Aplikacija za učenje i ponavljanje gradiva anatomije

Petković, Nikola

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:168:437664

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. 1615

APLIKACIJA ZA UČENJE I PONAVLJANJE GRADIVA ANATOMIJE

Nikola Petković

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1615

APLIKACIJA ZA UČENJE I PONAVLJANJE GRADIVA ANATOMIJE

Nikola Petković

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1615

Pristupnik:	**Nikola Petković (0036535584)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	doc. dr. sc. Tomislav Jagušt

Zadatak: **Aplikacija za učenje i ponavljanje gradiva anatomije**

Opis zadatka:

Potrebno je osmisliti i kreirati aplikaciju koja će studentima prve godine medicine, ali i svima ostalim korisnicima zainteresiranim za ovu temu, omogućiti učenje i vježbanje gradiva iz anatomije. Potrebno je proučiti postojeće slične aplikacije i dostupne izvore podataka te osmisliti način na koji se ti podaci mogu iskoristiti za generiranje sadržaja u aplikaciji. Nadalje, potrebno je kreirati prikladnu bazu podataka, popuniti je slikama i odgovarajućim opisima te generirati pitanja. Uz mogućnost samostalnog vježbanja aplikacija treba podržati i opciju natjecanja dva ili više igrača u obliku interaktivnog kviza. Studenti će osvojiti odgovarajući broj bodova ovisno o točnosti i brzini odgovaranja na pitanja, a bodovi i trenutni poredak igrača trebaju biti vidljivi na zasebnoj stranici unutar administracijskog sučelja.

Rok za predaju rada: 14. lipnja 2024.

Zahvaljujem mentoru doc. dr. sc. Tomislavu Jaguštu na savjetima i vođenju prilikom izrade ovog rada te se zahvaljujem kolegici s Medicinskog fakulteta Sveučilišta u Zagrebu, Lauri Ogrizek, na velikoj pomoći prilikom izrade baze pitanja potrebne za ovaj rad

Sadržaj

Uvod 1

1. Slične aplikacije 2

1.1. Kahoot 2

1.2. Quizlet Live 3

1.3. Quizizz 4

1.4. Mentimeter 5

2. Ideja rada i opis dijelova sustava 7

3. Model podataka 8

3.1. Relacijski model podataka 8

3.2. Opisi entiteta 9

3.3. Opis relacijskog modela podataka 10

4. Korištene tehnologije i alati 11

4.1. PostgreSQL 11

4.2. React.js 12

4.2.1. React Router DOM 13

4.2.2. axios 14

4.3. Spring Boot 14

4.4. Socket.io 16

4.5. GitHub 17

4.6. Heroku 17

5. Korištenje aplikacije 19

6. Daljnji razvoj 29

7. Mišljenja studenata Medicinskog fakulteta 30

Zaključak 31

Literatura 32

Sažetak 33

Summary 34

Skraćenice 35

Uvod

U današnjem digitalnom dobu, korištenje tehnologije u obrazovanju, a isto tako i u medicini, postaje sve važnije i prisutnije u svakidašnjem životu. U ovom završnom radu bavimo se razvojem obrazovne aplikacije namijenjene prvenstveno studentima prve godine medicine, ali i svima zainteresiranima za anatomiju, to jest medicinu općenito. Glavni cilj aplikacije je olakšati ponavljanje i vježbanje gradiva iz anatomije te bolje motivirati studente pružajući im novi način pripremanja za ispite.

Aplikacije je napravljena na način da postoji opcija samostalnog prolaska kroz pitanja kako bi studenti mogli ponoviti gradivo koje su prijašnje naučili. Također postoji izbor kreacije „sobe“ kojoj mogu pristupiti drugi studenti ili prijatelji te svi koji se nalaze u istoj „sobi“ odgovaraju na ista pitanja. Ovisno o točnosti i brzini odgovora svatko dobiva odgovarajući broj bodova. Pri završetku kviza prikazuje se tablica u kojoj se nalaze svi igrači i njihov odgovarajući broj bodova.

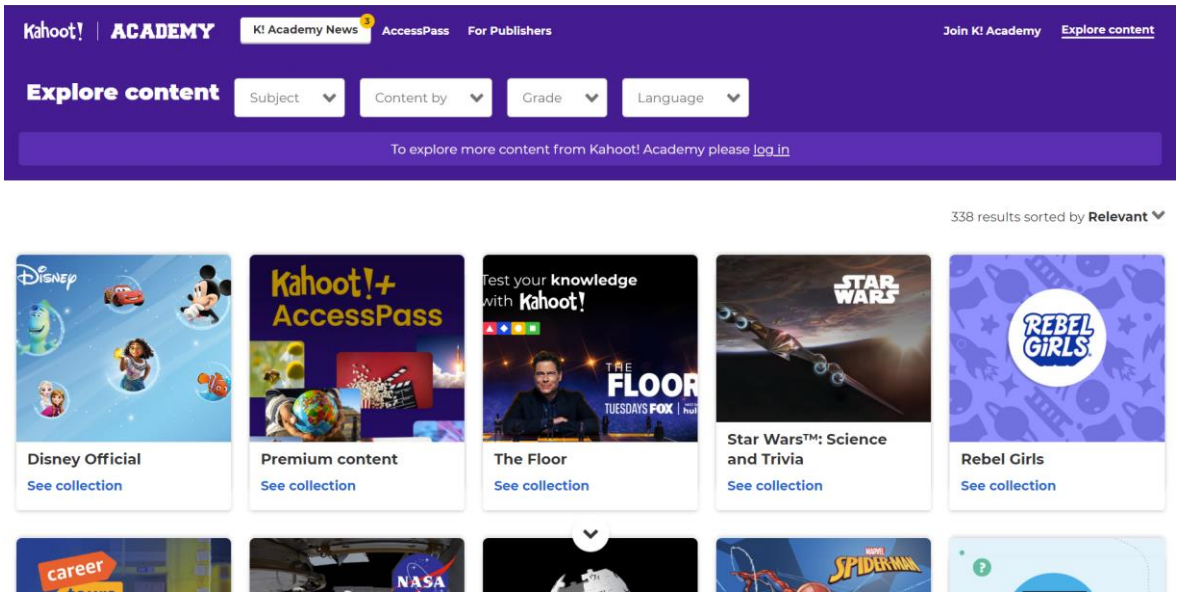
Baza pitanja je kreirana uz pomoć kolegice s Medicinskog fakulteta Sveučilišta u Zagrebu, Laure Ogrizek. Kolegica je poučena svojim iskustvom rješavanja ispita pomogla pri kreaciji pojedinog pitanja kako bi se osigurala što veća podudarnost sa stvarnim ispitima koji se polažu na fakultetu medicine. Kao izvor koristile su se knjige koje su trenutno dio literature Medicinskog fakulteta, a to su: *„Anatomija, Duale Reihe“[1], „Atlas of Anatomy, Latin Nomenclature“[2]* i *„Pocket Atlas of Human Anatomy“[3].*

U narednim poglavljima ovog rada, detaljno ćemo analizirati postojeće obrazovne aplikacije i aplikacije s kvizovima. Također će se prikazati koncepti, korištene tehnologije i alati koji su potrebni za izradu ovog rada.

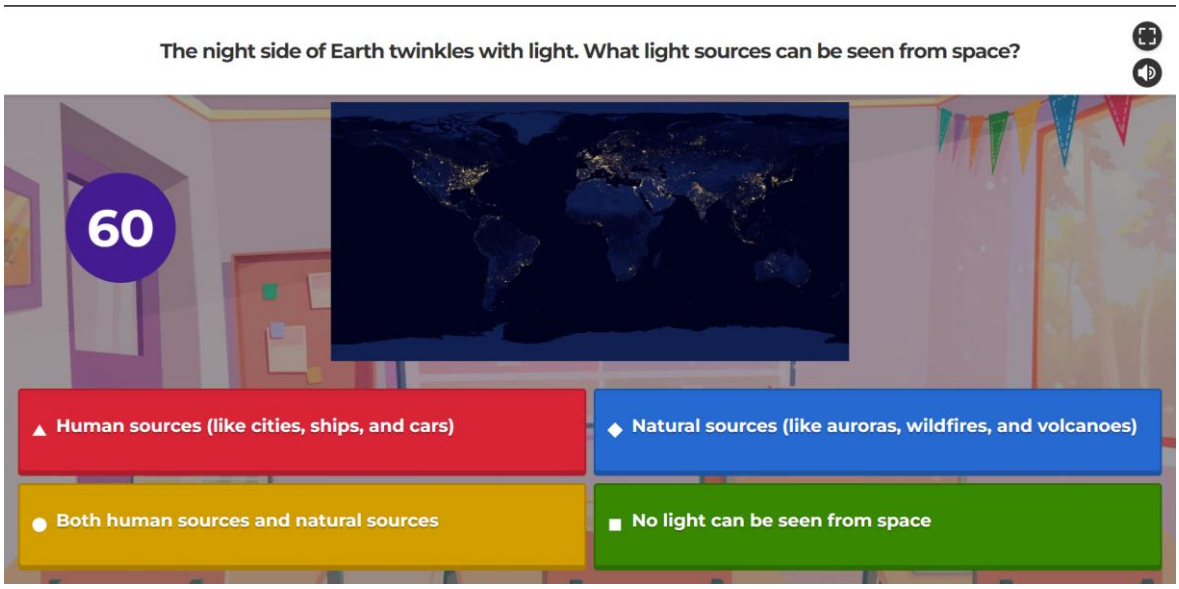
1. Slične aplikacije

1.1. Kahoot

Kahoot![4] je popularna obrazovna aplikacija koja je jedna od glavnih uzora za ovaj završni rad zbog primjene u obrazovanju još od osnovne škole i prikazuje kvizove kroz zanimljiviji način od rješavanja testova na papiru. *Kahoot!* omogućava igru uživo s više igrača, sudionici se pridružuju putem jedinstvenog broja koristeći mobitele, tablete ili računala. Rezultati se prikazuju odmah nakon svakog odgovorenog pitanja. Svako pitanje ima određeni vremenski period nakon kojeg se prikazuje odgovor i ažurira ljestvica igrača. Korisničko sučelje je intuitivno i jednostavno za upotrebu te je korišten veliki niz boja kako bi bio primjereniji za djecu kao i za odrasle.



