

# Sustav za učenje pomoću interaktivnih geografskih karata

---

Đurin, Matej

Master's thesis / Diplomski 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:168:416624>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-04-01**



*Repository / Repozitorij:*

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 610

**SUSTAV ZA UČENJE POMOĆU INTERAKTIVNIH  
GEOGRAFSKIH KARATA**

Matej Đurin

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 610

**SUSTAV ZA UČENJE POMOĆU INTERAKTIVNIH  
GEOGRAFSKIH KARATA**

Matej Đurin

Zagreb, lipanj 2024.

## DIPLOMSKI ZADATAK br. 610

Pristupnik: **Matej Đurin (0036526130)**  
Studij: Računarstvo  
Profil: Programsko inženjerstvo i informacijski sustavi  
Mentor: doc. dr. sc. Tomislav Jaguš

Zadatak: **Sustav za učenje pomoću interaktivnih geografskih karata**

### Opis zadatka:

Digitalni urođenici su osobe koje su odrasle uz stalnu prisutnost digitalnih tehnologija, kao što su pametni telefoni, računala i internet. Iz tog razloga primjena digitalnih alata nameće se kao logičan odabir u osmišljavanju uspješnih strategija u obrazovanju današnjih učenika, naravno uz obraćanje pažnje na pedagošku vrijednost upotrebe tehnologije, prilagodljivost sustava učenicima te održavanja ravnoteže između digitalnog i tradicionalnog učenja. U sklopu diplomskog rada potrebno je detaljno istražiti područje tehnologijom podržanog obrazovanja te osmisliti i razviti sustav za učenje pomoću interaktivnih geografskih karata, primjenjiv na različite predmete, potičući tako učenike na istraživanje i razumijevanje šireg konteksta gradiva koje uče. Potrebno je proučiti dostupne biblioteke za stvaranje interaktivnih karata, strukturu GeoJSON podataka, načine označavanja geografskih entiteta te slične obrazovne aplikacije s ciljem ostvarivanja što kvalitetnijeg vlastitog rješenja. Potrebno je osmisliti nekoliko vježbi prilagođenih različitim uzrastima, predmetima i razinama znanja. Sustav je potrebno testirati u praksi. Sustav treba prikupljati i pohranjivati podatke o uspješnosti rješavanja vježbi i pripadnih zadataka, radi naknadne analize.

Rok za predaju rada: 28. lipnja 2024.

*Zahvaljujem svim poznanicima, kolegama i prijateljima s kojima sam dijelio studentske dane. Zahvaljujem svojoj obitelji i djevojci Bernardi na pozitivnom stavu, podršci i vjeri u mene. Posebno zahvaljujem mentoru doc. dr. sc. Tomislavu Jaguštu na mentoriranju kroz čitav studij i nesebičnom dijeljenju znanja i ideja.*

# Sadržaj

|                                                                    |    |
|--------------------------------------------------------------------|----|
| Uvod.....                                                          | 1  |
| 1. Pregled područja .....                                          | 2  |
| 1.1. Tehnologijom podržano obrazovanje.....                        | 2  |
| 1.1.1. Načini na koje tehnologija podržava obrazovanje .....       | 3  |
| 1.1.2. Nedostaci i izazovi .....                                   | 4  |
| 1.2. Biblioteke za stvaranje interaktivnih geografskih karata..... | 8  |
| 1.2.1. Leaflet .....                                               | 8  |
| 1.2.2. OpenLayers .....                                            | 10 |
| 1.3. GeoJSON .....                                                 | 11 |
| 1.4. Pregled postojećih rješenja.....                              | 14 |
| 1.4.1. Seterra Geography.....                                      | 14 |
| 1.4.2. World Geography Games.....                                  | 17 |
| 2. Arhitektura sustava .....                                       | 19 |
| 2.1. Klijent .....                                                 | 19 |
| 2.2. Poslužitelj.....                                              | 20 |
| 2.2.1. API sloj .....                                              | 21 |
| 2.2.2. Poslovni sloj.....                                          | 23 |
| 2.2.3. Podatkovni sloj.....                                        | 24 |
| 2.3. Baza podataka .....                                           | 25 |
| 3. Model podataka.....                                             | 26 |
| 3.1. Opis relacija .....                                           | 27 |
| 4. Korištene tehnologije.....                                      | 36 |
| 4.1. Angular .....                                                 | 36 |

|        |                                    |    |
|--------|------------------------------------|----|
| 4.2.   | .NET.....                          | 36 |
| 4.3.   | SQL Server.....                    | 37 |
| 4.4.   | Microsoft Azure.....               | 37 |
| 5.     | Opis rada sustava .....            | 39 |
| 5.1.   | Korisnički procesi .....           | 39 |
| 5.1.1. | Registracija.....                  | 39 |
| 5.1.2. | Prijava .....                      | 41 |
| 5.1.3. | Odjava .....                       | 41 |
| 5.2.   | Učitelj.....                       | 42 |
| 5.2.1. | Upravljanje predmetima.....        | 42 |
| 5.2.2. | Upravljanje skicama ispita.....    | 44 |
| 5.2.3. | Upravljanje zadacima.....          | 46 |
| 5.2.4. | Tipovi zadataka.....               | 48 |
| 5.2.5. | Upravljanje ispitima.....          | 51 |
| 5.2.6. | Pregled i ispravljanje ispita..... | 53 |
| 5.3.   | Učenik.....                        | 54 |
| 5.3.1. | Rješavanje ispita .....            | 54 |
| 5.3.2. | Pregled rezultata ispita.....      | 56 |
| 6.     | Eksperiment i rezultati .....      | 58 |
|        | Zaključak.....                     | 61 |
|        | Literatura.....                    | 62 |
|        | Sažetak.....                       | 65 |
|        | Summary.....                       | 66 |

# Uvod

Sintagma „digitalni urođenici“, koju je skovao američki tehnolog Marc Prensky 2001. godine, odnosi se na osobe koje su rođene i odrasle uz stalnu prisutnost digitalnih tehnologija poput računala, pametnih telefona i interneta. Okruženost spomenutim digitalnim tehnologijama uvelike oblikuje njihov način razmišljanja, komuniciranja i učenja. Zbog toga se primjena digitalnih alata nameće kao logičan odabir u osmišljavanju uspješnih strategija u obrazovanju današnjih učenika, naravno uz obraćanje pažnje na pedagošku vrijednost upotrebe tehnologije, prilagodljivost sustava učenicima te održavanja ravnoteže između digitalnog i tradicionalnog učenja.

Cilj ovog diplomskog rada je istražiti područje tehnologijom podržanog učenja, s naglaskom na mogućnosti upotrebe geografskih karata u obrazovanju (npr. u predmetima kao što su geografija ili povijest) te na temelju tog istraživanja razviti sustav za učenje s pomoću interaktivnih geografskih karata koji je primjenjiv na različite predmete. Takav sustav može značajno doprinijeti učenju, potičući učenike na istraživanje i razumijevanje šireg konteksta gradiva koje uče.

Na početku diplomskog rada istraženo je područje tehnologijom podržanog obrazovanja, poznato kao e-učenje. Ovo područje obuhvaća različite metode i alate koji se koriste za unapređenje obrazovnog procesa putem digitalnih tehnologija. Kako bi razvijeni sustav bio što korisniji, prethodno su temeljito istražena i analizirana slična postojeća rješenja. Cilj je bio identificirati njihove prednosti i implementirati ih u sustav, dok se istovremeno pokušalo ispraviti njihove nedostatke. Proučene su dostupne biblioteke za stvaranje interaktivnih geografskih karata, struktura GeoJSON podataka i načini označavanja geografskih entiteta. Ova analiza imala je za cilj osigurati što kvalitetnije vlastito rješenje. Nakon istraživanja i analize, opisana je arhitektura implementiranog sustava, s korištenim tehnologijama. Zatim slijedi detaljan opis rada sustava, objašnjavajući sve ključne funkcionalnosti i njihove primjene. Na kraju diplomskog rada opisan je eksperiment koji je proveden s ciljem testiranja sustava u praksi te je dan zaključak diplomskog rada uz ideje daljnjeg razvoja i unaprjeđenja implementiranog sustava.



