

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 496

**Nadogradnja aplikacije za
interakciju studenata i nastavnika u
sustavu Edgar**

Vedran Bedeković

Zagreb, lipanj 2022.

ZAVRŠNI ZADATAK br. 496

Pristupnik: **Vedran Bedeković (0036523256)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: izv. prof. dr. sc. Igor Mekterović

Zadatak: **Nadogradnja aplikacije za interakciju studenata i nastavnika u sustavu Edgar**

Opis zadatka:

Potrebno je proučiti znanstvenu i stručnu literaturu i dostupne programe za interakciju s publikom u kontekstu predavanja na visokom učilištu. Klasificirati moguće scenarije interakcije (npr. pitalice s više ponuđenih odgovora, unos slobodnog teksta, mini-lekcije, itd.). Proučiti JazzQuiz modul sustava Moodle. Edgar je informacijski sustav za testiranje studenta i učenje razvijen na Fakultetu elektrotehnike i računarstva. U Edgar već postoji mogućnost ostvarivanja interakcije s publikom putem tzv. lecture quiz opcije koja omogućuje anonimno i neanonimnu interakciju, pri čemu nastavnik mora unaprijed pripremiti pitanja odnosno kviz koji će pokrenuti za vrijeme predavanja. Studentska aplikacija je izvedena kao jednostranična web-aplikacija pisana u razvojnom okviru Angular2. Potrebno je upoznati se s razvojnim okvirom Angular13, te nadograditi postojeći sustav i proširiti ga mogućnostima tako da omogućuje ad-hoc kvizeve uz miješane modele autentifikacije (poznat identitet, ali nije povezan s odgovorom) i odgovarajuće vizualizacije rezultata. Posebnu pozornost posvetiti izgledu aplikacije i korisničkom iskustvu na pametnim telefonima odnosno na manjim ekranima. Razmotriti uporabu svojstava progresivnih web-aplikacija. Razmotriti proširenje funkcionalnosti aplikacije kako bi se omogućilo studentima da postavljaju ad-hoc pitanja na predavanjima. Donijeti ocjenu ostvarenog pristupa te smjernice za budući razvoj.

Rok za predaju rada: 10. lipnja 2022.

SADRŽAJ

1. Uvod	1
2. Interakcija publike i predavača	2
2.1. Opis sustava	2
2.2. Korisnički zahtjevi	2
2.3. Pregled tehnologija	5
2.3.1. Postojeće rješenje	5
2.3.2. Angular	7
2.3.3. Socket.IO	7
2.3.4. Bootstrap	7
2.4. Postojeća slična rješenja	7
2.4.1. JazzQuiz	7
2.4.2. Kahoot	8
2.4.3. Mentimeter Quiz	9
3. Organizacija sustava interakcije	11
3.1. Uvod	11
3.2. Nastavnikova web-aplikacija	12
3.3. Studentova web-aplikacija	17
3.4. Edgar i <i>WebSocket</i>	19
3.4.1. Interakcijski servis	19
3.4.2. Nastavnikove rute	21
3.4.3. Studentove rute	22
3.4.4. <i>WebSocket</i>	22
3.5. Spremanje u bazu podataka	26
4. Korištene tehnologije	29
4.1. Angular13	29

4.2. TypeScript	29
4.3. Bootstrap	29
4.4. Socket.IO	30
4.5. Node.js	30
4.6. Express.js	30
4.7. Chart.js	30
5. Aplikacija za interakciju studenata i nastavnika u sustavu Edgar	31
5.1. Nastavnička aplikacija	31
5.2. Studentska aplikacija	40
6. Zaključak	46
Literatura	47

1. Uvod

Sve više se koriste programske podrške za upravljanje i održavanje interakcije predavača i publike. Cilj je poticati publiku za sudjelovanje u iznosu sadržaja. Dvosmjerna komunikacija pomaže predavaču u procjeni svog izlaganja i može na mjestu izmijeniti način predstavljanja ili ispraviti grešku. Edgar, informacijski sustav za testiranje studenta i učenje razvijen na Fakultetu elektrotehnike i računarstva, je opremljen sustavom za interakciju nastavnika i njegovih studenata, ali razvojem tehnologije došlo je vrijeme za njegovu nadogradnju.

U okviru rada provedeno je istraživanje moderne verzije razvojnog okvira Angular i popratnih tehnologija na kojima je stari sustav građen. Klasificirani su korisnički zahtjevi i prepoznate su nove značajke koje proširuju mogućnosti prošlog sustava. Proučeni su postojeći slični sustavi koji rješavaju problem interakcije publike i predavača u svrhu prepoznavanja mogućih proširenja postojećeg sustava. U radu je izložen kratki opis tih sustava.

Novi sustav interakcije i njegovi pojedinačni dijelovi su detaljno opisani. Nastavnikova i studentova web-aplikacija je građena od zasebnih komponenata i servisa. One dijele informacije međusobno preko poslužitelja sustava Edgar. On je unaprijeđen novim funkcionalnostima za održavanja interakcije i osigurava spremanje potrebnih entiteta u bazu podataka. Definiran je način autentifikacije prometa interakcije.

Tehnologije iz starog sustava se koriste i u novom, ali njihove moderne verzije. Novi sustav koristi i neke biblioteke koje nisu bile prisutne u starom sustavu. Svaka korištena tehnologija je izložena i objašnjena prema njihovim dokumentacijama.

Predočit će se korištenje sustava s nastavnikovog i studentovog stajališta, kako se pokreće interakcija i pridružuje na nju. Prikazan je tijek interakcije na različitim uređajima i ostvarene funkcionalnosti. Na kraju su navedene smjernice za budući razvoj sustava i njegova ocjena.

2. Interakcija publike i predavača

2.1. Opis sustava

Potrebno je izgraditi sustav koji omogućuje predavaču da ostvari interakciju s publikom putem interaktivnih digitalnih upitnika. Tako predavač može dobiti povratnu informaciju te potencijalno modificirati stil i sadržaj predavanja. Digitalni upitnici, kvizovi, mogu biti pripremljeni unaprijed ili napravljeni ad hoc, brzo i jednostavno, za vrijeme predavanja. Predavač mora imati omogućeni pregled svih zaprimljenih povratnih informacija, ovisno o kakvoj vrsti modela interakcije je predavač predvidio. Ti podaci se po želji predavača spremaju za kasniji pregled. Svaki član publike na kraju interakcije dobiva natrag poslane informacije na pregled. Dodavanje novog sadržaja za vrijeme aktivne interakcije, nebitno u kojoj fazi, treba biti jednostavno i ne smije prekinuti razmjenu poruka. Publika ne koristi samo jednu vrstu uređaja za interakciju sa sustavom pa je potrebno osigurati prilagodljivost sustava na različite veličine prikaza. Interakcija ne smije biti uništena nakon nepredviđenog pada sustava ili prijevremenog gubitka veze predavača. Nakon povratka u normalu sustav nastavlja s interakcijom. Treba se prihvaćati naknadno pridruženje novog člana publike ili povratak prijašnjeg člana.

Sustav se razmatra u kontekstu predavanja visokog sveučilišta pa je predavač nastavnik, a publika studenti. Utemeljen je na prijašnjem rješenju za zadani problem interakcije nastavnika i studenata, koji se nalazi na platformi Edgar. Nastavnik i student koriste zasebne aplikacije prilagođene njihovim ulogama u interakciji.

2.2. Korisnički zahtjevi

1. Priprema sadržaja interakcije

- Nastavnik unaprijed priprema pitalice, pitanja jednog ili više točnih

odgovora i pitanja kratkih odgovora, raspoređena po njegovoj zamisli. Nastali predložak sprema u bazu podataka za kasniji dohvat i/ili izmjene.

2. Priprema novog sadržaja (ad hoc) neposredno prije interakcije

- Prije samog početka interakcije nastavnik može izraditi potpuno novi predložak i pokrenuti interakciju. Omogućeno je spremanje predloška u bazu podataka za kasnije korištenje i izmjene.

3. Odabir modusa provjere identiteta za pojedinu interakciju

- Prije pokretanja interakcije, nastavnik odabire željeni modul provjere identiteta za zadanu interakciju. Omogućene su 3 vrste za odabir: provjera identiteta i povezani odgovori uz identitet, bez provjere identiteta i provjera samog identiteta bez povezanih odgovora.

4. Postavljanje pitanja nastavniku

- Studentima je omogućeno anonimno postavljanje pitanja nastavniku. Ta pitanja se prikazuju na nastavnikovom ekranu i on na njih odgovara usmeno pred cijelom publikom. Nastavnik može dozvoliti ili zabraniti postavljanje pitanja u bilo kojem trenutku interakcije.

5. Dodavanje novog sadržaja za vrijeme interakcije

- Nastavnik može dodati novi sadržaj i time ne prekida tijek interakcije. Dodani sadržaj se umeće u postojeći sadržaj na mjesto koje je nastavnik odredio. Dodavanje automatski ažurira redoslijed preostalog sadržaja.

6. Uvid u rezultate interakcije

- Nastavnik na kraju ima pregled sadržaja interakcije i popratnu grafičku statistiku. Ovisno o odabranom načinu autentifikacije, predavač može dodatno vidjeti:
 - tko je sve pristupio interakciji, tko je što odgovorio i bodovanje studenta
 - tko je pristupio interakciji, kada je pristupio i je li ostao do samog kraja interakcije
 - anonimna načina nema dodatnih informacija

7. Uvid u vlastite rezultate interakcije

- Studenti na kraju interakcije dobivaju povratne informacije o postignutim bodovima i pregled vlastitih odgovora na pitanja i njihove točnosti.

8. Spremanje rezultata interakcije za kasniji pregled i obradu

- Nastavnik može spremi cjelovit sadržaj interakcije u bazu podataka za kasniji uvid, uključujući novododani sadržaj, postignute rezultate i komentare zabilježene za vrijeme interakcije.

9. Prilagodba izgled aplikacije za različite veličine ekrana

- Studenti pristupaju interakciji na različitim uređajima pa je potrebno ostvariti prilagodbu aplikacije na različite veličine ekrana.

10. Otporan na pad poslužitelja ili veze sudionika.

- Unatoč kratkom padu poslužitelja, interakcija nastavlja gdje je i bila za vrijeme pada, sa svim do tad zaprimljenim podacima. Gubitak veze nastavnika ne prekida prihvatanje povratnih informacija. Studentski odgovori se čuvaju na poslužitelju te se u slučaju prekida veze ponovo šalju studentskoj aplikaciji kod ponovne uspostave veze.

tanja, nakon što unesu prikazanu šifru sobe. Te dvije web-aplikacije komuniciraju u stvarnom vremenu pomoću mrežne spojne točke (eng. *WebSocket*). Prema službenoj dokumentaciji [10], to je napredna tehnologija koja omogućuje otvaranje dvosmjerne veze između klijenta i poslužitelja i može slati poruke bez da je klijent to prvo zatražio. Studentima se na kraju kviza prikazuje ostvareni broj bodova, a nastavniku se prikazuje stranica s rezultatima kviza. Kviz je moguće držati pred autentificiranim ili anonimnim korisnicima. Poslužiteljska strana sustava Edgar je implementirana u Node.js i Express.js. On sadrži *WebSocket* zadužen za neprekidnu interakciju dvije web-aplikacije. Preciznije koristi se Socket.IO biblioteka koji, ako WS nije podržan, ima *fallback* na dugačke konekcije i *polling*. Podaci o interakciji i njezin sadržaj su pohranjeni u PostgreSQL bazi podataka.

The screenshot shows a quiz interface. At the top left, the identifier 'a5e5' is displayed in red. To its right is the text 'Old Q'. Further right are two green buttons: 'Show correct' and 'Next question'. Below this header, the question 'How's it going?' is shown. There are three radio button options: 'a Great', 'b Not so great', and 'c It's complicated'. The options are listed vertically with their respective letters in small grey boxes.

Slika 2.2: Stari sustav - Nastavnikov pogled na pitanje

The screenshot shows the same quiz interface from a student's perspective. The header 'a5e5' and 'Old Q' is present. The question 'How's it going?' is displayed. Below the question, there are three radio button options: 'a Great', 'b Not so great', and 'c It's complicated'. The option 'c It's complicated' is highlighted with a yellow background, indicating it is the selected answer.

Slika 2.3: Stari sustav - Studentov pogled na pitanje

2.3.2. Angular

Prije projektiranja i izrade samog sustava proučena je zadana tehnologija, Angular 13. Njome se proširuju mogućnosti naspram prijašnjeg sustava te olakša stvaranje i organiziranje novih komponenata aplikacije. Upoznavanje s razvojnim okvirom je započelo projektiranjem jednostavne web-aplikacije za praćenje zadataka, ravnano prema videomaterijalu [14] i popratnog materijala za učenje Angulara [9]. Angular je sveobuhvatan razvojni okvir i time sadrži vlastiti HTTP klijent. Službena dokumentacija [6] specificira da klijent koristi uočljivi (eng: *observable*) tip podataka za sve interakcije. Taj podatkovni tip obrađuje asinkrone događaje u Angularu.

