

Sadržaj

Uvod	1
1. Postojeća rješenja	3
1.1. Khan Academy	3
1.2. Photomath	4
1.3. Matific	4
1.4. Zaključak	5
2. Funkcionalni zahtjevi aplikacije	6
3. Arhitektura sustava	8
3.1. Pregled sustava	8
3.2. Primjena MVC arhitekture	8
4. Korištene tehnologije	11
4.1. Klijentska strana	11
4.1.1. React	11
4.1.2. Next.js	12
4.1.3. Chakra UI	13
4.1.4. React Query	14
4.1.5. Framer Motion	14
4.1.6. Zod	15
4.1.7. Tailwind CSS	15
4.2. Poslužiteljska strana	16
4.2.1. Spring Boot	16
4.2.2. Spring Security	16
4.2.3. Spring Data JPA	16
4.2.4. Jakarta Validation	17
4.2.5. JSON Web Token (JWT)	17
4.2.6. OpenAI API	19
5. Baza podataka	21
5.1. Supabase	21
5.2. Pravila pristupa i sigurnost	21
5.3. Pohrana datoteka	22
5.4. Vizualna relacijska shema	22
6. Korištenje aplikacije	23
6.1. Početna stranica	23

6.2.	Stranica za registraciju korisnika.....	24
6.3.	Uređivanje profila	25
6.4.	Nastavne cjeline.....	26
6.5.	Kviz	28
6.6.	Rang lista	30
7.	Potencijalna unaprjeđenja	31
7.1.	Napredna analitika	31
7.2.	Pristup sadržajima bez internetske veze.....	31
7.3.	Višejezična podrška	31
7.4.	Administratorsko sučelje i uređivanje sadržaja	31
7.5.	Mobilna aplikacija.....	32
	Zaključak	33
	Literatura	34
	Sažetak	36
	Summary.....	37

Uvod

Matematika je jedan od temeljnih predmeta u obrazovnom sustavu i važan je alat za razumijevanje brojnih prirodnih i tehničkih znanosti. Unatoč njenoj važnosti, mnogi učenici osnovnih i srednjih škola susreću se s brojnim izazovima prilikom savladavanja matematičkih sadržaja. Često se matematika doživljava kao apstraktna, teška i nezanimljiva, što dovodi do manjka motivacije i smanjenog uspjeha u učenju. Tradicionalni pristupi poučavanju, koji se uvelike oslanjaju na predavanja i standardne zadatke, ne uspijevaju uvijek potaknuti interes učenika ni zadovoljiti njihove individualne potrebe.

Upravo iz tog razloga raste potreba za modernim i interaktivnim rješenjima koja učenicima nude personalizirani pristup učenju, kombinirajući lekcije, kvizove i praćenje napretka. U ovom završnom radu razvijena je web-aplikacija **Math++**, osmišljena s ciljem da na zabavan i učinkovit način pomogne učenicima osnovnih i srednjih škola u učenju matematike. Neovisno o tome je li korisnik prilikom registracije odabrao osnovnu ili srednju školu te koji razred pohađa, aplikacija mu omogućuje pristup svim nastavnim cjelinama iz matematike za obje razine obrazovanja. Svaka cjelina sadrži dvadeset zadataka koji se povlače iz baze podataka, a zadaci su koncipirani tako da učenik bira jedan od četiri ponuđena odgovora. Nakon odabira, prikazuje se točan odgovor, što omogućuje brzo učenje kroz povratnu informaciju.

Korisnici imaju mogućnost označiti zadatke koje su uspješno savladali te tako pratiti svoj napredak kroz pojedine cjeline. Također, prije rješavanja zadataka za određenu cjelinu, korisnik može pristupiti poveznici koja ga vodi na vanjsku edukativnu platformu Edutorij [1], gdje se nalazi teorijski dio gradiva i dodatni zadaci za vježbu. Na taj način učenik može temeljito pripremiti gradivo prije samostalnog rješavanja zadataka u web-aplikaciji **Math++**.

Jedna od dodatnih funkcionalnosti aplikacije je **dnevni kviz**, koji se automatski generira svakog dana u ponoć te je prilagođen razini obrazovanja i razredu korisnika. Po završetku kviza, učenici dobivaju detaljan pregled rezultata – uključujući broj točnih odgovora, utrošeno vrijeme i analizu točnosti. Osim toga, korisnik ima pristup i **dnevnoj rang-listi** koja omogućuje usporedbu rezultata s vršnjacima. Na taj način potiče se natjecateljski duh i učenicima pruža dodatna motivacija za učenje.

U poglavlju **1. Postojeća rješenja** dan je pregled postojećih aplikacija za učenje matematike, kao što su Khan Academy [2], Photomath [3] i Matific [4], uz provedenu analizu njihovih prednosti i nedostataka u kontekstu hrvatskog obrazovnog sustava. Poglavlje **2. Funkcionalni zahtjevi aplikacije** definira konkretne funkcionalnosti koje aplikacija *Math++* mora podržavati kako bi zadovoljila potrebe korisnika osnovnih i srednjih škola. U poglavlju **3. Arhitektura sustava** opisana je tehnička struktura aplikacije, s naglaskom na primjenu obrasca MVC (engl. *Model-View-Controller*) i klijentsko-poslužiteljskog modela. Poglavlje **4. Korištene tehnologije** donosi detaljan opis svih alata i biblioteka koje su korištene na klijentskoj i poslužiteljskoj strani aplikacije. U poglavlju **5. Baza podataka** prikazana je organizacija baze podataka izrađene pomoću platforme Supabase [5], uključujući pravila pristupa, sigurnosne mehanizme i način pohrane datoteka. U poglavlju **6. Korištenje aplikacije** opisan je način korištenja svih funkcionalnih dijelova aplikacije *Math++*. Poglavlje **7. Potencijalna unaprjeđenja** iznosi prijedloge za daljnji razvoj aplikacije, poput implementacije mobilne verzije, višejezične podrške, napredne analitike i administracijskog sučelja, dok je u posljednjem poglavlju **Zaključak** dan zaključak te pregled i sažetak ključnih dijelova rada.

1. Postojeća rješenja

Učenje matematike mnogima predstavlja izazov, osobito kada se gradivo ne prezentira na zanimljiv i pristupačan način. S razvojem digitalnih tehnologija pojavile su se brojne aplikacije i platforme koje nastoje modernizirati pristup matematici te učenicima pružiti alate koji im mogu pomoći u savladavanju matematičkog gradiva. Ipak, iako postoje brojna rješenja koja na različite načine pristupaju problematici, malo ih je u potpunosti prilagođeno hrvatskom obrazovnom sustavu, a još manje ih uspješno kombinira elemente teorije, praktičnih zadataka i motivacije kroz interakciju i natjecanje.

U nastavku su predstavljena neka od najpoznatijih rješenja koja se koriste u edukaciji matematike, s naglaskom na njihove prednosti, ali i ograničenja kada je riječ o cjelovitom pristupu učenju matematike u kontekstu osnovnog i srednjoškolskog obrazovanja.

1.1. Khan Academy

Khan Academy [2] je jedna od najpoznatijih globalnih besplatnih obrazovnih platformi koja pokriva različite predmete, uključujući matematiku, što je prikazano na slici (Sl. 1.1). Platforma sadrži veliku bazu sadržaja i zadataka za osnovnu i srednju školu, pa čak i sveučilišnu razinu. Platforma nudi videolekcije, interaktivne zadatke, mogućnost praćenja napretka te individualizirani pristup putem preporuka zadataka. Glavni nedostatak platforme Khan Academy jest to što su sučelje i svi materijali dostupni isključivo na engleskom jeziku, a uz to platforma nije prilagođena hrvatskom kurikulumu. Također, Khan Academy ne nudi mogućnost pregleda rang-liste svih korisnika na kvizovima, već korisnici prate samo svoje osobne rezultate i bodove vezane uz vlastiti profil. Za razliku od nje, *Math++* omogućava upravo tu funkcionalnost – prikaz rang-liste svih sudionika, čime se dodatno potiče natjecateljski duh i motivacija za učenje.

Khan Academy				
MATH: PRE-K - 8TH GRADE	MATH: HIGH SCHOOL & COLLEGE	ECONOMICS	READING & LANGUAGE ARTS	SOCIAL STUDIES
Pre-K through grade 2 (Khan Kids)	Algebra 1	Macroeconomics	Up to 2nd grade (Khan Kids)	US history
Early math review	Geometry	AP®/College Macroeconomics	2nd grade	AP®/College US History
2nd grade	Algebra 2	Microeconomics	3rd grade	US government and civics
3rd grade	Integrated math 1	AP®/College Microeconomics	4th grade	AP®/College US Government & Politics
4th grade	Integrated math 2	See all Economics	5th grade	Constitution 101 NEW
5th grade	Integrated math 3		6th grade reading and vocab NEW	World History Project - Origins to the Present
6th grade	Algebra basics	SCIENCE	7th grade reading and vocab NEW	World History Project - 1750 to the Present
7th grade	Trigonometry	Middle school biology	8th grade reading and vocab NEW	World history
8th grade	Precalculus	Middle school Earth and space science	9th grade reading and vocab NEW	AP®/College World History NEW
Basic geometry and measurement	High school statistics	Middle school physics	10th grade reading and vocab NEW	Big History Project
See Pre-K - 8th Math	Statistics & probability	High school biology		Climate project NEW
	College algebra	High school chemistry		
	AP®/College Calculus AB			

Sl. 1.1 Različiti predmeti i cjeline koje pruža Khan Academy

1.2. Photomath

Photomath [3] je hrvatska aplikacija koja je stekla veliku popularnost zahvaljujući mogućnosti skeniranja matematičkih zadataka s pomoću kamere mobilnog uređaja te prikaza rješenja s detaljnim koracima. Korisnicima omogućuje jednostavno i brzo rješavanje različitih matematičkih problema, a posebno je korisna za razumijevanje načina dolaska do rješenja i samostalnu provjeru točnosti rezultata.

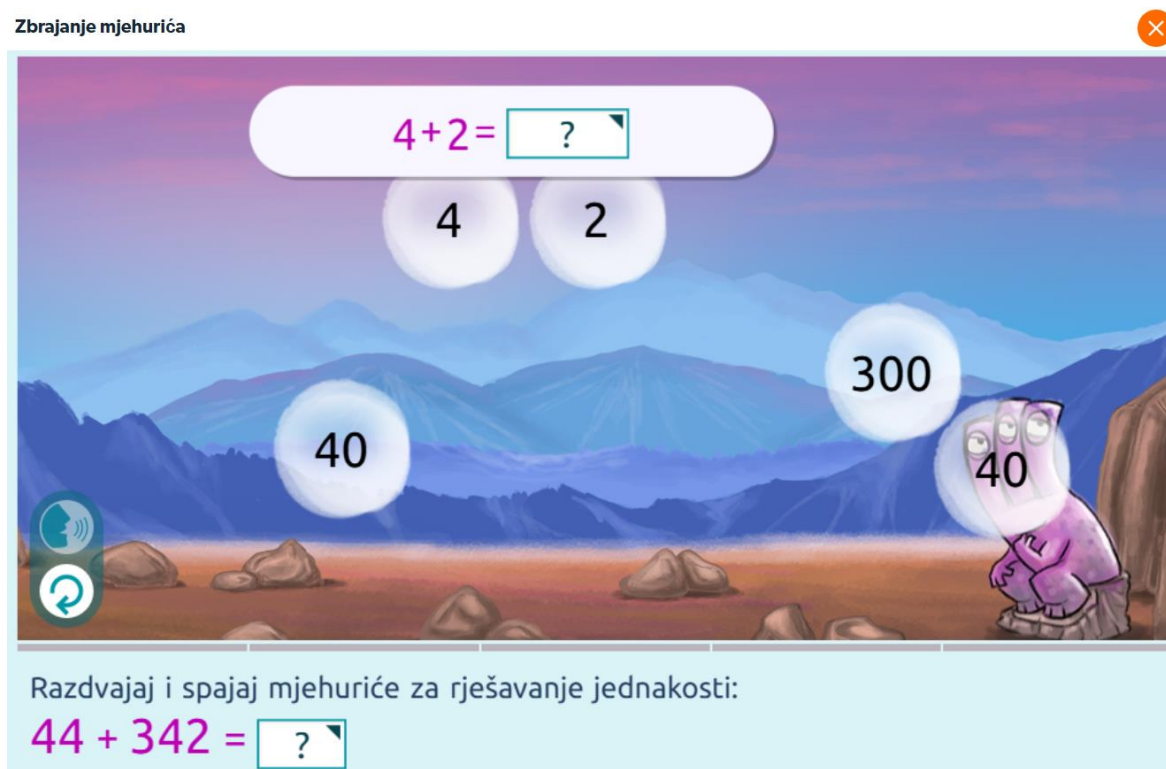
Unatoč svojim prednostima, Photomath ne nudi strukturirani pristup učenju matematike po razredima i nastavnim cjelinama, niti pruža funkcionalnosti poput praćenja napretka učenika ili uključivanja natjecateljskih elemenata koji bi dodatno motivirali korisnike.