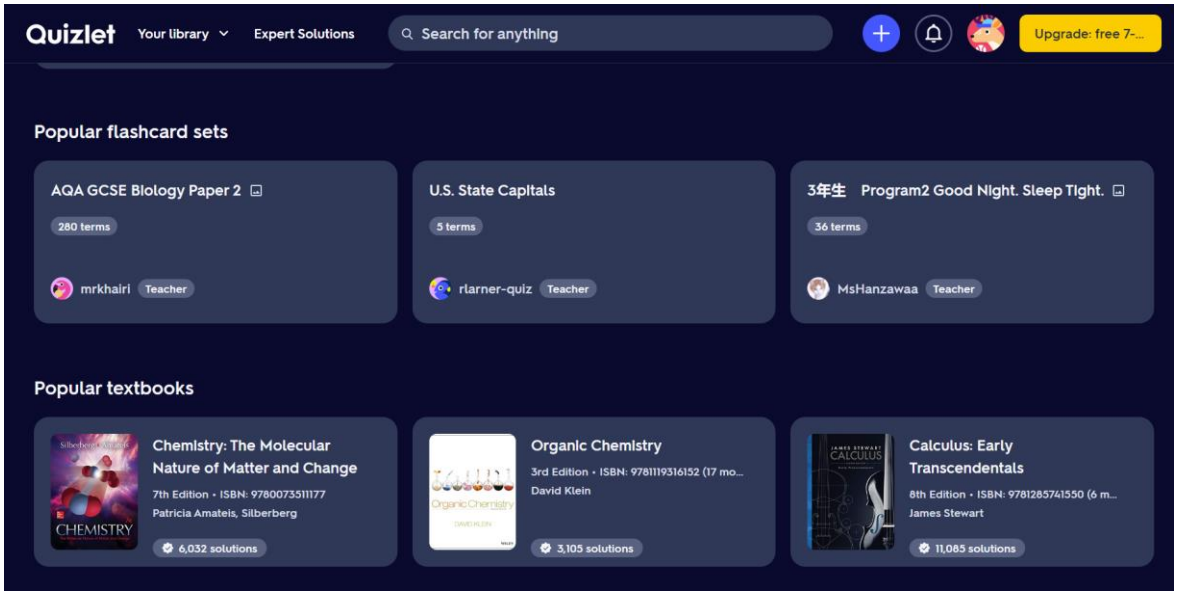
Slika 1.1 Kahoot! početna stranica



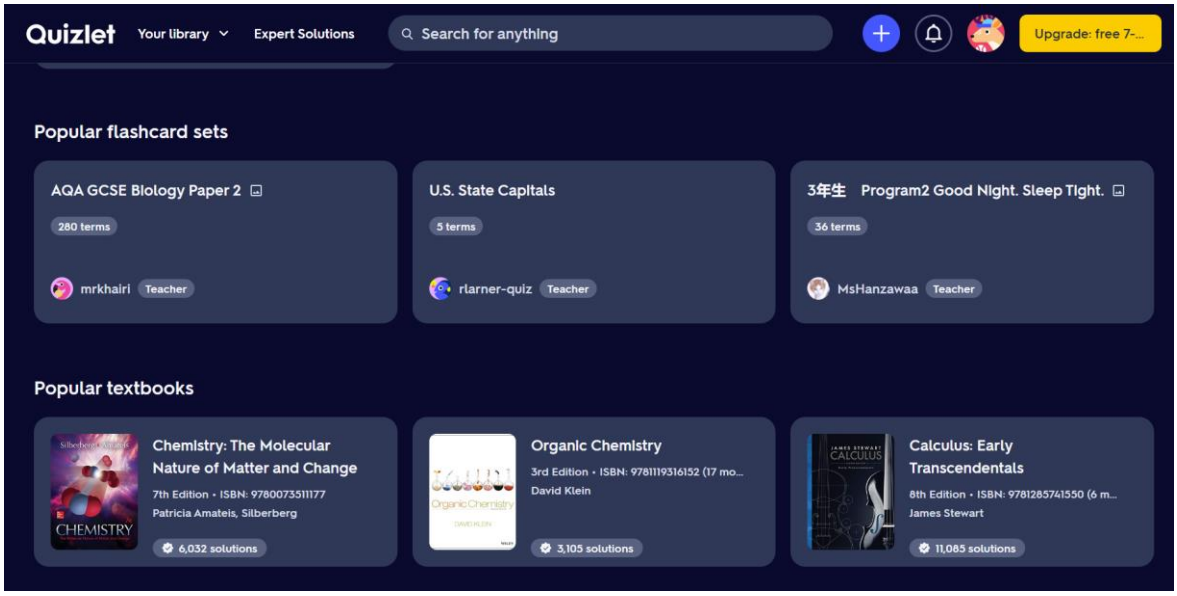
Slika 1.2 Primjer Kahoot! kviza

1.2. Quizlet Live

Quizlet Live[5] također je interaktivna obrazovna aplikacija kao i *Kahoot!*. Aplikacija omogućava korisnicima sudjelovanje u timskim kvizovima i igrama. Temeljni dio aplikacije su *flashcard*-ovi. Aplikacija se koristi za ponavljanje već naučenog gradiva koristeći već spomenute *flashcard*-ove. Jedna od prednosti je mogućnost kreacije pojedinog seta pitanja i odgovora za pojedini dio gradiva te dijeljenje tog kviza s kolegama. Ovime se *Quizlet Live* ne treba baviti proizvodnjom velike baze pitanja kako bi privukli korisnike, već svaki korisnik set pitanja i odgovora personalizira za svoje potrebe.



Slika 1.3 Quizlet početna stranica

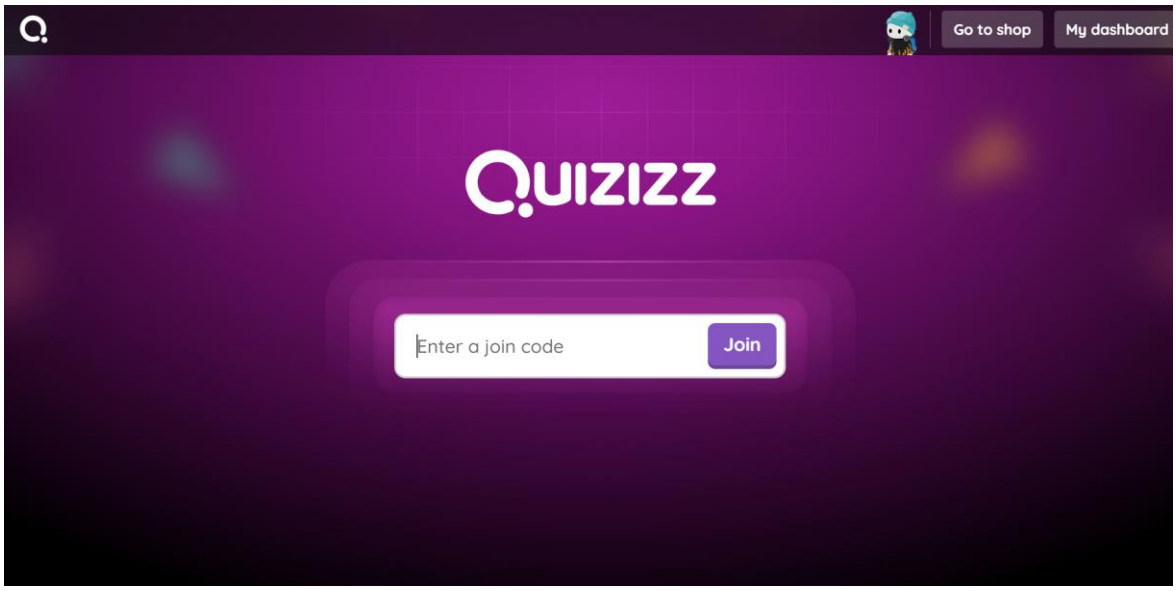


Slika 1.4 Primjer Quizlet kviza

1.3. Quizizz

Quizizz[6] obrazovna je platforma koja omogućava korisnicima kreaciju, dijeljenje i sudjelovanje u kvizovima u obliku igre. Aplikacije je napravljena kako bi privukla mlađe korisnike koristeći igrifikaciju obrazovnog materijala prilikom učenja. Naglasak je na samostalnom učenju i timskim aktivnostima. Korisnici jednostavno kreiraju kvizove s raznovrsnim tipovima pitanja, uključujući višestruki izbor, točno/netočno te otvorena pitanja. Također jedna od mogućnosti je

korištenje slika i videozapisa pri kreaciji pitanja kako bi se povećala interaktivnost. Korisnici se pridružuju putem jedinstvenog koda koristeći svoje uređaje. Aplikacije, isto kao i prijašnje, koristi sustav liste igrača i njihovih bodova osvojenih prilikom rješavanja kvizova.

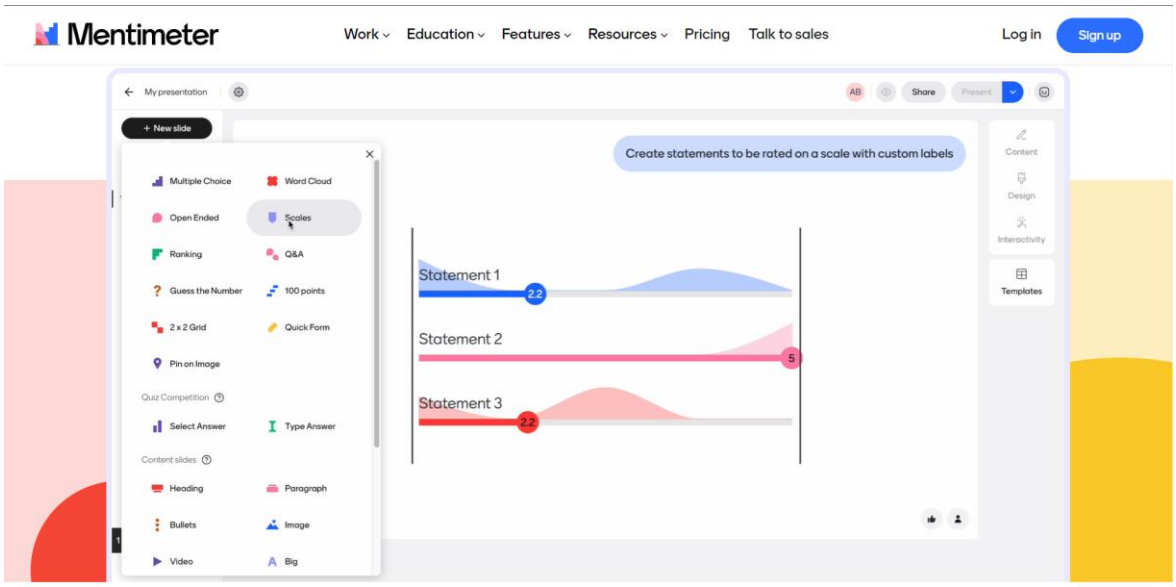


Slika 1.5 Pridruživanje Quizizz kvizu

1.4. Mentimeter

Mentimeter[7] je interaktivna aplikacija za stvaranje prezentacija i anketa u stvarnom vremenu, ali i kvizova, pitanjima i odgovorima i drugim oblicima interaktivnog sadržaja. Aplikacija se koristi i na Fakultetu elektrotehnike i računarstva kao sustav za sudjelovanje u kvizovima radi svoje mogućnosti za bodovanje pojedinog igrača na temelju točnosti i brzine te se ti podaci ažuriraju u stvarnom vremenu. Voditelj kviza ima uvid u ove podatke te ih može projicirati u dvorani u kojoj se rješava kviz. Jedna od prednosti ove aplikacije je integracija s prezentacijama, to jest aplikacija se može ukomponirati kao dio prezentacije te se time može koristiti kao lagani način testiranja gradiva koje se je toga dana predavalo.