# 1. Pregled područja

U ovom poglavlju daje se sveobuhvatan pregled područja obuhvaćenih diplomskim radom. Najprije se razmatra obrazovanje podržano tehnologijom, na temelju analiziranih članaka i istraživanja na tu temu. Zatim slijedi pregled dostupnih biblioteka za izradu interaktivnih geografskih karata, analiza GeoJSON strukture podataka te pregled postojećih aplikacija za učenje s pomoću interaktivnih geografskih karata.

## 1.1. Tehnologijom podržano obrazovanje

Promjene u modernoj tehnologiji brže su nego ikada prije [1], a svijet se razvija i tehnološki napredak cvjeta. Današnji svijet funkcionira tako da neprestano prati te promjene, a obrazovni sektor nije iznimka, bilježeći ogroman rast u posljednjih nekoliko godina. Stare metodologije podučavanja više nisu relevantne jer samo teoretsko znanje nije dovoljno za cjelokupni razvoj učenika. Praktični načini i nova učenja sada se uključuju u suvremene nastavne procese. S napretkom tehnologije, njezina integracija s učenjem može biti korisna za cijelo društvo. Korištenje novih i inovativnih tehnologija može učiniti nastavu zanimljivijom za učenike, što povećava njihovu pažnju i motivaciju za učenje [1].

U kontekstu tehnologijom podržanog obrazovanja, često se ističe pojam e-učenje (eng. e-learning). E-učenje je sustav učenja koji se temelji na formalnoj nastavi, ali uz pomoć elektroničkih resursa [2]. Iako se nastava može odvijati u učionici ili izvan nje, korištenje računala i interneta čini glavnu komponentu e-učenja. E-učenje se može nazvati i mrežnim prijenosom vještina i znanja, a njihov prijenos se vrši velikom broju primatelja u isto ili različito vrijeme. Ranije se e-učenje nije uvelike prihvaćalo jer se pretpostavljalo da takvom sustavu nedostaje ljudski element potreban za učenje. Međutim, s brzim napretkom tehnologije i napretkom u sustavima učenja, sada ga prihvaćaju mase. Pojam „e-učenje“ postoji tek od 1999. godine kada je riječ prvi put korištena na seminaru kognitivno bihevioralne terapije (CBT). U potrazi za točnim opisom novog sustava učenja, počeli su se pojavljivati i drugi pojmovi kao što su „mrežno učenje“ (eng. online learning) i „virtualno učenje“ (eng. virtual learning) [3].

### 1.1.1. Načini na koje tehnologija podržava obrazovanje

U radu „Kako tehnologija podržava poučavanje i učenje?“ Pooja Pant istražuje temu tehnologijom podržanog poučavanja i učenja te opisuje nekoliko načina na koje tehnologija podržava poučavanje i učenje [1].

#### Adaptivnost u obrazovnom sustavu

Uključivanje kreativnih strategija učenja i poučavanja može podići standarde obrazovanja za učenike. Resursi dostupni na internetu otvaraju nove horizonte za edukacijsko-tehnološke industrije. Pametne učionice opremljene naprednim softverom i audiovizualnom tehnologijom čine obrazovanje integriranijim. Ovaj novi sustav učenja pomaže učenicima da lakše vizualiziraju i razumiju teške koncepte. Predmeti poput matematike lakše se objašnjavaju s pomoću 3D dijagrama i videa. Umjesto tradicionalnih metoda kao što su puko učenje, uvođenje modernih metoda poput vizualizacije i videa pomoći će učenicima da bolje shvate temu. Navedene moderne metode omogućuju dublje razumijevanje gradiva. Dakle, korištenje tehnologije u nastavi i učenju definitivno se može smatrati kao adaptivan/prilagodljiv sustav učenja, koji je bolji i napredniji od tradicionalne nastave u učionici.

#### Pristupačnost

Digitalizirano obrazovanje stvara prilike kako za učenike tako i za nastavnike. Sa širim pristupom online učenju, obrazovanje je sada dostupno svima, a učenici mogu pristupiti informacijama i resursima za učenje iz udobnosti svojih domova. Nastavnici više nisu vezani za četiri zida učionice kako bi podučavali svoje učenike. U doba digitalnog učenja, učenici imaju na raspolaganju platforme poput Zoom<sup>1</sup>-a i Teams<sup>2</sup>-a te tehnologije poput podcasta i e-knjiga koje im olakšavaju učenje. Kako obrazovanje postaje pristupačnije, smanjuje se jaz između obrazovanih i neobrazovanih slojeva društva.

---

<sup>1</sup> **Zoom** - platforma za video konferencije, može se koristiti putem računala ili mobilne aplikacije. Omogućuje korisnicima online povezivanje u svrhu video konferencijskih sastanaka, webinarima i razgovora uživo. Tijekom krize uzrokovane Covid19 virusom, Zoom je doživio veliki porast popularnosti [4].

<sup>2</sup> **Microsoft Teams** – aplikacija koja nudi radni prostor za suradnju i komunikaciju u stvarnom vremenu, sastanke te dijeljenje datoteka i aplikacija [5].

## **Dostupnost i praktičnost**

Učenici koji su iz bilo kojeg razloga fizički vezani za svoj dom ili koji nemaju pristup obrazovnim institucijama, sada se mogu osloniti na online učenje. Mnoge obrazovne institucije, uključujući i najviše razine sveučilišta, nude online diplome, čiji troškovi su usporedivo manji od troškova tradicionalnih diploma. Također, s više dostupnih i povoljnih resursa, obrazovanje sada može dosegnuti i udaljenija područja zemlje, a digitalni alati učinili su online nastavu interaktivnom i zanimljivom za učenike.

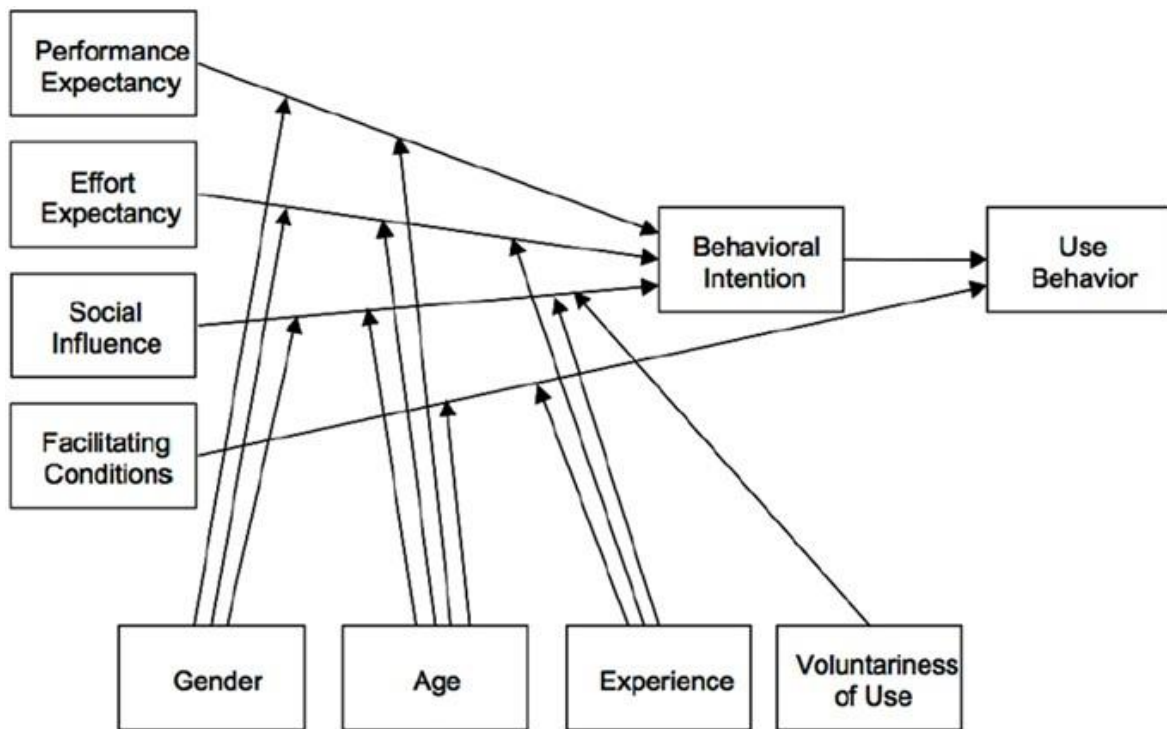
### **1.1.2. Nedostaci i izazovi**

Tehnologijom podržano obrazovanje uvelike je zaživjelo u razvijenim zemljama svijeta, omogućujući pristup obrazovnim resursima i alatima putem interneta i digitalnih tehnologija. Međutim, u zemljama u razvoju spomenute tehnologije još uvijek nisu dovoljno rasprostranjene.