1.3. Matific

Matific [4] je edukativna platforma namijenjena učenicima osnovne škole, a temelji se na pristupu učenju kroz interaktivne igre i zadatke. Primjer interaktivne igre je prikazan na slici (Sl. 1.2). Cilj joj je kroz zabavne i vizualno privlačne aktivnosti pomoći djeci da razviju matematičko razmišljanje i razumiju temeljne koncepte. Platforma je dostupna na više od 50 jezika, uključujući i hrvatski te je dostupna za besplatno testiranje u trajanju od sedam dana. Nakon probnog razdoblja, pristup se naplaćuje, što može ograničiti dostupnost svim korisnicima.

Jedna od značajki platforme je prilagodljivi put učenja, koji koristi inicijalne ispite kako bi putem vlastitog algoritma odredio optimalne zadatke za svakog pojedinog učenika. Također, Matific omogućuje izvođenje matematičkih kvizova uživo u razredu, u kojima se učenici mogu međusobno natjecati. Nastavnik pritom ima pristup tablici s rezultatima, što mu omogućuje da dodatno potakne učenike na rad i angažman.

Međutim, važno je naglasiti da je Matific strogo usmjeren na osnovnoškolski uzrast i ne podržava sadržaje prilagođene srednjoškolcima.



Sl. 1.2 Primjer interaktivne matematičke igre platforme Matific

1.4. Zaključak

Iako na tržištu već postoji niz aplikacija koje učenicima olakšavaju učenje matematike, analiza pokazuje da nijedna od njih ne nudi cjelovito rješenje koje je istovremeno prilagođeno hrvatskom obrazovnom sustavu, obuhvaća i osnovnu i srednju školu te je obogaćeno elementima interakcije i motivacije kroz natjecanje. Upravo iz tog razloga je u okviru ovog rada razvijena web-aplikacija *Math++*, s ciljem da objedini najbolje značajke postojećih rješenja i prilagodi ih konkretnim potrebama učenika osnovnih i srednjih škola u Hrvatskoj.

2. Funkcionalni zahtjevi aplikacije

Cilj ovog rada bio je razvoj web-aplikacije koja na zabavan i interaktivan način omogućava učenicima osnovnih i srednjih škola rješavanje različitih matematičkih zadataka omogućujući im personalizirano iskustvo, praćenje vlastitog napretka i postavljanje ciljeva. Aplikacija treba biti usklađena s hrvatskim obrazovnim sustavom te obuhvaćati nastavne cjeline svih razreda osnovne i srednje škole.

Aplikacija **Math++** mora ispunjavati sljedeće funkcionalne zahtjeve:

- **Registracija korisnika:** Svaki korisnik mora imati mogućnost izrade vlastitog korisničkog računa putem registracijskog obrasca, pri čemu odabire razinu obrazovanja (osnovna ili srednja škola) i pripadajući razred.
- **Prijava u sustav:** Korisnici s postojećim računom mogu se prijaviti u sustav putem svojih vjerodajnica (korisničko ime i lozinka).
- **Uređivanje korisničkog profila:** Korisnici imaju mogućnost uređivanja svog profila – mogu dodati ili promijeniti profilnu sliku te ažurirati osobne podatke kao što su ime, prezime, korisničko ime i adresa e-pošte.
- **Pristup nastavnim cjelinama:** Nakon prijave, korisniku su dostupne sve nastavne cjeline iz matematike, neovisno o tome je li riječ o osnovnoj ili srednjoj školi, čime se omogućuje dodatno istraživanje i samostalno učenje.
- **Rješavanje zadataka:** Za svaku nastavnu cjelinu dostupno je dvadeset zadataka koji se povlače iz baze podataka. Zadaci su višestrukog izbora s četiri ponuđena odgovora, a nakon svakog rješavanja prikazuje se točno rješenje uz mogućnost označavanja zadatka kao "naučenog".
- **Praćenje napretka:** Sustav korisnicima omogućuje uvid u vlastiti napredak kroz pojedine cjeline te pregled postignuća na dnevnom kvizu.
- **Povezivanje s vanjskim izvorima:** Korisnici imaju mogućnost pristupa vanjskim edukativnim sadržajima putem poveznica na vanjsku edukativnu platformu Edutorij [1], što im omogućuje teorijsku pripremu prije rješavanja zadataka.

- **Dnevni kviz:** Aplikacija automatski generira dnevni kviz koji je prilagođen razini obrazovanja i razredu korisnika. Kviz se sastoji od deset zadataka različitih težina te nakon rješavanja korisniku pruža detaljan pregled rezultata, uključujući broj točnih odgovora, vrijeme rješavanja i analizu uspješnosti.
- **Rangiranje korisnika:** Na temelju rezultata dnevnih kvizova, korisnicima se prikazuje dnevna rang-lista, čime se potiče natjecateljski duh i dodatna motivacija za kontinuirano učenje.

3. Arhitektura sustava

Aplikacija *Math++* razvijena je kao web-sustav temeljen na klijentsko-poslužiteljskoj arhitekturi. Sustav se sastoji od više međusobno povezanih komponenti koje zajedno omogućuju funkcionalnosti opisane u prethodnom poglavlju. Arhitektura je modularna i omogućuje lako održavanje te buduću nadogradnju aplikacije.

3.1. Pregled sustava

Sustav se sastoji od tri osnovne cjeline:

- **Klijentska strana (engl. *Frontend*)** – izrađena korištenjem biblioteke React [6] u kombinaciji s TypeScriptom [7], čime se postiže veća sigurnost tipova podataka i bolja organizacija koda. Klijentska strana predstavlja sučelje koje korisnici vide i s kojim izravno komuniciraju putem preglednika, a ona onda putem HTTP zahtjeva komunicira s poslužiteljskom stranom i dinamički prikazuje sadržaj korisnicima.
- **Poslužiteljska strana (engl. *Backend*)** – izrađena korištenjem okvira Spring Boot [8] u programskom jeziku Java. Poslužiteljska strana odgovorna je za obradu podataka, autentifikaciju korisnika, generiranje kvizova i komunikaciju s bazom podataka.
- **Baza podataka** – korišten je Supabase, moderna platforma otvorenog koda (engl. *open source*) temeljena na PostgreSQL-u, koja nudi mogućnosti klasične relacijske baze podataka. Osim toga, korišten je i Supabase Storage [9], koji omogućuje pohranu i upravljanje datotekama – u ovom slučaju za spremanje profilnih slika korisnika.

Detaljan pregled svih korištenih tehnologija i njihove uloge u razvoju aplikacije prikazan je u poglavlju 4. Korištene tehnologije.

3.2. Primjena MVC arhitekture

Arhitektonski obrazac MVC (engl. *Model–View–Controller*) [10] primijenjen je kao temelj organizacije sustava *Math++*, pri čemu se osobito koristi na strani poslužitelja (engl. *backend*), ali njegova struktura ima utjecaj i na način na koji klijent (engl. *frontend*) i poslužitelj međusobno komuniciraju. Ovakav pristup omogućuje jasno razdvajanje odgovornosti, veću preglednost koda te lakšu nadogradnju i održavanje sustava.

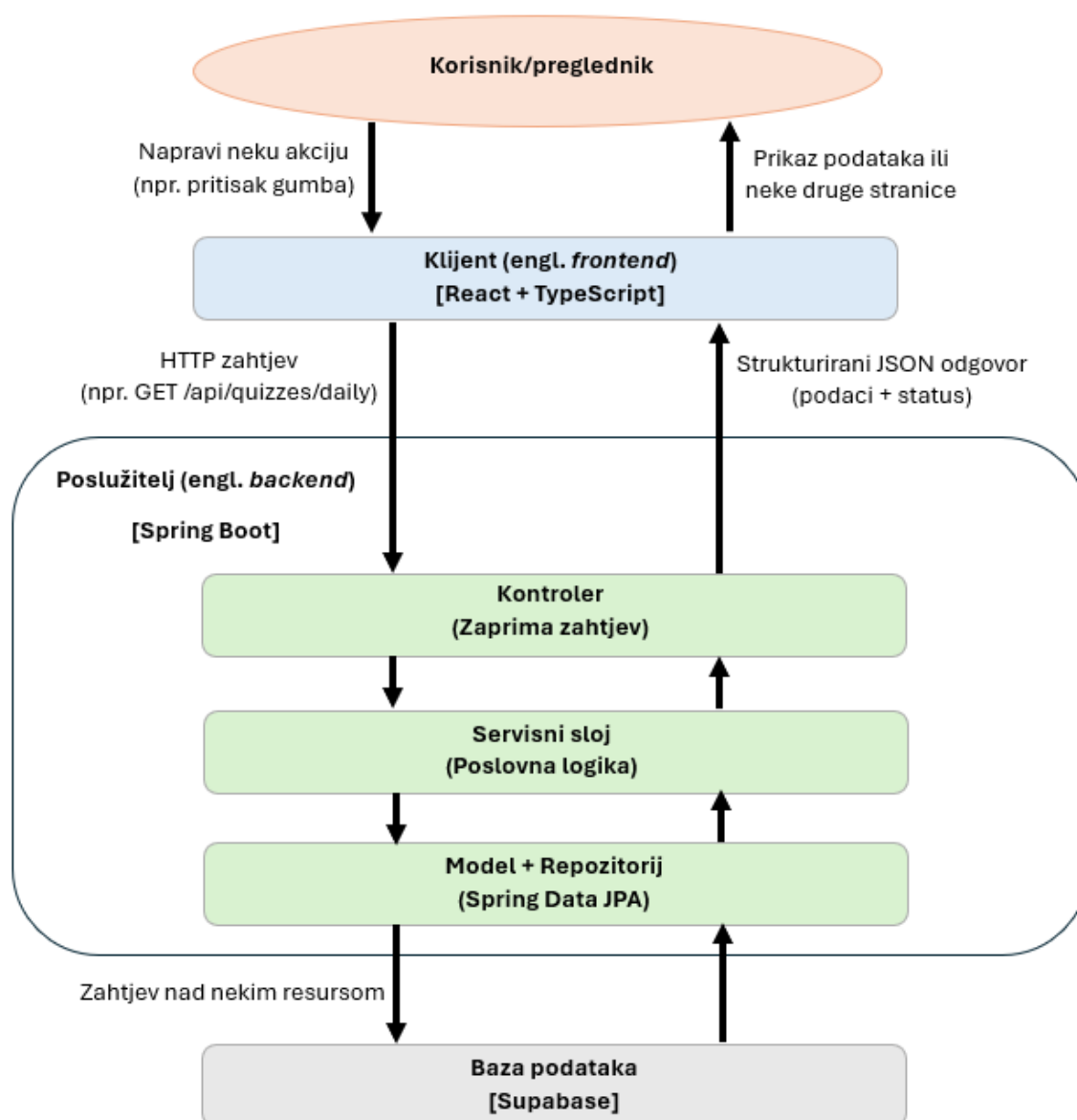
S obzirom na to da *Math++* funkcioniše kao web-sustav temeljen na klijentsko-poslužiteljskoj arhitekturi, obrazac MVC (engl. *Model–View–Controller*) implementiran je tako da poslužitelj predstavlja centralni logički sloj, dok klijent preuzima ulogu prikaza i interakcije s krajnjim korisnikom, tj. preglednikom.