2.3.3. Socket.IO

Socket.IO je biblioteka koja omogućava dvosmjernu komunikaciju klijenta i poslužitelja. Za dublje razumijevanje, implementirana je web-aplikacija za suradničko pisanje dokumenata u stvarnom vremenu, pomoću materijalu za učenje [8].

2.3.4. Bootstrap

Za prilagodljiv prikaz aplikacije na različitim uređajima koristit će se Bootstrap CSS razvojni okvir. On u prvi plan stavlja izgled aplikacije na mobilnim uređajima pa uvećava komponente po potrebi. Okvir će omogućiti ubrzan razvoj izgleda aplikacije i otvorenost na kasniju izmjenu.

2.4. Postojeća slična rješenja

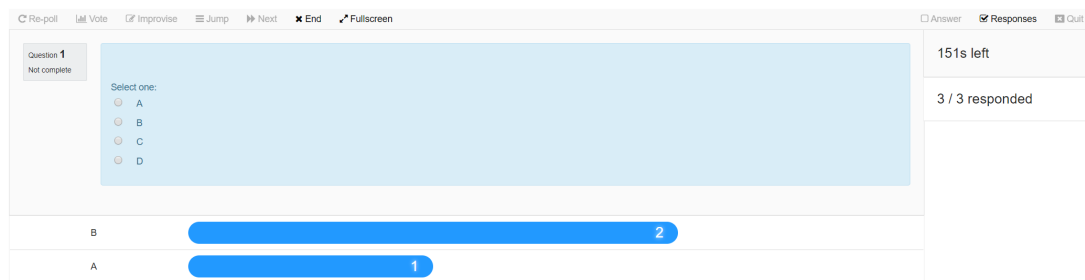
2.4.1. JazzQuiz

JazzQuiz [7] je dodatak za slobodan softver e-učenja Moodle. Ostvaruje izvođenje pripremljenog kviza i podržava dodavanje sadržaja. Sadržaj se dijeli na pitanja s više točnih odgovora i pitanja kratkih odgovora. Na primjer, za vrijeme kviza može se pokrenuti glasovanje na zadanu temu. Na kraju je dostupan pregled svih odgovora na pitanja te preuzimanje prisutnosti ili odgovora u CSV formatu.

Dodatak je namijenjen za korištenje u slučajevima kada se svi sudionici nalaze u istoj prostoriji. Stvaranje pitanja se odrađuje upisom teksta pitanja u odgovarajući unos pa odabirom željene vrste pitanja s padajuće liste. Uz postojeće

predloške improviziranih pitanja omogućeno je dodavanje vlastitih na korištenje za vrijeme kviza.

Za vrijeme predavanja, učitelj pokreće kviz i očekuje spajanje studenata. Kviz započinje kada učitelj odluči krenuti s prikazom prvog pitanja. Ovisno o postavkama, pitanje završava nakon isteka vremena ili kada učitelj to odluči. Spremljeni rezultati sadrže informacije za svako pitanje i može se vidjeti koji student je što odgovorio.



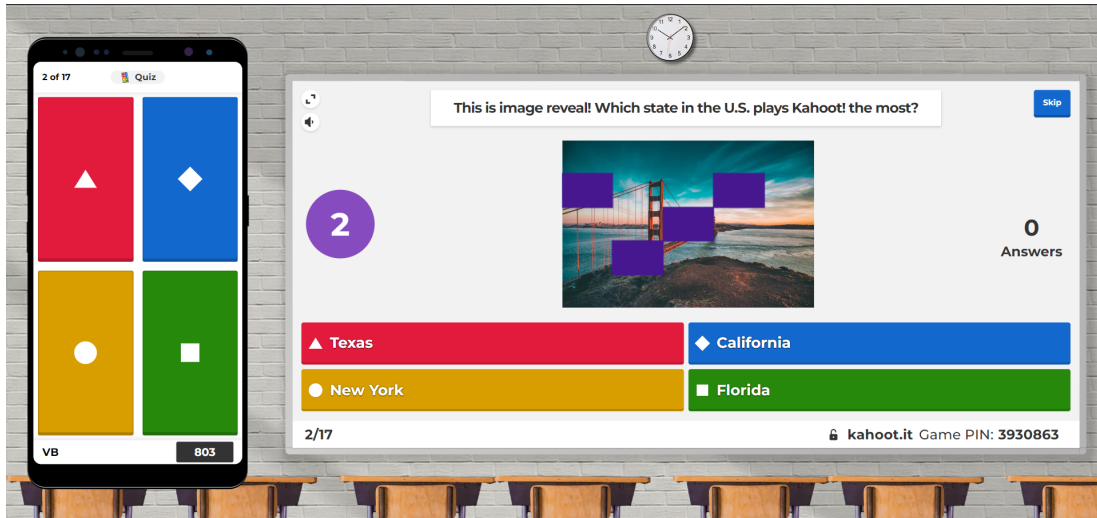
Slika 2.4: JazzQuiz pitanje

2.4.2. Kahoot

Kahoot [16] je platforma za učenje temeljena na konceptima preuzetim iz domene (računalnih) igara. Kahoot kvizovi imaju ubrzan tempo te nagrađuju publiku za točne i što brže odgovore. Pruža različite opcije u dizajnu i oblikovanju interakcije.

Pokretanje interakcije postiže se odabirom predloška drugog korisnika ili stvaranjem vlastitog. Svaki predložak može se naknadno urediti prije početka. Podržava pitanja s više ponuđenih odgovora, zadaci poretka zadanih pojmova, da/ne pitanja, ankete, pitanja postepenog otkrivanja slike i informativne slajdove. Svaki sadržaj koji očekuje odgovor se boduje za točnost odgovora i na brzini davanja odgovora. Na dijeljenom nastavničkom zaslonu prikazuju se pitanja i trenutno stanje bodova, a publici 90% prostora na ekranu zauzimaju gumbi za odgovore. Taj dizajn olakšava što bržu reakciju pri odgovaranju, ali predavač može omogućiti i prikaz pitanja na uređajima koje koristi publika. Publika ima nekoliko sekundi da pročita pitanje prije nego može poslati svoje odgovore. Na kraju svakog pitanja se prikazuju promjene pozicija publike na ljestvici. Davanje više točnih odgovora za redom pridonosi dodatnim bodovima. Na kraju interakcije predavač može preuzeti jednostavno izvješće za pregled.

Interakcija može biti dodijeljena kao zadatak studentima na platformi Microsoft Teams, koji ju mogu riješiti kada oni žele. Može se koristiti u svrhe udaljenog učenja ili ispitivanje studenata.



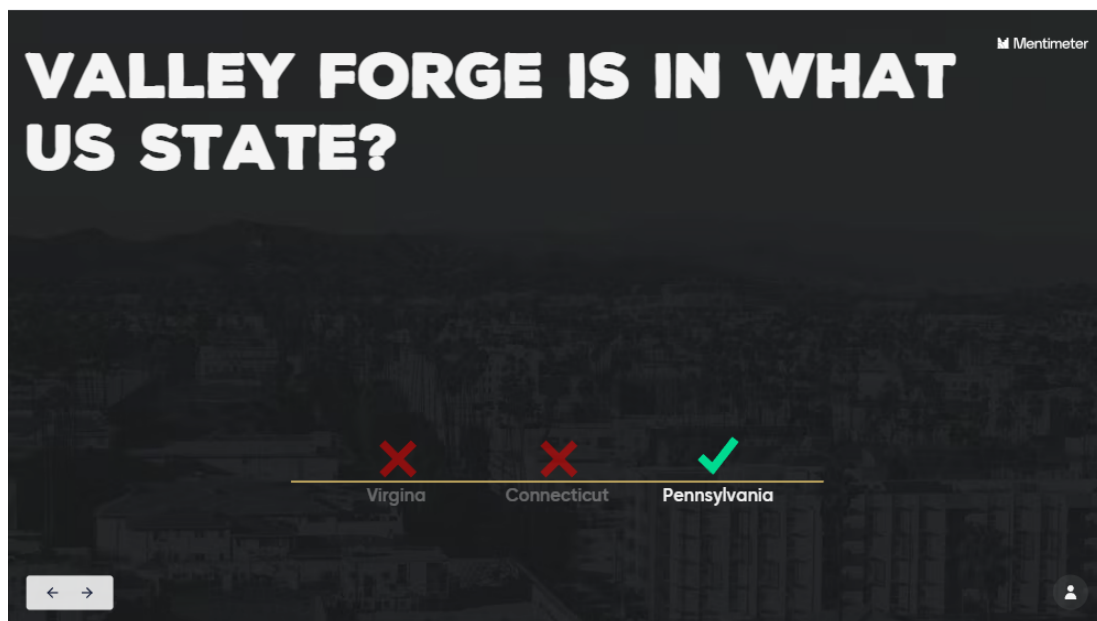
Slika 2.5: Kahoot demonstracija

2.4.3. Mentimeter Quiz

Mentimeterovi [17] prezentacijski kvizovi su namijenjeni za različite vrste publike, dizajnirani za dinamičnu interakciju i zabavu.

Pružene su usluge preuzimanja gotovih predložaka, ali omogućeno je i jednostavno stvaranje novih dodavanjem pitanja i označavanjem točnih odgovora. Primarna vrsta pitanja je pitanje s više ponuđenih odgovora, ali lako se dodaju pitanja različitih vrsta s Mentimeterovih ostalih usluga. To uključuje: ankete dinamičkih prikaza i grafikona, prikaz kolaža riječi gdje se učestali odgovori više ističu, "pitanja i odgovori" i informativni slajdovi.

Emma Cullen je u svom blogu [4] opisala tijek kviza. Publika se pridružuje interakciji upisom lozinke prikazane na prezentaciji. Svaki član publike dobiva svoju sliku i upisuje željeni nadimak. Prije nego što se omoguće odgovori, publika ima nekoliko sekundi da prvo pročita pitanje. Za točan odgovor se nagrađuju bodovi, a što brži odgovori dobivaju dodatne bodove. Na kraju pitanja se prikazuju točni odgovori i ljestvica rezultata.



Slika 2.6: Mentimeter pitanje

3. Organizacija sustava interakcije

3.1. Uvod

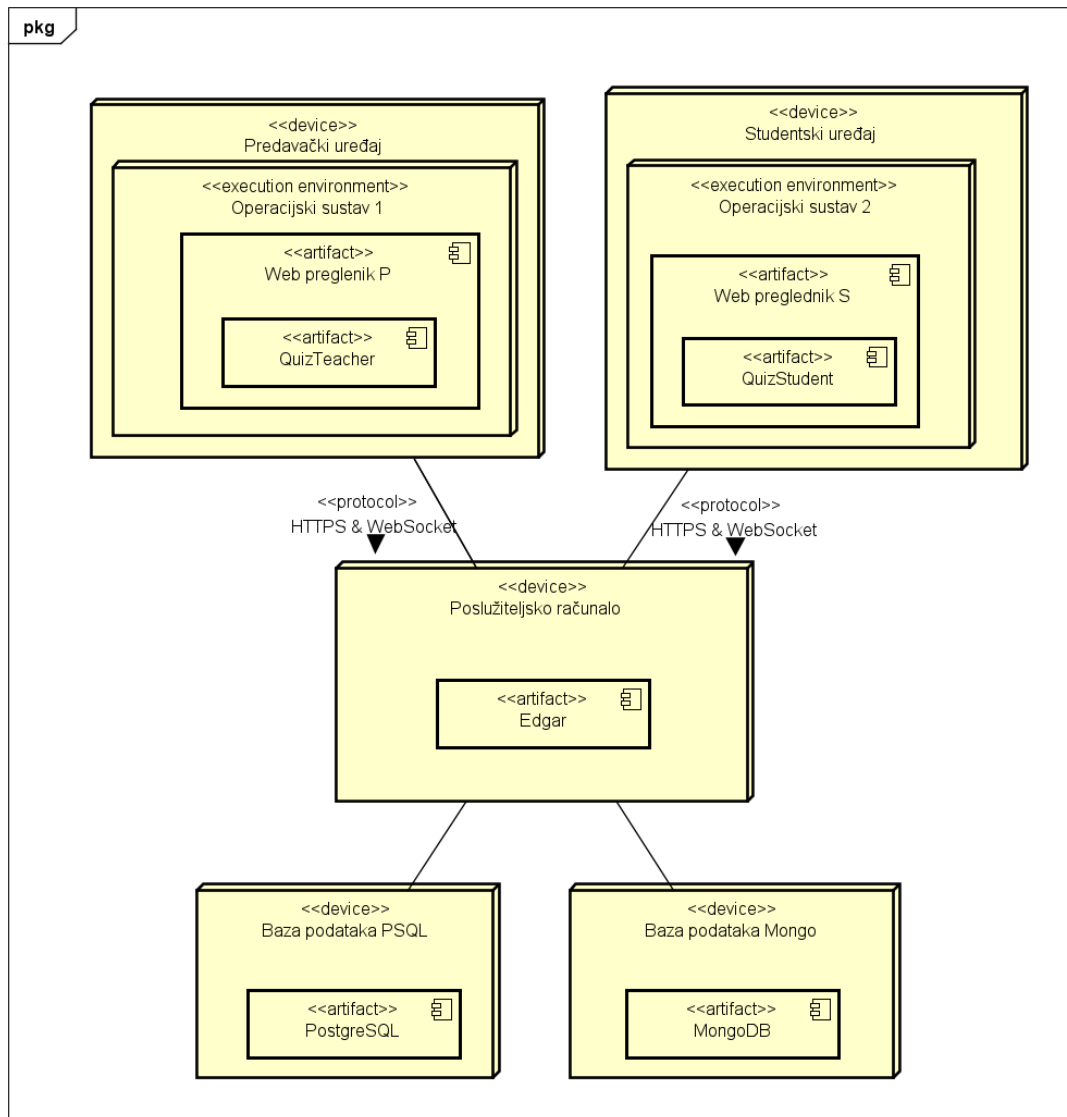
Sustav interakcije je podijeljen na dvije aplikacije na klijentskoj strani. Jedna aplikacija je namijenjena nastavniku za pokretanje i upravljanje interakcije, a druga služi studentima za pridruživanje na stvorenu interakciju i davanju odgovora na poslužena pitanja. Aplikacije komuniciraju s poslužiteljem HTTP zahtjevima za dohvaćanje informacija o aktivnim interakcijama i stvaranju novih. *WebSocket* protokol spaja nastavnikovu aplikaciju, preko poslužitelja, sa studentovom aplikacijom. Putem WS-a ostvaruje brzu interakciju i prijenos poruka između nastavnika i studenata, prikladno za akademsko predavanje i ispitivanje. Ako interakcija nije zadana kao anonimna, WS prihvaća samo autentificirane sjednice. U slučaju anonimne interakcije, autentificirana sjednica nije potrebna.

