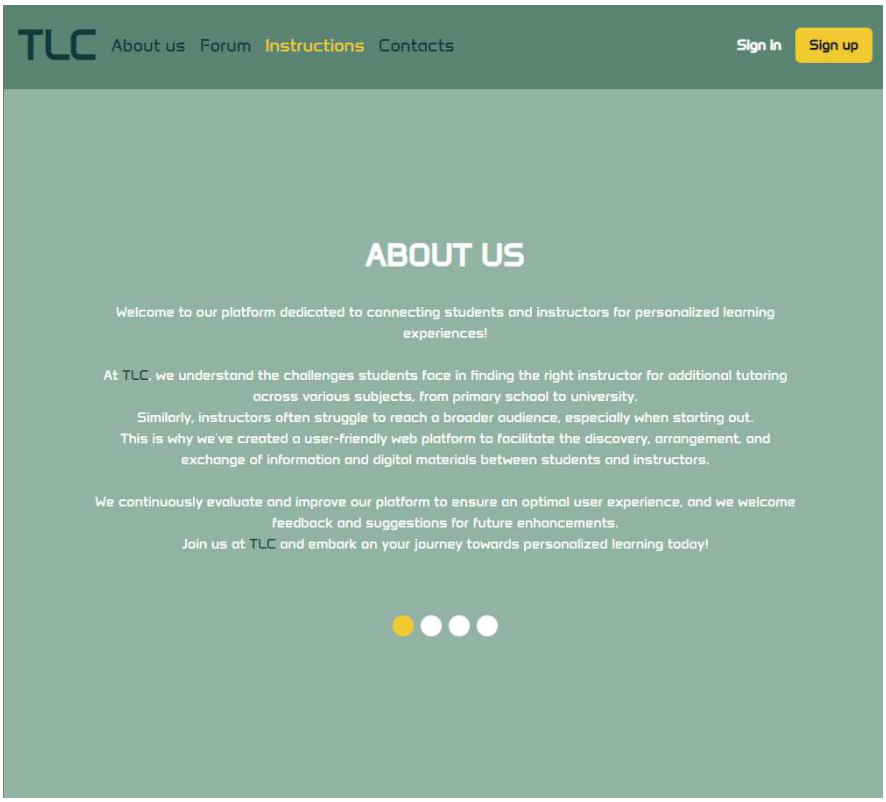
Sl. 22. Podnožje web aplikacije

5.4.1. Početna stranica

Početna stranica sadrži zaglavlje koje omogućuje navigaciju do kontakata te informacija o web aplikaciji i novostima. Klikom na jedan od navedenih izbora, korisnici se automatski pozicioniraju na odgovarajući dio stranice putem glatkog posmaka (engl. *smooth scroll)*. Vijesti i informacije o web aplikaciji se dinamično izmjenjuju svakih pet sekundi, a klikom na kružiće korisniku se omogućuje odabir sadržaja koji želi čitati. Početna stranica je prikazana na Sl. 23, a dio stranice na kojoj su informacije o web aplikaciji i novostima je prikazan na Sl. 24.



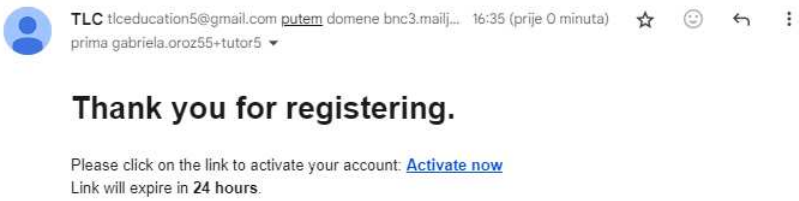
Sl. 23. Početna stranica



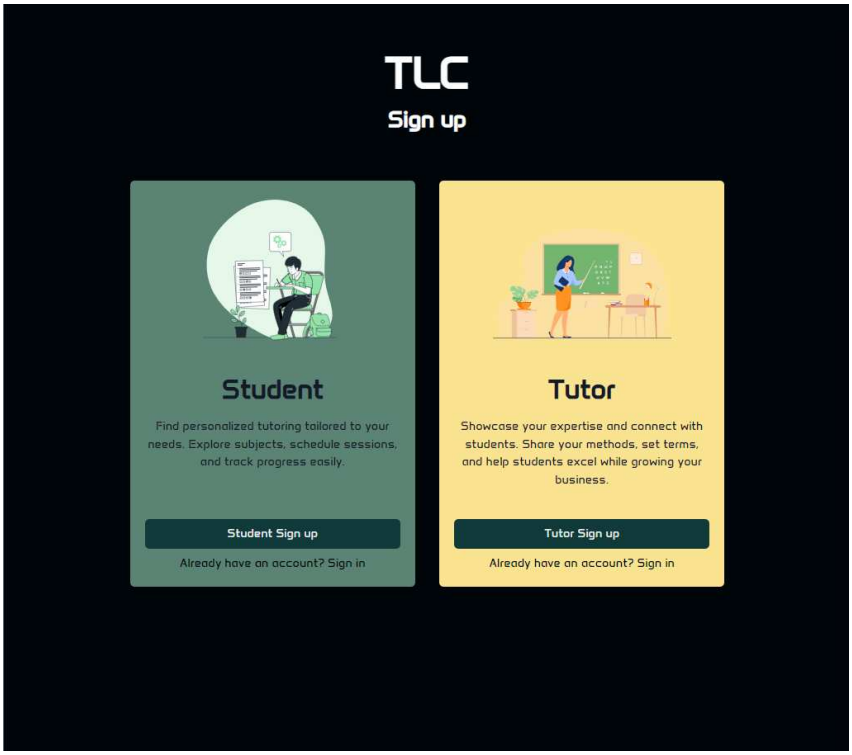
Sl. 24. Novosti i informacije o web aplikaciji na početnoj stranici

5.4.2. Registracija i prijava

Novi korisnik se može registrirati u web aplikaciju ulogom studenta ili instruktora. Prilikom registracije, korisnik unosi adresu e-pošte i lozinku. Nakon registracije, koristeći uslugu Mailjet korisniku se automatski šalje e-pošta za potvrdu registracije. Tek nakon što korisnik potvrdi svoju registraciju, omogućena mu je prijava u aplikaciju i korištenje svih njezinih funkcionalnosti. Primjer e-pošte koja se dobije za potvrdu registracije prikazana je na Sl. 25, a stranica za registraciju na Sl. 26.