Detaljan tok podataka kroz sustav – od korisničkog zahtjeva kroz preglednik do prikaza rezultata – ilustriran je na slici (Sl. 3.1).

U nastavku su opisane ključne uloge svakog sloja:

- **Model (engl. *Model*)** – predstavlja poslovne entitete, odnosno strukture podataka koje se koriste za rad s bazom. U aplikaciji su to klase poput *User*, *Quiz*, *QuizQuestion*, *UserQuiz*, koje definiraju attribute i odnose među podacima. Modeli su povezani s bazom podataka putem sloja Spring Data JPA, što omogućuje jednostavan pristup podacima.
- **Kontroler (engl. *Controller*)** – zadužen je za obradu dolaznih HTTP zahtjeva. U *Math++* aplikaciji, kontroleri definiraju REST API krajnje točke (engl. *endpoints*) koje klijent (engl. *frontend*) koristi za komunikaciju (npr. za dohvat zadataka, spremanje rezultata kviza ili upravljanje korisničkim podacima). Svaki zahtjev obrađuje odgovarajući kontroler, koji zatim koristi servisni sloj te vraća odgovor u JSON formatu.
- **Pogled (engl. *View*)** – u kontekstu REST API-ja, "pogled" nije tradicionalna HTML stranica, već predstavlja strukturirani JSON odgovor koji se šalje natrag klijentu (engl. *frontend*). Klijent zatim interpretira te podatke i prikazuje ih korisniku – čime zapravo postaje vidljivi sloj obrasca MVC.
- **Servisni sloj (engl. *Service*)** – iako izvorno nije dio obrasca MVC, u aplikaciji *Math++* dodatno je implementiran i ovaj sloj između kontrolera i modela. U njemu se nalazi poslovna logika poput različitih validacija i filtriranja podataka. Time se dodatno unapređuje modularnost sustava i omogućuje jednostavnije testiranje.

Primjenom ovako prilagođenog obrasca MVC omogućena je jasna podjela između korisničkog sučelja, logike sustava i upravljanja podacima, što rezultira boljom skalabilnošću i održivošću cijele aplikacije.



Sl. 3.1 Tok podataka kroz sustav

4. Korištene tehnologije

U ovom poglavlju navedene su tehnologije koje su korištene u razvoju web-aplikacije *Math++*. Sustav je razvijen kao moderna web-aplikacija s odvojenim klijentskim i poslužiteljskim dijelom, a za implementaciju su korišteni suvremeni alati i biblioteke koje omogućuju responzivno korisničko sučelje, stabilan rad poslužitelja te pouzdano upravljanje podacima. Tehnologije su podijeljene prema funkcionalnim cjelinama sustava: klijentska strana (engl. *frontend*) i poslužiteljska strana (engl. *backend*).

4.1. Klijentska strana

Klijentska strana aplikacije *Math++* izrađena je korištenjem biblioteke React [6] u kombinaciji s okvirom Next.js [11] te je implementirana u programskom jeziku TypeScript [7] radi veće sigurnosti i čitljivosti koda. Klijent je odgovoran za interakciju s korisnikom, prikaz sučelja te komunikaciju s poslužiteljskom stranom putem REST API-ja.

Osim temeljnog okruženja React/Next, aplikacija koristi niz dodatnih biblioteka čija će uloga biti detaljno objašnjena u nastavku ovog poglavlja.

4.1.1. React

React [6] je popularna Facebookova JavaScript biblioteka, namijenjena izradi dinamičkih i interaktivnih korisničkih sučelja, primarno za web-aplikacije. Temelji se na komponentnom pristupu, pri čemu se aplikacija gradi od manjih, višekratno iskoristivih dijelova (komponenti).

U web-aplikaciji *Math++*, React je iskorišten za:

- izradu modularnog sučelja koje se lako održava i proširuje
- dinamičko ažuriranje sadržaja bez ponovnog učitavanja stranice (SPA pristup – engl. *Single Page Application*, gdje se cijela aplikacija izvršava unutar jedne HTML stranice uz dinamičko učitavanje sadržaja putem API-ja)

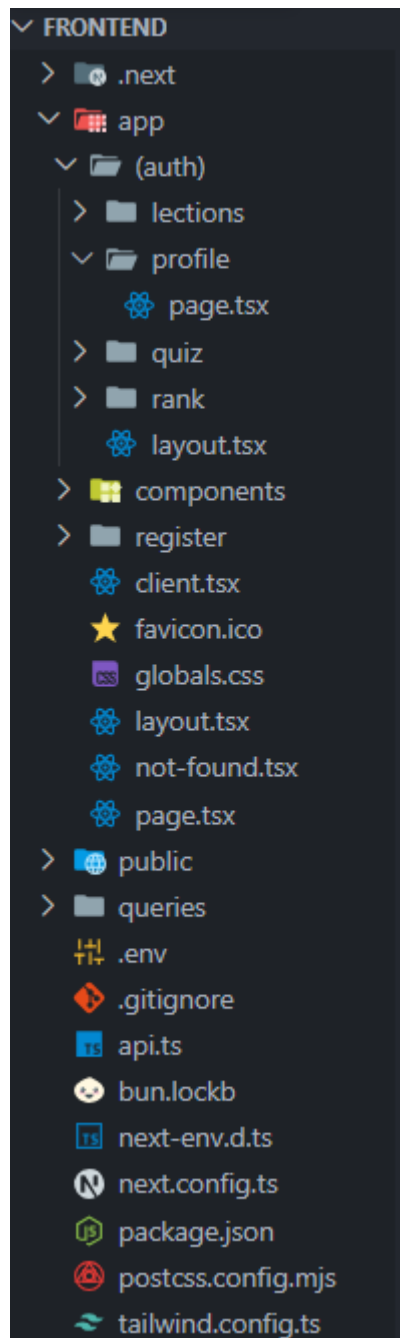
React se pokazao kao odličan izbor za razvoj klijentske strane aplikacije zahvaljujući svojoj fleksibilnosti, komponentnom pristupu i jednostavnoj integraciji s dodatnim bibliotekama poput Chakra UI [12], React Query [13] i Framer Motion [14].

4.1.2. Next.js

Next.js [11] je Reactov napredni okvir za razvoj (engl. *framework*) koji omogućuje razvoj bržih, optimiziranih i skalabilnih web-aplikacija. U okviru aplikacije *Math++*, Next.js je odabran kao temelj klijentskog sustava (engl. *frontend*) zbog svojih brojnih prednosti koje pojednostavljaju razvoj, poboljšavaju performanse i omogućuju bolju kontrolu nad prikazom i ponašanjem aplikacije.

Ključne značajke Next.js-a koje su korištene u razvoju aplikacije *Math++* uključuju:

- **Pojednostavljen sustav usmjeravanja (engl. *routing*)** – u okviru aplikacije Next.js koristi se *app* direktorij, unutar kojeg svaka *page.tsx* datoteka predstavlja krajnju točku (stranicu) dostupnu korisniku. Struktura putanja određuje se hijerarhijom poddirektorija – što je datoteka dublje ugniježđena, to je njezina ruta specifičnija. Također, ako je naziv poddirektorija obavijen zagradama (npr. *(auth)*) tada se taj direktorij koristi isključivo za organizaciju koda i ne sudjeluje u stvaranju URL putanje. Na taj se način omogućuje modularniji i pregledniji kod bez utjecaja na krajnju strukturu rute. Primjer takve strukture i oblikovanja putanja prikazan je na slici (Sl. 4.1), gdje se vidi da npr. datoteka *page.tsx* unutar direktorija „*app/(auth)/profile*“ definira stranicu dostupnu putem URL putanje „*/profile*“.
- **Prikazivanje na strani poslužitelja (engl. *server-side rendering*) i generiranje statičkih stranica (engl. *static site generation*)** omogućuju prikaz stranica na poslužitelju, bilo u trenutku zahtjeva ili unaprijed tijekom izrade aplikacije (engl. *building*). Ovo rezultira bržim učitavanjem sadržaja i boljim performansama, posebno kod dijelova aplikacije koji se ne mijenjaju često, čime se značajno poboljšava korisničko iskustvo.
- **Optimizacija performansi** – Next.js automatski optimizira pakete (engl. *bundles*), učitavanje komponenti i ostalih resursa. Također podržava odgođeno učitavanje komponenti (engl. *lazy loading*), što omogućuje brže prikazivanje samo onog sadržaja koji je korisniku trenutačno potreban.



Sl. 4.1 Struktura putanja u Next.js-u

4.1.3. Chakra UI

Chakra UI [12] je biblioteka komponenti za React koja omogućuje brzo i jednostavno oblikovanje responzivnog korisničkog sučelja. U aplikaciji *Math++* koristi se za izradu vizualno privlačnih elemenata poput gumba, obrazaca, kartica i ikona. Modularni pristup i jednostavna integracija ove biblioteke značajno ubrzavaju razvoj sučelja i poboljšavaju korisničko iskustvo.

4.1.4. React Query

React Query [13], danas poznat i kao TanStack Query, biblioteka je za upravljanje vanjskim podacima i asinkronim upitima u React aplikacijama. U aplikaciji *Math++* koristi se za dohvat i pred memoriranje (engl. *caching*) podataka s poslužitelja, kao što su informacije o korisniku. Pruža strukturiran način za rukovanje podacima dobivenih s poslužitelja te automatski upravlja stanjima poput učitavanja, pogrešaka i ponovnog dohvaćanja, čime pojednostavljuje razvoj i povećava pouzdanost aplikacije.

Primjer korištenja React Queryja za dohvat korisničkih podataka prikazan je na slici (Sl. 4.2).

```
import { useQuery } from "@tanstack/react-query";
import { useRouter } from "next/navigation";
import { apiCall } from "~/api";

export const useGetUser = () => {
  const router = useRouter();

  return useQuery({
    queryKey: ["user"],
    retry: false, // ako je false onda ako query fail-a neće se opet slati
    queryFn: async () => {
      const [data, status] = await apiCall(`/users/info`, { method: "GET" });

      if (status !== 200) {
        localStorage.removeItem("token");
        router.push("/");
        return null;
      }

      return data;
    },
  });
};
```

Sl. 4.2 Dohvat korisničkih podataka pomoću React Queryja

4.1.5. Framer Motion

Framer Motion [14] je biblioteka za animaciju u React aplikacijama koja omogućuje jednostavno stvaranje glatkih i modernih prijelaza koji značajno poboljšavaju vizualni dojam aplikacije, bez potrebe za složenim CSS animacijama. U aplikaciji *Math++* koristi se za animaciju prikaza komponenti na početnoj stranici.