Gulati [6] predstavlja pregled razvoja obrazovanja u pogledu otvorenog, udaljenog i tehnologijom podržanog učenja koje ima za cilj doseći obrazovno zakinite populacije svijeta. Gulati zaključuje da, iako razvoj otvorenog i udaljenog učenja u zemljama u razvoju teži pružanju jednakih i proširenih obrazovnih mogućnosti za ugrožene i siromašne populacije, nedostatak obrazovne i tehnološke infrastrukture, nedostatak obučениh nastavnika, negativni stavovi prema udaljenom učenju, društvena i kulturna ograničenja nametnuta djevojčicama i ženama te neprikladne odluke o politici i financiranju, doprinijele su produbljivanju jaza između bogatih i siromašnih, ruralnih i urbanih područja te između spolova. Autor je u radu identificirao značajne izazove s kojima se zemlje u razvoju susreću prilikom pokušaja da obrazovanje učine pristupačnijim korištenjem internetskih tehnologija za siromašniju populaciju. Autor rada ističe da, iako se udaljeno učenje i e-učenje smatraju jednostavno dostupnima, za ruralno siromašno stanovništvo u zemljama u razvoju nedostatak nastavnika, knjiga, učionica, novca i drugih resursa i dalje predstavlja značajan izazov. Gulati u svom radu [6] još zaključuje da u mnogim slučajevima gdje postoji ograničena IT infrastruktura, tradicionalne tehnologije poput tiskanog materijala, radija i televizije ostaju učinkovitije i pristupačnije za ruralne i ugrožene skupine.

U radu „Iskustvo nastavnika u implementaciji e-učenja u instituciji visokog obrazovanja Južnoafričke Republike“ [7] također su prepoznati i opisani neki od izazova integracije tehnologije u obrazovni sustav. Autori u radu [7] nastoje istražiti iskustva nastavnika s korištenjem e-učenja za podršku nastavi i učenju na jednom južnoafričkom sveučilištu. Teorija na kojoj se temeljila studija predstavljena u članku bila je Ujedinjena teorija prihvaćanja i upotrebe tehnologija (eng. Unified Theory of Acceptance and Use of Technology, UTAUT). UTAUT je poznat kao jedan od široko prihvaćenih teorijskih modela istraživanja prihvaćanja tehnologije [8]. Teorijski model UTAUT ispituje namjeru korisnika da koristi tehnološki sustav i naknadnu stvarnu upotrebu istog. UTAUT služi kao robusan osnovni model za ispitivanje prihvaćanja različitih tehnologija od strane korisnika u različitim društvenim i ekonomskim kontekstima. Razvio ga je istaknuti znanstvenik Viswanath Venkatesh kroz procjenu i konsolidaciju osam prethodno postojećih modela korištenja i prihvaćanja tehnologije [9]:

- Teorija promišljenog djelovanja
- Model prihvaćanja tehnologije
- Motivacijski model
- Teorija planiranog ponašanja
- Kombinirani model Modela prihvaćanja tehnologije i Teorije planiranog ponašanja
- Model korištenja osobnog računala
- Teorija difuzije inovacija
- Socijalno-kognitivna teorija



**Slika 1.1** Interakcija elemenata UTAUT modela [9]

Studija „Iskustvo nastavnika u implementaciji e-učenja u instituciji visokog obrazovanja Južnoafričke Republike“ [7] provedena je uz pomoć deset namjerno odabranih nastavnika i jednog IT stručnjaka na južnoafričkom sveučilištu. Podaci korišteni za odgovaranje na istraživačka pitanja prikupljeni su s pomoću polustrukturiranog intervjua. Razlog takvog tipa prikupljanja podataka bio je taj što su autori studije željeli čuti kako sudionici svojim riječima objašnjavaju svoja iskustva korištenja i prihvaćanja tehnologija e-učenja. Za implementaciju e-učenja u nastavi institucije, odabran je sustav za upravljanje učenjem (eng. Learning Management System, LMS) Moodle. Moodle je sustav za upravljanje učenjem (LMS) dizajniran kako bi edukatorima, administratorima i učenicima pružio jedinstven robustan, siguran i integriran sustav za stvaranje personaliziranih okruženja za učenje [10].

Nakon analize podataka prikupljenih iz polustrukturiranih intervjua sa sudionicima studije, utvrđena su dva glavna izazova nastavnika u implementaciji e-učenja unutar institucije [7]:

- tehnička podrška i obuka za e-učenje
- ICT infrastruktura i dostupnost interneta

## **Tehnička podrška i obuka za e-učenje**

Iako je sveučilište, u svrhu provedbe studije „Iskustvo nastavnika u implementaciji e-učenja u instituciji visokog obrazovanja Južnoafričke Republike“ [7], pružilo nastavnicima sudionicima studije obuku o korištenju sustava za upravljanje učenjem Moodle, oni su smatrali da je obuka bila nedovoljna. Sudionici studije naveli su kako je nedostajala kontinuirana tehnička podrška kada im je ona bila potrebna. U potporu toj tvrdnji, „Nastavnik B“ je izjavio: *„Neki predavači nisu sudjelovali ni na jednoj obuci i zbog toga im može nedostajati samopouzdanja za učinkovito korištenje Moodle-a. Na sveučilištu je puno nastavnika i stoga jedna ili dvije radionice za obuku možda nisu dovoljne. Sudjelovao sam na jednoj radionici i osjećam da još uvijek trebam naučiti kako raditi više različitih stvari s ciljem diversifikacije aktivnosti učenja na Moodle-u. Stoga je nužno kontinuirano održavati obuke kako bismo stekli više znanja i samopouzdanja u korištenju Moodle-a.“* [7]. Nastavnici sudionici studije vjeruju da e-učenje igra važnu ulogu u poboljšanju akademskih rezultata studenata. Međutim, istaknuli su da bi uspjeh studenata nedvojbeno porastao ako bi se njima kao predavačima pružala redovita tehnička podrška i obuka za korištenje tehnologija e-učenja.

## **ICT infrastruktura i dostupnost interneta**

Tehnološka revolucija omogućila je sveučilištima integraciju tehnologije u njihove obrazovne procese radi unapređenja nastave i učenja. Za postizanje tog cilja, povezanost je ključna. Wi-Fi povezanost će stoga osigurati da svaki student, bez obzira gdje se nalazi na sveučilištu, može pristupiti internetu. Rezultati studije „Iskustvo nastavnika u implementaciji e-učenja u instituciji visokog obrazovanja Južnoafričke Republike“ [7] pokazali su da je odabrana institucija implementirala Wi-Fi, no suočava se s izazovom ispunjavanja očekivanja studenata i osoblja za pouzdanom i brzom bežičnom povezanosti širom institucije. „IT stručnjak A“ izjavio je sljedeće: *„Dostupnost Wi-Fi-a po instituciji se povećala tijekom godina, posebno u uredskim prostorima i dvoranama za predavanja. Međutim, još uvijek postoji velika potreba za više hot spotova jer neka područja još uvijek nemaju pristup Wi-Fi-u.“* [7]. Spomenuta ograničenja značajno otežavaju implementaciju tehnologija e-učenja u odabranoj instituciji.