Sl. 25. E-pošta za potvrdu registracije



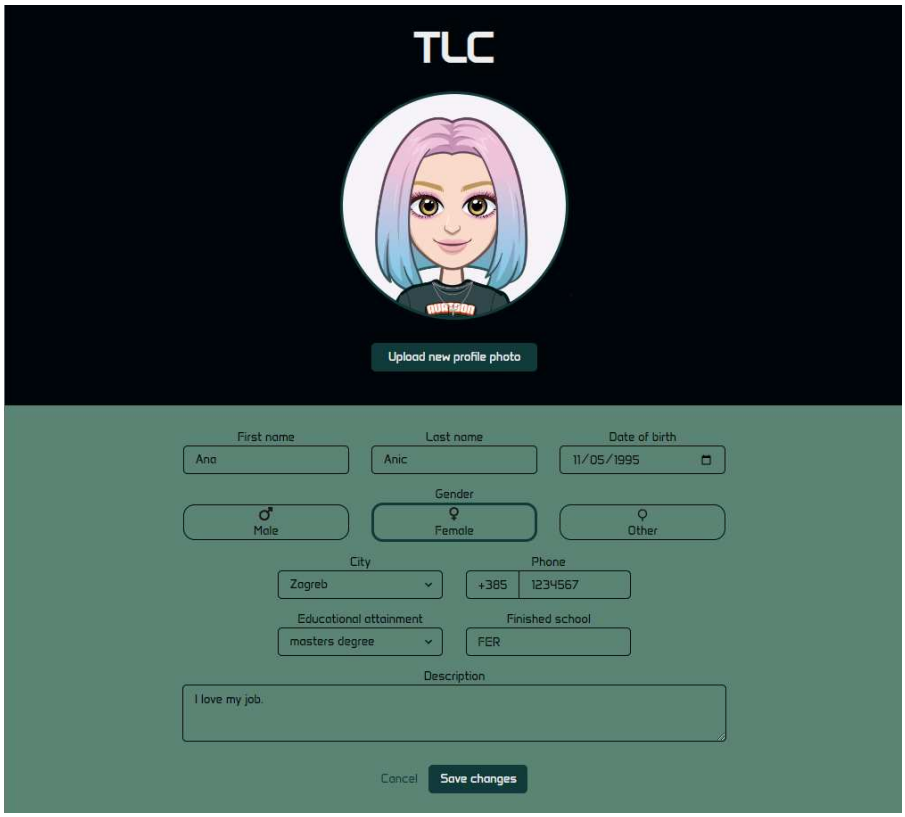
Sl. 26. Stranica za registraciju

5.4.3. Profil

Nakon što novi korisnik dovrši registraciju i potvrdi svoj račun, otvara mu se stranica za postavljanje profila. Ovisno o odabranoj ulozi, ispunjava osobne podatke. Svi korisnici unose ime, prezime, datum rođenja, spol, mjesto stanovanja, broj mobitela te postavljaju profilnu sliku. Studenti dodatno unose naziv škole ili fakulteta koji pohađaju te razred ili godinu. Instruktori dodatno unose svoje obrazovno postignuće, naziv završene škole ili fakulteta te opis o sebi.

U bilo kojem trenutku, korisnik se može vratiti na stranicu za postavljanje profila i izmijeniti svoje podatke.

Stranica za uređivanje profila prikazana je na Sl. 27.



Sl. 27. Stranica za uređivanje profila

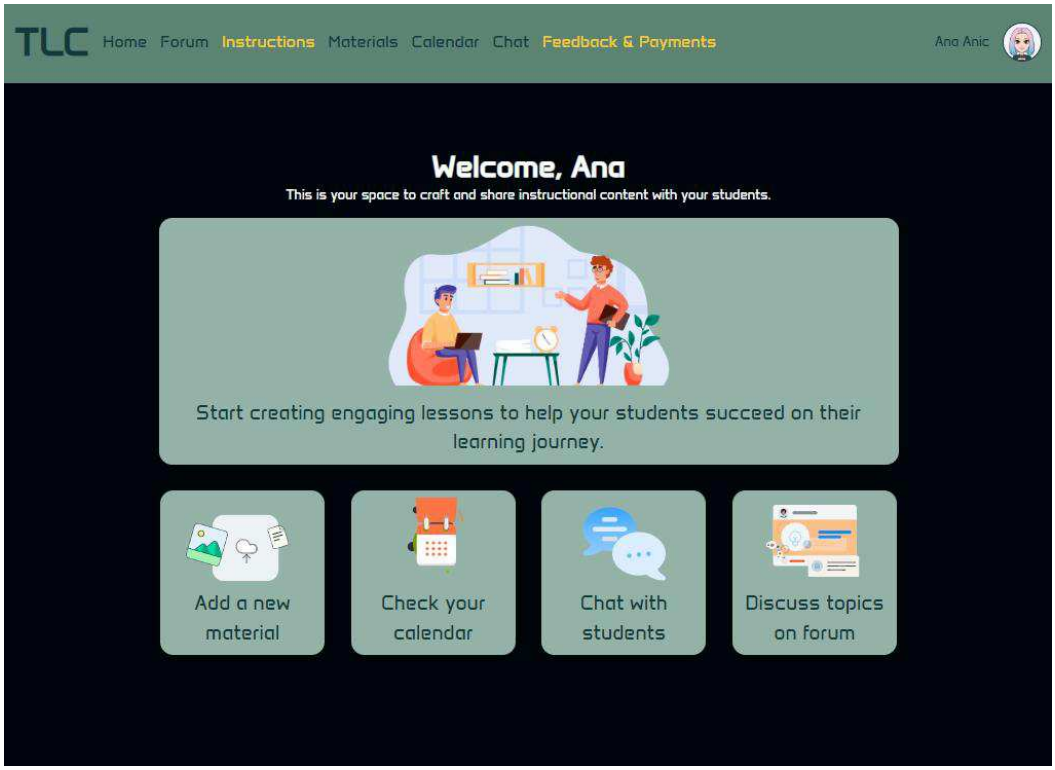
5.4.4. Početna stranica registriranog korisnika

Na početnoj stranici registriranog korisnika nalazi se pregled svih mogućnosti koje korisnik može iskoristiti u web aplikaciji.

Za studente to je pregled svih instrukcija, materijala koji su s njim/njom podijeljeni, kalendara s rezerviranim terminima instrukcija, foruma i kanala za komunikaciju s instruktorima, a dodatno u zaglavlju imaju opciju navigacije na stranicu za plaćanje i ocjenjivanje instruktora.

Za instruktore to je pregled osobnih instrukcija, materijala, kalendara sa slobodnim i rezerviranim terminima instrukcija, foruma i kanala za komunikaciju sa studentima, a dodatno u zaglavlju imaju opciju navigacije na stranicu za pregled plaćenih i ne plaćenih instrukcija te dobivenih ocjena od strane studenata. Izgled početne stranice za instruktore je prikazan na slici Sl. 28.