4.1.6. Zod

Zod [15] je biblioteka za validaciju podataka u TypeScriptu. U aplikaciji *Math++* koristi se za provjeru ispravnosti korisničkih unosa prilikom registracije, prijave i uređivanja profila. Omogućuje jednostavnu i pouzdanu provjeru podataka na temelju definiranih pravila, pri čemu automatski osigurava da podaci budu u skladu s očekivanim tipovima, što smanjuje mogućnost pogrešaka i doprinosi stabilnosti aplikacije. Kako je vidljivo na slici (Sl. 4.3), Zod shema definira pravila za validaciju korisničkih polja poput imena, prezimena, korisničkog imena, adrese e-pošte i lozinke.

```
import { z } from "zod";

const signUpSchema = z.object({
  first_name: z
    .string()
    .min(2, "Ime mora imati barem 2 znaka!")
    .regex(/^[A-ZČĆŠĐŽ][a-zA-ZČĆŠĐŽčćšđž]*$/, "Neispravan format imena!"),
  last_name: z
    .string()
    .min(2, "Prezime mora imati barem 2 znaka!")
    .regex(/^[A-ZŠČĆĐŽ][a-zA-ZČĆŠĐŽčćšđž]*$/, "Neispravan format prezimena!"),
  username: z
    .string()
    .min(4, "Korisničko ime mora imati barem 4 znaka!")
    .regex(/^[a-zA-Z][a-zA-Z0-9._]*$/, "Neispravan format korisničkog imena!"),
  email: z.string().email("Neispravan email format!"),
  password: z.string().min(6, "Lozinka mora imati barem 6 znakova!"),
});
```

Sl. 4.3 Zod shema za validaciju podataka pri registraciji korisnika

4.1.7. Tailwind CSS

Tailwind CSS [16] je CSS okvir koji omogućuje brzo i učinkovito stiliziranje sučelja izravno unutar HTML/JSEX elemenata, bez potrebe za pisanjem zasebnih CSS datoteka. U aplikaciji *Math++* koristi se za definiranje izgleda elemenata, rasporeda, boja i responzivnosti.

4.2. Poslužiteljska strana

Poslužiteljska strana aplikacije *Math++* razvijena je korištenjem okvira Spring Boot [8], jednog od najraširenijih rješenja za izgradnju web-aplikacija u programskom jeziku Java. Poslužitelj aplikacije strukturiran je kao REST API koji komunicira s klijentom i bazom podataka, obrađuje korisničke zahtjeve te upravlja poslovnom logikom.

Web-aplikacija *Math++* koristi niz tehnologija i biblioteka koje osiguravaju stabilnost, sigurnost, validaciju, autentifikaciju te upravljanje pohranom podataka. U nastavku su opisani najvažniji moduli korišteni u razvoju poslužiteljske strane.

4.2.1. Spring Boot

Spring Boot [8] je razvojni okvir temeljen na platformi Spring, koji pojednostavljuje konfiguraciju i pokretanje Java aplikacija. U aplikaciji *Math++* koristi se za izgradnju REST API-ja, automatsko povezivanje zahtjeva s metodama kontrolera, injektiranje ovisnosti i organizaciju slojeva unutar aplikacije (kontroleri, servisi, repozitoriji, modeli), o čemu je detaljnije pisano u poglavlju 3.2 Primjena MVC arhitekture.

Modul *spring-boot-starter-web* omogućuje obradu HTTP zahtjeva, dok modul *spring-boot-starter-data-jpa* upravlja komunikacijom s bazom podataka.

4.2.2. Spring Security

Spring Security [17] koristi se za upravljanje autentifikacijom, autorizaciju korisnika te za zaštitu API-ja. Osigurava pristup zaštićenim resursima i omogućuje integraciju s JSON Web Tokenima (JWT) za autentifikaciju putem tokena. Zahvaljujući ovoj biblioteci, samo ovlašteni korisnici mogu pristupiti određenim dijelovima sustava (npr. profil, rezultati kvizova).

4.2.3. Spring Data JPA

Spring Data JPA [18] omogućuje jednostavan rad s bazom podataka korištenjem JPA (engl. *Java Persistence API*). U aplikaciji se koristi za mapiranje entiteta na PostgreSQL tablice, automatsko generiranje SQL upita i upravljanje podacima o korisnicima, kvizovima, lekcijama. Uz pomoć anotacija (*@Entity*, *@Repository*, *@Query*), baza podataka se koristi na siguran i strukturiran način, bez pisanja ručnog SQL-a.

4.2.4. Jakarta Validation

Za provjeru ispravnosti podataka na poslužitelju koristi se Jakarta Validation API [19]. Omogućuje dodavanje anotacija (npr. `@NotNull`, `@Email`, `@Size`) na podatkovne modele kako bi se osigurala pravila validacije već prije same obrade podataka. Na taj se način sprječava unos neispravnih ili nepotpunih podataka u bazu.

4.2.5. JSON Web Token (JWT)

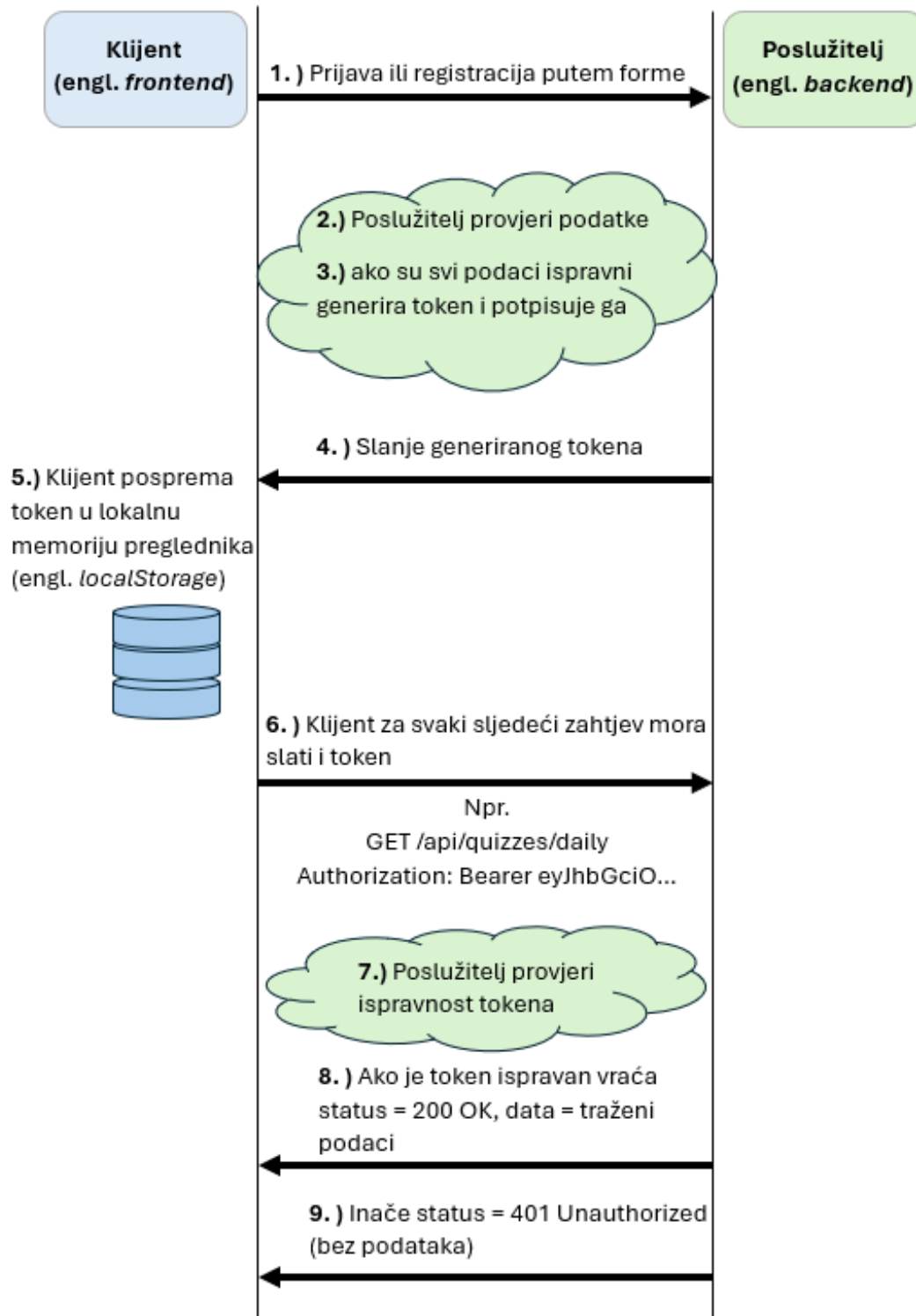
Za autentifikaciju korisnika u aplikaciji *Math++* koristi se sustav JSON Web Token (JWT) [20], implementiran putem biblioteka *jjwt-api*, *jjwt-impl* i *jjwt-jackson*. Ovaj mehanizam omogućuje sigurno prepoznavanje korisnika bez potrebe za stalnim slanjem korisničkih podataka poput korisničkog imena i lozinke.

Prilikom registracije ili prijave, poslužiteljska strana (engl. *backend*) izdaje JWT token, koji uključuje ID korisnika. U tim trenucima klijentska strana (engl. *frontend*) ne mora slati token, jer ga sustav tek generira i vraća klijentu kao odgovor, ako je sve prošlo uspješno.

Nakon uspješne prijave ili registracije, dobiveni token se sprema u lokalnu memoriju preglednika (engl. *local storage*). Token se zatim s klijenta automatski šalje u zaglavlju svakog sljedećeg HTTP zahtjeva prema zaštićenim resursima (npr. dohvat profila, rješavanje kviza, uređivanje podataka).

Na strani poslužitelja svaki se token provjerava kako bi se utvrdila njegova valjanost – provodi se provjera potpisa, provjerava se korisnički identitet i gleda je li token istekao. Rok trajanja tokena definiran je na 24 sata, nakon čega korisnik mora ponovno obaviti autentifikaciju.

Detaljan prikaz toka autentifikacije putem JWT tokena prikazan je na slici (Sl. 4.4).



Sl. 4.4 Tok autentifikacije putem JWT tokena

4.2.6. OpenAI API

U web-aplikaciji *Math++* svakodnevni kvizovi generiraju se automatski korištenjem OpenAI API-ja, čime se osigurava raznolik i dinamičan sadržaj prilagođen znanju korisnika. Ova integracija omogućuje aplikaciji da ne ovisi isključivo o unaprijed definiranim zadacima u bazi podataka, već koristi napredni jezični model za generiranje pitanja u stvarnom vremenu.

Generiranje kvizova implementirano je na poslužiteljskoj strani aplikacije (engl. *backend*), gdje se koristi HTTP komunikacija za slanje zahtjeva prema OpenAI API-ju. Sadržaj kviza generira se na temelju razine obrazovanja i razreda korisnika, pri čemu se prvo dohvaćaju relevantne nastavne teme za dani razred. Te se informacije zatim uključuju u posebno formulirani tekstualni zahtjev (engl. *prompt*) koji se šalje prema jezičnom modelu.

U tekstualnom zahtjevu se jasno navodi:

- da se generira **točno 10 matematičkih pitanja**
- da svako pitanje mora imati **četiri ponuđena odgovora**
- da postoji **samo jedan točan odgovor**, označen vrijednošću *correctAnswerIndex* između 0 i 3
- da svi odgovori moraju biti **nedvosmisleni i jasno različiti**
- da se **pozicija točnog odgovora mora mijenjati** između pitanja.