Na Edgaru su definirane javne i zaštićene rute HTTP zahtjeva. Javne rute su na raspolaganju svima, ali ovisno o vrsti interakcije mogu biti dostupne samo prijavljenim studentima. Pod zaštićene rute spadaju svi zahtjevi koje nastavnik može napraviti za dohvat informacija o interakcijama. Poslužitelj sadrži servis interakcija, koji pomaže u obradi informacija, određuje potrebnu razinu autentičnosti pristupnika interakciji i stvara sigurnosni snimak interakcije za nastavnika u određenim trenucima.

Na bazi podataka PostgreSQL su definirane dvije dodatne tablice za spremanje potrebnog sadržaja. Jedna služi za spremanje predložaka interakcije i opće informacija o njima za kasnije preuzimanje te pokretanje. Druga tablica sadrži rezultate i zabilježenu prisutnost završene interakcije za kasniji dohvat i obradu. MongoDB se koristi implicitno za držanje podataka o sjednicama, koje se dohvaćaju uz *WebSocket* poruke.

Povezanosti svih tih dijelova sustava je prikazan na slici 3.1 dijagrama raz-

mještaja svih dijelova sustava interakcije.



Slika 3.1: Dijagram razmještaja - Cijeli sustav interakcije

3.2. Nastavnikova web-aplikacija

Početna stranica Angular SPA aplikacije, prikazana u dijagramu komponenti 3.2, sadrži jednu komponentu za preusmjeravanje aplikacije u dva stanja: pokretanje interakcije i provođenje interakcije.

Komponenta pokretanja interakcije služi za pregledavanje spremljenih predložaka i rezultata. Kod stvaranju komponente, ona traži podatke o spremljenim predlošcima i rezultata sa servisa pokretanja kviza. Postoji koncept javnih predložaka, dostupni svakom prijavljenom nastavniku. Dohvat tih predložaka mora

eksplicitno zatražiti korisnik. Svaki vlastiti element, poput spremljenih predložaka i rezultata, se može pregledati i brisati. Dohvaćeni javni predlošci mogu se samo pregledati i pokretati. U sebi sadrži komponentu za stvaranje novog ili podešavanje postojećeg predloška. Odabirom određenog predloška traže se njegovi podaci sa servisa. Nakon uspješnog dohvata, otvara se komponenta za manipulaciju predloška, popunjena dohvaćenim podacima. Spojene su dvije funkcije koje se pokreću ovisno koje odaslane događaje je vratila komponenta za manipulaciju predloška. Spremanje predloška je postavljeno da modificira odabrani predložak i šalje promjene servisu. Pokretanje interakcije šalje servisu naredbu da stvori sobu interakcije i preusmjerava na glavnu komponentu za vođenje interakcije kod nastavnika. Moguće je pokrenuti oba događaja odjednom. Stvaranje novog predloška se radi otvaranjem komponente za manipulacijom podataka, ali bez zadavanja početnih vrijednosti. To će otvoriti prazan predložak. Događaj pokretanja radi isto, samo događaj spremanja šalje naredbu spremanja novog predloška servisu. Odabirom javnog predloška za pregled se isto otvara komponenta za modificiranje predloška s popunjenim podacima, ali sada spremanje ne mijenja javni predložak. Spremanje šalje servisu naredbu spremanja novog predloška, što kao rezultat daje nastavniku novi vlastiti predložak za uporabu. Odabir spremljenih rezultata, zatraže se pripadni podaci sa servisa i otvara komponentu pregleda spremljenih rezultata. U danom trenutku može biti otvorena samo jedna potkomponenta. Prije pokretanja interakcije, nastavnik može odabrati njezinu vrstu s padajuće liste i ovisno o vrsti interakciji će moći pristupiti samo prijavljeni studenti ili bilo tko ima šifru nastale sobe. Ako se nastavnik nađe u početnoj komponenti, a ima nedovršenu interakciju, nudit će mu se opcija brzog povratka na nju.

Komponenta manipulacije predloška je forma kojom se stvaraju novi predlošci ili izmjenjuju postojeći. Ako dobije početne vrijednosti na njezinom stvaranju, onda će forma biti popunjena. Forma direktno mijenja naslov predloška, njegov opis i pretpostavljeno bodovanje svakog pitanja. Pretpostavljeno bodovanje služi za automatsko popunjavanje bodovanja pojedinog pitanja, ali i dalje je moguće to izmijeniti pojedinačno. Broj pitanja i najveći mogući postignuti bodovi se prikazuju, ali se ne mogu direktno promijeniti. Ti podaci se izračunavaju svakom promjenom pitanja u formi. Svako pitanje predloška se prikazuje komponentom pitanja. Ona ne odašilje događaje, samo prima podatke za pravilan prikaz pitanja u formi predloška. Svako pitanje može promijeniti poziciju, urediti i obrisati. Moguće je i dodavanje novih pitanja. Dodavanje ili uređivanje pitanja otvara

komponentu manipulacije pitanja. Pri dodavanju je komponenta manipulacije pitanja prazna i potrebno ju nadopuniti podacima, a kod uređivanja komponenta je popunjena zaprimljenim podacima i spremna na izmjenu. Forma predloška ima pet gumba za odašiljanje događaja ili izvršavanja internih akcija forme. Spremanje i pokretanje, samo spremanje i samo pokretanje su tri različite kombinacije odaslanih događaja. Te akcije su zabranjene ako nisu sva predviđena polja popunjena i ako nema niti jednog pitanja u predlošku ili je otvorena komponenta manipulacije pitanja. Opcija vraćanja na praznu formu briše sva dodana pitanja i korisnikove unose. Ta opcija je isključena na uređivanju predloška. Odustajanje od predviđene akcije otvorene forme se odvija odašiljanjem događaja zatvaranja forme roditeljskoj komponenti i brisanjem unesenih podataka.

Komponenta pitanja služi za prikaz svih implementiranih vrsta pitanja i njihovih odgovora. Roditeljska komponenta joj daje sve podatke o pitanju. Pitanju s ponuđenim odgovorima može biti naznačena točnost pojedinog odgovora, a pitanju slobodnog odgovora mogu biti prikazani svi poslani odgovori, ako joj je zadana ta opcija. Postoji jedan odašiljač događaja u komponenti, glavne svrhe da obavijesti roditeljsku komponentu da je bio pritisnut gumb prikaza dohvaćenih odgovora na pitanju slobodnih odgovora.

Komponenta manipulacije pitanja je forma za dodavanje novih ili uređivanje postojećih pitanja. Ako se na otvaranju komponente ne unesu podaci o postojećem pitanju, prikazat će se prazna forma. Prilaže se unos teksta pitanja s mogućnošću uređivanja HTML elementima. Nastavnik može odabrati na kojoj poziciji u redoslijedu će se nalaziti novo pitanje. Bodovanje je ispunjeno pretpostavljenim vrijednostima zadane interakcije, ali moguće su izmjene bodovanja za trenutno pitanje. Odabirom vrste pitanja s ponuđenim odgovorima otvaraju se dodatno dva unosa za tekst odgovora. Isto se mogu urediti HTML oznakama. Broj odgovora je dinamičan. Može se dodavati i brisati proizvoljan broj odgovora. Svaki odgovor može biti označen kao točan. Ako je nastavnik izabrao više od jednog točnog odgovora, pitanje postaje pitanje s više točnih odgovora. Moguće je stvoriti pitanje bez točnih odgovora. To pitanje postaje poput male ankete i bode se na isti način kao i pitanja otvorenog odgovora. Ako su sva potrebna polja popunjena može se odaslati novo ili izmijenjeno pitanje roditeljskoj komponenti na obradu. Forma otvorena za dodavanje novog pitanja ima opciju brisanja svih unesenih vrijednosti. Odašiljanje događaja odustajanja od manipulacije pitanja roditeljskoj komponenti je predviđeno za zatvaranje forme.

Komponenta pregleda spremljenih rezultata se pokreće davanjem podataka o

spremljenoj interakciji. Prikazuje jedno pitanje s točnim ili poslanim odgovorima i koliko studenata je odgovorilo te definira navigaciju na sljedeće i prethodno pitanje. Ispod pitanja se prikazuje tablica bitnih informacija o studentima, ovisno o kojoj vrsti interakcije je bilo riječ. Tip interakcije na kojem se prate studentovi odgovori omogućuje njihov detaljan pregled otvaranjem odgovarajuće komponente odabirom studenta i prosljeđivanjem njegovih podataka njoj.

Komponenta pregleda detalja o studentu zahtjeva podatke određenog studenta prije stvaranja. Ona prikazuje pitanja kako ih student vidi pomoću komponente prikaza pojedinog pitanja iz studentovih detalja. Vidljivo je jedno pitanje, a pregled ostalih se ostvaruje navigiranjem na prethodno ili sljedeće pitanje.

Komponenta pogleda pitanja sa studentove perspektive služi za usporedbu studentovih odgovora i točnih odgovora na pitanju. U slučaju pitanja otvorenog tipa prikazuje se ispunjeno polje unosa teksta sa studentovim odgovorom, ako ga je dao.

Glavna komponenta aplikacije zaslužena za provođenje interakcije služi za prikaz i upravljanje sadržajem tijekom interakcije. Ulaskom u komponentu, ona stvara pretplate na podacima iz servisa zaslužen za odvod i dovod podataka interakcije. Ti podaci se šalju u stvarnom vremenu s poslužitelja. Promjenom tih podataka ažurira se status i statistika interakcije. Na zaglavlju je otvorena komponenta zaglavlja. Njoj se šalju podaci o stanju interakcije i ona odašilje događaje za obradu glavnoj komponenti. Prije početka interakcije, očekuje se ulazak korisnika u nastalu sobu interakcije. Prikaz tog stanja je ostvaren komponentom sobe za čekanje. Njoj se prosljeđuju informacije o sobi i broju povezanih studenata. Ako je omogućena opcija postavljanja pitanja, pridošla pitanja se prikazuju ispod zaglavlja, jedno po jedno. Nastavnik može odbaciti sva pitanja ili prijeći na sljedeće postavljeno pitanje. On ima opciju umetanja novog pitanja, za vrijeme interakcije, pomoću komponente manipulacije pitanja. Pitanje ne može biti umetnuto na poziciju koja je manja od rednog broja trenutnog pitanja. Trenutno pitanje se prikazuje pomoću komponente prikaza pitanja. Na završetku pitanja vidi se stupčasti dijagram, koji stvaramo prosljeđivanjem svih rezultata trenutnog pitanja. Na njemu se vidi raspodjela poslanih odgovora i njihova točnost. Na kraju interakcije prikazuje se komponenta rezultata unosom svih podataka interakcije i opcija spremanja rezultata za kasniji pregled. Podaci prije prikaza moraju biti posluženi sa servisa interakcije, poredani po bodovima i inicijalima studenata. Pri dohvatom događaja spremanja, šalje podatke servisu interakcije za spremanje u bazu podataka. Komponenta tijekom interakcije održava njezinu

sigurnosnu kopiju u slučaju pada poslužitelja.