Slika 1.6 Poredak igrača po bodovima u Mentimetru



Slika 1.7 Mentimeter kreacija kviza

2. Ideja rada i opis dijelova sustava

Ideja ovog završnog rada bila je razviti sustav sličan ranije navedenim aplikacijama koje bi prvenstveno služilo za ponavljanje gradiva anatomije studentima Medicinskog fakulteta, također omogućiti jednostavniji način prikaza znanja iz anatomije budućim studentima kako bi se prikazao dio gradiva koji se podučava i samim time djelomično olakšali izbor prilikom izbora fakulteta.

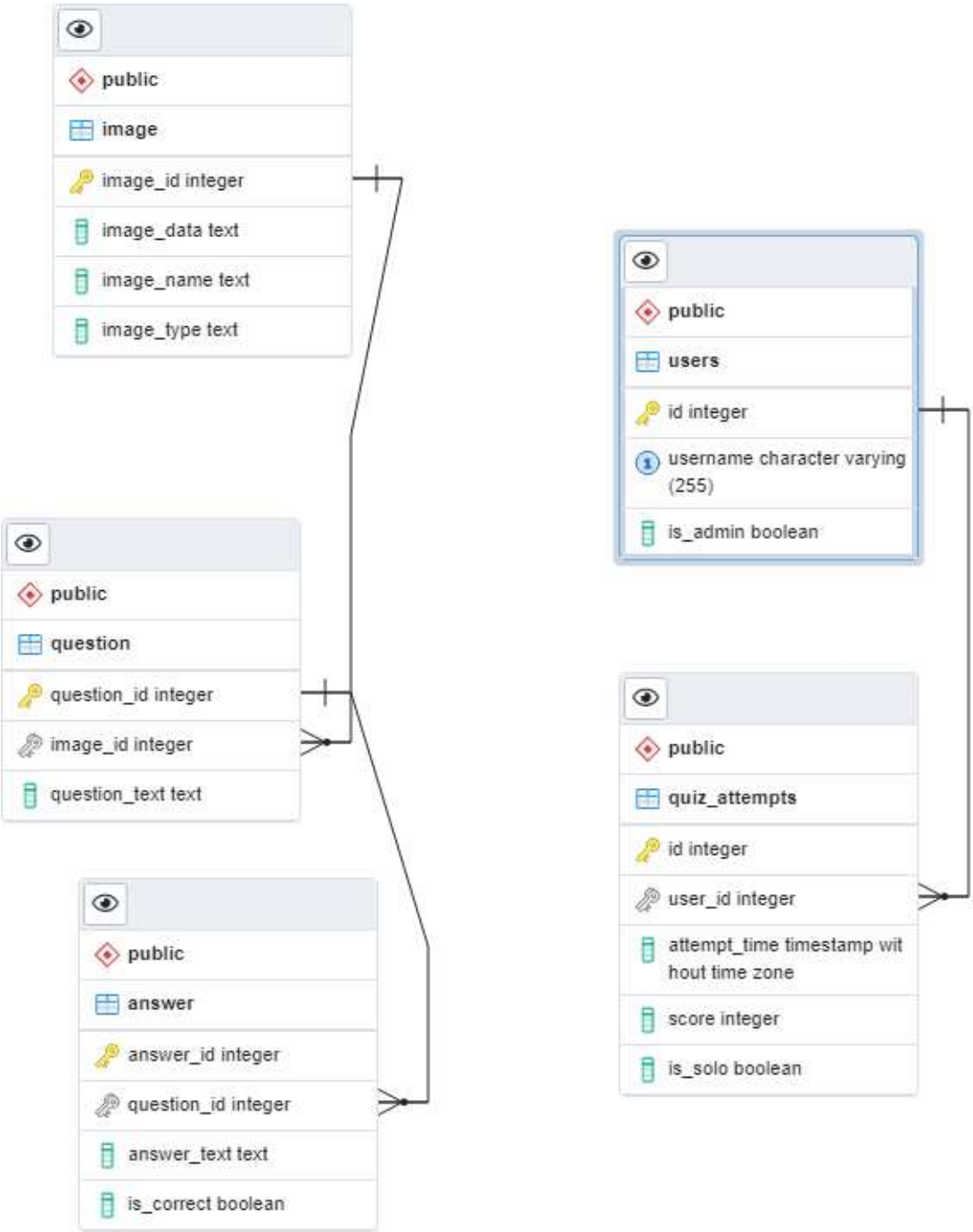
Aplikaciji je moguće pristupiti putem mobitela, tableta i računala putem interneta te prilikom pristupanju aplikaciji prikazan je izbornik koji nudi opcije za pokretanje kviza koje korisnik samostalno rješava i opcije za kreaciju „sobe“ ili pristupanje već kreiranoj „sobi“.

Pokretanjem samostalnog rješavanja kviza korisniku se prikazuju pitanja iz baze pitanja koja je popunjena pitanjima i odgovorima po uzoru na vrste pitanja koja se pojavljuju na ispitima kolegija „*Anatomija*“ na Medicinskom fakultetu Sveučilišta u Zagrebu. Korisnik klikom odabire odgovor te taj odgovor poprimi zelenu ili crvenu boju, ovisno o tome je li točan ili ne. Baza ima 150 pitanja podjednako podijeljenih po dijelovima gradiva kako se obrađuju na već spomenutom kolegiju, to jest 50 pitanja za A1 (opća anatomija, anatomija leđa i udova), 50 pitanja za A2 (anatomija trupa) i 50 pitanja za A3 (anatomija glave i vrata).

Kreiranjem sobe omogućava se pristup drugim korisnicima za paralelno rješavanje istih pitanja, a pristupanjem sobi prikazuje se forma za upis jedinstvenog koda. Svi korisnici, to jest igrači, koji se nalaze u istoj sobi paralelno odgovaraju na pitanja te ovisno o točnosti i brzini odgovora dodjeljuju im se bodovi koji se, prilikom završetka kviza, prikazuju svim igračima i rangiraju po bodovima.

3. Model podataka

3.1. Relacijski model podataka



Slika 3.1 Relacijski model podataka

3.2. Opisi entiteta

Image (slika)

- **imageId** – šifra slike (PK)
- imageData – tekstualni oblik slike
- imageName – naziv slike
- imageType – tip slike

Question (pitanje)

- **questionId** – šifra pitanja (PK)
- imageId – šifra slike na koju se odnosi pitanje (FK)
- questionText – tekst pitanja

Answer (odgovor)

- **answerId** – šifra odgovora (PK)
- questionId – šifra pitanja na koje se odnosi odgovor (FK)
- answerText – tekst odgovora
- isCorrect – zastavica za označavnje točnosti odgovora

User (korisnik)

- **id** – šifra korisnika (PK)
- username – korisničko ime
- isAdmin – zastavica za administratora

Quiz Attempts (pokušaji rješavanja kviza)

- **id** – šifra pokušaja (PK)
- userId – šifra korisnika na kojeg se odnosi pokušaj (FK)
- attemptTime – vrijeme rješavanja
- score – broj bodova
- isSolo – zastavica ovisno o vrsti kviza koji je rješavan

3.3. Opis relacijskog modela podataka

Baza podataka sastoji se od tri tablice: „image“, „question“ i „answer“. Svaka tablica ima svoj zaseban privatan ključ u obliku identifikacijskog broja koji se automatski generira prilikom novog unosa. Aplikacija je osmišljena na način da svako pojedino pitanje je vezano uz sliku koja je u tablici spremljena koristeći tekstualni prikaz slike (imageData), kao na primjer base64. ImageName služi za lakše snalaženje u bazi podataka, a imageType je potreban kako bi se slika u *front-end*-u pravilno prikazala u obliku slike, a ne u obliku teksta kako je spremljena u bazi. Svaki odgovor uz sebe ima vezan answerText koji sadrži tekst pojedinog odgovora i zastavicu isCorrect za označavanje točnih i netočnih odgovora.

4. Korištene tehnologije i alati

Razvijeni sustav sastoji se od tri glavna dijela:

- Baza podataka
- Poslužitelj – program koji dostavlja uslugu drugim programima koji su spojeni na njega preko komunikacijskog kanala
- Klijent – program koji pristupa poslužitelju tražeći uslugu

Model arhitekture korištene u ovom sustavu naziva se klijent-poslužitelj. Klijent, u ovom slučaju web-preglednik, pruža grafičko korisničko sučelje za korištenje aplikacije. Korisnici putem svog web-preglednika šalju i primaju podatke od poslužitelja, gdje je baza podataka također smještena. Njihova međusobna komunikacija odvija se putem HTTP protokola, koji određuje format i sadržaj poruka (zahtjeva i odgovora).

Komunikacija između klijenta i poslužitelja kada klijent pošalje zahtjev poslužitelju i čeka njegov odgovor. Poslužitelj odgovara s rezultatima koji ovise o vrsti zahtjeva, a svaki odgovor se sastoji od statusnog koda i odgovarajućeg resursa. Poslužitelj ne mora biti svjestan svakog klijenta, već samo odgovara na zahtjeve, neovisno o njihovom podrijetlu.

Kako bi se ostvarila komunikacija u stvarnom vremenu između klijenata koristi se *Stack.io*. Ova platforma omogućuje integraciju funkcionalnosti u stvarnom vremenu u aplikacije koristeći WebSocket protokol za brzu razmjenu podataka između više klijenata koji, u određenim uvjetima, ovise jedni o drugima.