Prikaz sadržaja tekstualnog zahtjeva koji se šalje OpenAI API-ju vidljiv je na slici (Sl. 4.5). Tako formulirani zahtjev omogućuje da generirani sadržaj bude dosljedan, tehnički ispravan i prikladan za automatsku obradu unutar aplikacije. Kao odgovor model vraća strukturirani JSON niz koji se automatski parsira i pohranjuje u bazu podataka kao sadržaj dnevnog kviza.

```
String prompt = String.format("""
Ti si iskusni profesor matematike. Generiraj točno 10 matematičkih pitanja za %s. razred %s.
Pitanja moraju biti %s.

VAŽNE SPECIFIKACIJE:
1. Pitanja moraju biti prilagođena uzrastu učenika
2. Težina pitanja mora odgovarati razredu
3. Svako pitanje mora imati TOČNO JEDAN točan odgovor
4. Točan odgovor ne smije uvijek biti na istoj poziciji
5. Odgovori moraju biti jasno različiti i nedvosmisleni
6. Izbjegavati pitanja gdje bi više odgovora moglo biti točno

FORMAT PITANJA:
- Tekst pitanja (jasno i konkretno, max 2 rečenice)
- 4 ponuđena odgovora (samo jedan točan)
- Točan odgovor označen indexom (0-3)

Vrati SAMO JSON array u točno ovom formatu :
[
  {
    "text": "Koliko je 7 + 8?",
    "options": ["13", "14", "15", "16"],
    "correctAnswerIndex": 2
  },
  ...
]

VAŽNO: Vrati SAMO JSON array, bez ikakvih dodatnih znakova, komentara ili markdown oznaka.
JSON array mora biti valjan i moći će se parsirati direktno.
""", grade, levelName, lessonText);
```

Sl. 4.5 Tekstualni zahtjev koji se šalje OpenAI API-ju

Za komunikaciju s OpenAI servisom koristi se model *GPT-4o*, s postavljenom temperaturom od 0.5 kako bi se uravnotežila kreativnost i konzistentnost odgovora. Povratni JSON rezultat koji se dobiva od OpenAI API-ja obrađuje se na poslužiteljskoj strani aplikacije. Tijekom obrade provodi se validacija kako bi se osiguralo da generirani sadržaj odgovara očekivanom formatu. Sustav provjerava postoji li točno deset zadataka, je li za svaki zadatak navedeno točno četiri ponuđena odgovora te je li indeks točnog odgovora unutar dozvoljenog raspona vrijednosti (od 0 do 3). Ako bilo koji od tih uvjeta nije zadovoljen, odgovor se smatra neispravnim, a sustav automatski prijavljuje grešku i sprječava prikaz takvog kviza korisniku. Na taj se način održava pouzdanost i kvaliteta prikazanih pitanja.

5. Baza podataka

Baza podataka predstavlja temelj aplikacije *Math++*, jer omogućuje trajnu pohranu svih ključnih informacija vezanih uz korisnike, kvizove, lekcije i zadatke. Za razvoj baze podataka korišten je Supabase [5] – moderna platforma otvorenog koda (engl. *open source*) temeljena na PostgreSQL-u, koja kombinira relacijsku bazu, REST API pristup te mogućnosti upravljanja datotekama.

U nastavku su prikazane glavne značajke baze podataka i njezina organizacija.

5.1. Supabase

Supabase [5] je platforma koja nudi sve što je potrebno za poslužiteljsku stranu aplikacije (engl. *backend*) bez potrebe za ručnim podešavanjem sustava. Temelji se na PostgreSQL-u, što osigurava pouzdanu i strukturiranu pohranu podataka, a uz to automatski generira REST API za svaku tablicu te podržava detaljna pravila pristupa kroz Row-Level Security [21].

U aplikaciji *Math++* Supabase se koristi za:

- Sigurnu pohranu podataka (korisnici, kvizovi, lekcije, zadaci)
- Pohranu profilnih slika korisnika kroz Supabase Storage
- Preciznu kontrolu pristupa putem Row-Level Security (RLS)

5.2. Pravila pristupa i sigurnost

Supabase podržava mehanizam Row-Level Security (RLS) [21], koji omogućuje detaljnu kontrolu pristupa podacima na razini pojedinačnih redova unutar tablica.

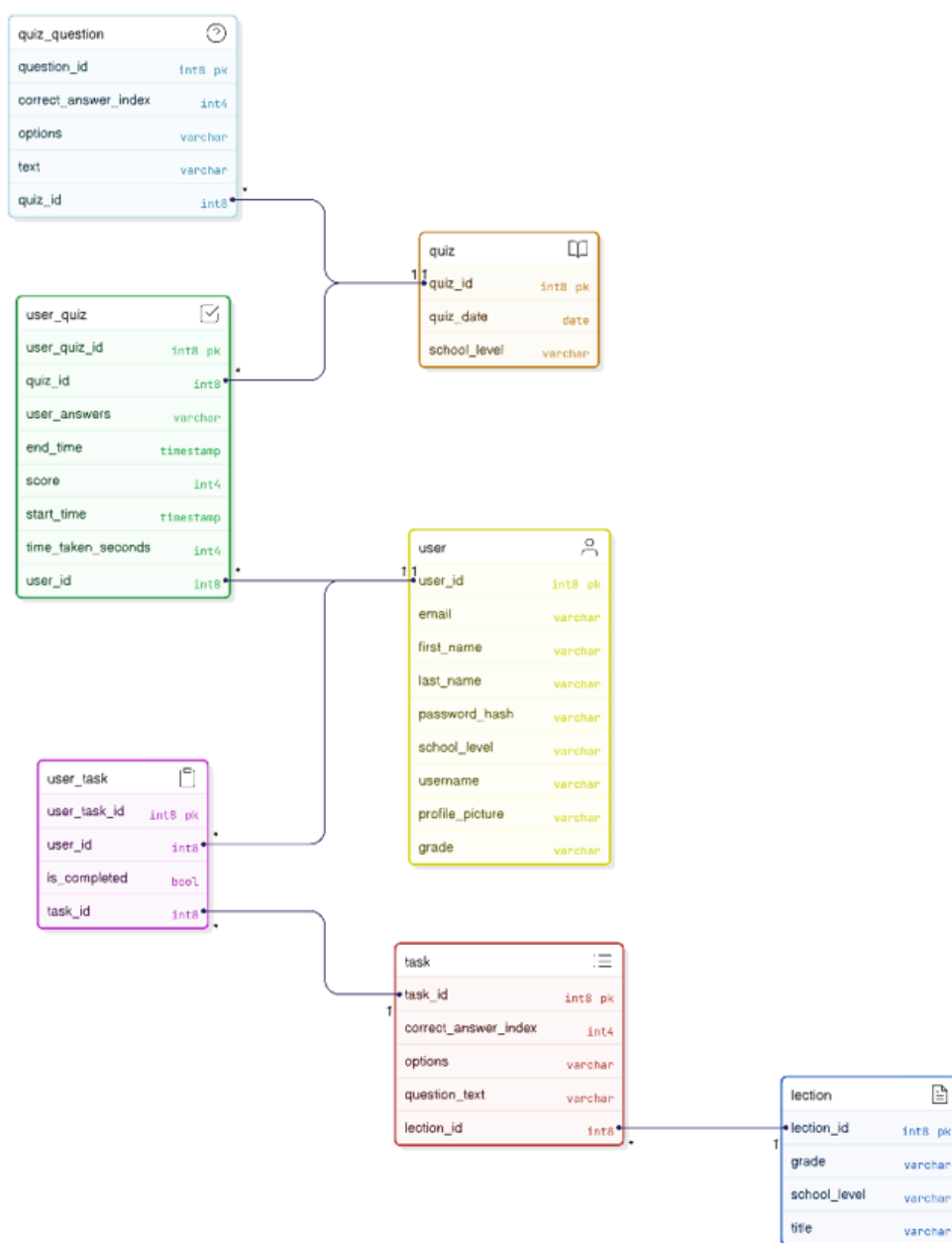
U aplikaciji *Math++*, RLS je trajno omogućen na svim tablicama te nijedno pravilo pristupa nije definirano. Ovakva konfiguracija predstavlja svjesnu sigurnosnu odluku: nijedan korisnik – uključujući i prijavljene korisnike – nema pristup podacima direktno putem REST API-ja. Ovaj pristup osigurava potpunu privatnost i zaštitu sadržaja te jamči da se podacima može pristupiti samo ako se u budućnosti odluče postaviti precizna pravila koja dopuštaju određene radnje. Trenutačno jedini aktivan pristup bazi podataka dopušten je isključivo putem poslužiteljske strane (engl. *backend*), koja koristi sigurni servisni ključ za komunikaciju sa Supabaseom. Time se dodatno pojačava kontrola pristupa i sprečava bilo kakva manipulacija s klijentske strane.

5.3. Pohrana datoteka

Za spremanje profilnih slika korisnika koristi se Supabase Storage [9], koji funkcioniра kao sustav za pohranu objekata (engl. *object storage*). Svaki korisnik ima jedinstveni naziv slike koja se sprema unutar definiranog koša (engl. *bucket*), a onda se slike dohvaćaju putem generiranih URL-ova.

5.4. Vizualna relacijska shema

Struktura i povezanost tablica prikazana je na slici (Sl. 5.1), koja prikazuje glavne entitete i njihove odnose.



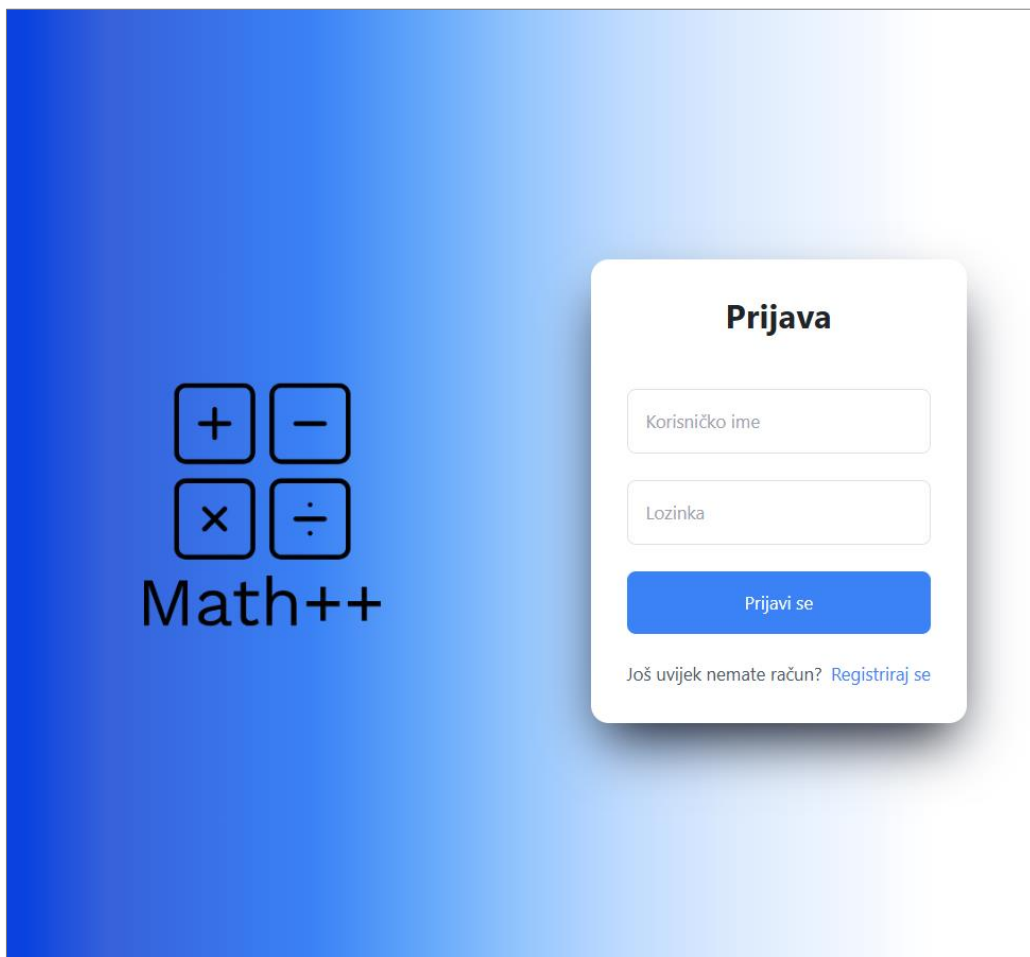
Sl. 5.1 Relacijska shema baze podataka

6. Korištenje aplikacije

U ovom poglavlju opisano je korištenje aplikacije kroz prikaz svake njezine stranice. Objasnjene su osnovne funkcionalnosti i svrha pojedinih dijelova sučelja.

6.1. Početna stranica

Na početnoj stranici, korisniku koji nije prijavljen prikazuje se forma za prijavu u sustav. Prijava se vrši unosom korisničkog imena i lozinke, nakon čega korisnik dobiva token od poslužitelja koji se pohranjuje u memoriju preglednika. Izgled stranice prikazan je na slici (Sl. 6.1).