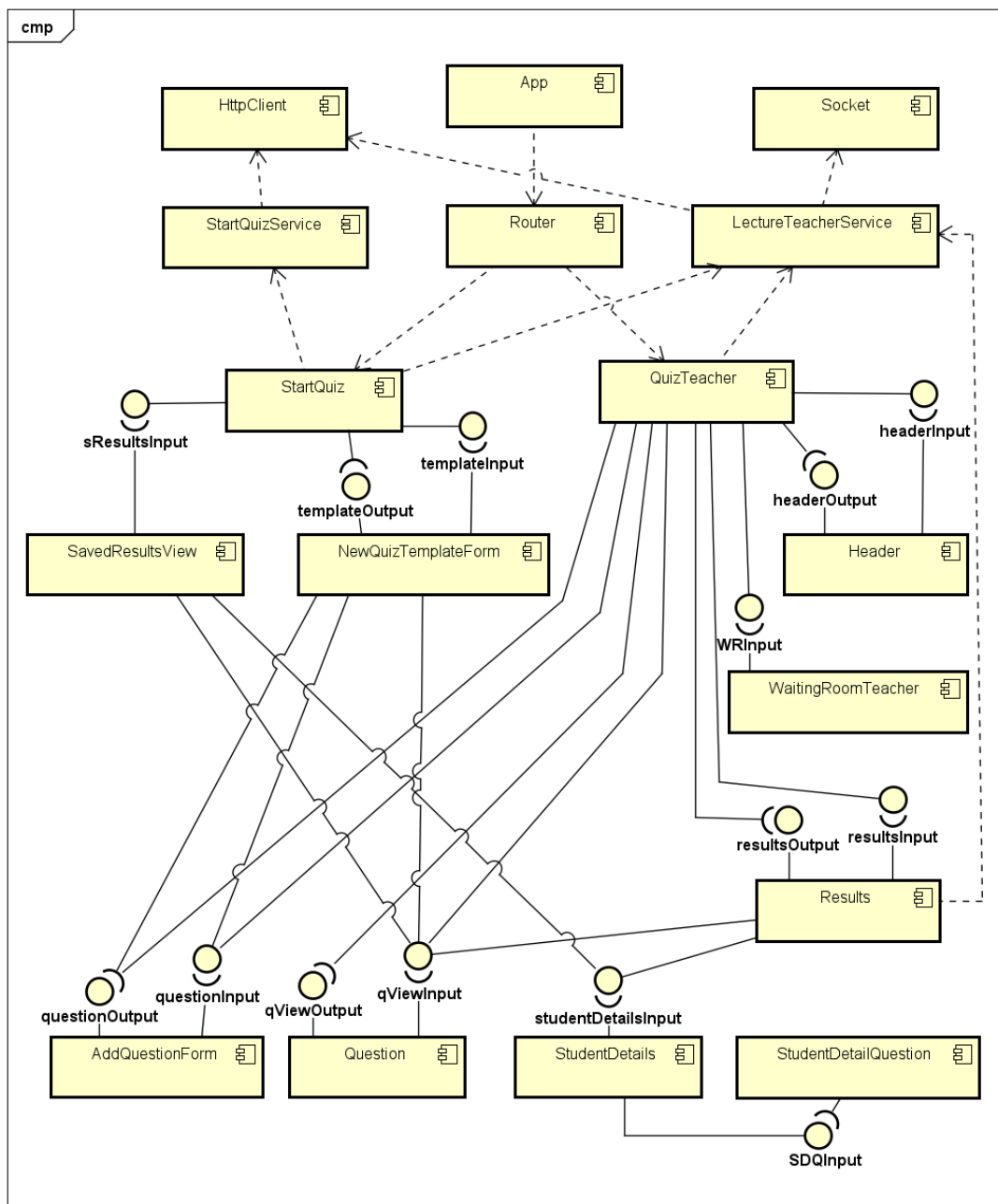
Komponenta zaglavlja prikazuje naslov interakcije i šifru sobe. Glavna funkcija komponente je dinamično prikazivanje akcija omogućena nastavniku u različitim stanjima interakcije. Pritiskom na određeni gumb odašilje se događaj glavnoj komponenti za daljnju manipulaciju prikaza i stanja interakcije.

Komponenta sobe za čekanje poslužuje informacije za pridruživanje na interakciju. Informira o šifri sobe i broju trenutno spojenih korisnika.

Komponenta rezultata se otvara na kraju kviza, nakon dohvata i obrade svih informacija. Ona izlaže sva pitanja i popratna bodovanja. Na početku je prikazano prvo pitanje iz interakcije, a ostala predavač dohvaća i prikazuje putem kontrola za navigaciju u kvizu. Ispod pitanja s ponuđenim odgovorima je prikazan stupčasti dijagram s raspodjelom odgovora. Pitanja su prikazana komponentom pitanja. Ovisno o vrsti interakcije, prikazuje se prikladna tablica. Anoniman tip interakcije ne sadrži popratnu tablicu. Tablica prvo nije vidljiva, sve dok nastavnik pritisne gumb za njezin prikaz. Tako ju isto može i sakriti. Odabirom studenta u tablici otvara komponentu prikaza detalja tog studenta. Gumb spremanja rezultata odašilje događaj spremanja glavnoj komponenti.

Glavna svrha servisa interakcije nastavnika je slanje i zaprimanje *WebSocket* poruka i popratnih podataka s poslužitelja. U tu svrhu koristi Socket servis iz "ngx-socket-io" modula. Za sve podatke koje očekuje iz ulaznih *WebSocket* poruka stvara *observable* podatak na koji se komponente mogu pretplatiti, ako im je potreban. Odašiljanje *WebSocket* poruka se izvrši kada neka komponenta to zatraži uz popratnu šifru sobe i podataka. Servis definira jedan HTTP zahtjev, koji služi za spremanje rezultata interakcije.

Servis pokretanja interakcije traži podatke s poslužitelja i šalje mu njihove izmjene. On na zahtjev komponenti dohvaća spremljene predloške, javne predloške, spremljene rezultate, spremanje izmjena predloška, spremanje novih predloška, brisanje predloška ili rezultata, postavljanje predloška javnim i stvaranje interakcije na serveru prema zadanom predlošku i tipu interakcije.



Slika 3.2: Dijagram komponenti - Nastavnikova aplikacija

3.3. Studentova web-aplikacija

Početna stranica aplikacije koristi komponentu preusmjeravanja iz jedne komponente u ostale. Po pokretanju aplikacije prvo je otvorena komponenta spajanja na interakciju pa se nakon nje preusmjerava na glavnu komponentu interakcije studenta.

Komponenta spajanja na interakciju je forma za unos šifre sobe tražene inte-

rakcije. Koristi servis spajanja na interakciju za slanje HTTP zahtjeva o postojanju unesene sobe. Ako tražena interakcija ne postoji, prikazat će se pravilna poruka na zaslonu. Ako postoji, ali interakcija zahtjeva prijavu studenta prikazat će se poruka i poveznica na prijavu. Nakon prijave će student biti preusmjeren natrag na ovu komponentu. Unosom prihvaćene šifre sobe, aplikacija preusmjerava na glavnu komponentu interakcije studenta.

Glavna komponenta studentove interakcije sadržava sve ostale komponente potrebne za prikaz izgleda i slanje informacija interakcije. Na zaglavlju se nalazi komponenta zaglavlja, gdje se prikazuju potrebne informacije o interakciji za studenta. Prije prikaza prvog pitanja, otvorena je komponenta sobe za čekanje. Ako je dopušteno, glavna komponenta će prikazati formu za postavljanje pitanja nastavniku. Pitanja su prikazana u komponenti pitanja. Za pravilno učitavanje pitanja potrebno je predati podatke dobivene *WebSocket* porukom komponenti pitanja. Pitanja odašilju promjene odgovora, koja se prenašaju servisu pa na kraju poslužitelju preko *WebSocket* veze. Na kraju interakcije dobiva se postignuti broj bodova i otvara se komponenta pregleda svih pitanja i koje odgovore je student poslao na zadano pitanje. Toj komponenti su potrebna sva pitanja, pridruženi točni odgovori i studentovi odgovori. Kada nastavnik izbriše interakciju s poslužitelja, studente se isključuje s *WebSocket* veze te se prikaže poruka o završetku interakcije i poveznica na početnu stranicu Edgara.

Komponenta zaglavlja prikazuje osnovne informacije o interakciji, poput šifre sobe, naslova i oznake tipa interakcije.

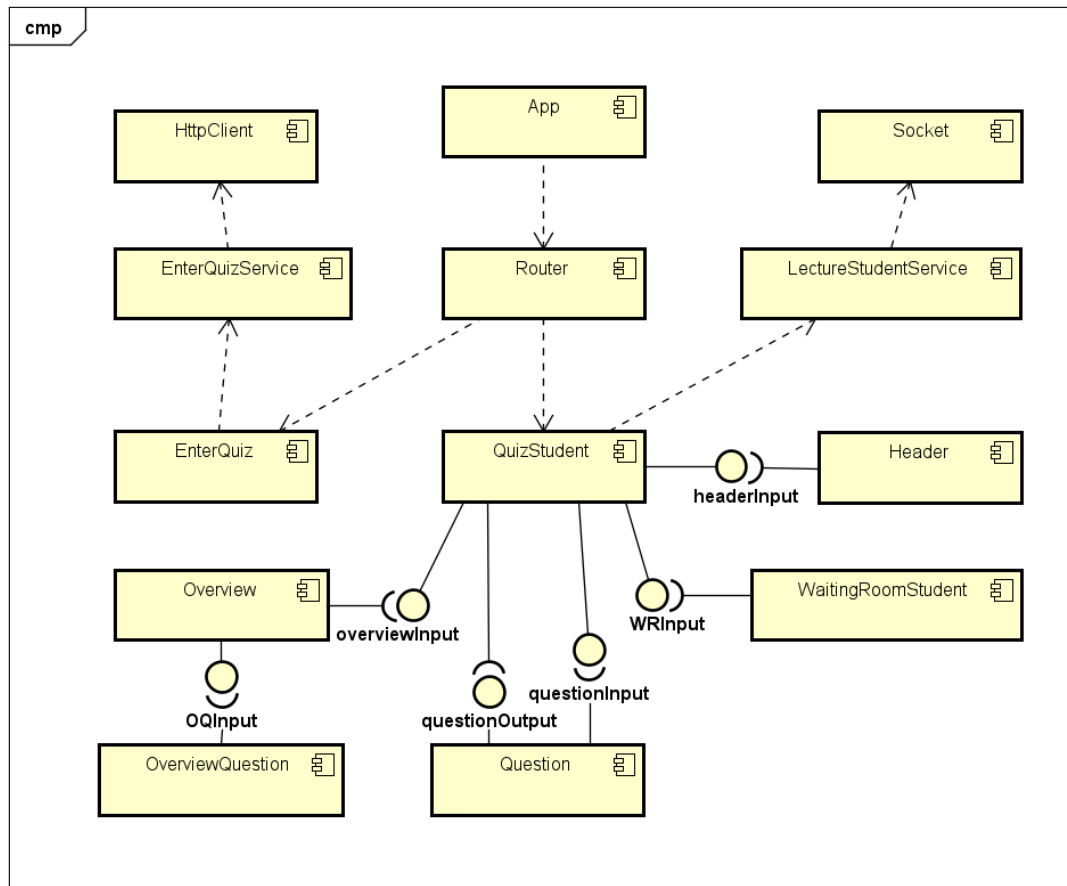
Komponenta sobe za čekanje traži od glavne komponente samo šifru sobe. Ona je glavna oznaka da interakcija još nije službeno započela. Interakcija je moguća u ovom stanju ako je nastavnik dozvolio postavljanje pitanja.

Komponenta pitanja prikazuje pitanje i omogućuje biranje odgovora ili unos vlastitog ovisno o vrsti pitanja. Nakon davanja odgovora, pitanje odašilje promjene glavnoj komponenti za daljnje usmjeravanje.

Komponenta pregleda predstavlja studentu usporedbu točnih odgovora s danim odgovorima po pitanju. Ona očekuje ulazne podatke svih pitanja i studentove odgovore na svako pitanje, ako ih ima. Student se pomiče na određeno pitanje pomoću navigacijskih gumbova. Pitanja su prikazana pomoću komponente pregleda pitanja. Potrebni su joj podaci o pitanju i studentov odgovor na to pitanje.

Servis interakcije ima ulogu slanja poruka odgovora nastavniku i primanje poruka promjene stanja kviza preko protokola *WebSocket*. On generira jedinstveni niz znakova koji predstavlja njegov identitet u interakciji. Pomoću njega je

moгуće ponovo se pridružiti kvizu bez gubitka dotad poslanih odgovora.



Slika 3.3: Dijagram komponenti - Studentova aplikacija

3.4. Edgar i *WebSocket*

Sustav Edgar je proširen novim rutama za dohvat podataka i posluživanju studentove i nastavnikove web-aplikacije. Novi servis služi kao pomoćna biblioteka funkcionalnosti i validacija *WebSocket* prometa. On sadrži podatkovni tip, koji sprema instance interakcije na poslužiteljevoj memoriji. Koristi se ista *WebSocket* biblioteka "Socket.io" s prošlog sustava i funkcionalnost je proširena novim porukama, funkcijama i autentifikacijom. Novi sustav postoji u paraleli sa starim i nisu međusobno povezani.