4.1. PostgreSQL

U ovom projektu, za upravljanje bazom podataka koristi se *PostgreSQL* sustav za relacijske baze podataka. *PostgreSQL* je poznat po robusnosti, fleksibilnosti i usklađenosti s ACID svojstvima, što osigurava integritet podataka čak i u slučaju sistemskih kvarova i pada sustava. Sustav također podržava, osim osnovnih tipova podataka, napredne tipove podataka poput JSON, XML i drugih,

što ga čini fleksibilnim pri implementaciji baze podataka. Sustav je napisan u programskom jeziku *C* i jedan je od najpopularnijih sustava za baze podataka koji se koriste u sadašnjem svijetu[8].

4.2. React.js

Za *front-end* dio sustava koristi se *JavaScript* biblioteka *React*. *React* je razvijen od strane *Facebook*-a za izgradnju brzih i interaktivnih *SPA* korisničkih sučelja, UI, za web i mobilne aplikacije. Temelji se na integraciji komponenti, gdje se aplikacija gradi od manjih, samostalnih komponenti koje se mogu koristiti u svakom trenutku pri razvoju web stranice. Ovime se omogućava ponovna upotreba i modularnost koda. Osnovni dio komponente su *HTML* elementi koji služe kao osnovni izgled web stranice. Korištenjem *CSS*-a elementima se pridružuju parametri kojima se upravlja kompletan izgled stranice. Svaka komponenta može imati svoje stanje i svojstva kojima se komponenta generira i time može promijeniti izgled, to jest promijeniti određene elemente, stranice. Svaka komponenta pisana je korištenjem *JSX* sintakse, iako se može koristiti i osnovni *JavaScript*, koja spaja *XML* i *JavaScript* kao cjelinu. Tijekom kompletnog razvoja *front-end* dijela aplikacije korišten je uređivač teksta *Visual Studio Code*.

Kompletna srž *React* aplikacije nalazi se u dokumentu *main.jsx* koji kreira jedinstveni identifikator *root* i pokreće cijeli ostatak aplikacije.

```
ReactDOM.createRoot(document.getElementById('root')).render(
  <BrowserRouter>
    <App />
  </BrowserRouter>
)
```

Ovime se poziva *BrowserRouter* kojim se s lakoćom pozivaju zahtjevi s *back-enda*, ovisno o parametrima, ali i druge komponente koje su potrebne u različitim vremenima prikazivanja. Unutar *BrowserRoutera* poziva se *App*, to jest komponenta koja se nalazi u dokumentu nazvanom *App.jsx[9]*.

```
function App() {

  return(
    <Routes>
      <Route path="/" element={<Home/>}></Route>
      <Route path="/quiz" element={<Question/>}></Route>
      <Route path="/create-room" element={<CreateRoom/>}></Route>
      <Route path="/join-room" element={<JoinRoom/>}></Route>
      <Route path="/waiting-room/:roomId/:playerName"
element={<WaitingRoom/>}></Route>
      <Route path="/start-room/:roomId/:playerName"
element={<StartRoom/>}></Route>
      <Route path="/quiz-room/:roomId" element={<QuizRoom/>}></Route>
      <Route path="/scoreboard/:roomId"
element={<Scoreboard></Scoreboard>}></Route>
    </Routes>
  )

}
```

Kôd 4.1 – *App.jsx* komponenta aplikacije

Ovom funkcijom podešavaju se pozivi svake komponente ovisno o pozivu pojedine putanje. U putanjama se također koriste i parametri poput *roomId* i *playerId* koji se mogu dalje koristiti pri personaliziranju stranice za pojedinog korisnika, ali se također mogu koristiti i pri pozivu zahtjeva potrebnih pri pozivu *servera* i baze podataka.

4.2.1. React Router DOM

Za već spomenuto upravljanje navigacijom koristi se *React Router DOM*, kao što se vidi u kôdu (Kôd 4.1). *React Router DOM* popularna je biblioteka koja služi za upravljanje rutama u React aplikacijama te omogućava izgradnju jednostraničnih aplikacija s višestrukim prikazima. Biblioteka je posebno dizajnirana za React aplikacije, što osigurava jednostavnu integraciju i optimalne performanse[10].

```
import { Route, Routes } from "react-router-dom"
```

4.2.2. axios

React aplikacija sama po sebi služi za kreaciju i prikaz komponenata i njihovim upravljanjem te je potrebna *axios* biblioteka kako bi se omogućilo slanje i primanje HTTP zahtjeva.

```
try {
            const response = await
http.get(`/api/questions/${index}`)
            console.log(response.data)
            if (response.data) {
               const data = await response.data
               setQuestionDetails(data)
            } else {
               throw new Error('Failed to fetch question
details')
            }
         } catch (error) {
            console.log(error)
         }
```

Kôd 4.2 – Slanje HTTP get zahtjeva

U ovom odsječku kôda vidi se primjer korištenja axios biblioteke kako bi se ostvarila komunikacija *front-end*-a i *back-end*-a. Varijabla *http* sadrži baseURL koji označava originalnu putanju na temelju koje se odvija daljnja komunikacija, a .get metoda označava da se radi o get HTTP zahtjevu. U odgovoru aplikacija dobije tražene elemente ili pogrešku u slučaju lošeg zahtjeva.

4.3. Spring Boot

Za back-end dio sustava koristi se Spring Boot, radni okvir otvorenog koda temeljen na Java programskom jeziku, a razvijen od strane Vmware-a. Spring Boot olakšava izgradnju aplikacija zasnovanih na Spring Framework-u, s minimalnom potrebom za konfiguraciju. Automatska konfiguracija izvodi se pomoću dodanih biblioteka dodanih u dependecies te se odmah mogu početi koristiti prilikom pisanja kôda. Spring Initializerom jednostavno se kreiraju početne konfiguracije poput naziva projekta, izbora jezika i početnih dependecy-ja. Spring

Boot aplikacija omogućuje klijentima da ostvare komunikaciju s poslužiteljem i izvode potrebne operacije.

```
public class Question {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long questionId;

    @JoinColumn(name = "image_id", nullable = false)
    private Long imageId;

    @Column(nullable = false)
    private String questionText;

}
```

Kôd 4.3 – Model za tablicu Question

Modeli u Spring Boot aplikacijama prikazuju tablice u bazi podataka i dodavanjem anotacija mogu se dodatno konfigurirati podaci vezani uz pojedini model. Anotacija `@Id` označava da je ta varijabla zaslužna za identitet u tablici, dok anotacija `@Column(nullable = false)` označava da taj stupac mora sadržavati vrijednost, to jest ne smije biti prazan.

```
public class QuestionService {
    @Autowired
    private QuestionRepository questionRepository;

    public Question saveQuestion(Question question) {
        return questionRepository.save(question);
    }

    public Question getQuestionById(Long questionId) { return
questionRepository.findQuestionByQuestionId(questionId); }

    public long getTotalQuestions() { return
questionRepository.countQuestions(); }
}
```

Kôd 4.4 – Service za tablicu Question

Service sloj služi za generiranje metoda koje će se dalje koristiti u aplikaciji. Ovdje su definirane sve metode koje se koriste te se ovime izbjegava višestruko pisanje istog kôda.

```
public ResponseEntity<?> getQuestionDetails(@PathVariable
Long questionId) {
        Question question =
questionService.getQuestionById(questionId);
        if (question == null) {
            return new ResponseEntity<>("Question not
found.", HttpStatus.NOT_FOUND);
        }

        Image image =
imageService.getImageById(question.getImageId());
        if (image == null) {
            return new ResponseEntity<>("Image not found.",
HttpStatus.NOT_FOUND);
        }

        List<Answer> answers =
answerService.getAnswersByQuestionId(questionId);

        QuestionDetailsDTO questionDetails = new
QuestionDetailsDTO(question, image, answers);

        return new ResponseEntity<>(questionDetails,
HttpStatus.OK);
    }
```

Kôd 4.5 – Controller tablice Question

Konačno aplikacija sadrži controllere koji služe za upravljanje funkcijama koje su potrebne za rad web stranice. Ovaj dio kôda primjer je controllera koji služi za dohvat svih potrebnih detalja za svako pitanje[11].

4.4. Socket.io

Za komunikaciju u stvarnom vremenu između klijenata koristi se *Socket.io*, popularna biblioteka koja omogućava WebSocket komunikaciju. Biblioteka je napisana u jeziku *JavaScript*, a razvijena je od strane Automattic-a. Sastoji se od

dvije komponente: klijenta i servera. Ovaj protokol osigurava komunikaciju između više klijenata u slučaju kada je to potrebno, kao što je slučaj kada više korisnika rješava isti kviz.

```
socket.on('joinRoom', (roomId, playerName) => {
      socket.join(roomId);
      console.log(`User ${playerName} joined room
${roomId}`);

      if (!rooms[roomId]) {
            rooms[roomId] = [];
      }
      rooms[roomId].push(playerName);

      console.log('Update players.')
      io.to(roomId).emit('updatePlayers', rooms[roomId]);
      console.log(rooms[roomId])
   });
```

Kôd 4.6 – server strana protokola

Ovaj kôd primjer je kako na strani servera funkcionira usklađivanje više korisnika. Server korisnike spaja na jedan socket i zatim prilikom slanja zahtjeva tom socketu taj zahtjev dobiju svi korisnici koji se nalaze unutar istog socketa. S klijentske strane na ove zahtjeve odrade se funkcije ovisno o vrsti zahtjeva koji je poslan[12].

4.5. GitHub

Za upravljanje verzijama i dijeljenje kôda korišten je GitHub, platforma za pružanje programske potpore Git repozitorija. GitHub prati razlike u kôdu napravljene prilikom svake nadoknade i izmjene. Također pruža usluge poslužitelja i time omogućava pristup kôdu s bilo kojeg mjesta. GitHub su razvili Chris Wanstrath, Tom Preston-Werner, Scott Chacon i P. J. Hyett[13].