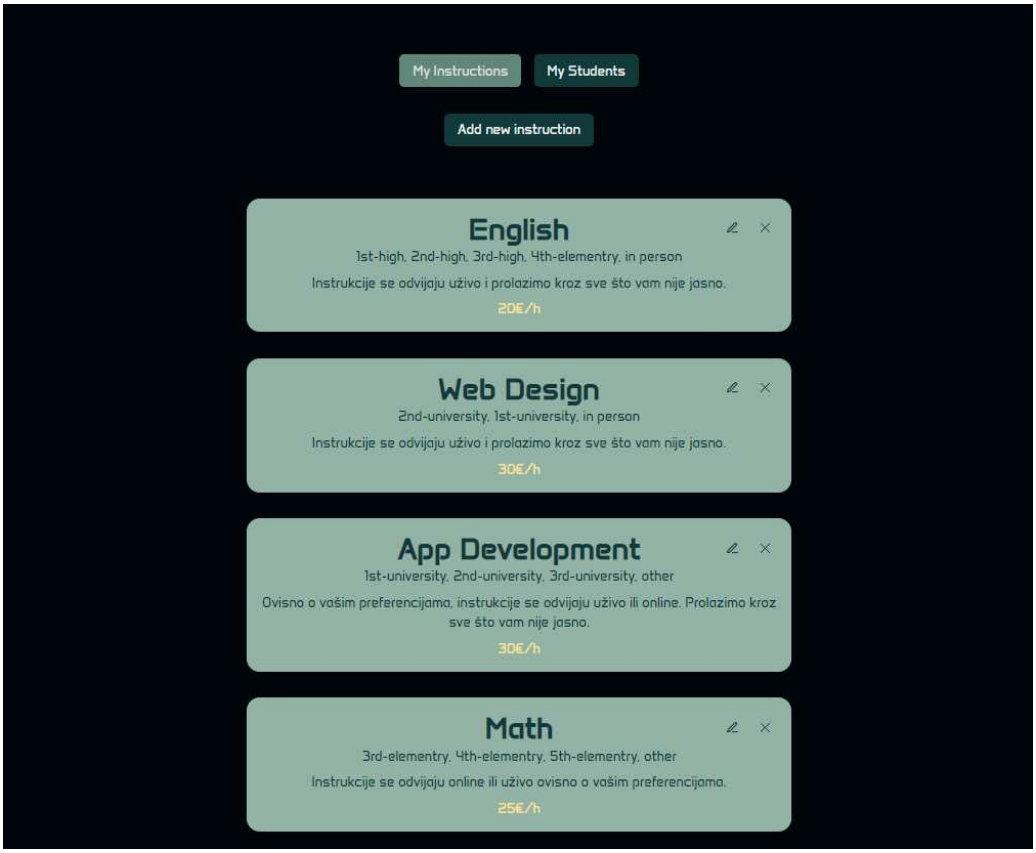
Adminima se nudi opcija dodavanja novih vijesti na početnu stranicu.



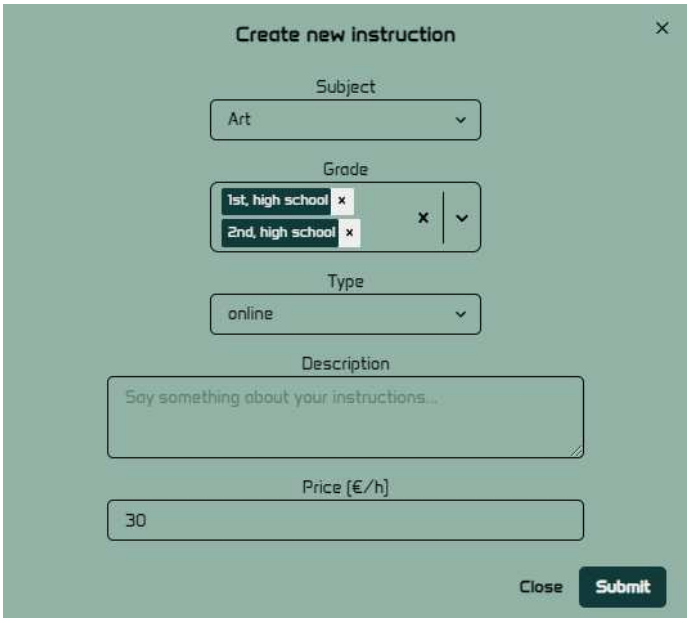
Sl. 28. Početna stranica instruktora

5.4.5. Instrukcije

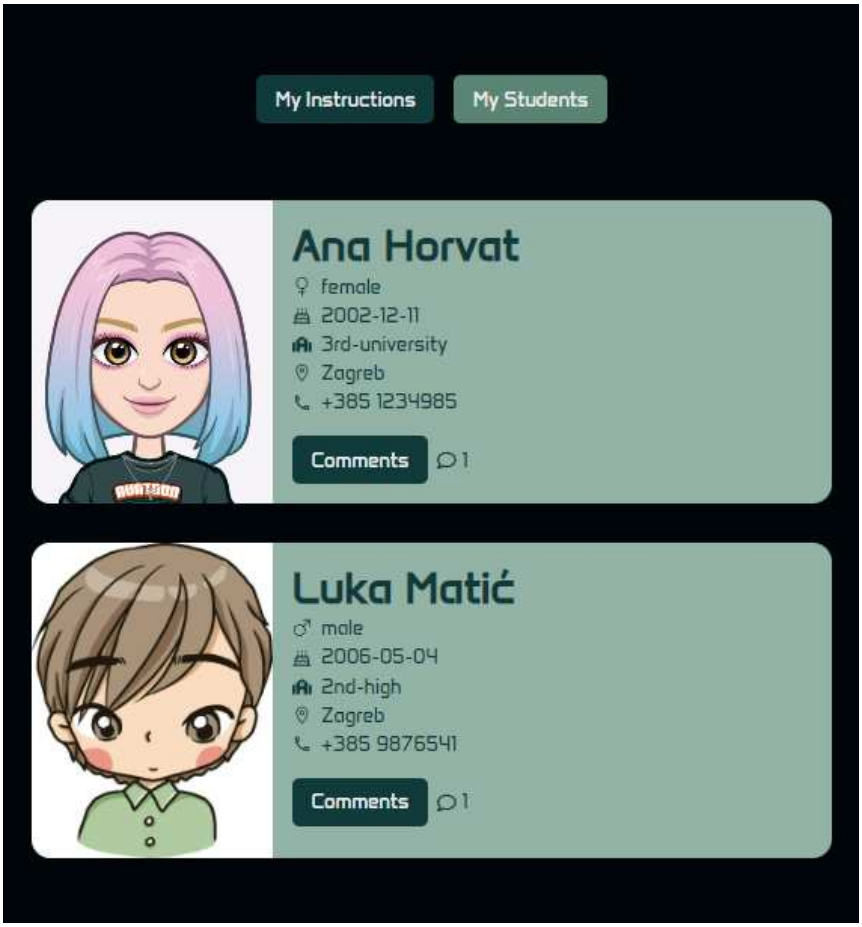
Na stranici za instrukcije, instruktori imaju mogućnosti kreiranja, uređivanja, brisanja i pregledavanja svojih instrukcija. Prilikom kreiranja nove instrukcije, instruktori unose naziv predmeta koji će predavati, razrede za koje su instrukcije namijenjene, način predavanja (online, uživo, drugo), opis te cijenu. Ovo su također opcije koje se mogu mijenjati prilikom uređivanja postojeće instrukcije. Osim toga, instruktori mogu pregledavati svoje studente i dodavati komentira o njima. Komentare na studente može vidjeti samo onaj instruktor koji je kreirao taj komentar. Instruktorova stranica za instrukcije je prikazana na Sl. 29, kreiranje nove instrukcije na Sl. 30., pregledavanje studenata na Sl. 31, a komentiranje studenta na Sl. 32.