## 1.2. Biblioteke za stvaranje interaktivnih geografskih karata

Kako bi sustav za učenje s pomoću interaktivnih geografskih karata razvijen u sklopu ovog diplomskog rada bio što intuitivniji njegovim korisnicima i nudio što više funkcionalnosti, ključno je prethodno istražiti dostupne biblioteke za stvaranje interaktivnih geografskih karata te odabrati onu koja najbolje odgovara zahtjevima sustava. Važno je detaljno istražiti svaku opciju, uzimajući u obzir čimbenike poput jednostavnosti korištenja, mogućnosti prilagodbe, podrške za različite vrste kartografskih podataka i načina njihovog označavanja na karti te kompatibilnosti s drugim tehnologijama koje će se koristiti u implementaciji sustava. U nastavku slijedi pregled i analiza dviju biblioteka za stvaranje interaktivnih geografskih karata, kako bi se identificirala i odabrala najbolja opcija za implementaciju sustava za učenje.

### 1.2.1. Leaflet

Leaflet je vodeća JavaScript biblioteka otvorenog koda za interaktivne karte prilagođene mobilnim uređajima. Podržavaju ga neke od vodećih svjetskih kompanija poput GitHub-a i Meta-e. Prvi je put objavljen 2011. godine [11]. Budući da je riječ o projektu otvorenog koda, članovi zajednice su dobrodošli da doprinesu razvoju repozitorija i učine ga još većim nego što već jest [12]. Leaflet je dizajniran imajući na umu jednostavnost, učinkovitost i upotrebljivost. Učinkovito radi na svim većim platformama za stolna računala i mobilne uređaje. Također, proširiv je s mnoštvom dodataka (eng. plugin), ima dobro dokumentiran API i lako čitljiv izvorni kod [11].

Leaflet izvorno podržava slojeve WMS<sup>3</sup>-a (eng. Web Map Service), GeoJSON<sup>4</sup> slojeve, vektorske<sup>5</sup> slojeve i slojeve pločica (eng. tile layer). Mnoge druge vrste slojeva

---

<sup>3</sup> **WMS** - standardni protokol koji opisuje kako poslužiti bilo koju georeferenciranu kartografsku sliku putem interneta, a koju obično generira kartografski poslužitelj koristeći podatke iz baze podataka geografskog informacijskog sustava (GIS). Standard protokola razvio je Open Geospatial Consortium (OGC) i prvi je put objavljen 1999. godine [13].

<sup>4</sup> **GeoJSON** - format za kodiranje raznih struktura geografskih podataka [14]. Detaljnije o GeoJSON-u pisano je u poglavlju 1.3.

<sup>5</sup> **Vektorska grafika** - oblik računalne grafike u kojoj se vizualne slike stvaraju izravno iz geometrijskih oblika definiranih na kartezijskoj ravnini, poput točke, linije, krivulje i poligona [15].

podržane su putem dodataka. Na slici 1.2 prikazan je JavaScript programski kod stvaranja karte pomoću Leaflet biblioteke.

```
var map = L.map('map').setView([51.505, -0.09], 13);
L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {
  maxZoom: 19,
  attribution:
    '&copy; <a href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>',
}).addTo(map);

var marker = L.marker([51.5, -0.09]).addTo(map);
var circle = L.circle([51.508, -0.11], {
  color: 'red',
  fillColor: '#f03',
  fillOpacity: 0.5,
  radius: 500,
}).addTo(map);
var polygon = L.polygon([
  [51.509, -0.08],
  [51.503, -0.06],
  [51.51, -0.047],
]).addTo(map);

marker.bindPopup('<b>Hello world!</b><br>I am a popup.').openPopup();
circle.bindPopup('I am a circle.');
```

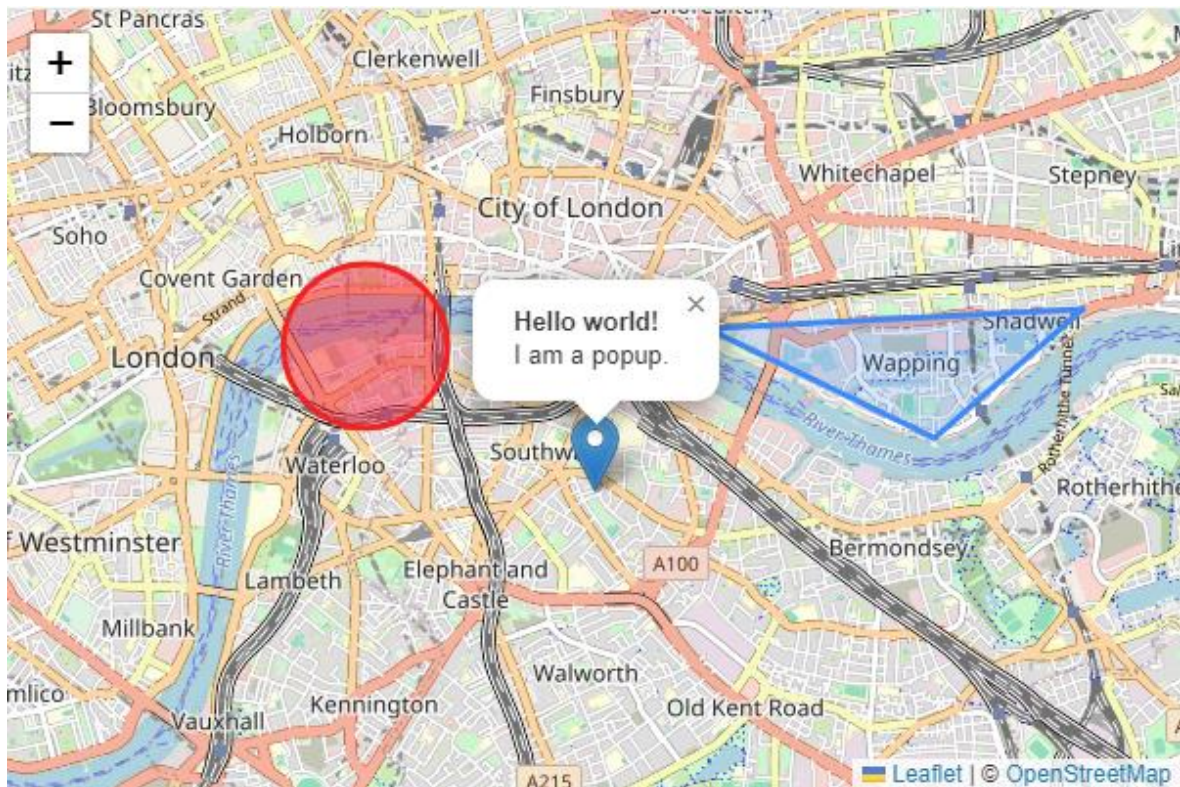
Slika 1.2 Stvaranje karte korištenjem Leaflet biblioteke [16]

Promatrani programski kod inicijalizira kartu, postavlja joj pogled na željene geografske koordinate te odabire razinu zumiranja. Zatim dodaje sloj pločica dobivenih s OpenStreetMap<sup>6</sup>-a. Osim slojeva pločica, programski kod na kartu dodaje i po jednu oznaku (eng. marker), krug i poligon. Svakom od tih objekata dodan je pripadni skočni prozor s tekstualnim ili HTML sadržajem, koji se otvara klikom na objekt na karti. Rezultat promatranog programskog koda prikazan je na slici 1.3.

---

<sup>6</sup> **OpenStreetMap** - besplatna karta cijelog svijeta koja se može uređivati. OpenStreetMap pokrenut je 2004. godine u Ujedinjenom Kraljevstvu. Uključuje podatke o cestama, zgradama, adresama, trgovinama i poduzećima, točkama interesa, željeznicama, stazama, prijevozu, korištenju zemljišta, prirodnim obilježjima i još mnogo toga [17].





Slika 1.3 Karta dobivena pomoću programskog koda sa slike 1.2 [16]

## 1.2.2. OpenLayers

OpenLayers je visokoučinkovita JavaScript biblioteka prepuna značajki za izradu interaktivnih karata na webu [18]. Može prikazivati pločice karte, vektorske podatke i oznake učitane iz bilo kojeg izvora na bilo kojoj web stranici. OpenLayers je JavaScript biblioteka otvorenog koda, objavljena pod BSD licencom s dvije klauzule također poznata kao FreeBSD<sup>7</sup>.