Sl. 6.1 Početna stranica

6.2. Stranica za registraciju korisnika

Ako osoba još nema kreiran korisnički račun, može ga izraditi putem registracijske forme, čiji je izgled prikazan na slici (Sl. 6.2). Forma zahtjeva unos osnovnih podataka: ime, prezime, korisničko ime, adresu e-pošte te lozinku koja mora sadržavati najmanje šest znakova. Prilikom unosa provodi se validacija svih polja, uključujući provjeru jedinstvenosti korisničkog imena i adrese e-pošte, kako bi se spriječilo njihovo višestruko korištenje. Nakon ispravnog unosa svih obaveznih podataka, korisnik dodatno odabire razinu obrazovanja koju pohađa te pripadajući razred.

Math++

Registracija

Ime

Prezime

Korisničko ime

Email

Lozinka

Odaberite razinu svojeg obrazovanja:

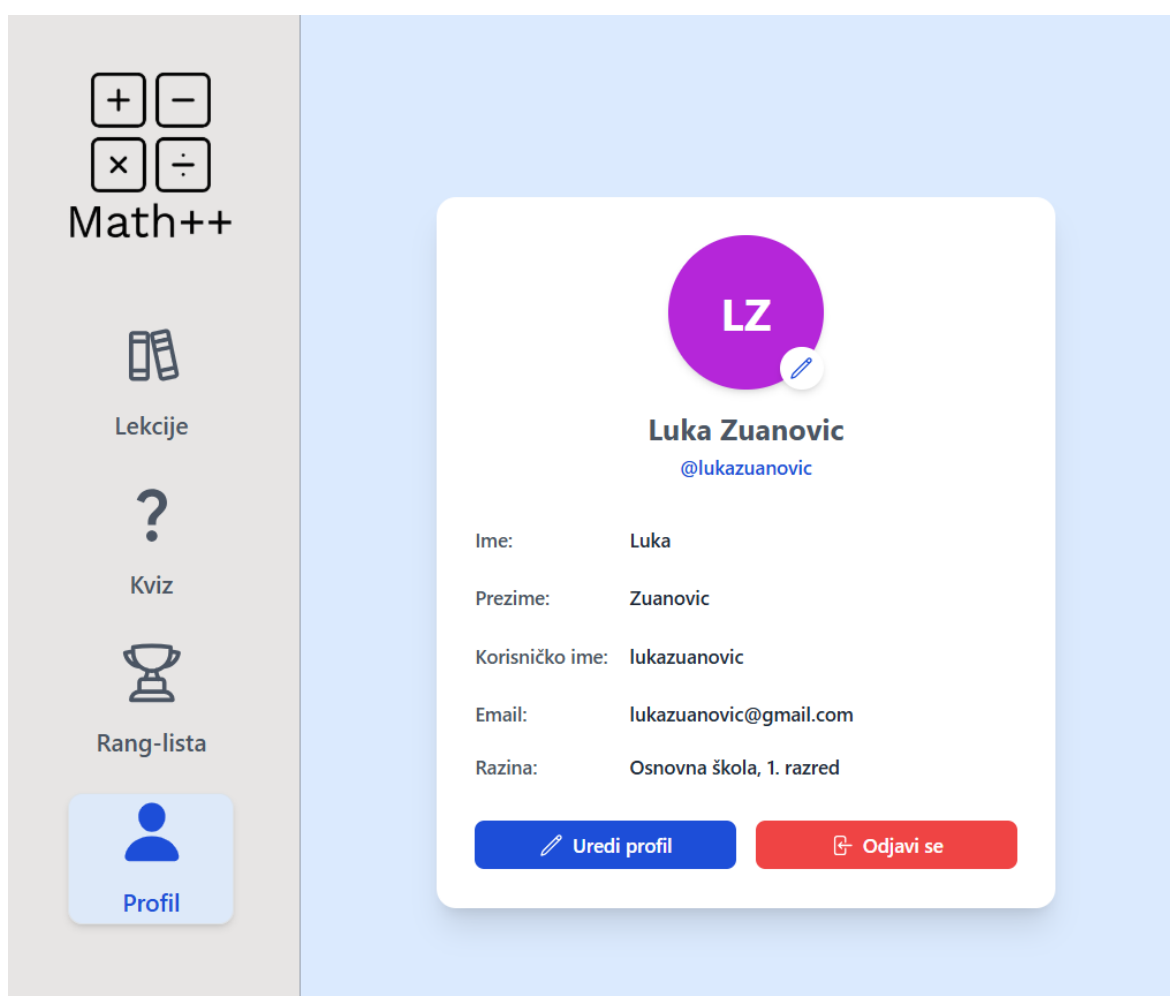
Odaberite svoj razred:

Već imate račun? [Prijavi se](#)

Sl. 6.2 Stranica za registraciju

6.3. Uređivanje profila

Korisnik na stranici za uređivanje profila, čiji je izgled prikazan na slici (Sl. 6.3), ima mogućnost izmjene osnovnih podataka, uključujući ime, prezime, korisničko ime i adresu e-pošte. Prilikom svake promjene provodi se validacija unesenih podataka, identična onoj tijekom registracije, s ciljem osiguranja ispravnosti i jedinstvenosti korisničkog imena i adrese e-pošte. Osim toga, korisnik može dodati novu profilnu sliku ako je ranije nije imao ili promijeniti postojeću. Na istoj stranici dostupna je i opcija odjave iz sustava, pri čemu se token briše iz memorije preglednika.

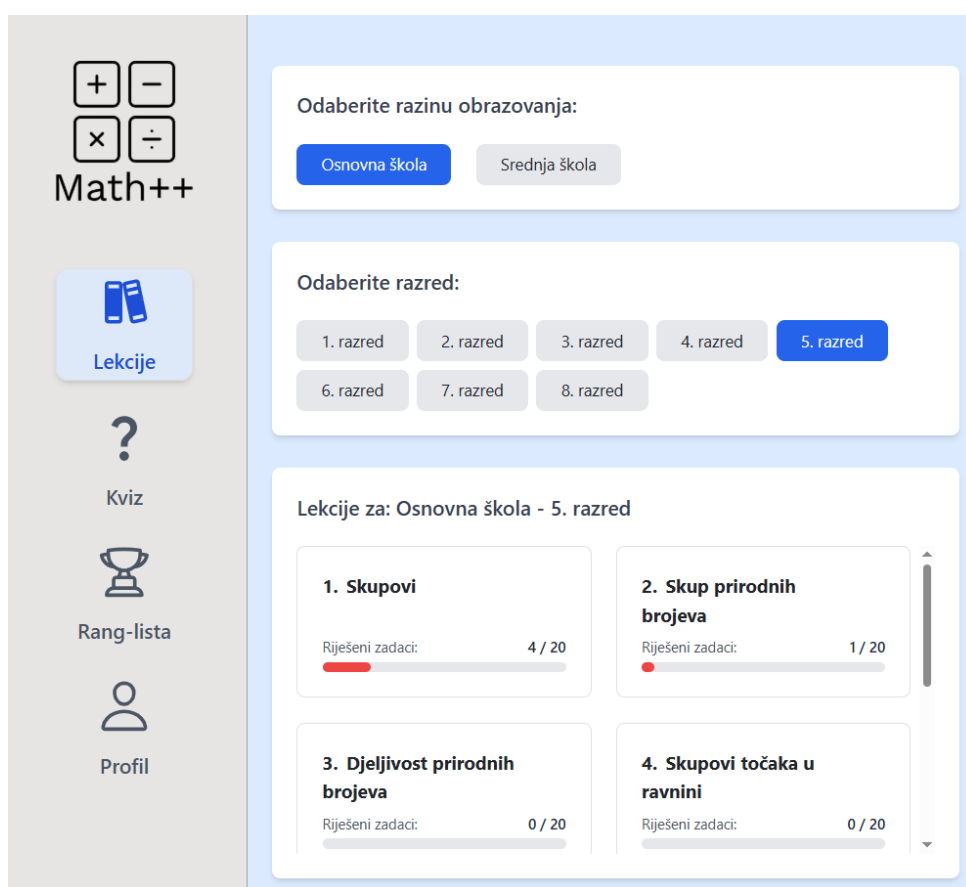


Sl. 6.3 Stranica za uređivanje profila

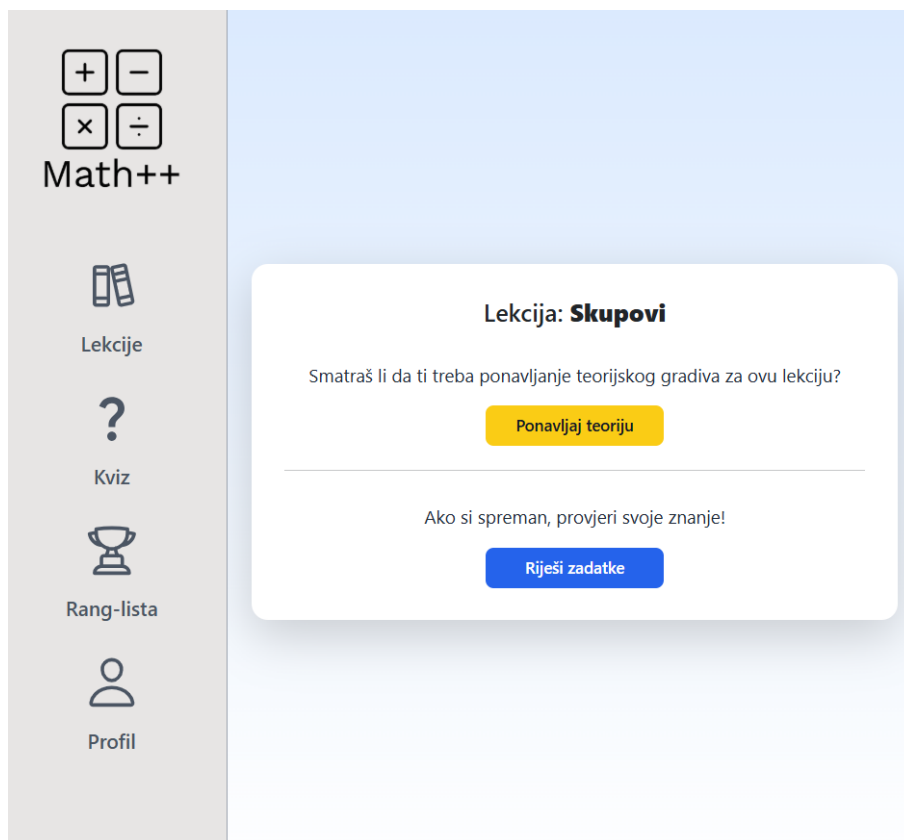
6.4. Nastavne cjeline

Nakon prijave korisniku su dostupne sve nastavne cjeline iz matematike, neovisno o razini obrazovanja (osnovna ili srednja škola), čime se omogućuje dodatno istraživanje i samostalno učenje. Prikaz dostupnih nastavnih cjelina vidljiv je na slici (Sl. 6.4). U popisu cjelina, na dnu svake cjeline prikazan je broj zadataka od ukupno dvadeset koje je korisnik označio kao naučene, što omogućuje uvid u vlastiti napredak.