Sl. 29. Pregledavanje i upravljanje instrukcijama za instruktore



Sl. 30. Kreiranje nove instrukcije

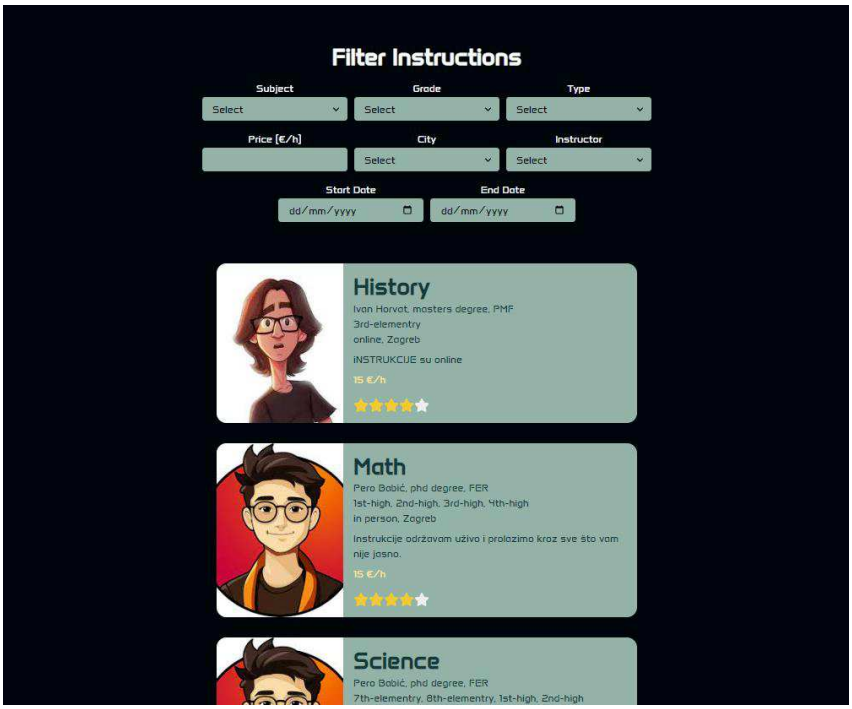


Sl. 31. Pregledavanje studenata

Sl. 32. Komentiranje studenta

Studenti i neregistrirani korisnici mogu pregledavati sve dostupne instrukcije poredane prema prosječnoj ocjeni instruktora koji ih predaje. Također, imaju mogućnost pretraživanja instrukcija po nazivu predmeta, razredu, načinu predavanja, cijeni, lokaciji

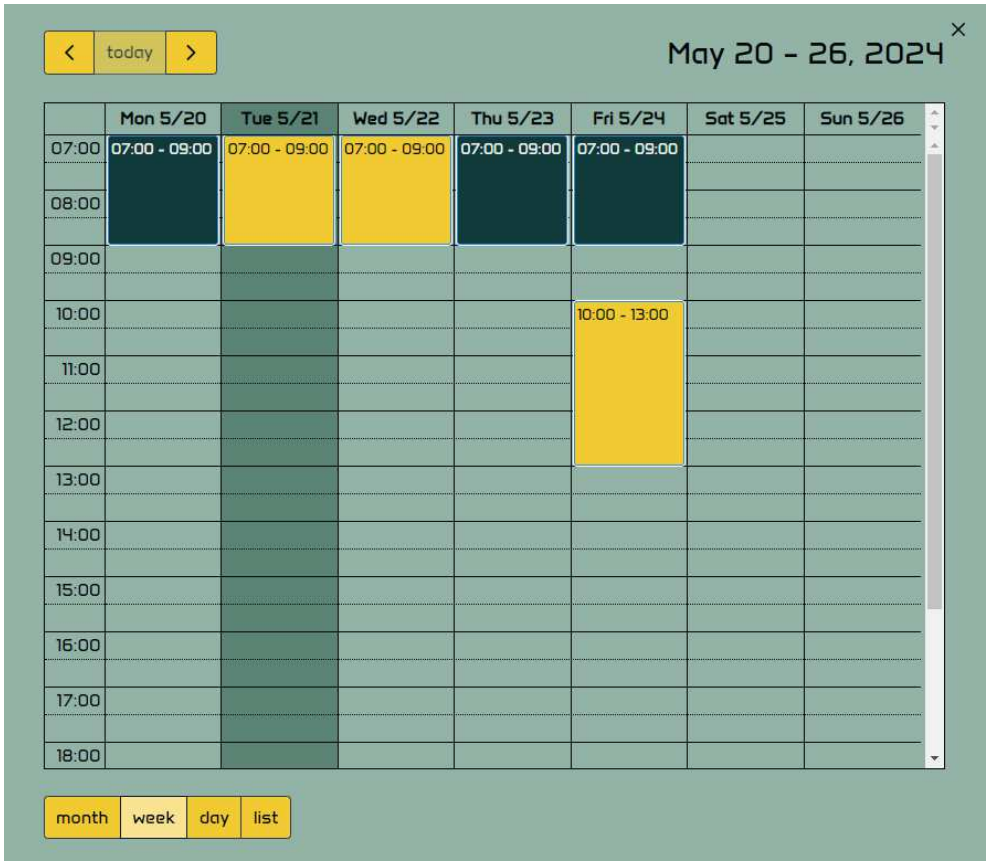
održavanja instrukcija, instruktoru i dostupnim datumima. Klikom na određenu instrukciju, studenti dobivaju detaljan uvid u instruktora, instrukciju, javne materijale instruktora te dostupne termine. Stranica instrukcija za studente i neregistrirane korisnike je prikazana na Sl. 33, a uvid u instruktora, instrukcije, javne materijale i kalendar na slici Sl. 34.



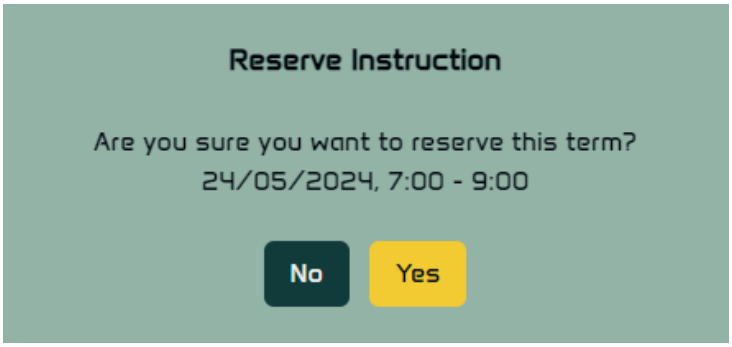
Sl. 33. Stranica instrukcija za studente i neregistrirane korisnike

Sl. 34. Uvid u instruktora, instrukcije, javne materijale i kalendar

Kada korisnici odaberu pregled kalendara, rezervirani javni termini će biti prikazani u žutoj boji, a dostupni termini zelenom bojom. Registrirani studenti mogu rezervirati dostupne termine klikom na željeni termin. Nakon što student rezervira termin, termin će se prikazati u kalendaru studenta i u kalendaru instruktora. Primjer kalendara koji korisnik pregledava je prikazan na Sl. 35, a rezerviranje termina je prikazano na Sl. 36.