Sposobnosti i značajke OpenLayers biblioteke ne razlikuju se uvelike od prethodno promatranih značajki Leaflet biblioteke. OpenLayers može dohvatiti pločice geografske karte iz OpenStreetMap-a, Bing-a, MapBox-a, Stadia Maps-a ili bilo kojeg drugog izvora

---

<sup>7</sup> **FreeBSD** – jedna od popularnih licenci za OpenSource projekte. U principu samo kaže sljedeće: "radi što god želiš, ali nemoj mene kriviti ako nešto pođe po zlu!" [19].

[20]. Na karti može prikazivati vektorske podatke iz GeoJSON, TopoJSON, KML, GML, Mapbox vektorskih pločica i drugih formata.

### 1.3. GeoJSON

GeoJSON je vrlo popularan format podataka među mnogim GIS tehnologijama i uslugama. Jednostavan je, lagan i jasan te je često podržan od strane biblioteka za stvaranje interaktivnih geografskih karata s ciljem prikazivanja vektorskih podataka [21]. Dokumentom [22], koji je proizvod Radne skupine za inženjering interneta (eng. Internet Engineering Task Force, IETF), GeoJSON je definiran kao format za kodiranje različitih geografskih podatkovnih struktura koristeći JavaScript Object Notation (JSON<sup>8</sup>). GeoJSON objekt može predstavljati regiju prostora (geometriju), prostorno ograničen entitet (značajku, eng. Feature) ili popis značajki (kolekciju značajki, eng. FeatureCollection). GeoJSON podržava sljedećih sedam tipova geometrija:

- točka (Point)
- linija (LineString)
- poligon (Polygon)
- skup točaka (MultiPoint)
- skup linija (MultiLineString)
- skup poligona (MultiPolygon)
- kolekcija geometrija (GeometryCollection)

GeoJSON format bavi se geografskim podacima u najširem smislu. Sve što ima kvalitete ograničene geografskim prostorom može biti Feature objekt, bez obzira na to je li riječ o fizičkoj strukturi ili nije. Detaljna analiza GeoJSON formata na temelju primjera kolekcije značajki sa slike 1.4 slijedi u nastavku.

---

<sup>8</sup> **JSON** - lagani format za pohranu i prijenos podataka [23]. Pravila sintakse JSON-a: podaci su u parovima ime/vrijednost, podaci odvojeni zarezima, vitičaste zagrade drže objekte, uglate zagrade drže polje.

```

{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [
          102, 0.5
        ]
      },
      "properties": {
        "prop0": "value0"
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [ 102, 0 ], [ 103, 1 ], [ 104, 0 ], [ 105, 1 ]
        ]
      },
      "properties": {
        "prop0": "value0",
        "prop1": 0
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [ 100, 0 ], [ 101, 0 ], [ 101, 1 ], [ 100, 1 ], [ 100, 0 ]
          ]
        ]
      },
      "properties": {
        "prop0": "value0",
        "prop1": {
          "this": "that"
        }
      }
    }
  ]
}

```

**Slika 1.4** Kolekcija značajki u GeoJSON formatu

Promatrana kolekcija značajki sadrži polje koje se sastoji od tri objekta značajki. Svaki objekt značajke je GeoJSON objekt koji ima član „type“ s vrijednošću „Feature“. Unutar svakog od objekta značajke nalazi se „Geometry“ objekt. On predstavlja točke, krivulje i površine u koordinatnom prostoru. Vrijednost člana „type“ objekta „Geometry“ mora biti jedan od sedam prethodno spomenutih tipova geometrije podržanih u GeoJSON formatu. Konkretno, u promatranom primjeru opisane su geometrije tipa točka, linija i poligon.

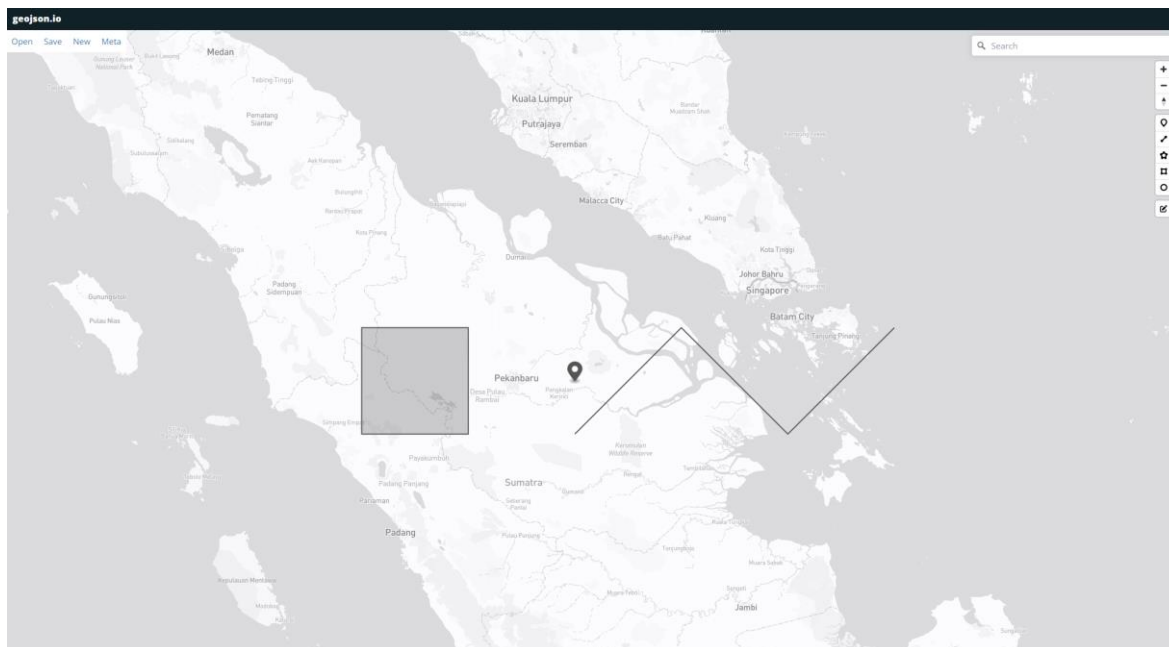
GeoJSON „Geometry“ objekt bilo kojeg tipa, osim kolekcija geometrija (eng. GeometryCollection) ima člana s imenom „coordinates“. Vrijednost člana „coordinates“ je polje (eng. array). Struktura elemenata u ovom polju određena je tipom geometrije. Pozicija je temeljni geometrijski konstrukt. Član „coordinates“ objekta „Geometry“ sastoji se od:

- jedne pozicije u slučaju geometrije tipa točka (eng. Point)
- polja pozicija u slučaju geometrije tipa linija (eng. LineString) ili tipa skup točaka (eng. MultiPoint)
- polja pozicija geometrije tipa linija (eng. LineString) u slučaju geometrije tipa poligon (eng. Polygon) ili tipa skup linija (eng. MultiLineString),
- polja koordinata geometrije tipa poligon (eng. Polygon) u slučaju geometrije tipa skup poligona (eng. MultiPolygon)

Pozicija je zapravo polje brojeva. Pozicija može imati isključivo dva ili tri elementa. Prva dva elementa su geografska dužina i širina, točno tim redom, definirani pomoću decimalnih brojeva. Nadmorska visina je opcionalni treći element polja. Pozicije u promatranom primjeru sastoje se isključivo od geografske dužine i širine.

Članovi koji nisu opisani specifikacijom u dokumentu [22], mogu se koristiti u GeoJSON objektu kao strani članovi. Podrška za strane članove može varirati između implementacija i nije definiran normativni model obrade stranih članova. Stoga, implementacije koje previše ovise o korištenju stranih članova mogu doživjeti smanjenu interoperabilnost s drugim implementacijama. U primjeru, član „properties“ je strani član objekta značajke. Kada je vrijednost stranog člana objekt, što je i slučaj u promatranom primjeru, svi članovi tog objekta su također strani članovi.

Na slici 1.5 prikazan je prethodno opisani GeoJSON objekt na geografskoj karti u alatu GeoJson.io. GeoJson.io je popularan web alat otvorenog koda za pretvaranje, uređivanje i kreiranje GeoJSON datoteka [24]. Redom, s lijeva prema desno, prikazane su značajke s tipom geometrije poligon, točka i linija.