4.6. Heroku

Za hosting i implementaciju aplikacije koristi se Heroku, platforma koja omogućava programerima laganu implementaciju, upravljanje i skaliranje

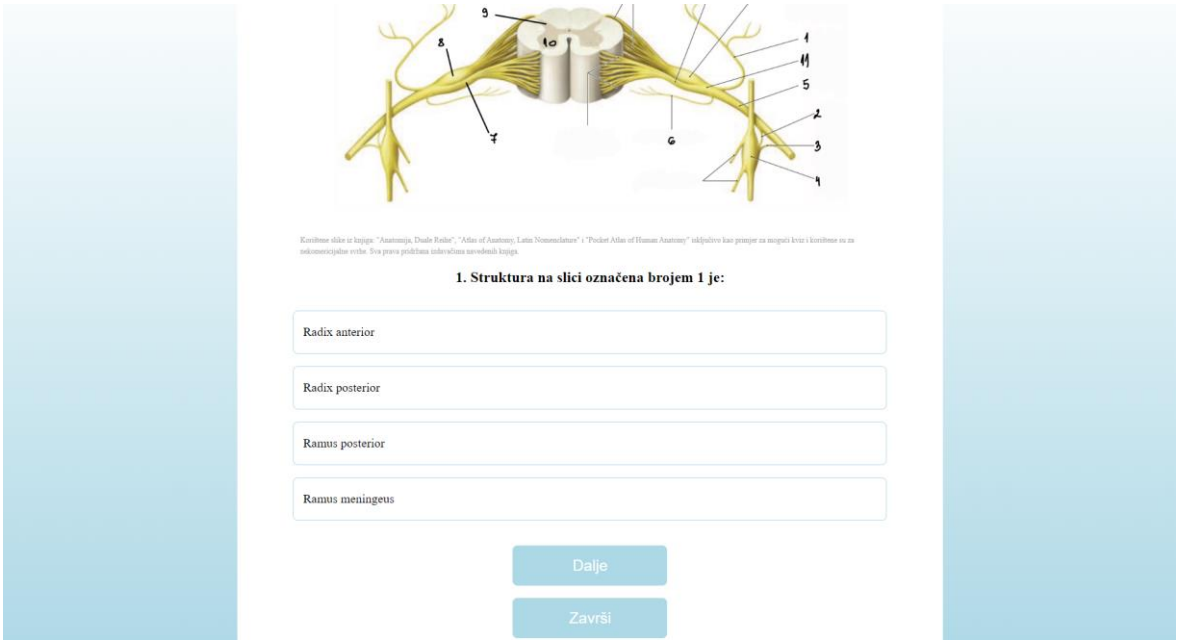
aplikacije. Platforma podržava različite programske jezike i pruža integrirane alate koji se koriste za testiranje, razvoj i implementaciju aplikacije[14].

5. Korištenje aplikacije

Aplikacije se dijeli na dva dijela, samostalno rješavanje i grupno rješavanje kviza. Izborom tipke „Započni“ pokreče se samostalno rješavanje kviza. Kviz ima 150 pitanja kako bi se pokrilo znanje potrebno za predmet „Anatomija“.

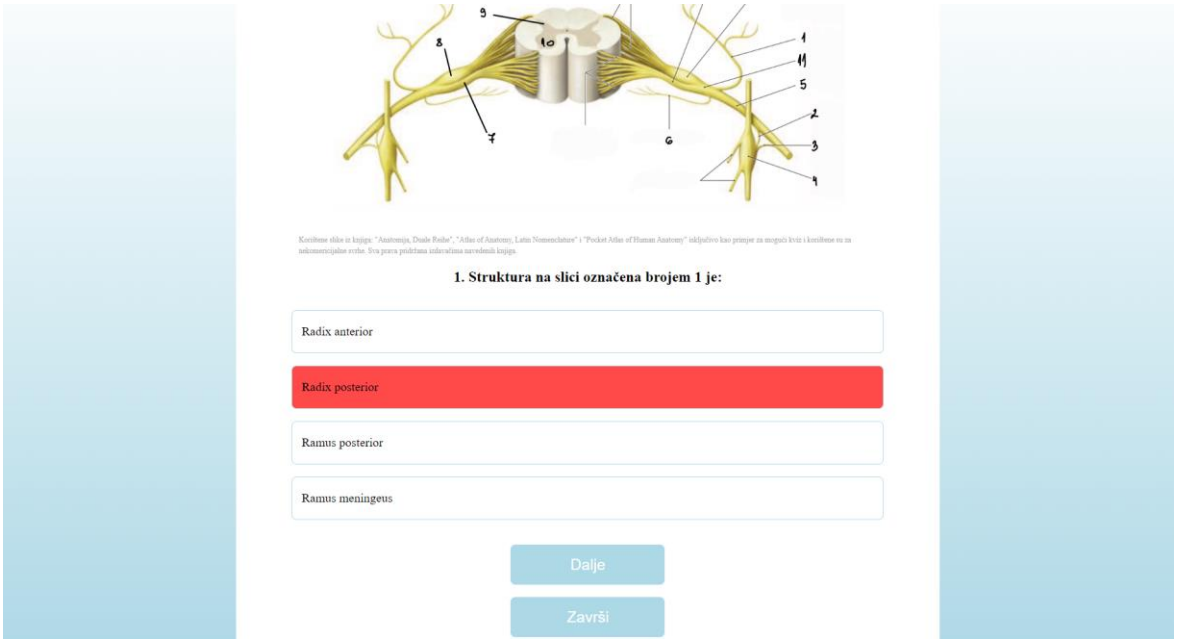
Slika 5.1 Početna stranica aplikacije

Svako pitanje u kvizu oblikovano je u obliku slike, pitanja i ponuđenih odgovora. Ovaj sistem pitanja prevladava na ispitima koji se polažu na Medicinskom fakultetu Sveučilišta u Zagrebu te se time pokušava što više preličiti način rješavanja kako bi se postigla bolja priprema za stvarni ispit.



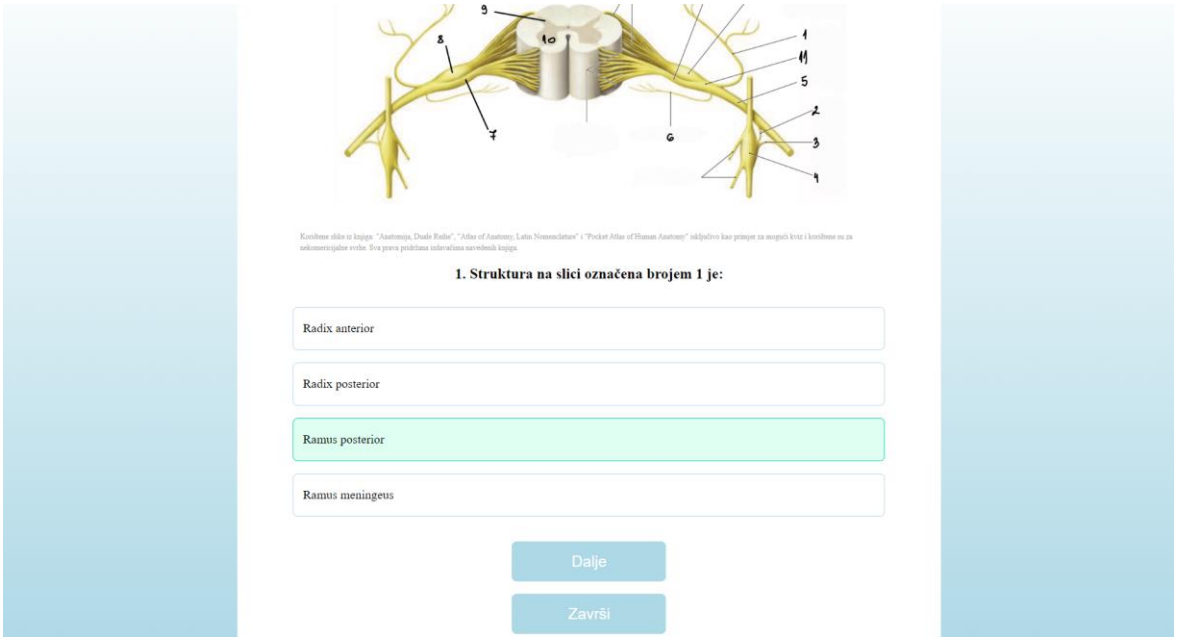
Slika 5.2 Primjer pitanja

Odabirom na pojedini odgovor isključuje se mogućnost ponovnog odabira i, ovisno o točnosti odgovora, odgovor poprimi zelenu, za točan, ili crvenu boju, za netočan odgovor. Slika segmenta kralježnice korištena kao primjer iz knjige [1].



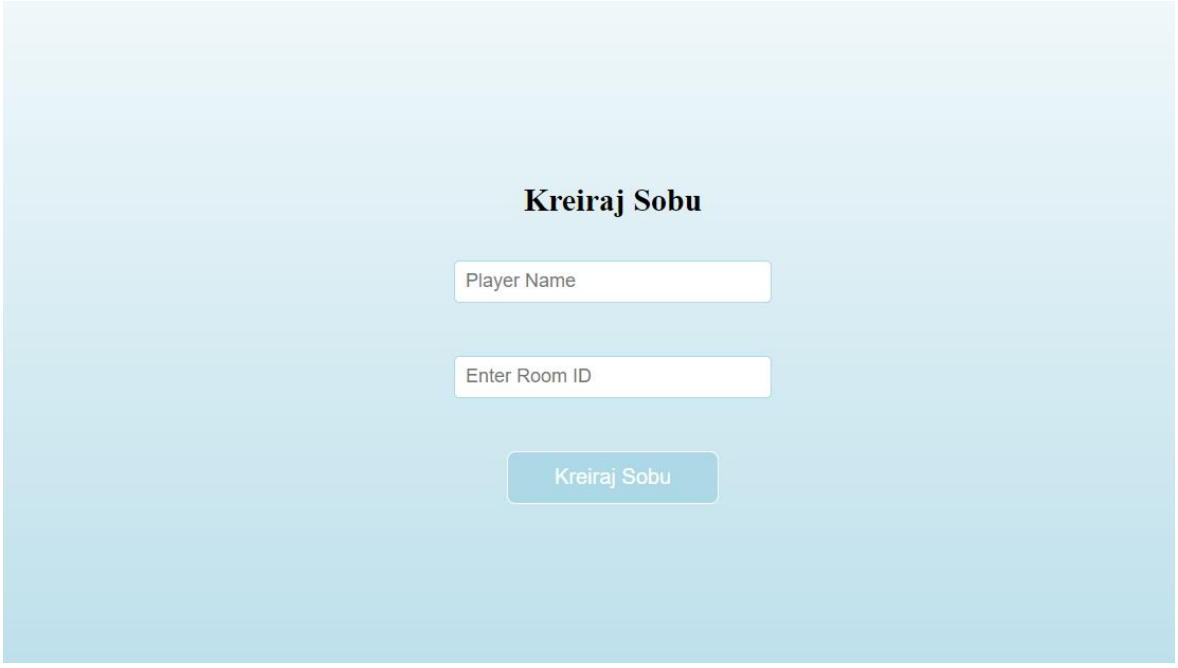
Slika 5.3 Primjer netočnog odgovora

Slika segmenta kralježnice korištena kao primjer iz knjige [1].