Sl. 35. Pregled kalendara po tjednu



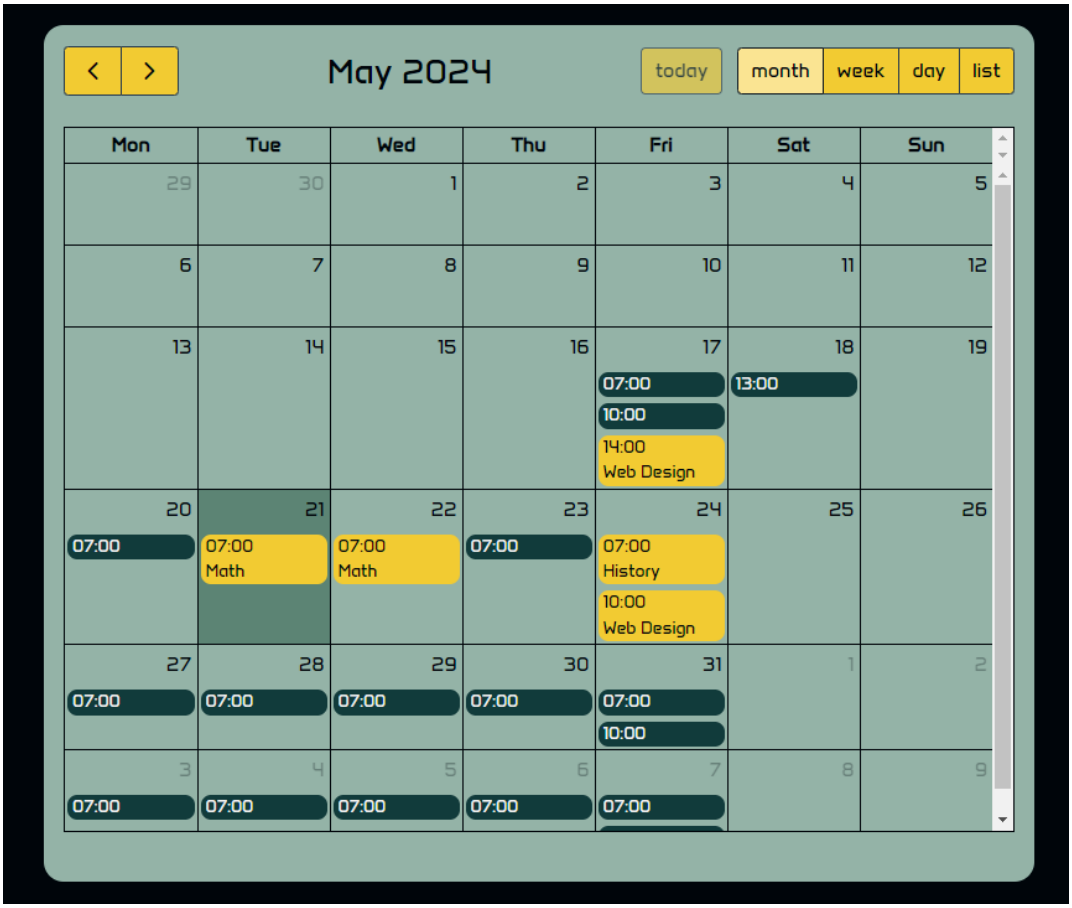
Sl. 36. Rezerviranje termina

5.4.6. Kalendar

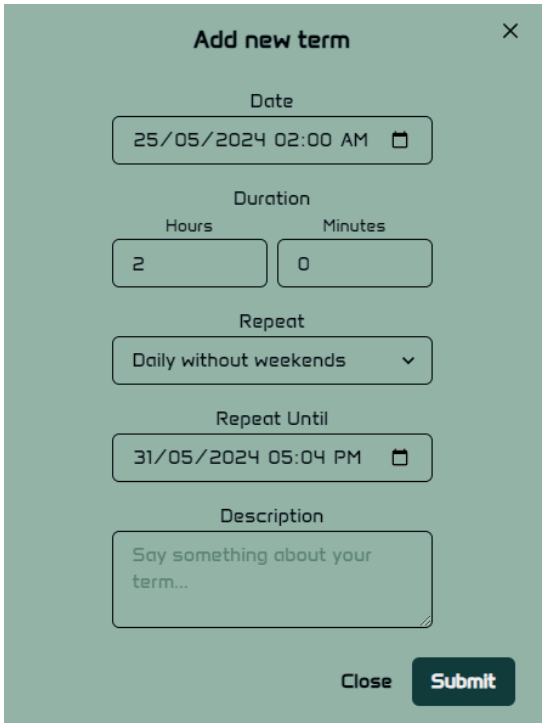
Svi registrirani korisnici imaju mogućnost pregledavanja svojeg kalendara po mjesecu, tjednu, danu ili dnevnom redu. Instruktori koriste interaktivni kalendar temeljen na

FullCalendar biblioteci za dodavanje slobodnih termina klikom na određeni datum i vrijeme unutar kalendara, nakon čega popunjavanju informacije o terminu, uključujući trajanje, ponavljanje termina i opis. Kada je termin rezerviran, bit će prikazan žutom bojom, dok će dostupni termini biti zelene boje. Instruktori i studenti mogu klikom na termin dobiti više informacija o njemu, a također ga mogu i otkazati, što će obavijestiti drugu osobu e-poštom. Otkazivanje je moguće najkasnije 12 sati prije početka termina.

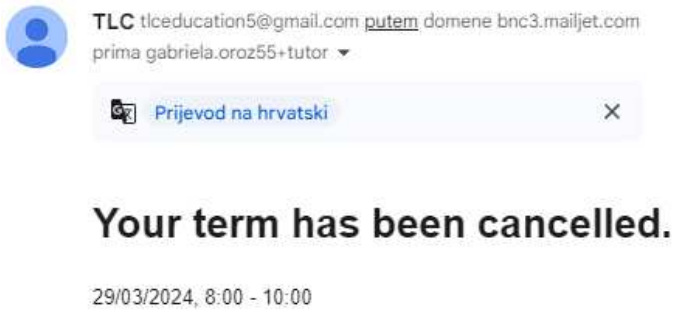
Primjer mjesečnog prikaz kalendara instruktora može se vidjeti se na Sl. 37, dodavanje novog termina na Sl. 38, a e-pošta o otkazanom terminu na Sl. 39.