3.4.1. Interakcijski servis

Servis definira rječnik 3.1 za spremanje svih podataka o aktivnoj interakciji. Svaka instanca je zapisana pod ključem, koji je jednak šifri sobe. U njemu se

bilježi naslov, broj pitanja, u kojoj fazi se nalazi interakcija, na kojem pitanju se nalazi, sva pitanja, bodovanje, dozvolu postavljanja pitanja, vrstu interakcije i korisničke podatke priključenih studenata. Servis računa i šalje statistiku trenutnog pitanja preko *WebSocket* protokola. Definirana je funkcija koja šalje predavaču sigurnosnu kopiju instance interakcije u slučaju pada sustava. Preko servisa se stvara nova instanca interakcije i generira jedinstvena šifra te interakcije za vrijeme njezinog trajanja. Njime se validira nadolazeći *WebSocket* promet pomoću omotanih sjednica na svakoj poruci.

Kôd 3.1: Struktura podataka s početnim vrijednostima instance kviza

```
service.instancedRooms[newRoomID] = {
    quizHandler: "",
    quizType: type,

    questionsAllowed: false,
    quizOngoing: true,

    examTitle: quizTemplate.title,
    noOfQuestions: quizTemplate.numberOfQuestions,
    password: "",

    questions: quizTemplate.questions,
    grading: quizTemplate.gradingArray,
    defaultGrading: quizTemplate.defaultGrading,

    currentQuestionOrdinal: 0,
    currentQuestionAnsweredCount: 0,
    questionActive: true,
    currentFreeTextAnswers: [],

    userCount: 0,
    usersInfo: new Object(),
    userSocketDict: new Object()
}
```

3.4.2. Nastavnikove rute

Određene su nove rute za spremanje i dohvat podataka bitne nastavniku za pokretanje sustav interakcije. Svaka nova ruta zahtjeva autentificiranu sjednicu i ulogu učitelja na sustavu Edgar.

- **/quiz/teacher**
 - Služi za dohvaćanje nastavnikove web-aplikacije.
- **/quiz/teacher/getQuizTemplate/:id**
 - Dohvaća jedan određeni predložak s baze podataka prema zadanom identifikacijskom broju.
- **/quiz/teacher/myQuizTemplates**
 - Zatražuje sve spremljene predloške jednog nastavnika s baze podataka prema identifikacijskom broju korisnika, dobivenog sa sjednice.
- **/quiz/teacher/createQuizRoom**
 - Prosljeđuje servisu interakcije predložak iz kojeg će generirati novu instancu interakcije i vratiti šifru sobe.
- **/quiz/teacher/postQuizTemplate**
 - U bazu sprema poslani novonastali predložak.
- **/quiz/teacher/updateQuizTemplate/:id**
 - Ažurira postojeći predložak s izmijenjenim podacima.
- **/quiz/teacher/deleteQuizTemplate/:id**
 - Briše predložak sa zadanim identifikacijskim brojem s baze podataka.
- **/quiz/teacher/publishQuizTemplate/:id**
 - Predložak sa zadanim identifikacijskim brojem mijenja u javno dostupan.
- **/quiz/teacher/publicQuizTemplates**
 - Dohvat općih informacija svih javnih predložaka drugih korisnika.
- **/quiz/teacher/getPublicQuizTemplate/:id**
 - Dohvat određenog javnog predloška.

- **/quiz/teacher/saveQuizResults**
 - Spremanje podataka o završenoj interakciji na bazu podataka.
- **/quiz/teacher/mySavedQuizResults**
 - Dohvaćanje općih informacija o spremljenim rezultatima za općeniti pregled.
- **/quiz/teacher/getSavedQuizResult/:id**
 - Dohvaćanje specifičnog spremljenog rezultata za detaljan pregled.
- **/quiz/teacher/deleteSavedQuizResult/:id**
 - Brisanje spremljenog rezultata nakon odrađenog pregleda.

3.4.3. Studentove rute

Studentove rute nisu samo namijenjene prijavljenim studenima, već i anonimnim korisnicima. Potvrda autentičnosti studenta dolazi naknadno, ako to zadaje tražena vrsta interakcije.

- **/quiz/student**
 - Služi za dohvaćanje studentove web-aplikacije.
- **/quiz/student/getInstanceExistsInfo/:room**
 - Poziva se servis interakcije da provjeri postojanje sobe i vrste interakcije. Ako zahtjev nema validiranu sjednicu, a vrsta interakcije to zahtjeva, vraća se greška.

3.4.4. *WebSocket*

Provođenje više interakcija istovremeno je omogućeno grupiranjem konekcija u posebne sobe. Soba je kanal na koji se korisnici mogu spojiti ili odspojiti. Time će svaka interakcija biti lokalizirana i neće slučajno prelijevati u druge interakcije. Svaka zaprimljena poruka sadrži šifru sobe s kojom komunicira. Na ključnim događajima, poziva se funkcija iz servisa interakcije za stvaranje i slanje sigurnosne kopije instance nastavnika.

Poruke nastavnikovog klijenta

handlerJoin Poruka dojavljuje spajanje nastavnika na instancu interakcije. U slučaju pada sustava, instanca interakcije više nije u sustavu pa se šalje

zadnji spremljeni snimak interakcije i ponovo ju pokreće. Nastavnikov *socket* se spaja na zadanu sobu i vraćaju mu se poruke potrebne za nastavak interakcije, ako se ponovo priključuje na nju.

deleteRoom Nastavnik je službeno završio interakciju i briše podatke sobe s poslužitelja. Prije brisanja se šalje poruka svim studentima u sobi o toj akciji.

newQuestion Umeće se novo pitanje u instancu interakcije. Provodi se ažuriranje podataka i raspodjela pitanja. Ovu akciju nije potrebno obavijestiti studente.

nextQuestion Naglašuje korak na sljedeće pitanje u nizu. Vraća se na početno stanje statistike trenutnog pitanja i šalje poruku studentima da je došlo do promjene pitanja uz priloženo pitanje.

stopCurrentQuestion Trenutno pitanje više neće prihvaćati odgovore. Potrebno obavijestiti sve studente u sobi. Nastavniku se vraća statistika trenutnog pitanja.

questionsAllowed Mijenja se dozvola za postavljanje pitanja. Studentima se šalje poruka o novom stanju.

done Interakcija je prošla sva pitanja. Studenti dobivaju poruku o završetku i poruku sa svim pitanjima na interakciji. Boduju se studenti i ažuriraju im se podaci. Svaki student dobiva posebnu poruku svojih rezultata i odgovora. Nastavnik dobiva poruku sa svim podacima uključenih korisnika.

Poruke studentovog klijenta

joinRoom Poruka kojom se student pridružuje u zadanu sobu. Ako soba ne postoji, vraća se poruka za ponovi pokušaj. Uspješnim pridruživanjem u sobu vraćaju se informacije i trenutno stanje interakcije. Ako se student prvi put pridružuje sobi, stvara se nova podatkovna struktura koja sprema njegove podatke i odgovore. Student koji se pridružio usred interakcije dobiva poruke o trenutnom pitanju i stanju. Pridruživanjem na kraju kviza je moguće samo ako se student pridružio bar jednom prije kraja. U tom slučaju dobiva sva pitanja, svoje odgovore i bodove.

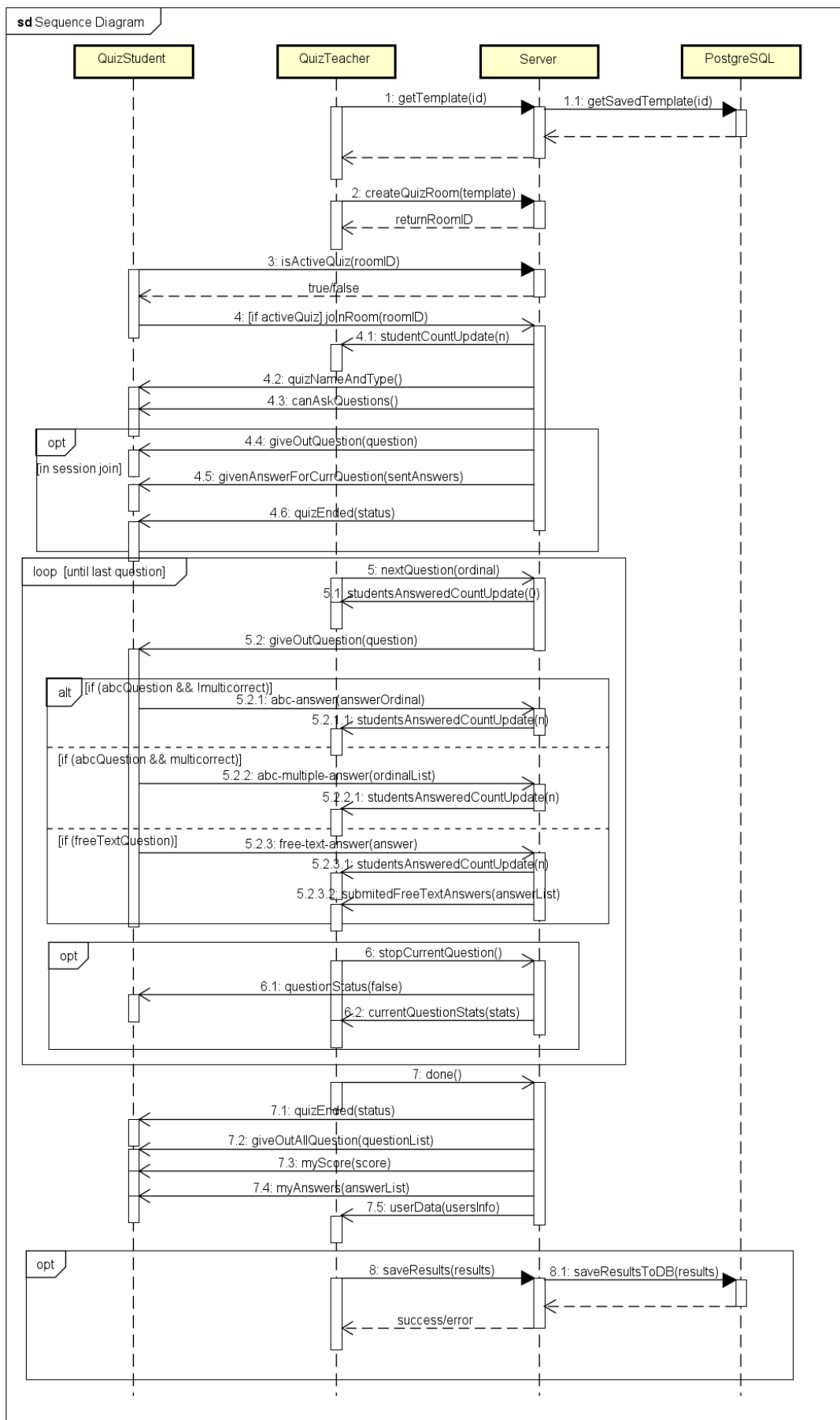
abc-answer Poslan ili je izmijenjen odgovor na pitanje s ponuđenim odgovorima. Nastavniku se šalje promjena zbroja poslanih odgovora.

abc-multiple-answer Poslani ili su izmijenjeni odgovori na pitanje s više točnih odgovorima. Nastavniku se šalje promjena zbroja poslanih odgovora.

free-text-answer Poslan odgovor slobodnog tipa. Nastavniku se šalje promjena zbroja poslanih odgovora i sami odgovor.

askQuestion Student je poslao pitanje namijenjeno nastavniku. Isto pitanje se proslijeđuje nastavniku.

Općeniti tijek informacija od početka pa do kraja kviza je prikazan na sekvencijskom dijagramu 3.4. Na njemu se mogu vidjeti najčešće *WebSocket* poruke u interakciji te dohvat i spremanje podataka.



Slika 3.4: Sekvencijski dijagram - Općeniti tijek kviza

Autentifikacija

Za autentifikaciju poruka koristi se postojeći sustav sjednice na Edgaru, omo-
tan oko *WebSocket* poruka, prema službenom primjeru [13]. Time svaka poruka
poslana s klijenta sadrži informacije o sjednici korisnika. Svakoј nastavnikovoj
poruci se provjerava autentičnost. Studentove poruke se provjeravaju samo ako
interakcija nije postavljena na anoniman tip. Neki korisnički podaci iz sjednice
se koriste u interakciji.

3.5. Spremanje u bazu podataka

Na PostgreSQL bazi podataka su stvorene dvije nove tablice za spremanje predlo-
žaka kviza i rezultata interakcije. Samo nastavnik može pristupiti i manipulirati
vlastite entitete. Tablica **Predložci kviza** 3.1 sadrži općenite podatke za opis
predložka, autorstvo i sami predložak u JSON formatu. Tablica **Spremljeni
rezultati** 3.2 služi za spremanje svih podataka završene interakcije za kasniji
pregled. Uz JSON podatke, sprema općenite informacije o interakciji poput nas-
lova, vrste i broja sudionika. Obje tablice spremaju vrijeme kada je novi entitet
dodan ili izmijenjen i ime korisnika, koji je obavio tu akciju. Identifikacijski
brojevi oba entiteta se samostalno generiraju u seriji i ne unose se izravno.