Slika 5.4 Primjer točnog odgovora

Odgovorom na zadnje pitanje kviz završava i vraća se na početnu stranicu. Korisnik može također, i pokrenuti zajednički kviz s kolegama odabirom „Kreiraj sobu“ što ga prebaci na ekran gdje je potrebno unijeti ime igrača i broj sobe. Slika segmenta kralježnice korištena kao primjer iz knjige [1].



Slika 5.5 Stranica za kreaciju sobe

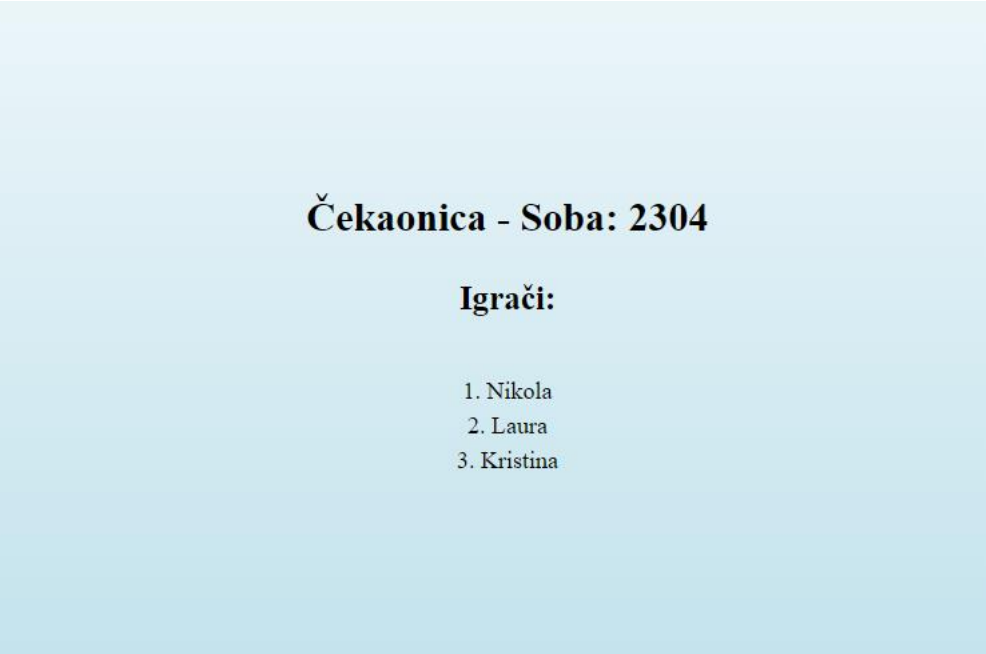
Nakon unosa potrebnih podataka odabirom tipke „Kreiraj Sobu“ korisnika se prebacuje na stranicu „Čekaonice“ dok se pridružuju ostali korisnici.

Slika 5.6 Čekaonica s jednim igračem

Nakon pridruživanja drugih igrača lista igrača se odmah osvježava i prikazuje imena ostalih igrača.

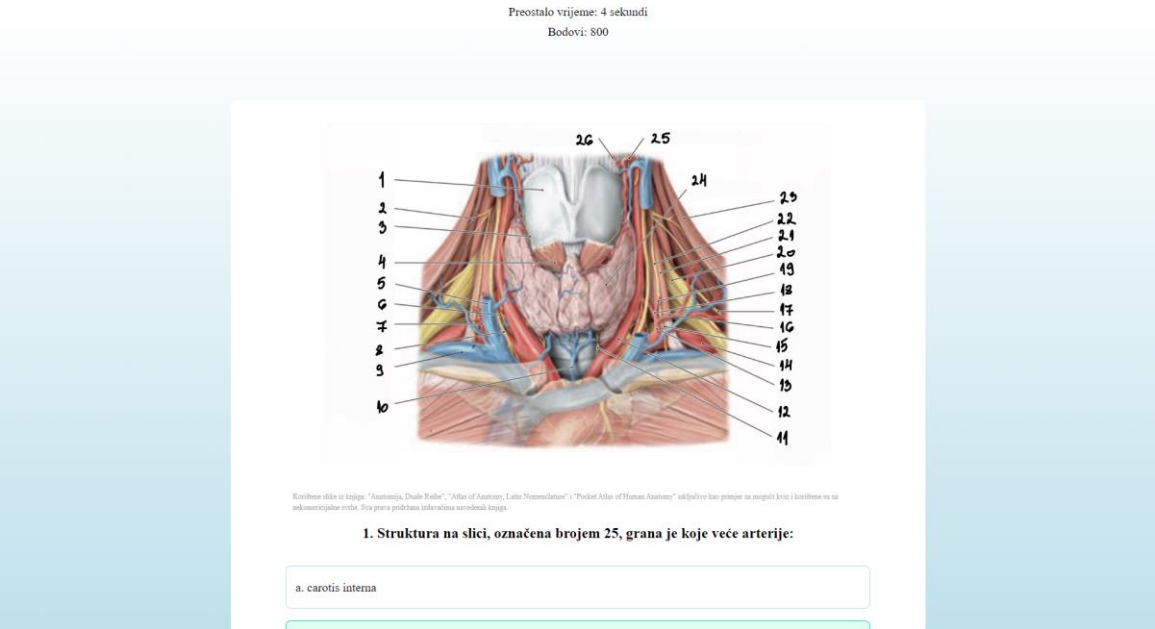
Slika 5.7 Čekaonica s više igrača

Samo korisnik koji je kreirao sobu može pokrenuti kviz odabirom na „Započni kviz“, ostali korisnici nemaju tu mogućnost.



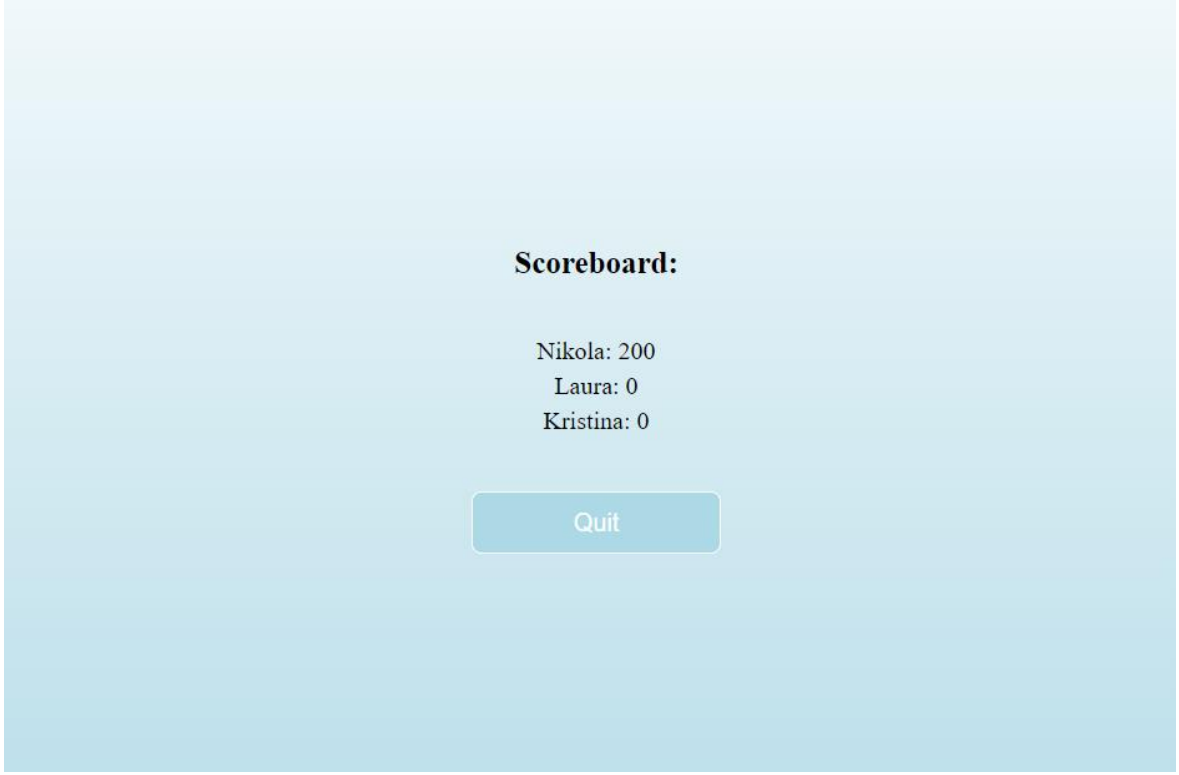
Slika 5.8 Čekaonica igrača koji se pridružio

Svi korisnici koji se nalaze u istoj sobi u isto vrijeme rješavaju kviz te ovisno o točnosti odgovora dobiju određeni broj bodova. Za vrijeme rješavanja kviza svakom igraču je prikazano preostalo vrijeme i broj bodova koje trenutno imaju. Po isteku vremena prikaže se točan odgovor i isključi se opcija ponovnog odabira odgovora. Na kraju kviza lista svih igrača i njihovih bodova se prikazuje svima.