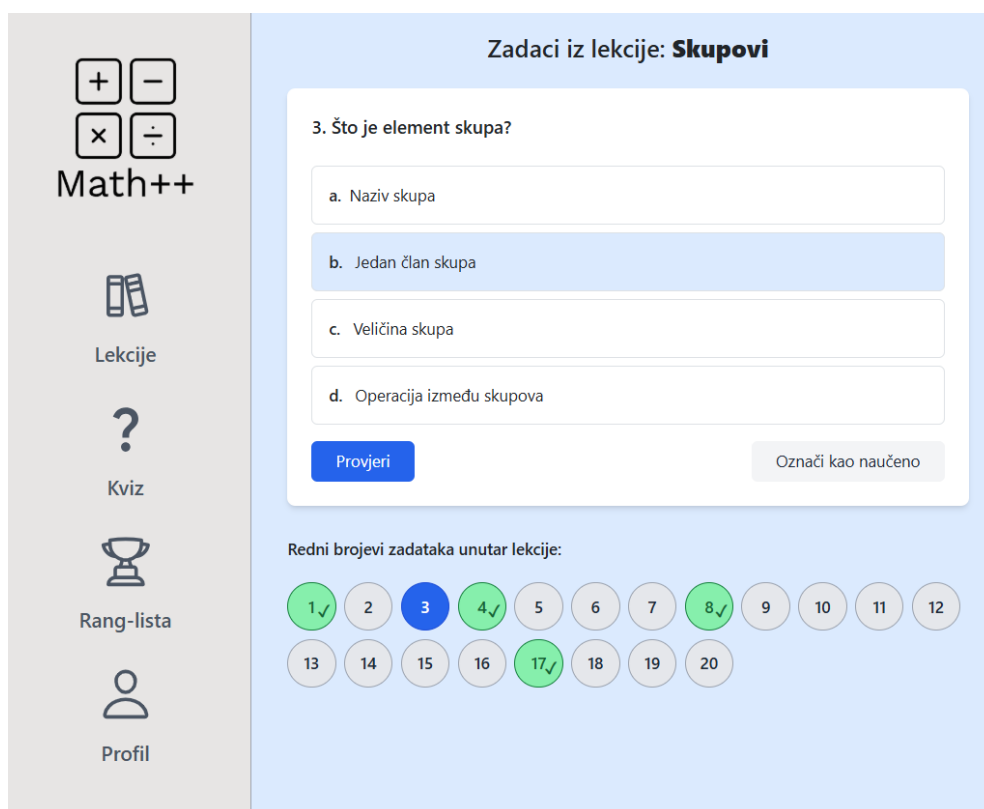
Odabirom određene nastavne cjeline, korisniku se prikazuje stranica s dvije opcije: gumb „Ponavljaj teoriju“ i gumb „Riješi zadatke“, što je vidljivo na slici (Sl. 6.5). Gumb „Ponavljaj teoriju“ preusmjerava korisnika na odgovarajuću cjelinu na vanjskoj edukativnoj platformi Edutorij [1], pružajući teorijsku pripremu prije rješavanja zadataka. Pritiskom na gumb „Riješi zadatke“ korisnik pristupa stranici za rješavanje zadataka prikazanoj na slici (Sl. 6.6). Zadaci se povlače iz baze podataka i predstavljeni su u obliku višestrukog izbora s četiri ponuđena odgovora. Korisnik ima na raspolaganju gumb „Provjeri“ za provjeru točnosti odgovora te gumb „Označi kao naučeno“ kojim može označiti zadatak kao naučen, čime se evidentira korisnikov napredak.



Sl. 6.4 Stranica s nastavnim cjelinama



Sl. 6.5 Stranica s izborom između ponavljanja teorije i rješavanja zadataka



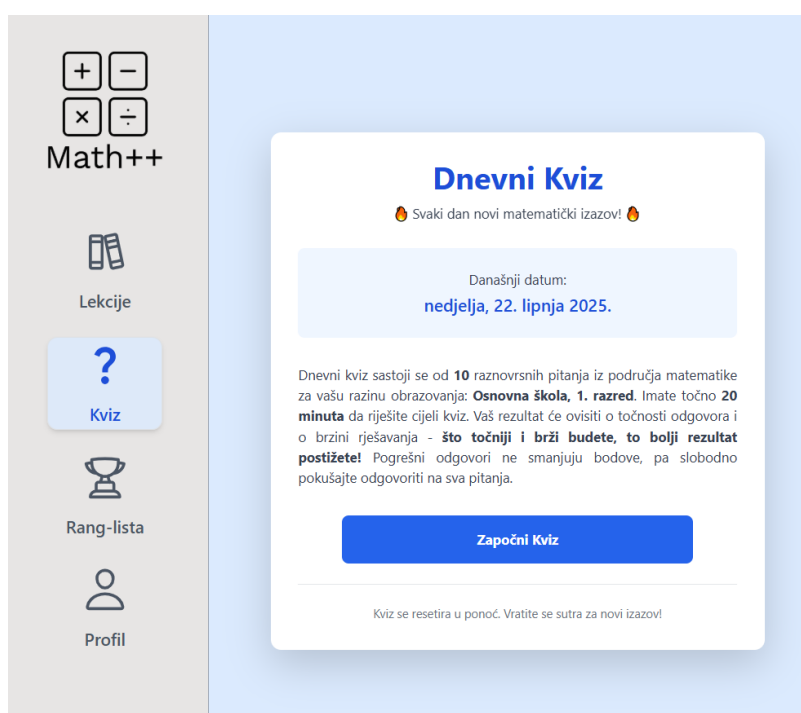
Sl. 6.6 Stranica za rješavanje zadataka

6.5. Kviz

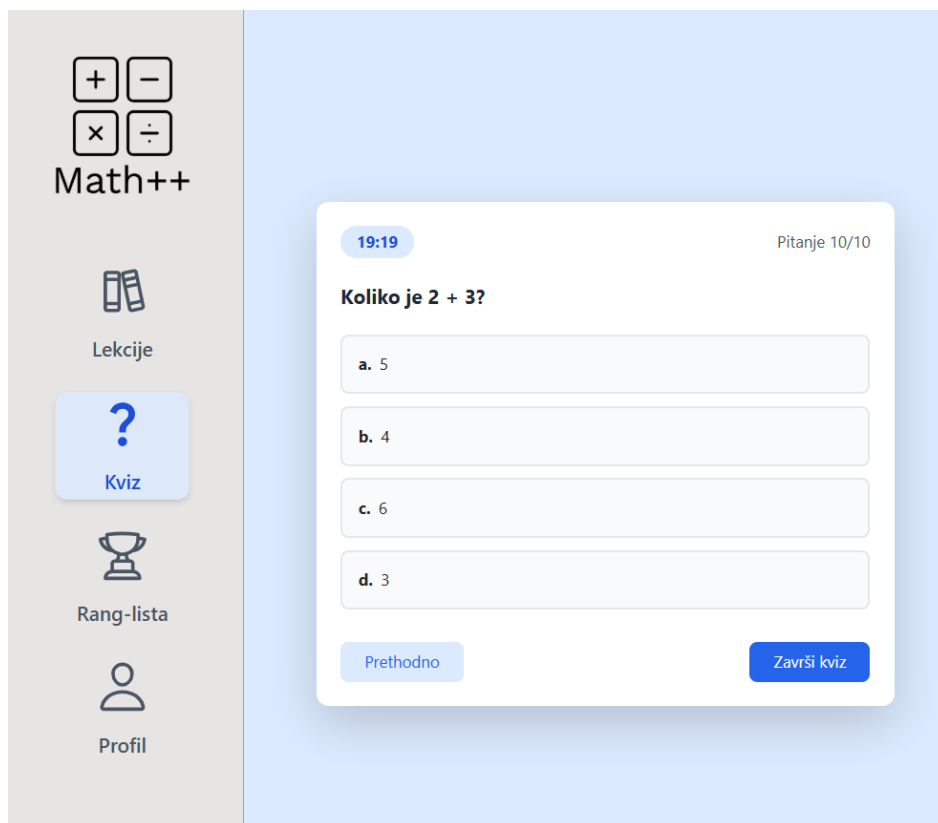
Aplikacija svakodnevno automatski generira kviz koji se sastoji od deset zadataka te je prilagođen korisnikovoj razini obrazovanja i pripadajućem razredu. Zadaci se generiraju korištenjem OpenAI API-ja, koji omogućuje stvaranje sadržaja temeljenog na naprednom jezičnom modelu *GPT-4o*. Detaljan opis postupka generiranja kvizova nalazi se u poglavlju 4.2.6 OpenAI API. Prije početka rješavanja kviza korisniku se prikazuje stranica s osnovnim informacijama o kvizu te gumb „Započni kviz“, čijim pritiskom započinje rješavanje, što je prikazano na slici (Sl. 6.7).

Nakon što je korisnik započeo kviz prikazuje mu se sučelje za rješavanje zadataka. Svako pitanje nudi četiri odgovora. Za rješavanje kviza predviđeno je maksimalno 20 minuta, a preostalo vrijeme kontinuirano se prikazuje u gornjem lijevom kutu. Kada korisnik dođe do posljednjeg pitanja, dostupna mu je opcija „Završi kviz“, kojom se odgovori šalju na poslužitelj (engl. *backend*) – prikaz ove stranice dan je na slici (Sl. 6.8).

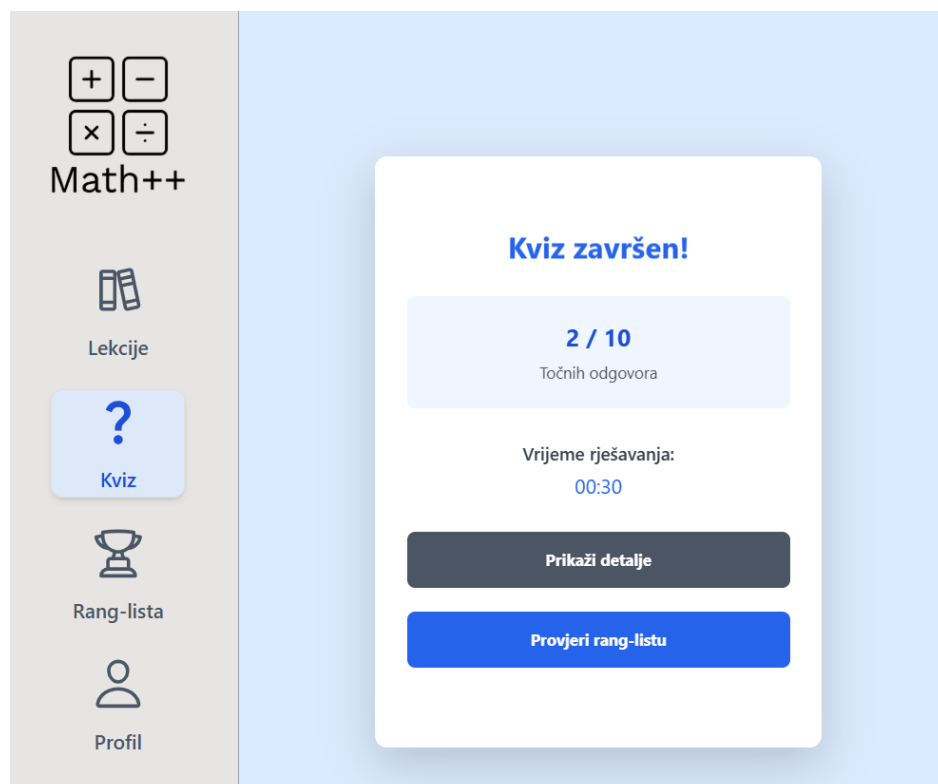
Po završetku kviza korisnika se preusmjerava na stranicu s rezultatima, gdje je prikazano ukupno potrošeno vrijeme te broj točno riješenih zadataka, vidljivo na slici (Sl. 6.9). Uz osnovni pregled rezultata dostupna su i dva dodatna gumba: „Prikaži detalje“, koji otvara analizu svih pojedinačnih pitanja, uključujući točan odgovor i korisnikov odabir te gumb „Provjeri rang-listu“, kojim korisnik pristupa stranici s dnevnom rang-listom.