Table 3.1: Tablica spremljenih predložaka

Predložci kviza		
id	INT	Jedinstveni ID predložka.
autor_id	INT	Jedinstveni ID autora koji ga povezuje s korisnikom aplikacije.
naslov	VARCHAR	Dani naslov predlošku.
broj pitanja	INT	Zbroj svih pitanja u predlošku.
opis	VARCHAR	Proizvoljan opis predložka za lakše snalaženje.
je javan	BOOL	Dozvola za preuzimanje predložka ostalim korisnicima.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Predlošci kviza		
predložak kviza JSON	JSONB	Podaci predloška spremljeni u JSON formatu. Sadrži: informacije o pitanjima, bodovanje za svako pitanje i pretpostavljeno bodovanje.
vrijeme nastanka	TIMESTAMP	Vrijeme prvog spremanja podatka u bazu.
vrijeme izmjene.	TIMESTAMP	Vrijeme izmjene podatka u bazi.
izmijenio korisnik	VARCHAR	Ime korisnika koji je manipulirao podatkom.

Table 3.2: Tablica spremljenih rezultata

Spremljeni rezultati		
id	INT	Jedinstveni ID rezultata.
autor_id	INT	Jedinstveni ID autora koji ga povezuje s korisnikom aplikacije.
naslov	VARCHAR	Naslov korištenog predloška.
vrsta kviza	VARCHAR	Odabrana vrsta interakcije
broj sudionika	INT	Broj studenata koji su pristupili kvizu.
spremljeni rezultati JSON	JSONB	Svi podaci interakcije spremljeni u JSON formatu. Sadrži: informacije o pitanjima iz predloška, informacije o novo dodanim pitanjima, bodovanje za svako pitanje i informacije svakog studenta (ime, prezime, ostvareni bodovi, odgovori na pitanja, kada se pridružio kvizu i je li ostao do kraja kviza).

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Spremljeni rezultati		
vrijeme nastanka	TIMESTAMP	Vrijeme prvog spremanja podatka u bazu.
vrijeme izmjene	TIMESTAMP	Vrijeme izmjene podatka u bazi.
izmijenio korisnik	VARCHAR	Ime korisnika koji je manipulirao podatkom.

4. Korištene tehnologije

4.1. Angular13

Angular [1] je aplikacijski razvojni okvir, baziran na programskom jeziku TypeScript i razvijen u Googleu. Koristi se za razvoj jednostraničnih web-aplikacija, koje se mogu poslužiti na različitim uređajima. U Angularu se aplikacije projektiraju iz korijenskog modula, koji može sadržavati različite komponente za enkapsulaciju funkcionalnosti. Servisi služe za dijeljenje funkcionalnosti preko više komponentata. Komponenta se sastoji od razreda, metapodataka i HTML predloška. Nastavnikova i studentova web-aplikacija su razvijene u tom razvojnem okruženju i dijele sličnu strukturu.

4.2. TypeScript

TypeScript [15] je strogo tipizirani programski jezik, proširenje JavaScript jezika. Prikladan je za pisanje projekata velikih opsega. Kao proširenje JavaScripta, on podržava postojeće JavaScript programe. Prije izvršavanja se prevđa u JavaScript. Izgrađene statičke datoteke web-aplikacije se spremaju na poslužitelju Edgar u JavaScript jeziku i tako se poslužuju.

4.3. Bootstrap

Projektiran [2] u tvrtki Twitter, Bootstrap je CSS razvojni okvir za projektiranje dizajna web stranice, koji se može prilagoditi za različite rezolucije. Sadrži HTML, CSS i obrasce bazirane u JavaScriptu za tipografiju, forme, gumbove, navigaciju i ostale komponente sučelja. Njime se osigurava prilagodljiv izgled studentove i nastavnikove Angular aplikacije, prvo dizajnirane za mobilne uređaje pa zatim za veće uređaje.

4.4. Socket.IO

Socket.IO [12] je biblioteka koja pruža dvosmjernu komunikaciju između klijenta i poslužitelja. Izgrađen je na *WebSocket* protokolu i uključuje zamjenski protokol baziran na HTTP-u. U slučaju gubitka veze klijent odmah pokušava ponovo uspostaviti vezu. To se implementira pomoću mehanizma otkucaja srca, gdje se periodički provjerava status veze. Socket.IO je temelj interakcije u projektu, zaslužna za održavanje pravovremene interakcije te razmjenu poruka između nastavnika i studenata.

4.5. Node.js

Node.js [11] je izvršno okruženje za izvršavanje programskog koda pisanog u JavaScriptu izvan internetskog preglednika. Koristi arhitekturu događaja za asinkrono izvođenje procesa. Taj dizajn optimizira propusnost i skalabilnost u web-aplikacijama. Servis interakcije, koji se nalazi u sustavu Edgar, se izvršava u Node.js-u.

4.6. Express.js

Express.js [5] je poslužiteljski razvojni okvir za Node.js. On se koristi za razvoj poslužiteljskog dijela web-aplikacija i API-ja. Edgar je razvijen pomoću te tehnologije i sustav interakcije je samo nadodan na postojeću infrastrukturu, stvaranjem vlastitog API-a za komuniciranje s bazom podataka.

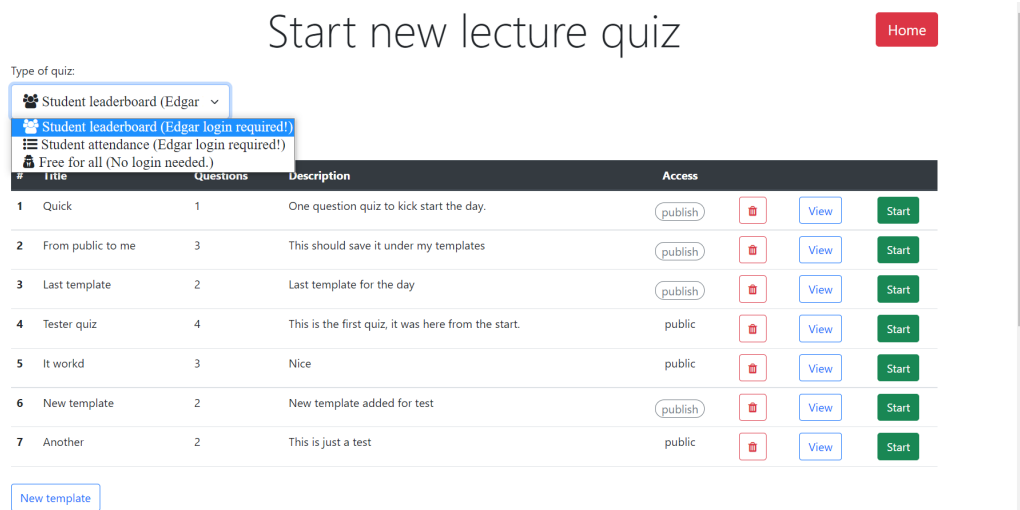
4.7. Chart.js

Chart.js [3] je JavaScript biblioteka za prikaz grafikona na web-stranicama. U svrhu ovog projekta koristi se za prikaz raspodjele predanih odgovora na stupčastom dijagramu u nastavnikovoj Angular aplikaciji.

5. Aplikacija za interakciju studenata i nastavnika u sustavu Edgar

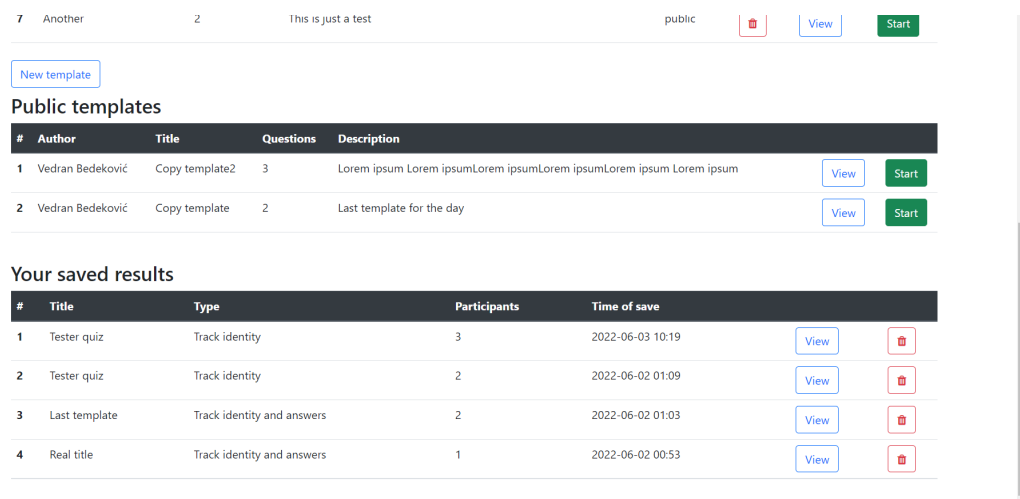
5.1. Nastavnička aplikacija

Nastavnik je na početku dočekan s opcijama pokretanja kviza vidljivih na slici 5.1 ili pregleda spremljenih podataka. Prije pokretanja može promijeniti vrstu kviza u padajućoj listi. Rangiranje studenata prati odgovore autentificiranih studenata i omogućuje nastavniku detaljan pregled danih odgovora. Postoji anonimni kviz gdje nastavnik dobiva samo općenite informacije o rezultatima kviza. Ova vrsta kviza ne zahtijeva prijavu u sustav Edgar i time omogućuje spajanje publike van akademskog okruženja. Implementirana nova vrsta kviza je mješavina prošlih dviju. Zahtijeva prijavu u sustav, ali ne veže odgovore za specifičnog studenta. Nastavniku posluži ako želi zadržati anonimnost odgovora i uz to pratiti tko je sve prisustvovao predavanju.



Slika 5.1: Nastavnikova početna stranica - 1. dio

U prvom planu je tablica spremljenih predložaka. Ona prikazuje opće informacije o spremljenom predlošku. Predložak se može postaviti javnim i time prepustiti ostalim nastavnicima na korištenje. Uz to se predložak može brisati, uređivati ili detaljno pregledati te izravno pokrenuti s početne stranice. Brisanje i objavljivanje zahtjeva dodatnu potvrdu prije izvršavanja. Nastavnik može dodatno zatražiti javne predloške i prikazat će mu se slična tablica vidljivoj na slici 5.2. Pošto je to javan predložak, prikazan je autor predloška i omogućene su samo opcije pregleda i izravnog pokretanja kviza.



Slika 5.2: Nastavnikova početna stranica - 2. dio

Pritiskom na "New template" otvara se forma za stvaranje novog predloška prikazana na slici 5.3. Dodavanje pitanja otvara podformu za unos novog pitanja.

Popunjen predložak se može vidjeti na slici 5.6. Svako dodano pitanje može naknadno promijeniti poziciju, biti uređeno i obrisano. Pravilan predložak mora imati popunjena sva polja za unos i sadržavati barem jedno pitanje. Nakon ispunjavanja svih uvjeta nastavnik može spremi predložak u bazu podataka i pokrenuti. On ne mora nužno obaviti obje akcije. Sporedne opcije omogućuju izbor između samog spremanja predloška ili samog pokretanja, čime će predložak biti zaboravljen nakon kviza.

Start new lecture quiz Home

Type of quiz: Student leaderboard (Edgar) ▾

New quiz template

Title:

Description:

Default grading: Correct / Answered: No answer: Wrong: Partial:

#Q:

Max points:

[Add question](#)

[Save and start](#)
[Just save](#)
[Just start](#)
[Reset](#)
[Cancel](#)

Slika 5.3: Forma za stvaranje predloška

Pregled predloška otvara istu, samo popunjenu formu kao za dodavanje novog predloška, malo promijenjenog izgleda i opcija, prikazana na slici 5.6. Izmjene se na jednak način mogu pokrenuti i spremi ili se odabere samo jedna od tih opcija.

Forma pitanja, prikazana na slici 5.4, omogućuje unos teksta pitanja i bodovanja. Ovisno o kojoj vrsti pitanja je riječ prikazat će se različite opcije za bodovanje i dodavanje odgovora. Forma označena za pitanje ponuđenih odgovora se može vidjeti na 5.5. To stanje forme dodatno traži popunjene unose za tekst odgovora koliko god ih bilo. Nastavnik može odrediti proizvoljan broj odgovora. Može se označiti više od jednog točnog odgovora, ali nije nužno niti jednog označiti. U tom slučaju pitanje nema točnih odgovora i koristi sistem bodovanja kao pitanje otvorenog odgovora, gleda se samo je li student odgovorio.

New question

Select ordinal:

Grading for: Answered: No answer:

HTML Question Text:

☐ ABC Question

Add Reset Cancel

Slika 5.4: Stvaranje novog pitanja

New question

Select ordinal:

Grading for: Correct: No answer: Wrong: Partial:
ignored if one correct.

HTML Question Text:

☒ ABC Question

Answers: Correct:

☐

☐

+

Add Reset Cancel

Slika 5.5: Stvaranje novog ABC pitanja

Odabirom spremljenih rezultata iz tablice, prikazane na dnu slike 5.2, otvara se pregled tih rezultata kao na slici 5.7. Nastavnik može pregledati pitanja s kviza i vidjeti njihovo bodovanje. Ispod njih se nalazi tablica s prikladnim rezultatima ovisno o vrsti kviza. Anonimni kviz nije moguće spremiti pa se neće prikazati u tablici spremljenih rezultata. Kviz u kojem se prati prisutnost i što je student odgovorio omogućuje detaljan pregled pojedinog studenta. Detaljan pregled studenta, prema slici 5.8, prikazuje informacije o studentu i postignuti broj bodova. Dodatno, nastavnik se kreće po pitanjima i uspoređuje točne i dane odgovore.