Sl. 37. Mjesečni prikaz kalendara instruktora



Sl. 38. Dodavanje novog termina

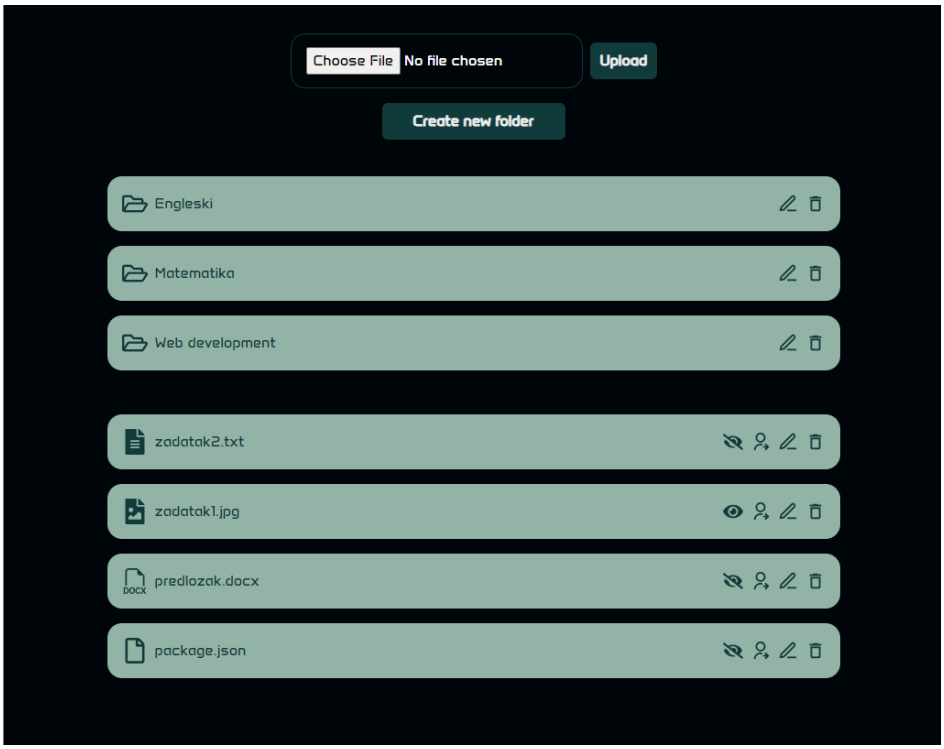


Sl. 39. E-pošta o otkazanom terminu

5.4.7. Materijali

Studenti imaju mogućnost pregledavati i preuzimati sve materijale koji su javni ili su s njima podijeljeni, dok ih instruktori mogu dodavati, uređivati, brisati, činiti privatnima ili javnima te dijeliti sa svojim studentima do određenog roka. Također, instruktori mogu organizirati svoje materijale unutar određenih direktorija. Za pohranu materijala koristi

se Supabase, a preuzimanje se ostvaruje klikom na ime materijala. Stranica materijala za instruktore prikazana je na Sl. 40, a dijeljenje materijala sa studentima na Sl. 41.



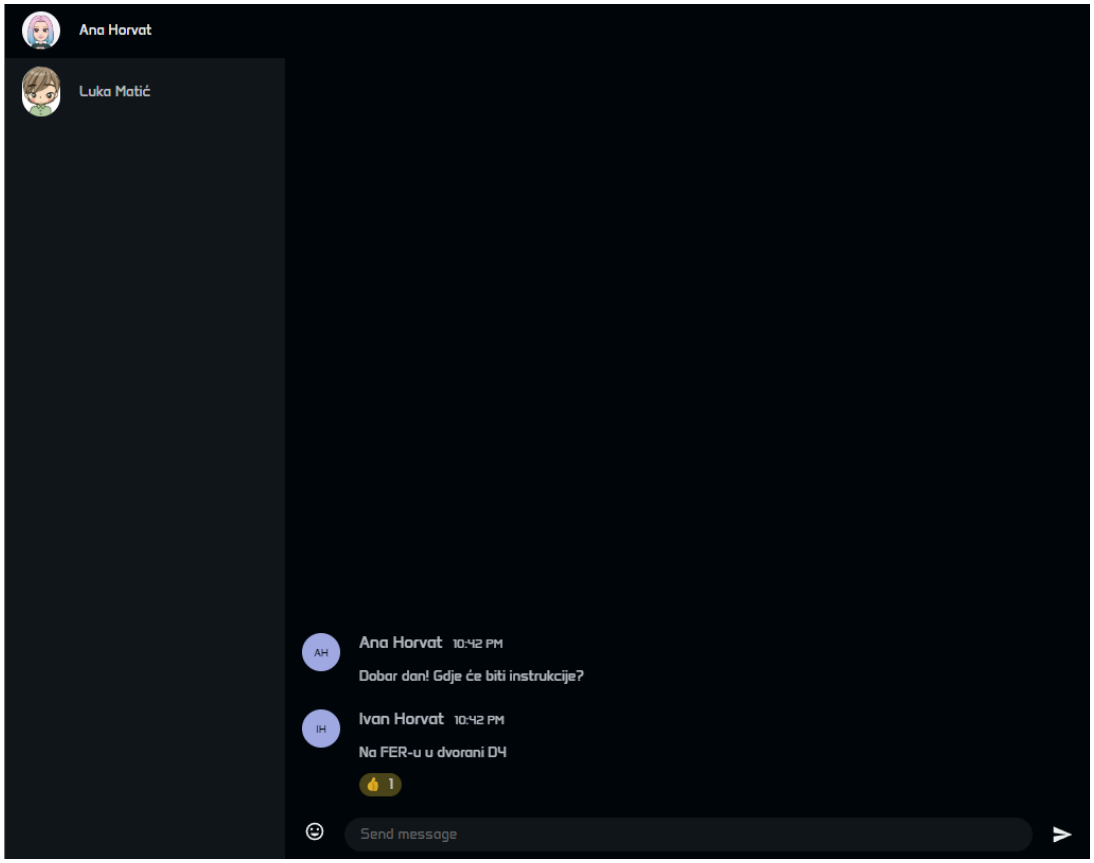
Sl. 40. Stranica materijala za instruktore

Sl. 41. Dijeljenje materijala sa studentima

5.4.8. Stranica za komunikaciju

Komunikacija između studenata i instruktora odvija se na stranici za komunikaciji koja koristi PubNub za interakciju u stvarnom vremenu (engl. *real-time*). To znači da poruke

koje se šalju između korisnika stižu primatelju odmah nakon što su poslane, čime se postiže brza i efikasna komunikacija. Osim što korisnici mogu slati poruke, mogu i reagirati emotikonima. Primjer izgleda komunikacije između studenta i instruktora je prikazan na Sl. 42.