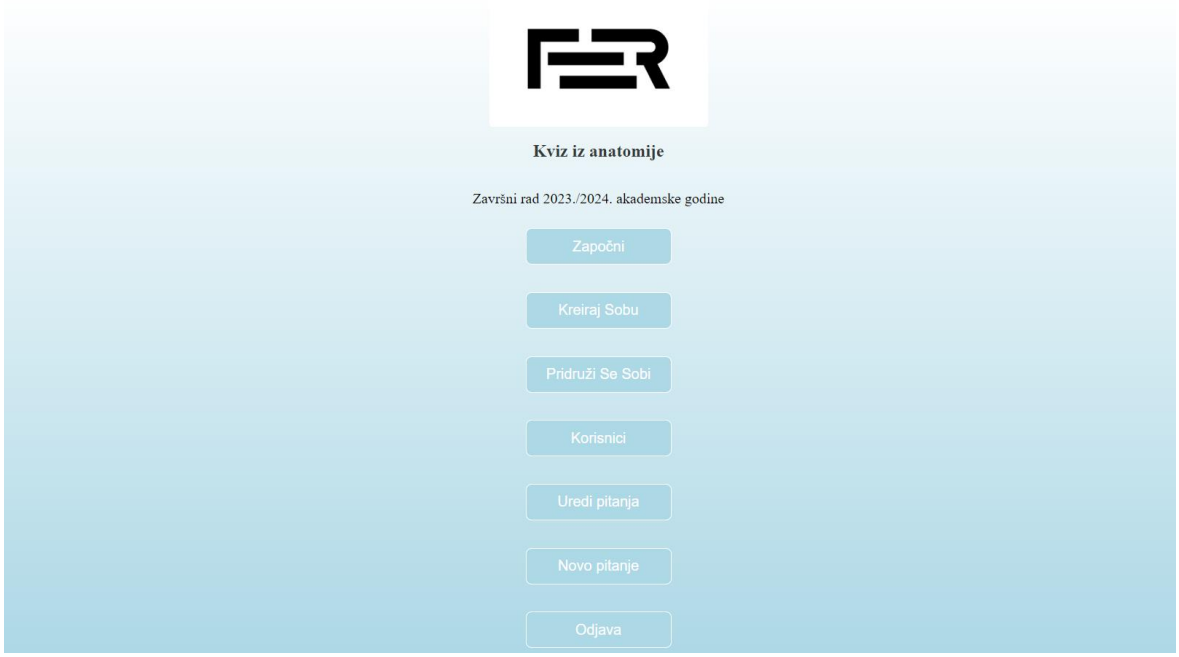
Slika 5.9 Prikaz preostalog vremena i bodova

Slika strukture korištena kao primjer iz knjige [1].

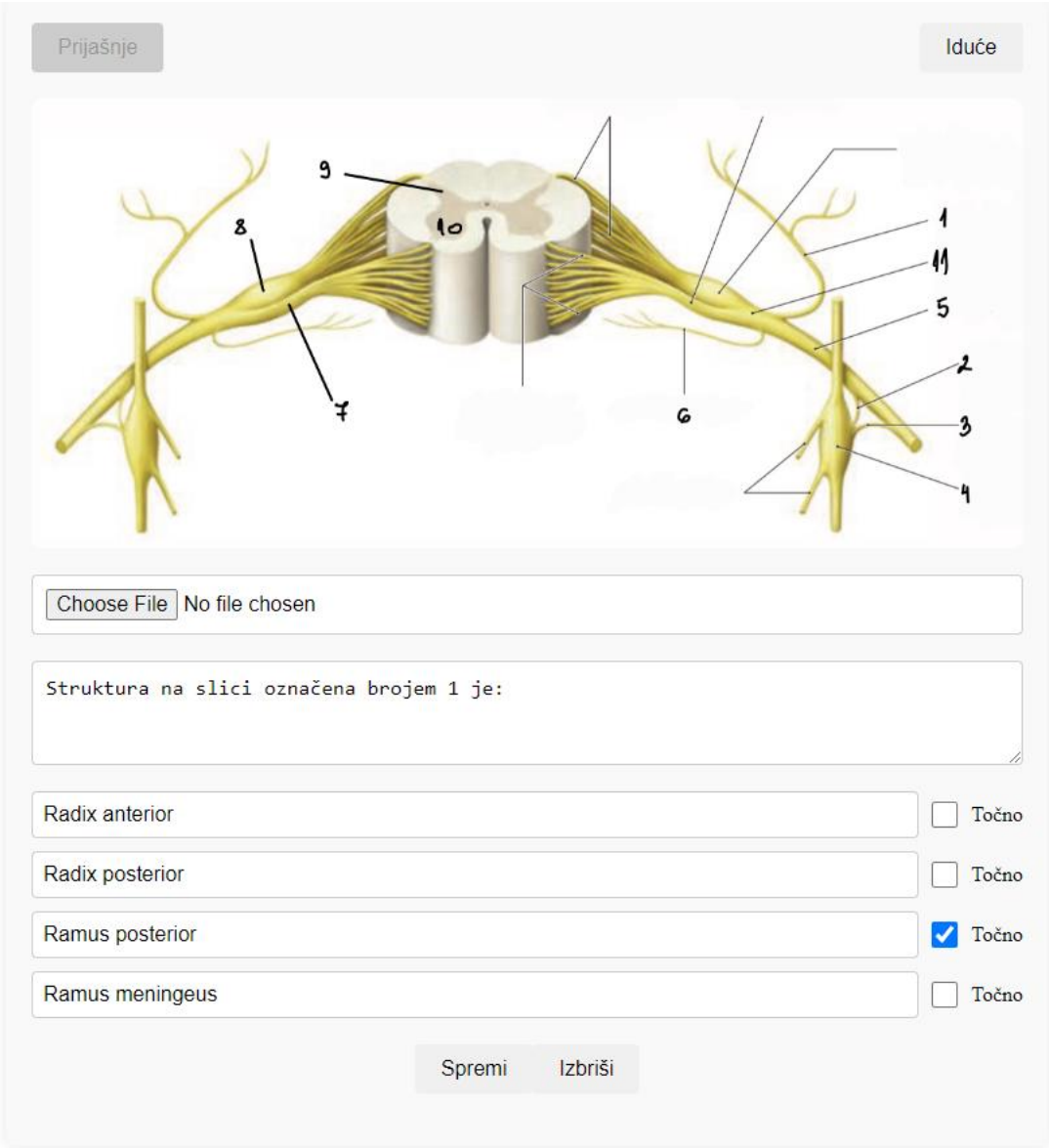


Slika 5.10 Prikaz liste igrača i njihovih bodova

Kada se prijavi administrator prikažu se dodatne opcije, opcija za uređivanje pitanja i odgovora, opcija za dodavanje novih pitanja te opcija za pregled svih korisnika i prikaz njihovih pokušaja.

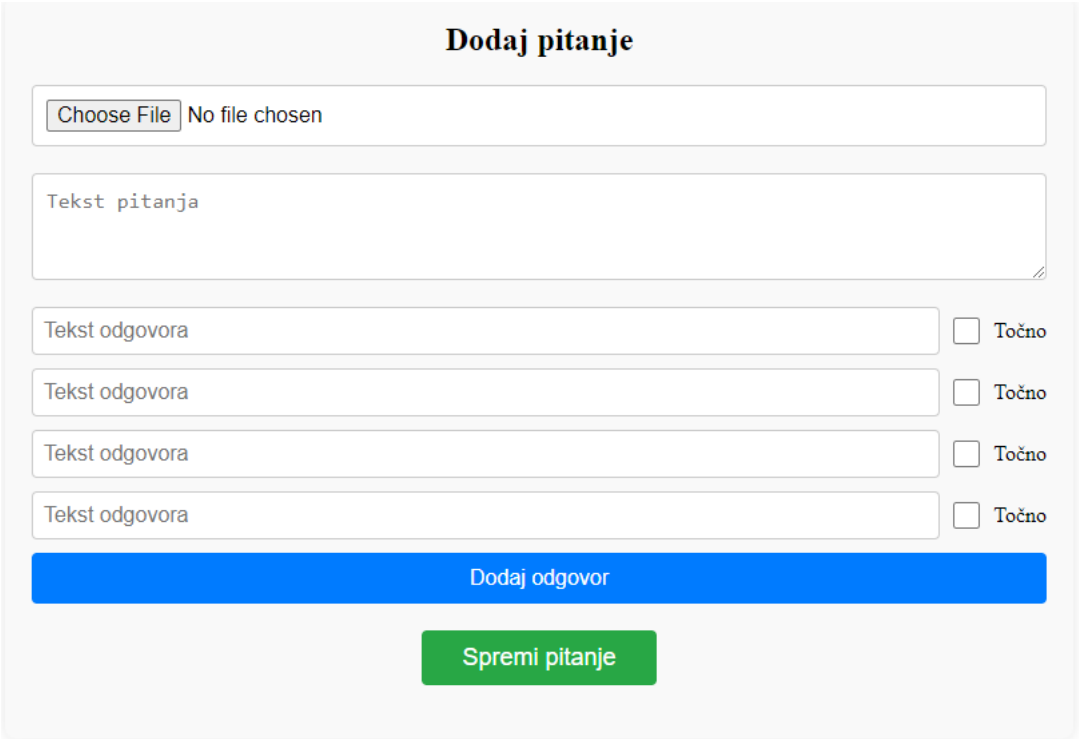


Slika 5.11 Administratorsko sučelje



Slika 5.12 Uređivanje pitanja

Administrator može promijeniti sliku, pitanje, ponuđene odgovore i zastavicu ovinso o tome koji odgovor je točan.



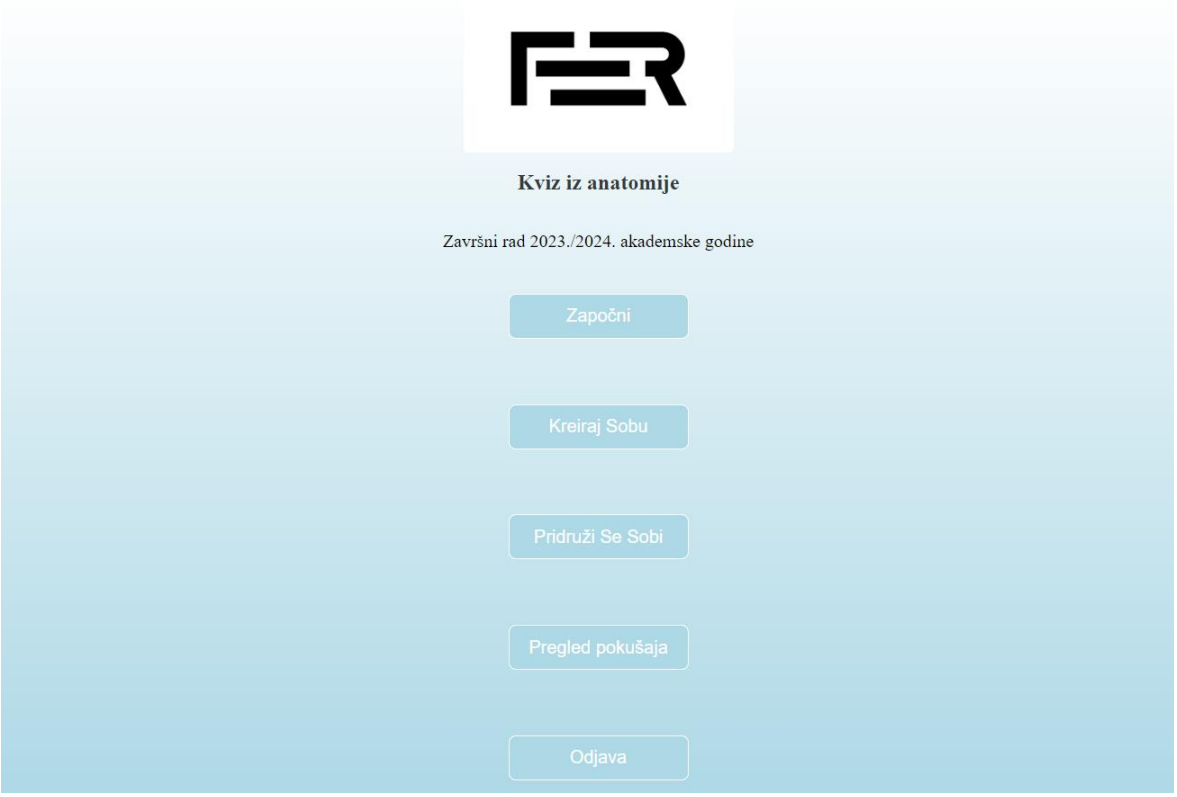
Slika 5.13 Dodavanje pitanja

Administrator pri dodavanju pitanja mora odabrati sliku i ispuniti tekst pitanja i odgovora koji će biti ponuđeni te označiti koji odgovor je točan.