Slika 1.5 Prikaz GeoJSON objekta sa slike 1.4 u alatu GeoJson.io

## 1.4. Pregled postojećih rješenja

Proučavanjem i detaljnim pregledom dostupnih aplikacija i sustava za učenje s pomoću interaktivnih geografskih karata, nastoji se uočiti njihove prednosti i nedostatke. Svrha ovog pregleda je uočiti koje su komponente tih sustava učinkovite, a koje manje učinkovite, te prikupljena saznanja iskoristiti u implementaciji vlastitog obrazovnog sustava, čineći ga što kvalitetnijim i prilagođenijim potrebama korisnika (učitelja i njegovih učenika). U nastavku poglavlja provedena je analiza dvaju sustava za učenje s pomoću interaktivnih geografskih karata.

### 1.4.1. Seterra Geography

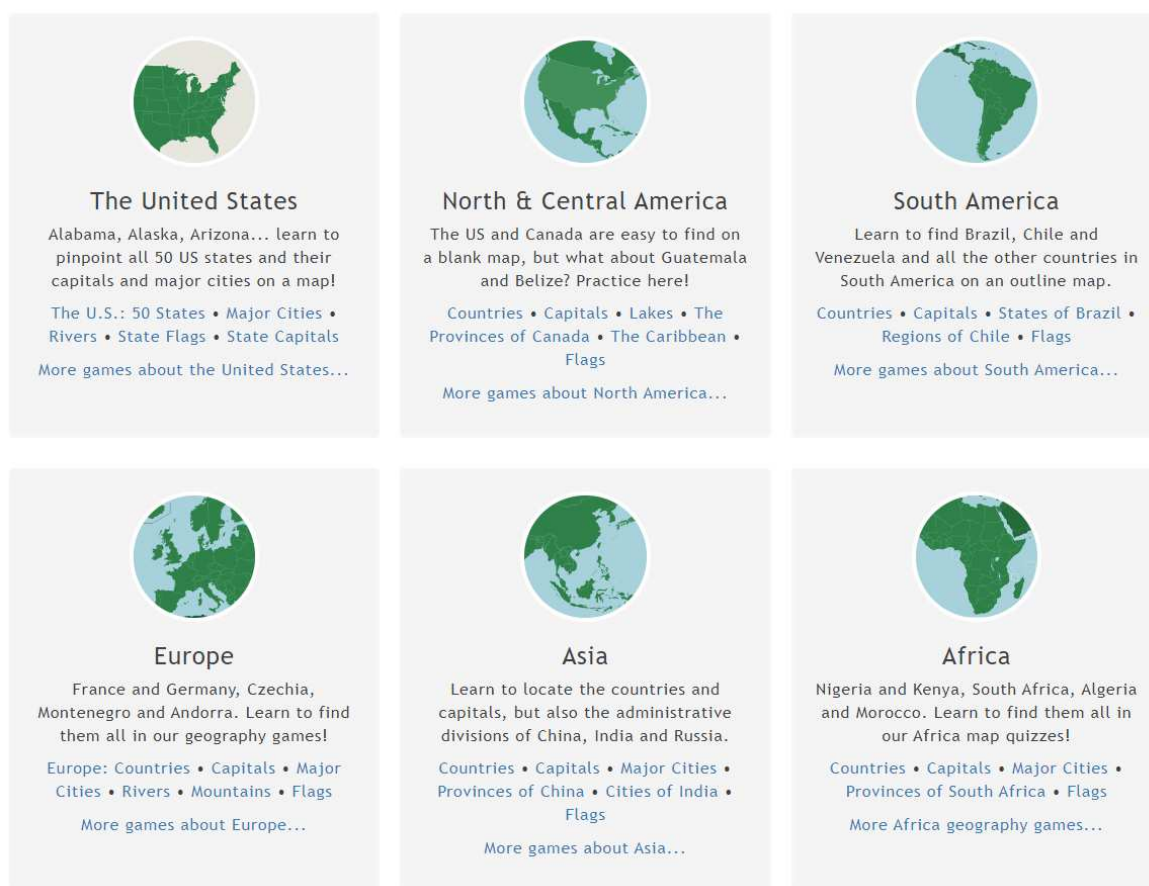
Seterra<sup>9</sup> je besplatna kvizaška igra s geografskim kartama, dostupna online u web pregledniku i kao aplikacija za iOS i Android mobilne uređaje [25]. Seterra trenutno ima više od tri milijuna jedinstvenih posjetitelja svaki mjesec. Nudi preko 400 kvizaških igri s

---

<sup>9</sup> Seterra - <https://www.geoguessr.com/quiz/seterra>

geografskim kartama na preko 40 jezika. Kvizovi s geografskim kartama prilagodljivi su tako da korisnici mogu odabrati željene lokacije koje će biti obuhvaćene kvizom. Prilagođeni kvizovi mogu se lako dijeliti s prijateljima ili učenicima.

Seterra nudi zabavne kvizove koji pomažu u upoznavanju država svijeta, glavnih gradova, administrativnih jedinica država, zastava, rijeka, jezera, planina i značajnih geoloških obilježja. Na najvišoj razini, kvizovi su grupirani prema kontinentima, što je vidljivo na slici 1.6.



**Slika 1.6** Seterra – podjela kvizova prema kontinentima

Iako je kvizaška igra Seterra prvenstveno namijenjena učenju i ponavljanju geografskih znanja, određeni kvizovi učinkovito integriraju geografske i povijesne informacije, pružajući korisnicima sveobuhvatniji edukativni doživljaj. Primjer jednog takvog kviza vidljiv je na slici 1.7.

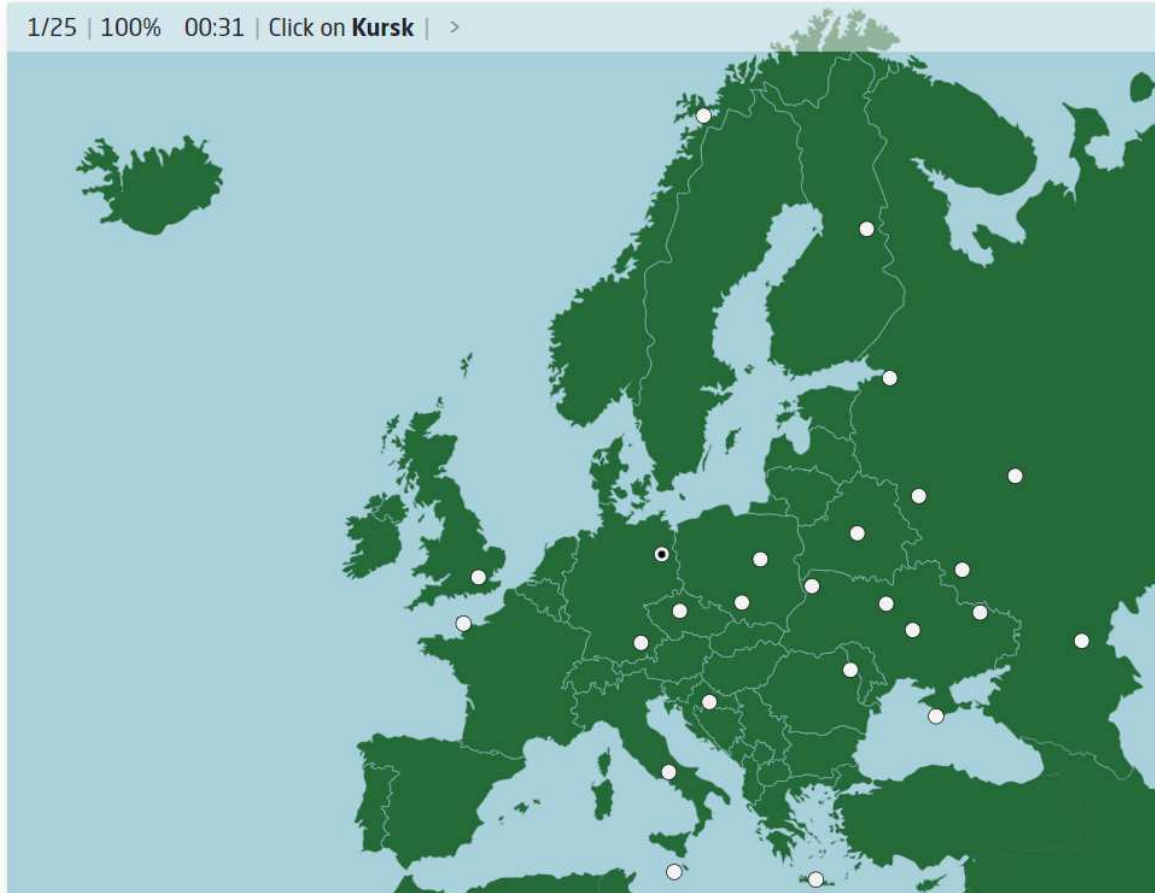


## Europe: WW2 Events - Map Quiz Game

Anthropoid, Auschwitz, Berlin, Crete, Dnieper, Jasenovac, Jassy-Kishinev, Kharkov, Kursk, Kyiv, Leningrad, Malta, Minsk, Monte Cassino, Moscow, Munich Betrayal, Narvik, Normandy, Sevastopol, Smolensk, Stalingrad, Suomussalmi, The Blitz, Volhynia, Warsaw Ghetto (25)