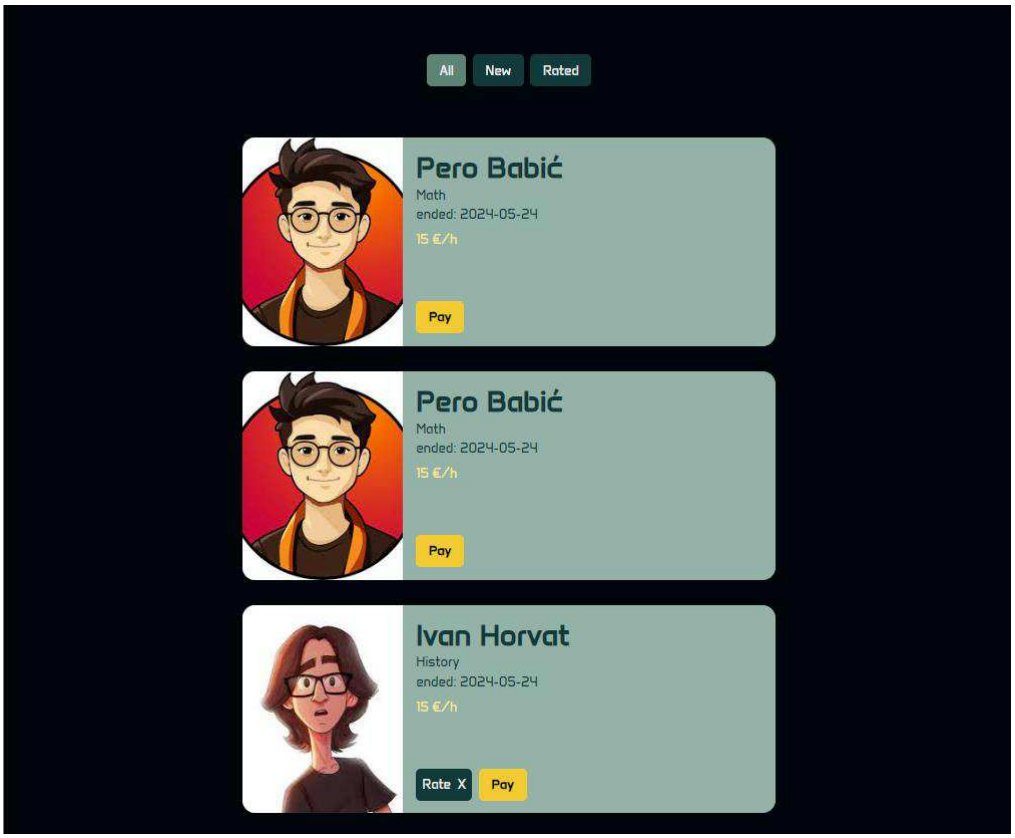


Sl. 42. Razgovor između studenata i instruktora

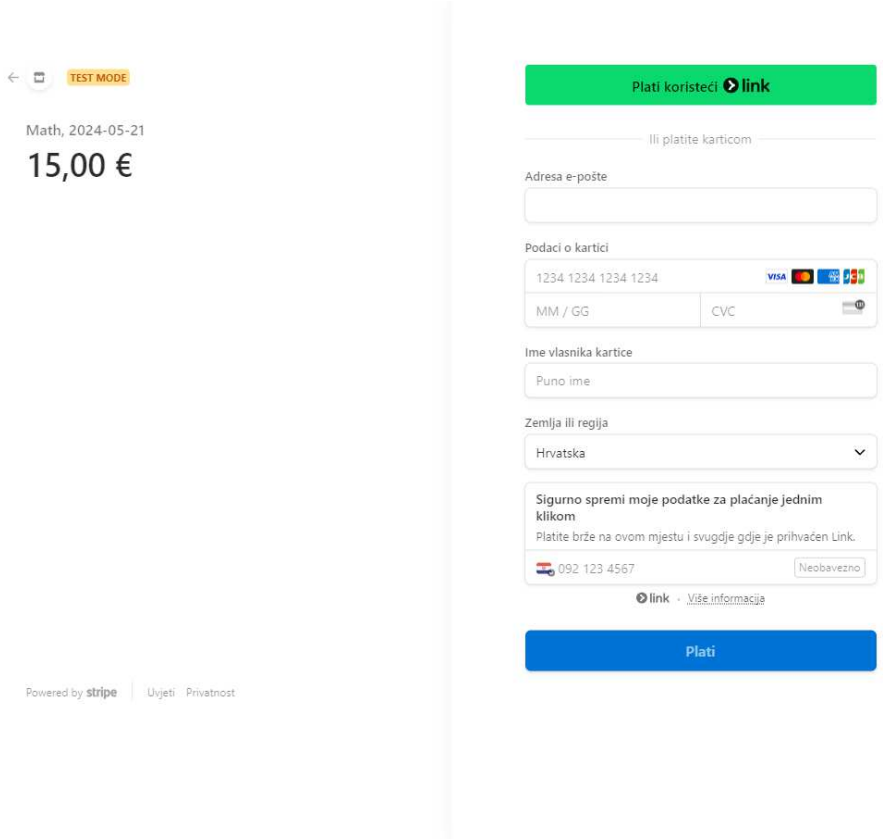
5.4.9. Stranica za plaćanje i ocjenjivanje

Nakon završetka termina, studentima se na ovoj stranici prikazuje opcija za plaćanje i ocjenjivanje instruktora. Za plaćanje se koristi Stripe, omogućavajući siguran i jednostavan proces plaćanja putem različitih metoda. Također, studenti mogu ocijeniti instruktora zvjezdicama od 1 do 5 i komentirati svoju ocjenu. Instruktori mogu vidjeti tko je platio, tko nije te dobivene ocjene i komentare od strane studenata. Stranica za

plaćanje i ocjenjivanje instruktora je prikazana na Sl. 43, a primjer plaćanja termina putem Stripea na slici Sl. 44.



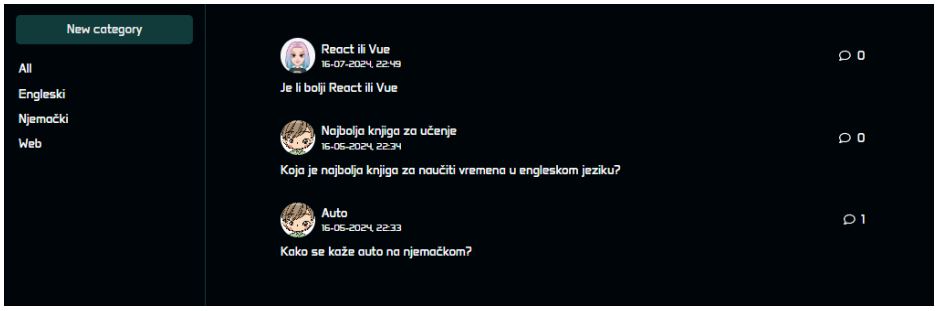
Sl. 43. Stranica za plaćanje i ocjenjivanje instruktora