Slika 5.14 Lista korisnika

Slika 5.15 Korisnikovi pokušaji

Administrator ima opciju da se prikažu svi korisnici kao što je vidljivo na sllici 5.14 te odabirom pojedinog korisnika ima pristup svim pokušajima rješavanja kviza koje je korisnik pokrenuo. Ako je korisnik samostalno rješavao kviz bodovi su broj pitanja na koje je korisnik točno odgovorio, a u slučaju da je korisnik rješavao kviz s kolegama bodovi su postignuti bodovi za vrijeme kviza.



Slika 5.16 Korisnikovo sučelje

Ako se prijavi običan korisnik dodaje se opcija za pregled vlastitih pokušaja koja izgleda isto kao i kod administratorskog sučelja, ali korisnik može pristupiti samo vlastitim pokušajevima.

6. Daljnji razvoj

Zbog ograničenog vremena pisanja rada postoje ideje za daljnji razvoj budućim verzijama. Trenutno aplikacija funkcionira na ekranima prijenosnih računala, no mijenjanjem veličine ekrana dolazi do nenamjernih promjena u izgledu aplikacije. Naravno uvijek je moguća nadogradnja baze podataka, to jest dodavanje dodatnih slika, pitanja i odgovora. Trenutno se u bazi nalaze 53 slike, 150 pitanja i 600 mogućih odgovora. Problem s dodavanjem pitanja je da sva pitanja treba proučiti kolega/ica s Medicinskog fakulteta kako bi sva pitanja ostala vjerodostojna onima koji se pojavljuju na ispitima. Slike se mogu poboljšati kvalitetom i samim time i njihov prikaz unutar aplikacije. Kako su i sada pitanja podjednako raspoređena da prate gradivo kolegija „Anatomija“ može se dodati verzija kviza koja bi bila određena po cjelini koja se obrađuje što je, kao već spomenuto i u uvodu, A1, A2 i A3. Također bi se mogla dodati opcija dodavanja kolega da se vide njihovi postoci riješenosti i opcija poziva sudjelovanja u kvizu kako bi se olakšalo grupno sudjelovanje.

7. Mišljenja studenata Medicinskog fakulteta

Aplikacija je prikazana maloj grupi studenata, njih šest, te su mišljenja uglavnom ista. Sviđa im se jednostavni izgled aplikacije, što je i bilo u cilju da se izgled kviza preslika na sadržaj, a to je anatomija ljudskog tijela i to se postiže donekle sterilnim, to jest jednostavnim izgledom aplikacije. Sretni su s opcijom zajedničkog rješavanja kviza i misle kako bi ih to moglo potaknuti i samostalno učenje kako bi mogli pobijediti svoje kolege prilikom idućeg rješavanja kviza. Svidjela im se ideja budućeg unaprjeđenja da se izdvoje određena pitanja iz kviza ovisno o tome koju cjelinu trenutno rade. Mišljenje svih ispitanika je da bi aplikacija s još dodatnim radom za unaprjeđenje bila odličan alat za učenje i ponavljanje gradiva.

Dodatno ispitivanje trebalo bi se provesti iduće godine kako bi studenti mogli koristiti aplikaciju za vrijeme učenja kako bi se provjerila uspješnost aplikacije kao alata za pomoć pri učenju.

Zaključak

U okviru ovog završnog rada razvijena je interaktivna aplikacija za učenje anatomije, koja je prvenstveno namijenjena studentima prve godine medicine, ali i svima zainteresiranima za ovu temu. Aplikacija omogućava ponavljanje i vježbanje gradiva kroz samostalno i grupno rješavanje kviza. Ugrađene su funkcionalnosti za kreaciju sobe, pridruživanje sobi i praćenje stanja bodova igrača.

Za potrebe implementacije, korištene su moderne tehnologije kao što su React.js za front-end, Spring Boot za back-end, PostgreSQL za upravljanje bazom podataka, te Socket.io za komunikaciju u stvarnom vremenu.

Testiranje bi se trebalo dodatno provesti kako bi se pratila uspješnost aplikacije kroz godinu.

U budućnosti, aplikacija se može dodatno unaprijediti kroz nekoliko mogućih proširenja. Plan je i dalje da se aplikacija testira među studentima medicine kako bi se procijenila uspješnost svih integriranih funkcionalnosti i algoritama. Povratne informacije od stvarnih korisnika omogućit će bolje razumijevanje korisničkog iskustva i pomoći u optimizaciji aplikacije za još bolje rezultate u učenju anatomije.

Testiranje i prikupljanje povratnih informacija omogućit će preciznije određivanje težine zadataka na temelju statistike rješavanja, što će dodatno unaprijediti prilagodljivost kvizova potrebama korisnika. Konačni cilj aplikacije je pružiti učinkovit alat za učenje anatomije, koji će ne samo pomoći studentima u usvajanju gradiva, već će i potaknuti širi interes za medicinu i anatomiju kroz interaktivan i natjecateljski pristup.

Literatura

[1] Aumuller, G., Aust, G., Engele, J., Kirsch, J., Maio, G., Mayerhofer, A., Mense, S., Reissig, D., Salvatter, J., Schmidt, W., Schmitz, F., Schulte, E., Spanel-Borowski, K., Wennemuth, G., Wolf, W., Wurzinger, L. J., Zilch, H. G. *Anatomija Duale Reihe*. 3. izdanje. Zagreb: Medicinska Naklada, 2018.

[2] Gilroy, A., MacPherson, B., Wikenheiser, J., Schuenke, M., Schulte, E., Schumacher, U., Voll, M., Wesker, K., Zeberg, H. *Atlas of Anatomy, Latin Nomenclature*. 4. izdanje. New York, Thieme Medical Publishers, Inc., 2020.

[3] Dauber, W., Feneis, H. *Pocket Atlas of Human Anatomy*. 4. izdanje. New York, Thieme Medical Publishers, Inc., 2000.

[4] Kahoot!. Poveznica: https://kahoot.com/; pristupljeno 8. lipnja 2024.

[5] Quizlet Live. Poveznica: https://quizlet.com/live; pristupljeno 8. lipnja 2024.

[6] Quizizz. Poveznica: https://quizizz.com/?lng=en; pristupljeno 8. lipnja 2024.

[7] Mentimeter, Wikipedia. Poveznica: https://en.wikipedia.org/wiki/Mentimeter; pristupljeno 8. lipnja 2024.

[8] PostgreSQL, Wikipedia. Poveznica: https://en.wikipedia.org/wiki/PostgreSQL; pristupljeno 8. lipnja 2024.

[9] React (JavaScript library), Wikipedia. Poveznica: https://en.wikipedia.org/wiki/React_(JavaScript_library); pristupljeno 8. lipnja 2024.

[10] React Router DOM, Home. Poveznica: https://reactrouter.com/; pristupljeno 8. lipnja 2024.

[11] Spring Boot, Wikipedia. Poveznica: https://en.wikipedia.org/wiki/Spring_Boot; pristupljeno 8. lipnja 2024.

[12] Socket.io, Wikipedia. Poveznica: https://en.wikipedia.org/wiki/Spring_Boot; pristupljeno 8. lipnja 2024.

[13] GitHub, Wikipedia. Poveznica: https://en.wikipedia.org/wiki/GitHub; pristupljeno 8. lipnja 2024.

[14] Heroku, Wikipedia. Poveznica: https://en.wikipedia.org/wiki/Heroku; pristupljeno 8. lipnja 2024.

Sažetak

Aplikacija za učenje i ponavljanje gradiva anatomije

U sklopu ovog završnog rada razvijena je aplikacija za učenje anatomije, koja je prvenstveno namijenjena studentima prve godine medicine, ali i svim drugim zainteresiranim korisnicima. Cilj aplikacije je olakšati ponavljanje i vježbanje gradiva iz anatomije kroz kvizove. Sustav se sastoji od korisničkog sučelja za studente, gdje mogu rješavati kvizove, i baze podataka. Ključne tehnologije korištene u razvoju aplikacije uključuju React.js za front-end, Spring Boot za back-end, PostgreSQL za bazu podataka i Socket.io za komunikaciju u stvarnom vremenu.

Ključne riječi: React.js, Spring Boot, PostgreSQL, Socket.io, anatomija, kviz.

Summary

Application for Learning and Reviewing Anatomy

As part of this thesis, an application for learning anatomy was developed, primarily aimed at first-year medical students but also for all interested users. The goal of the application is to help with learning and practice of anatomy through quizzes. The system consists of a user interface for students, where they can solve quizzes, and the database. The key technologies used in the development of the application include React.js for the front-end, Spring Boot for the back-end, PostgreSQL for the database, and Socket.io for real-time communication.

Keywords: React.js, Spring Boot, PostgreSQL, Socket.io, anatomy, quiz.

Skraćenice

HTTP	*hyper text transfer rotocol*	protokol prijenosa hiperteksta
ACID	*atomicity, consistency, isolation, durability*	atomičnost, konzistentnost, izolacija, trajnost
SPA	*single-page application*	aplikacija na jednoj stranici
UI	*user interface*	korisničko sučelje