CREATE CHALLENGE

1/25 | 100% | 00:31 | Click on **Kursk** | >



Battle of Berlin, April 16 - May 2, 1945 One of the last major offensives of the European theatre of the war. It is also known as the Fall of Berlin.

**Slika 1.7** Seterra – kviz o bitkama u Drugom svjetskom ratu na području Europe

Tema kviza su događaji Drugog svjetskog rata, odnosno lokacije najznačajnijih bitki Drugog svjetskog rata na području Europe. Korisnik treba između ponuđenih lokacija na karti odabrati onu točnu. Nakon ispravnog odgovora korisnika, ispod karte pojavljuje se tekst koji detaljnije objašnjava sam povijesni događaj. Navedeni kviz omogućuje učenje povijesti i geografije tako što svakom povijesnom događaju pridružuje točku na karti koja označava njegovu geografsku lokaciju. Tako se povijesni događaji stavljaju u kontekst njihove

lokacije u svijetu, pružajući dublje razumijevanje njihovog značaja i utjecaja. U svrhu praćenja rezultata korisnika, tijekom provedbe kviza prati se proteklo vrijeme i korisnikov postotak točnosti rješavanja kviza. Još jedan primjer kviza koji učinkovito integrira geografske i povijesne informacije je kviz o karti Europe 1914. godine, neposredno prije Prvog svjetskog rata. Kroz ovaj kviz korisnik uči o tadašnjoj podjeli Europe na države, nazivima država i njihovim granicama, dobivajući uvid u političku situaciju tog vremena.

Iako Seterra nudi preko 400 kvizova temeljenih na geografskim kartama, nedostatak mogućnosti stvaranja vlastitih kvizova može odvratiti određeni broj učitelja koji traže sustav za učenje s pomoću interaktivnih geografskih karata za uporabu u sklopu nastavnog procesa. Također, interaktivnost geografskih karti unutar kvizova poprilično je ograničena, budući da se interakcija korisnika s kartom uvijek svodi na odabir jedne od ponuđenih mogućnosti.

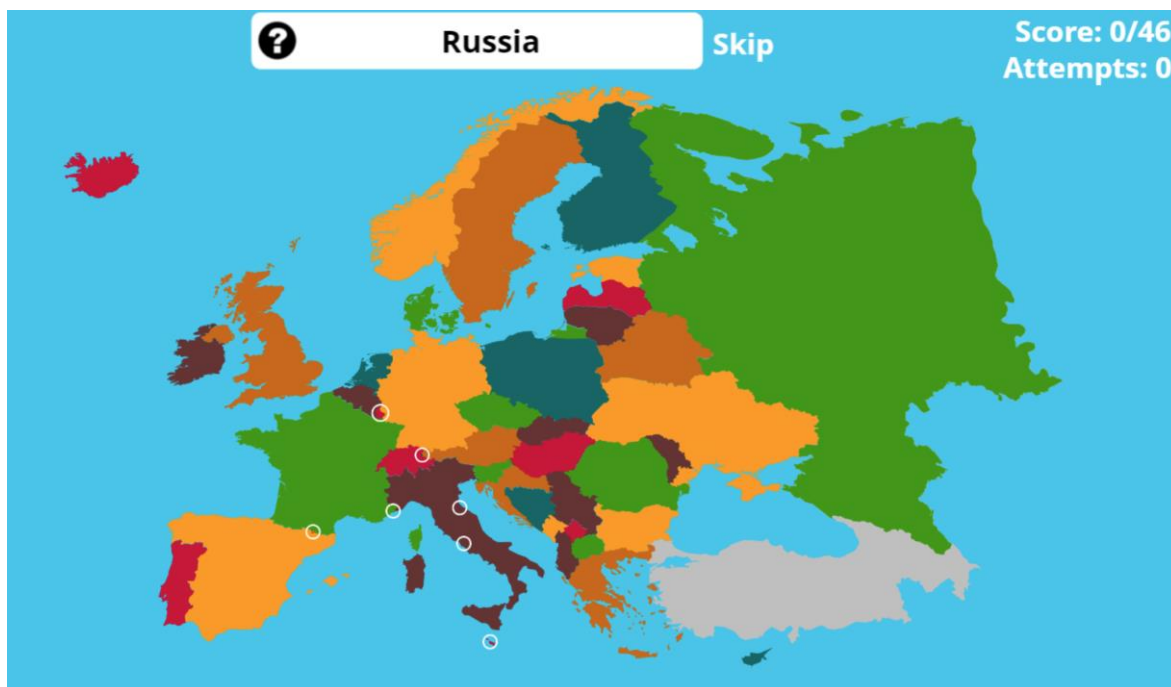
#### **1.4.2. World Geography Games**

World Geography Games<sup>10</sup> je besplatna web aplikacija koja pruža zabavne i edukativne kvizaške igre za djecu, učenike, studente, odrasle i starije osobe s ciljem unaprjeđenja vlastitog geografskog znanja [26]. Kvizovi uključuju pitanja o državama, glavnim gradovima, zastavama, regijama, vodama, planinama, pustinjama, metropolitanskim područjima i drugim temama. Primjer kviza koji ispituje prepoznavanje država Europe vidljiv je na slici 1.8. Prije samog pokretanja kviza, korisnik ima mogućnost pregleda geografske karte sa svim označenim rješenjima. Zahvaljujući spomenutoj funkcionalnosti, korisnik može jednostavno učiti s karte, a zatim i provjeriti naučeno s pomoću kviza.

---

<sup>10</sup> World Geography Games - <https://world-geography-games.com/>





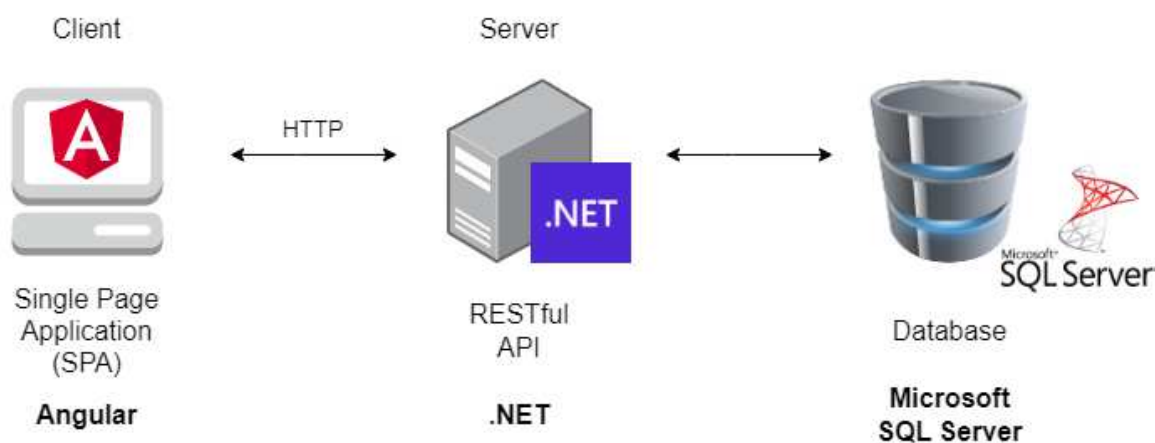
**Slika 1.8** World Geography Games – kviz prepoznavanja država Europe

World Geography Games, zbog svog dizajna i jednostavnosti korištenja, čini se najprikladnijim za djecu i učenike nižih razreda osnovne škole. Interaktivnost geografske karte u igrama ograničena je na odabir jedne od ponuđenih opcija na karti. Nemogućnost stvaranja vlastitih kvizova značajan je nedostatak aplikacije, no unatoč tome, aplikacija može biti korisna u nastavnom procesu.