Sl. 6.7 Stranica s osnovnim informacijama o kvizu



Sl. 6.8 Prikaz posljednjeg pitanja kviza



Sl. 6.9 Stranica s rezultatima kviza

6.6. Rang lista

Na temelju rezultata ostvarenih u dnevnim kvizovima, korisnicima se prikazuje dnevna rang-lista s dvadeset najuspješnijih učenika, čime se potiče natjecateljski duh i dodatna motivacija za redovito učenje. Rangiranje se prvenstveno temelji na točnosti odgovora, dok je vrijeme rješavanja sekundarni kriterij. U slučajevima kada više korisnika ostvari jednak broj bodova i isto vrijeme, oni dijele isto mjesto na listi.

Korisniku se prikazuje isključivo rang-lista koja obuhvaća korisnike istog stupnja obrazovanja i razreda, čime se osigurava relevantna usporedba unutar odgovarajuće razine. Prikaz stranice s rang-listom dan je na slici (Sl. 6.10).

Dnevna Rang Lista
"Poredak 20 najboljih učenika na današnjem kvizu"

Kriteriji rangiranja:
1. Točnost odgovora (broj bodova)
2. Brzina rješavanja (kraće vrijeme = bolji rang)
3. Rangiranje se vrši unutar istog razreda (Osnovna škola - 1. razred)
U slučaju istog broja bodova i vremena, korisnici dijele mjesto

Osnovna škola - 1. razred 22. lipnja 2025.

		lukazuanovic Rezultat: 2 / 10	Vrijeme: 0:30
		marko12345 Rezultat: 1 / 10	Vrijeme: 0:10

Već ste završili današnji kviz!
[Prikaži moj rezultat na rang listi.](#)

Sl. 6.10 Prikaz dnevne rang-liste

7. Potencijalna unaprjeđenja

Iako web-aplikacija *Math++* u svom trenutačnom izdanju već nudi kvalitetno i funkcionalno okruženje za učenje matematike, postoje brojni načini na koje bi se sustav mogao dodatno unaprijediti kako bi se poboljšalo korisničko iskustvo, proširile mogućnosti sustava i povećala skalabilnost. U nastavku su navedena neka od najvažnijih potencijalnih unaprjeđenja.

7.1. Napredna analitika

Uvođenjem dublje analitike rezultata (npr. prepoznavanje tipičnih pogrešaka na kvizovima) moguće je dodatno prilagoditi sadržaj korisnicima. Sustav bi mogao detektirati slabosti i prema tome istaknuti lekcije koje bi bilo korisno ponovo proći i ponoviti, čime bi se povećala učinkovitost učenja.

7.2. Pristup sadržajima bez internetske veze

Dodavanjem mogućnosti preuzimanja lekcija i zadataka u PDF formatu te njihovog rješavanja u okruženju bez internetske veze (engl. *offline*), zadaci bi postali dostupni i korisnicima koji nemaju stalan pristup internetu, kao i onima koji žele sadržaje pohraniti lokalno na svojim računalima ili mobilnim uređajima.

7.3. Višejezična podrška

Uvođenjem višejezične podrške aplikacija bi postala dostupna širem krugu korisnika te bi se u budućnosti mogla koristiti i izvan hrvatskog govornog područja. Za početak bi bilo korisno omogućiti učenicima korištenje aplikacije na engleskom jeziku.

7.4. Administratorsko sučelje i uređivanje sadržaja

Razvoj posebnog administratorskog sučelja omogućio bi lakše upravljanje zadacima, lekcijama, korisnicima i analizom sustava. Trenutačno se svi sadržaji moraju dodavati izravno kroz bazu podataka, a dodatno sučelje omogućilo bi lakšu prilagodbu nastavnicima ili sistemskim administratorima.

7.5. Mobilna aplikacija

Razvoj mobilne verzije aplikacije za operacijske sustave Android i Apple iOS omogućio bi pristup zadacima i kvizovima s bilo kojeg mjesta, u bilo kojem trenutku. Korištenjem istog poslužiteljskog sustava (engl. *backend*) moguće je proširiti postojeće funkcionalnosti na mobilne uređaje bez dupliciranja logike.

Zaključak

Web-aplikacija *Math++* rezultat je detaljno promišljenog i tehnički utemeljenog pristupa rješavanju jednog od najvažnijih izazova suvremenog obrazovanja – kako učenicima učiniti učenje matematike zanimljivijim, učinkovitijim i dostupnijim. Aplikacija objedinjuje teorijsko znanje, praktične zadatke i elemente motivacije kao što su dnevni kvizovi i rang-liste, čime se značajno podiže angažiranost korisnika i osigurava kontinuitet u učenju.

U razvoju *Math++* aplikacije korišten je širok spektar suvremenih tehnologija koje zajedno čine stabilan, skalabilan i siguran sustav. Na klijentskoj strani korištene su biblioteke i okviri React, Next.js, Chakra UI, Tailwind CSS, React Query, Framer Motion i Zod. Na poslužiteljskoj strani primijenjen je Spring Boot uz module Spring Security, Spring Data JPA i Jakarta Validation. Za autentifikaciju korisnika korišten je JSON Web Token (JWT) sustav. Kao baza podataka korišten je Supabase, moderna platforma temeljena na PostgreSQL-u. Detalji o svakoj od navedenih tehnologija i biblioteka opisani su u prethodnim poglavljima rada.

Arhitektonski obrazac MVC (engl. *Model–View–Controller*) doprinio je boljoj organizaciji aplikacije i njezinoj održivosti, omogućujući jasnu podjelu između slojeva aplikacije. Korištenjem REST API komunikacije između klijenta i poslužitelja omogućena je fleksibilnost u dohvaćanju i prikazu podataka, kao i priprema za buduće nadogradnje, uključujući razvoj mobilnih aplikacija i administracijskih sučelja.

Poseban naglasak stavljen je na prilagodbu hrvatskom obrazovnom sustavu, čime *Math++* nadopunjuje praznine koje ostavljaju postojeća globalna rješenja poput Khan Academyja, Photomatha i Matifica. Osim same funkcionalnosti, projekt je poslužio kao primjer uspješne primjene teorijskih znanja stečenih tijekom studija – od razvoja softverskih rješenja i upravljanja bazama podataka, do implementacije sigurnosnih mehanizama, oblikovanja korisničkog sučelja i organizacije projektnog koda.

U budućnosti, aplikacija *Math++* ima značajan potencijal za daljnji razvoj. Navedena su proširenja kao što su napredna analitika korisničkih rezultata, razvoj mobilne aplikacije za operacijske sustave Android i Apple iOS, podrška za više jezika, kao i sučelje za administraciju i uređivanje sadržaja. Time bi se *Math++* mogao dodatno pozicionirati kao sveobuhvatno rješenje za digitalno učenje matematike, ne samo u Hrvatskoj nego i izvan njezinih granica.

Literatura

- [1] *Edutorij*, Nacionalni repozitorij otvorenih obrazovnih sadržaja. Poveznica: <https://edutorij.e-skole.hr>; pristupljeno 16. lipnja 2025.
- [2] *Khan Academy*, Khan Academy. Poveznica: <https://www.khanacademy.org/>; pristupljeno 11. lipnja 2025.
- [3] *Photomath*, Photomath. Poveznica: <https://www.photomath.com>; pristupljeno 12. lipnja 2025.
- [4] *Matific*, Matific. Poveznica: <https://www.matific.com>; pristupljeno 12. lipnja 2025.
- [5] *Supabase documentation*, Supabase. Poveznica: <https://supabase.com/docs>; pristupljeno 16. lipnja 2025.
- [6] *React* – A JavaScript library for building user interfaces. Poveznica: <https://react.dev/>; pristupljeno 16. lipnja 2025.
- [7] *TypeScript* – TypeScript Documentation. Poveznica: <https://www.typescriptlang.org/>; pristupljeno 16. lipnja 2025.
- [8] *Spring Boot* – Spring Boot Reference Documentation. Poveznica: <https://spring.io/projects/spring-boot>; pristupljeno 16. lipnja 2025.
- [9] *Supabase Storage* – Store and serve any type of digital content. Poveznica: <https://supabase.com/docs/guides/storage>; pristupljeno 16. lipnja 2025.
- [10] *MVC Framework – Introduction*, GeeksforGeeks. Poveznica: <https://www.geeksforgeeks.org/software-engineering/mvc-framework-introduction/>; pristupljeno 16. lipnja 2025.
- [11] *Next.js* – The React Framework for Production. Poveznica: <https://nextjs.org/>; pristupljeno 16. lipnja 2025.
- [12] *Chakra UI* – Simple, Modular and Accessible UI Components. Poveznica: <https://chakra-ui.com/>; pristupljeno 16. lipnja 2025.
- [13] *TanStack Query (React Query)* – Powerful asynchronous state management for TS/JS. Poveznica: <https://tanstack.com/query/latest>; pristupljeno 16. lipnja 2025.
- [14] *Framer Motion* – A production-ready motion library for React. Poveznica: <https://www.framer.com/motion/>; pristupljeno 16. lipnja 2025.
- [15] *Zod* – TypeScript-first schema validation library. Poveznica: <https://zod.dev>; pristupljeno 16. lipnja 2025.
- [16] *Tailwind CSS* – Rapidly build modern websites without ever leaving your HTML. Poveznica: <https://tailwindcss.com>; pristupljeno 16. lipnja 2025.
- [17] *Spring Security Reference* – Securing Spring Applications. Poveznica: <https://spring.io/projects/spring-security>; pristupljeno 16. lipnja 2025.

- [18] *Spring Data JPA* – Reference Documentation. Poveznica: <https://spring.io/projects/spring-data-jpa>; pristupljeno 16. lipnja 2025.
- [19] *Jakarta Bean Validation* – Specification. Poveznica: <https://beanvalidation.org/>; pristupljeno 16. lipnja 2025.
- [20] *JSON Web Tokens* – Introduction to JWT. Poveznica: <https://jwt.io/introduction>; pristupljeno 16. lipnja 2025.
- [21] *Row Level Security (RLS)* – Supabase Auth Guides. Poveznica: <https://supabase.com/docs/guides/auth/row-level-security>; pristupljeno 16. lipnja 2025.

Sažetak

Aplikacija za interaktivno učenje matematike za osnovnu i srednju školu

U ovom radu prikazan je proces razvoja web-aplikacije *Math++*, namijenjene učenicima osnovnih i srednjih škola za interaktivno učenje matematike. Aplikacija omogućuje pristup zadacima iz svih nastavnih cjelina, dnevnim kvizovima generiranim putem umjetne inteligencije, praćenju napretka te prikazu rang-liste koja dodatno potiče natjecateljski duh među korisnicima. Razvijena je korištenjem modernih tehnologija poput Reacta, Next.js-a i Spring Boota. Naglasak je stavljen na prilagodbu hrvatskom obrazovnom sustavu i poticanje motivacije kroz elemente igre. Provedena je usporedba s postojećim rješenjima kao što su Khan Academy, Photomath i Matific, pri čemu je napravljena analiza prednosti i nedostataka svakog od njih. Rad obuhvaća funkcionalne zahtjeve aplikacije, arhitekturu sustava, korištene tehnologije, bazu podataka te mogućnosti budućeg razvoja i unaprjeđenja.

Ključne riječi: web-aplikacija, učenje matematike, React, Next.js, Spring Boot, Supabase, kviz, rang-lista, kurikulum

Summary

Application for interactive mathematics learning for primary and secondary school

This paper presents the development of the web-application *Math++*, designed for primary and secondary school students to support interactive learning of mathematics. The application provides access to exercises from all curriculum units, daily quizzes generated by artificial intelligence, progress tracking, and a ranking list that further encourages a competitive spirit among users. It was developed using modern technologies such as React, Next.js, and Spring Boot. The emphasis was placed on alignment with the Croatian educational curriculum and on enhancing motivation through gamification elements. A comparison was conducted with existing solutions such as Khan Academy, Photomath, and Matific, with an analysis of the advantages and limitations of each. The paper covers the functional requirements of the application, system architecture, used technologies, database design, and possibilities for future development and improvements.

Keywords: web-application, mathematics learning, React, Next.js, Spring Boot, Supabase, quiz, ranking list, curriculum