View and edit before starting

Title:

Tester quiz

#Q:

4

Description:

This is the first quiz, it was here from the start.

Max points:

8

Default grading:

Correct / Answered:

0

No answer:

0

Wrong:

0

Partial:

0

1.) Is this the first question?

a Yes

b Hello World!

c No

Grading: 2 / 0 / -1 / 0.5

↑

Edit

Remove

↓

2.) What is this?

a Code

b Pain

c **A question**

Grading: 2 / 0 / -1

↑

Edit

Remove

↓

3.) This is a question. Can it be formatted with **html elements**?

a No

b Yes

Grading: 3 / 0 / -1.5

↑

Edit

Remove

↓

4.) What do you think of this project?

Answers:

👤 Student answers show up here

Grading: 1 / 0

↑

Edit

Remove

↓

Add question

Save and start

Just save

Just start

Back

Slika 5.6: Pregled i izmjena predloška

35

Start new lecture quiz

Home

Results for quiz: Last template

Close

From: 2022-06-02 01:03

Student leaderboard

2.) 4 / 0 / -2

Max: 12

One more q

<

a

 Ne

b

 Da

>

#	Name	Surname	Points	%	
1	Mislav	LastName2487	8	66.67%	Details
2	Mario	LastName5959	6	50%	Details

Slika 5.7: Pregled spremljenog rezultata

Start new lecture quiz

Home

Results for quiz: Last template

Close

From: 2022-06-02 01:03

Student leaderboard

Details for:

Mislav LastName2487

Score: 66.67%

Back

Overview: 1.

One question

a

 No

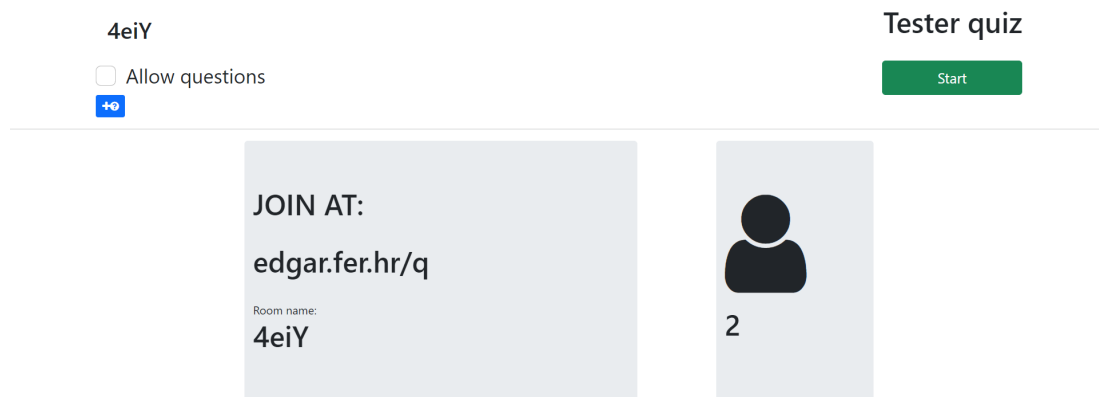
b

 Yes

>

Slika 5.8: Detaljan pregled studentovih odgovora

Nakon pokretanja kviza iz nekog predloška, nastavniku se predstavlja početno stanje kviza. Na zaglavlju se prikazuju sve radnje koje može činiti u danom trenutku. U tom stanju je omogućen prikaz samo prvog pitanja. Prikazana je šifra sobe na koju se studenti mogu spojiti i broj pridruženih studenata ažuriran u stvarnom vremenu. Pregled opisanog je prikazan na slici 5.9.



Slika 5.9: Početak kviza

Nastavnik na zaglavlju ima opciju dozvole ili zabrane postavljanja pitanja u bilo kojem trenutku interakcije. Postavljena pitanja se prikazuju ispod zaglavlja i sadrži opcije odbacivanja svih do sad postavljenih pitanja ili prelazak na sljedeće postavljeno pitanje. Izgled je prikazan na slici 5.10.




Slika 5.10: Prikaz studentovog pitanja

Na aktivno pitanje, vidljivo na slici 5.11, se prihvaćaju poslani odgovori. Vidi se tekst pitanja i ponuđeni odgovori. Nastavnik uz to ima uvid u broj danih odgovora kroz broj trenutno spojenih studenata. Može zaustaviti prikupljanje odgovora i prikazati statistiku zadanog pitanja ili odmah prebaciti na sljedeće pitanje. Kada dođe do zadnjeg pitanja to će biti naznačeno promjenom gumba "*Next question*" u "*End quiz*".

Slika 5.11: Prikaz pitanja

4eiY

☐ Allow questions




 Student leaderboard



What do you think of this project?

Answers:



Slika 5.12: Pitanje bez odgovora

38

IzMU

Tester quiz

☐ Allow questions

Stop answers

End test

Student attendance

3 / 3

What do you think of this project?

Answers:

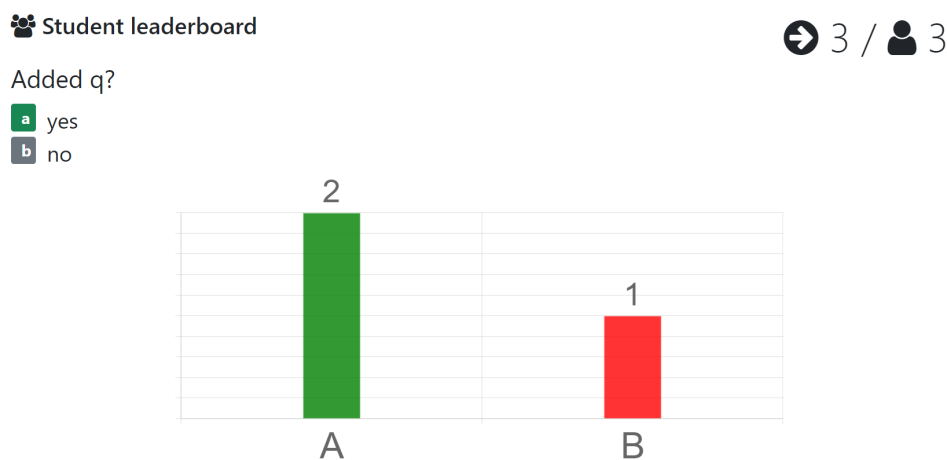
Good

Bad

Great

Slika 5.13: Pitanje s odgovorima

Ako se nastavnik odluči zadržati na pitanju nakon prekida prikupljanja odgovora, prikazat će se stupčasti dijagram raspodjele odgovora. Točan odgovor će biti označen na pitanju i na dijagramu će biti naznačen zelenim stupcem, kao na slici 5.14.



Slika 5.14: Prikaz statistike

U bilo kojem trenutku nastavnik može otvoriti formu 5.4 za dodavanje novog pitanja. Na formi je dodatno ograničenje da ne može odabrati redni broj pitanja manjeg ili jednakog od trenutnog. Bodovanje je popunjeno prema pretpostavljenim vrijednostima bodovanja kviza, ali moguće su izmjene.

Dolaskom do kraja kviza prikazat će se rezultati sukladni s vrstom kviza, isto kao i u spremljenim rezultatima. Prvo je vidljiv samo pregled pitanja. Tablica se može dodatno prikazati i sakriti pritiskom gumba. Nastavnik ne mora odmah

obraditi podatke, već ih može spremi za kasniji pregled. Opcije prikaza i spremanja nisu moguće za anonimni kviz. Rezultati kraja kviza su vidljivi na slikama 5.15 i 5.16. Pritiskom "Home" gumba, briše se instanca kviza, odspajaju se svi studenti i nastavnička aplikacija se preusmjerava na početnu stranicu Edgara.

IzMU

Tester quiz

☐ Allow questions

Home

☰ Student attendance

3.) 3 / 0 / -1.5

Max: 8

←

This is a question. Can it be formatted with **html elements?**

→

a No

b Yes

Show Attendance

Save Attendance

Slika 5.15: Kraj kviza i prikaz rezultata

Hide Attendance

Save Attendance


Attendance list

#	Name	Surname	From the start	Late join at q#	Present at the end
1	Marko	LastName2486	✓	—	✓
2	Mislav	LastName2487	✓	—	✓
3	Mario	LastName5959	✓	—	✓

Slika 5.16: Tablica prisutnosti na kraju kviza

5.2. Studentska aplikacija


Otvaranjem aplikacije, studenta se traži unos pravilne šifre za željeni kviz. U slučaju da kviz nije anonimni i student nije prijavljen, zatražuje se prijava u sustav. Nakon prijave se preusmjerava natrag na formu za unos šifre kviza. Prikaz forme je vidljiv na slici 5.17.



Enter

Slika 5.17: Unos šifre

Unosom šifre postojeće sobe, student je preusmjeren u kviz. Ako je kviz u tom trenutku već započeo, studentu je poslužno trenutno pitanje i bilježi se kao kasno pristupanje. Početak kviza je vidljiv na slici 5.18. Na zaglavlju je prikazana šifra sobe, naslov kviza i ikona trenutne vrste interakcije.

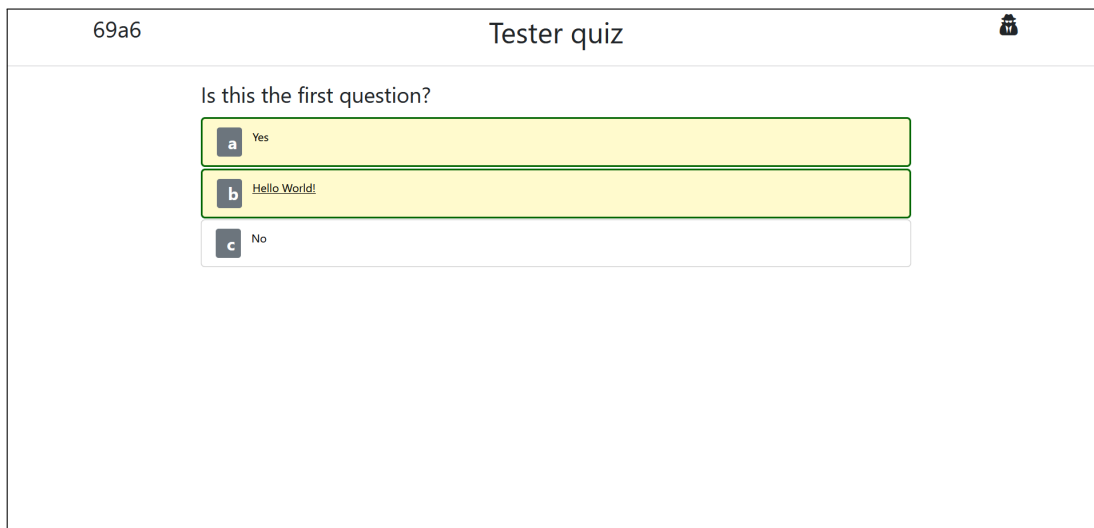
69a6 

Tester quiz

Welcome to room:
69a6
Get ready!

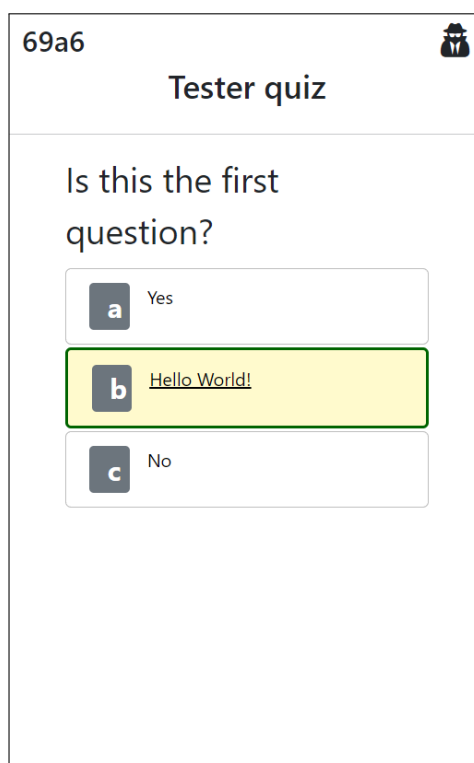
Slika 5.18: Početak kviza prikazan na mobilnom uređaju

Kada nastavnik otvori pitanje s ponuđenim odgovorima, student može odabrati i mijenjati svoje odgovore sve dok je pitanje aktivno. Taj prikaz se može vidjeti na slikama 5.19 i 5.20.



The screenshot shows a desktop view of a quiz application. At the top, there is a header bar with the text "69a6" on the left, "Tester quiz" in the center, and a small icon of a person on the right. Below the header, the main content area displays the question "Is this the first question?". Underneath the question, there are three answer options, each in a yellow box with a green border. Option 'a' is "Yes", option 'b' is "Hello World!", and option 'c' is "No". Option 'b' is currently selected, indicated by a green highlight.