Sl. 44. Primjer plaćanja termina putem Stripea

5.4.10. Forum

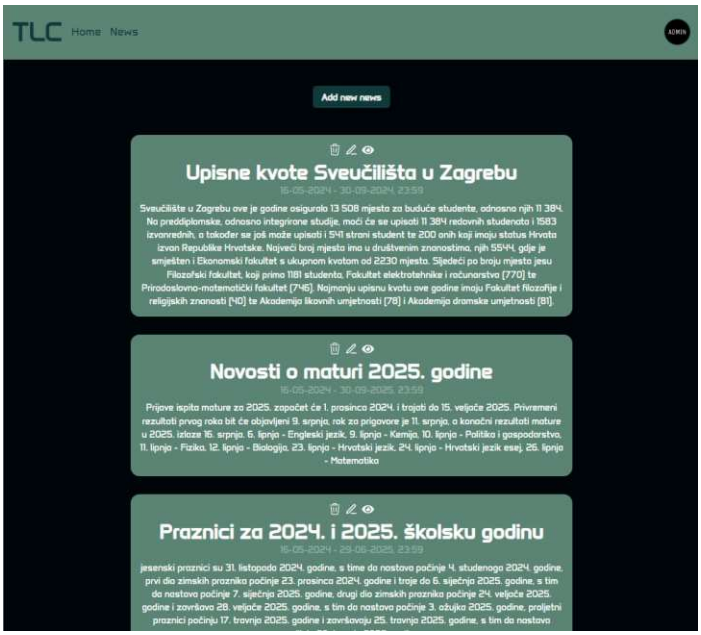
Neregistrirani korisnici mogu pregledavati sve kategorije, objave i komentare postavljene na forumu, dok registrirani korisnici mogu dodavati i nove. Objave su poredane od najnovije prema najstarijoj i pored svake se može vidjeti koliko ima komentara. Izgled foruma za registrirane korisnike prikazan je na Sl. 45.



Sl. 45. Izgled foruma za registrirane korisnike

5.4.11. Novosti

Administrator može kreirati, uređivati i brisati novosti za početnu stranicu. Također ih može učiniti javnima ili privatnima te odrediti datum do kojeg se mogu prikazivati na početnoj stranici. Stranica za novosti kojoj ima pristup samo administrator prikazana je na Sl. 46.



Sl. 46. Stranica za novosti kojoj ima pristup samo administrator

Zaključak

Cilj ovog rada bio je izraditi web aplikaciju koja će olakšati pronalazak, dogovor te razmjenu podataka i digitalnih materijala između studenata i instruktora. Svim korisnicima omogućeno je pregledavanje početne stranice, foruma i dostupnih instrukcija, dok je registriranim korisnicima, ovisno o njihovoj ulozi, na određen način omogućeno upravljanje instrukcijama, materijalima, kalendarom, međusobnim komuniciranjem, plaćanjem, ocjenjivanjem i novostima.

Izrada ove web aplikacije je bila izazovna, no dobar izbor tehnologija je olakšao i ubrzao razvoj. Next.js je omogućio razvoj *frontenda* i *backenda* unutar istog projekta, a Chakra UI je omogućila brzo i jednostavno stiliziranje stranica i njihovih komponenti kao što je to i integracija FullCalendara omogućila za izradu kalendara. Platforme i usluge poput Supabase, Stripe, Mailjet i PubNub su omogućile fokusiranje na razvoj glavnih funkcionalnosti bez potrebe za izradom vlastitih rješenja za sve usluge.

Daljnji razvoj aplikacije mogao bi rezultirati kompleksnijom aplikacijom koja bi nudila automatizirane preporuke instrukcija na temelju interesa i napretka studenata, integraciju s dodatnim alatima za online instrukcije poput digitalne ploče (npr. Whiteboard) i video poziva te podršku za više jezika. Ove nadogradnje bi poboljšale korisničko iskustvo i povećale funkcionalnosti aplikacije.

Literatura

[1] »MyTutor«, https://www.mytutor.co.uk/, [Pristup 21.5.2024]

[2] »MyTutor«, https://nimbleapproach.com/success-stories/mytutor/, [Pristup 21.5.2024]

[3] »Tutor Hunt«, https://www.tutorhunt.com/, [Pristup 22.5.2024]

[4] »Preply«, https://preply.com/, [Pristup 22.5.2024]

[5] »React«, https://en.wikipedia.org/wiki/React_(JavaScript_library), [Pristup 12.6.2024]

[6] »Next.js Rendering«, https://nextjs.org/docs/pages/building-your-application/rendering, [Pristup 11.6.2024]

[7] »Next.js Routing«, https://nextjs.org/docs/app/building-your-application/routing, [Pristup 20.5.2024]

[8] P. Kadam, »Navigativing Next.js: Choosing Between App Routing and Page Routing«, 10.8.2023., https://medium.com/@kadampritesh46/navigating-next-js-choosing-between-app-routing-and-page-routing-85e1c08d8921, [Pristup 28.5.2024]

[9] »Next.js«, https://nextjs.org/, [Pristup 20.5.2024]

[10] »Chakra UI«, https://v2.chakra-ui.com/, [Pristup 20.5.2024]

[11] »Chakra UI Button«, https://v2.chakra-ui.com/docs/components/button, [Pristup 20.5.2024]

[12] »PostgreSQL«, https://www.postgresql.org/, [Pristup 22.5.2024]

[13] P. Scott, »What is PostgreSQL? Everything You Need to Know«, 2.2.2024., https://www.percona.com/blog/what-is-postgresql-used-for/, [Pristup 28.5.2024]

[14] »PostgreSQL Data Types«, https://www.javatpoint.com/postgresql-datatypes, [Pristup 29.5.2024]

[15] »FullCalendar«, https://fullcalendar.io/, [Pristup 21.5.2024]

[16] »PubNub«, https://www.pubnub.com/docs, [Pristup 20.5.2024]

[17] »Mailjet«, https://dev.mailjet.com/, [Pristup 20.5.2024]

[18] M. Kosisochukwu, »Intro to Supabase - the next generation of serverless development«, 17.4.2024., https://bejamas.io/hub/serverless-database/supabase, [Pristup 29.5.2024]

[19] N. Bouchard, »What Is Stripe and How Does it Work?«, 1.3.2024., https://unito.io/blog/what-is-stripe/, [Pristup 21.5.2024]

[20] S. Saliba, »Adding Payments to your Web Application: An Introduction to Stripe Integration«, 12.8.2021., https://medium.com/sipios/adding-payments-to-your-web-application-an-introduction-to-stripe-integration-3abd7354206a, [Pristup 20.5.2024]

[21] N. Raval, »TypeScript vs JavaScript: Know The Difference«, 14.12.2023., https://radixweb.com/blog/typescript-vs-javascript, [Pristup 29.5.2024]

[22] V. Sivan, »Build your web apps using Next.js«, 29.1.2023., https://codemaker2016.medium.com/build-your-web-apps-using-next-js-5083162e42d6, [Pristup 11.6.2024]

Sažetak

Naslov

Web-aplikacija za pronalazak i upravljanje uslugama podučavanja

Sažetak

Ovaj rad opisuje izradu i funkcionalnosti web aplikacije za pronalazak i upravljanje uslugama podučavanja namijenjena studentima i instruktorima.

Ključne riječi

Next.js, React, SSR, CSR, Web aplikacija

Summary

Title

A web application for finding and managing tutoring services

Summary

This paper describes the development and functionalities of a web application for finding and managing tutoring services intended for students and instructors.

Keywords

Next.js, React, SSR, CSR, Web application