Slika 5.19: Prikaz pitanja i odabira odgovora



The screenshot shows a mobile view of the same quiz application. The header bar at the top contains "69a6" on the left, "Tester quiz" in the center, and a small icon of a person on the right. The main content area displays the question "Is this the first question?". Below the question, there are three answer options, each in a white box with a gray border. Option 'a' is "Yes", option 'b' is "Hello World!", and option 'c' is "No". Option 'b' is currently selected, indicated by a green highlight.

Slika 5.20: Prikaz pitanja i odabira odgovora na mobilnom uređaju

Odgovor na pitanje otvorenog odgovora se ne može promijeniti. Njegov prikaz 5.21 se sastoji od teksta pitanja i polja za unos odgovora.

69a6 


Tester quiz

What do you think of this project?

Enter your answer here...

Submit answer

Slika 5.21: Prikaz pitanja slobodnog odgovora na mobilnom uređaju

VcRk 

Tester quiz

Is this the first question?

a Yes

b Hello World!

c No

Ask a question:

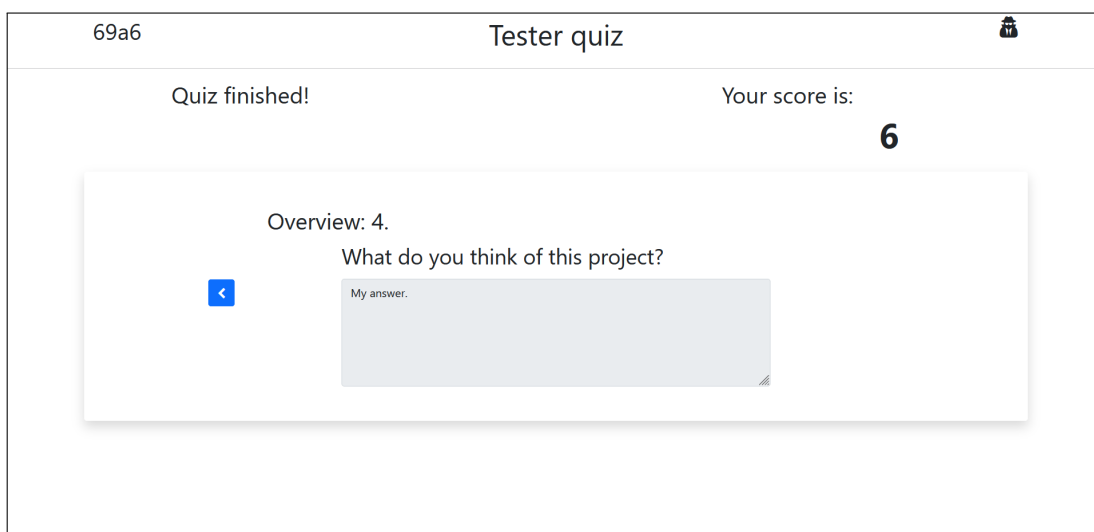
Write question here

Ask

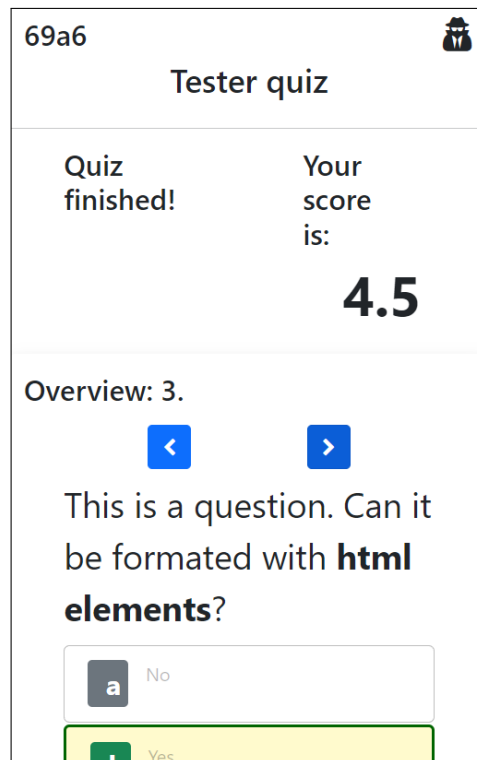
Slika 5.22: Prikaz forme za postavljanje pitanja ispod aktivnog pitanja na mobilnom uređaju.

Studenti mogu postavljati pitanja za vrijeme interakcije, ako je nastavnik to omogućio. Forma 5.22 za unos pitanja se dinamično prikaže ili sakrije ovisno što nastavnik odluči.

Na kraju kviza studenti dobivaju ostvarene bodove i mogu pregledati svoja rješenja te usporediti ih s točnim, ako ih ima. Pregled je ostvaren pogledom na jedno pitanje i navigacijom na ostale. Student vidi svoje dane odgovore i točne odgovore. Slika 5.23 prikazuje pregled na računalu, a slika 5.24 na mobilnom uređaju.

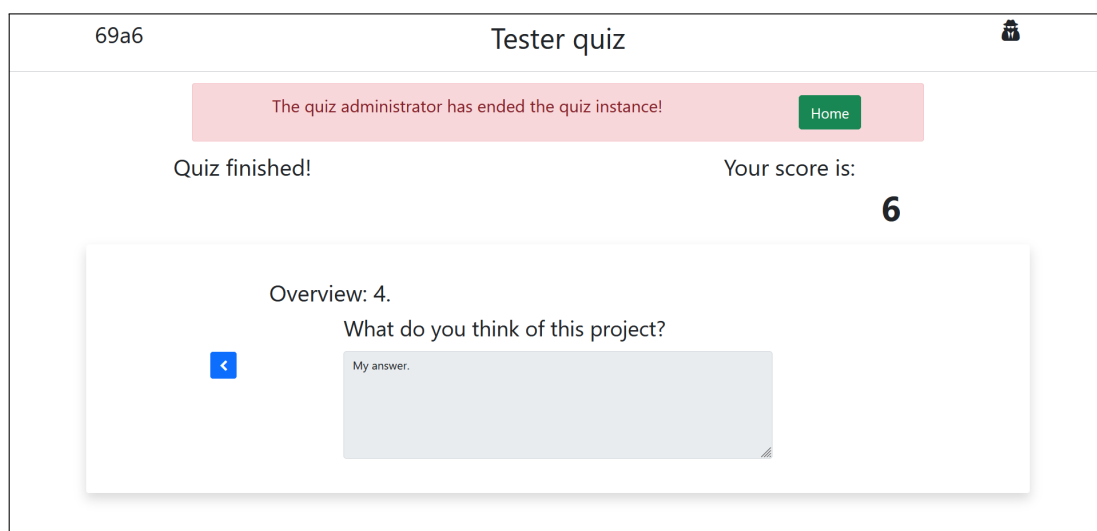


Slika 5.23: Pregled i kraj kviza



Slika 5.24: Pregled i kraj kviza na mobilnom uređaju

Nakon što nastavnik izađe s interakcije, studentima se prekida *WebSocket* veza i iskače im prozor s tom obavijesti. Na njemu mogu pritisnuti gumb za povratak na početnu stranicu Edgara.



Slika 5.25: Nastavnik je završio interakciju

6. Zaključak

Tehnologije se brzo razvijaju i potrebno je održavati sustave u skladu s tim procesom. Razvojem Angulara došlo je do promjena u ustroju, knjižnicama i obrascima pisanja koda odnosno sam razvojni okvir je evoluirao te je stare projekte potrebno osvježiti i izmijeniti u skladu s novim normama i funkcionalnostima, kako bi održali njegov životni vijek i otvorili ga na laku buduću nadogradnju. Projektirani sustav uspješno održava duh svog prethodnika i proširuje ga novim mogućnostima. Uspješno se provodi sinkrona interakcija između nastavnika i njegovih studenata, otporna na gubitak veze sudionika i pada sustava. Nastavnička aplikacija je fleksibilnija naspram prošlog sustava. Njome se sada može lako dodati i izmijeniti sadržaj aktivnog kviza. Sakupljeni rezultati su sada integrirani u Angular aplikaciji i otvoreni su na daljnja proširenja. Studenti mogu anonimno postavljati pitanja nastavniku i poslužuje im se vlastiti pregled rezultata na kraju interakcije. Nastavnik može dobiti identitet priključenih studenata i sadržati anonimnost njihovih odgovora postavljanjem kviza na novi model autentifikacije. Aplikacije mijenjaju raspodjelu i veličinu elemenata sučelja, ovisno u kojoj rezoluciji se prikazuju.

Za daljnji razvoj se preporučuje redizajn aplikacija, pogotovo nastavnikove aplikacije za prilagođen prikaz i upravljanje kvizom na mobilnom uređaju. Razmotriti proširenje podržanih vrsta sadržaja poput zadataka na dopunjavanje, zadaci permutacije zadanih pojmova i videomaterijala. Istražiti i implementirati drugi način pridruživanja studenta na interakciju, kao preko obavijesti koje šalju stranice na internetskim preglednicima.

LITERATURA

- [1] Angular. Getting started with angular, veljača 2022. URL <https://angular.io/start>. 2. lipnja 2022.
- [2] Bootstrap. Get started with bootstrap, 2022. URL <https://getbootstrap.com/docs/5.2/getting-started/introduction/>. 2. lipnja 2022.
- [3] Chart.js. Chart.js, 2022. URL <https://www.chartjs.org/>. 2. lipnja 2022.
- [4] Emma Cullen. How to host a mentimeter quiz like a pro!, svibanj 2018. URL <https://www.mentimeter.com/blog/awesome-presentations/how-to-host-a-mentimeter-quiz-like-a-pro>. 15. ožujka 2022.
- [5] Express.js. Express - node.js web application framework, 2022. URL <https://expressjs.com/>. 2. lipnja 2022.
- [6] Google. Communicating with backend services using http, 2021. URL <https://angular.io/guide/http>. 15. ožujka 2022.
- [7] Sebastian Gundersen, George Schaathun, i André Storhaug. Jazzquiz, veljača 2021. URL https://moodle.org/plugins/mod_jazzquiz. 10. ožujka 2022.
- [8] Seth Gwartney. How to create a real-time app with socket.io, angular, and node.js, siječanj 2021. URL <https://www.digitalocean.com/community/tutorials/angular-socket-i>. 13. travnja 2022.
- [9] Asim Hussain. Angular crash course, 2020. URL <https://codecraft.tv/courses/angular/quickstart/overview/>. 15. ožujka 2022.
- [10] Mozilla Developer Network. The websocket api (websockets), lipanj 2022. URL https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API. 5. lipnja 2022.

- [11] Node.js. About node.js, 2022. URL <https://nodejs.org/en/about/>. 2. lipnja 2022.
- [12] Socket.IO. Introduction - what socket.io is, lipanj 2022. URL <https://socket.io/docs/v4/>. 2. lipnja 2022.
- [13] Socket.IO. How to use with express-session, 2022. URL <https://socket.io/how-to/use-with-express-session>. 30. svibnja 2022.
- [14] Traversy Media. Angular crash course, svibanj 2021. URL <https://www.youtube.com/watch?v=3dHNOWTI7H8>. 15. ožujka 2022.
- [15] TypeScript. What is typescript?, 2022. URL <https://www.typescriptlang.org/>. 2. lipnja 2022.
- [16] Morten Versvik, Johan Brand, i Jamie Brooker. Kahoot, 2013. URL <https://kahoot.com/schools/how-it-works/>. 11. ožujka 2022.
- [17] Johnny Warström i Niklas Ingvar. Mentimeter, 2014. URL <https://www.mentimeter.com/features/quiz-presentations>. 15. ožujka 2022.

Nadogradnja aplikacije za interakciju studenata i nastavnika u sustavu Edgar

Sažetak

Interakcijski sustavi postaju sve više zastupljeni u akademskim sredinama i dolazi do zahtjeva za proširenje njihovih mogućnosti. Provedeno je istraživanje razvojnog okruženja Angular13 i postojećih sličnih rješenja za razvoj novog sustava interakcije. Razvijeni sustav je nadogradnja postojećeg sustava za interakciju s publikom na platformi Edgar. Funkcionira kao kviz u stvarnom vremenu, u kojem nastavnik upravlja i dodaje sadržaj, a studenti daju povratne informacije. Interakcija je ostvarena implicitnom razmjenom podataka studentove i nastavnikove Angular aplikacije, koje komuniciraju s poslužiteljem preko *WebSocket* protokola kojeg upravlja JavaScript biblioteke *Socket.IO*.

Ključne riječi: Edgar, sustav za interakciju s publikom, kviz, Angular, Socket.IO, WebSocket, Node.js, JavaScript

Upgrade of the student-teacher interaction application in the Edgar system

Abstract

Interactive systems are becoming ever more present in the academic scene, and as a result, demand for the expansion of their capabilities. To implement a new system, research on the Angular13 framework and other similar existing solutions was conducted. The developed system is an upgrade of the existing audience interaction system on the Edgar platform. It functions as a real-time quiz, in which the teacher manages and adds content and students give feedback. The interaction is handled implicitly with the exchange of data from the student's and teacher's Angular application to the server through the *WebSocket* protocol, managed by the JavaScript library *Socket.IO*.

Keywords: Edgar, audience interaction system, quiz, Angular, Socket.IO, WebSocket, Node.js, JavaScript