## 2. Arhitektura sustava

U sklopu diplomskog rada razvijen je sustav za učenje s pomoću interaktivnih geografskih karata. Sustav je zamišljen kao web aplikacija, implementirana u arhitekturi klijent-poslužitelj (eng. client-server). Navedena arhitektura dijeli aplikaciju na dva dijela, klijenta i poslužitelja s pripadnom mu bazom podataka:

- **Klijent** – program koji šalje zahtjeve za podacima prema poslužitelju preko mreže (obično putem HTTP, odnosno HTTPS protokola) te ih prikazuje korisniku [27]
- **Poslužitelj** – program koji pruža središnju funkcionalnost arhitekture, prihvaća zahtjeve klijenta, izvršava traženi zadatak i vraća rezultate klijentu, prema potrebi [28]
- **Baza podataka** – organizirana kolekcija strukturiranih informacija ili podataka, obično elektronički pohranjenih u računalnom sustavu [29]



Slika 2.1 Arhitektura klijent-poslužitelj

### 2.1. Klijent

Klijent, poznat i pod pojmom „front-end aplikacija“, je aplikacija koja se izvodi unutar web preglednika na računalu samog korisnika. U sklopu diplomskog rada, klijentska aplikacija razvijena je uz pomoć standardnih tehnologija za web razvoj: HTML, CSS i

JavaScript. Zbog svoje kompleksnosti, spomenute tehnologije iskorištene su unutar radnog okvira (eng. framework) Angular. Korištenje radnih okvira u razvoju aplikacije pomaže efikasnijoj organizaciji koda, ponovnoj iskoristivosti komponenti i olakšava implementaciju naprednih funkcionalnosti. Korišteni Angular radni okvir razvija aplikaciju u SPA arhitekturi (eng. Single Page Application). SPA arhitektura donosi promjenu u paradigmi razvoja web aplikacija, pružajući brže i uglađenije korisničko iskustvo. Više o radnom okviru Angular i SPA arhitekturi nalazi se u poglavlju 4.1. Jedna od zadaća takvog implementiranog klijenta jest komunikacija s poslužiteljem putem HTTP/HTTPS protokola. Protokol HTTPS (eng. Hypertext Transfer Protocol Secure) je sigurna verzija HTTP-a, mrežnog protokola aplikacijskog sloja bez očuvanja stanja. HTTP je primarni protokol koji se koristi za slanje podataka između klijenta i poslužitelja [30]. Nakon što klijent pošalje zahtjev poslužitelju, očekuje povratni odgovor s HTTP statusnim kodom i JSON objektom. Dobiveni JSON objekt klijent zatim prikazuje korisniku kroz korisničko sučelje (eng. User Interface, UI).

Interaktivne geografske karte, unutar klijentske aplikacije, implementirane su s pomoću Leaflet JavaScript biblioteke. Leaflet je vodeća JavaScript biblioteka otvorenog koda za interaktivne karte prilagođene mobilnim uređajima [11]. Odabrana je Leaflet biblioteka zato što je jednostavna za korištenje, besplatna i kompatibilna s odabranim Angular radnim okvirom klijentske aplikacije. Opširnije o samoj biblioteci pisano je u poglavlju 1.2.1 u sklopu pregleda dostupnih biblioteka za stvaranje interaktivnih geografskih karata.

Klijentska aplikacija postavljena je na Microsoft Azure, što omogućava njeno pouzdano i skalabilno izvršavanje u oblaku (eng. Cloud computing). Microsoft Azure je platforma za računalstvo u oblaku i infrastruktura koju je stvorio Microsoft za izgradnju, implementaciju i upravljanje aplikacijama i uslugama putem globalne mreže podatkovnih centara kojima također upravlja Microsoft [31].

## **2.2. Poslužitelj**

Poslužitelj, poznat i pod pojmom „back-end aplikacija“, je aplikacija koja se izvodi na sigurnom poslužiteljskom računalu. U sklopu diplomskog rada, poslužiteljska aplikacija razvijena je u programskom jeziku C#, koristeći .NET platformu za razvoj aplikacija.

Platforma .NET je besplatna platforma otvorenog koda, osmišljena i podržana od strane Microsofta, namijenjena za izradu mnogih vrsta aplikacija. Platforma .NET može pokrenuti programe napisane na više različitih jezika, dok je C# najpopularniji [32]. Više o .NET platformi nalazi se u poglavlju 4.2.

Poslužiteljska aplikacija implementirana je kao višeslojni (eng. Multi-layer) RESTful API te je također postavljena na Microsoft Azure.

Aplikacijsko programsko sučelje (eng. Application Programming Interface, API) može se smatrati ugovorom o usluzi između dvije aplikacije, klijentske i poslužiteljske. Ovaj ugovor definira međusobnu komunikaciju tih dviju aplikacija, koristeći zahtjeve i odgovore [33].

REST (eng. Representational state transfer) je skup arhitektonskih ograničenja za razvoj web aplikacijskih programskih sučelja. Kada se zahtjev klijenta podnese putem RESTful API-ja, on prenosi prikaz stanja resursa podnositelju zahtjeva ili krajnjoj točki. Ove informacije ili prikaz najčešće se isporučuju u JSON (eng. JavaScript Object Notation) formatu putem HTTP protokola [34], što je i slučaj u implementaciji poslužitelja u sklopu diplomskog rada.

Višeslojna aplikacijska arhitektura olakšava upravljanje složenim modernim aplikacijama i omogućava rad na agilniji način. Navedena arhitektura sastoji se od više slojeva, od kojih svaki ima specifičnu odgovornost. Odvajanje slojeva i odgovornosti na ovaj način čini razvoj i održavanje aplikacije jednostavnijim. Poslužiteljska aplikacija podijeljena je na tri osnovna sloja: API, poslovni i podatkovni sloj.

### **2.2.1. API sloj**

API sloj poslužiteljske aplikacije predstavlja gornji sloj implementirane troslojne arhitekture. Ovaj sloj je odgovoran isključivo za izlaganje funkcionalnosti aplikacije i ne uključuje poslovnu logiku i pristup podacima. Sastoji se od kontrolera (eng. controller) koji osluškiju zahtjeve na konfiguriranim krajnjim točkama (eng. endpoint).

Na API sloju implementirana je autentifikacija korisnika s pomoću kolačića (eng. Cookie). Po uspješnoj prijavi korisnika u sustav, poslužiteljska aplikacija stvara kolačić s podacima poput globalno jedinstvenog identifikatora korisnika i njegovoj ulozi te ga vraća klijentu. Dobiveni kolačić se pohranjuje u web preglednik. Na slici 2.2 prikazan je

programski kod kontrolera za prijavu korisnika u sustav, koji po uspješnoj prijavi stvara kolačić i vraća ga klijentu.

```
[HttpPost]
[Route("Login")]
0 references
public async Task<ActionResult> Login(LoginDto loginDto)
{
    var account = await this._authService.Login(loginDto);
    var response = new Response();

    var userRole = (account.Role == (int)UserRole.Teacher) ? "Teacher" : "Student";

    var claims = new List<Claim>()
    {
        new Claim(ClaimTypes.NameIdentifier, Convert.ToString(account.Id)),
        new Claim(ClaimTypes.Email, account.Email),
        new Claim(ClaimTypes.Role, userRole),
    };

    var identity = new ClaimsIdentity(claims, CookieAuthenticationDefaults.AuthenticationScheme);
    var principal = new ClaimsPrincipal(identity);
    await HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme, principal, new AuthenticationProperties()
    {
        IsPersistent = true,
        AllowRefresh = true,
        ExpiresUtc = DateTimeOffset.UtcNow.AddMinutes(120)
    });

    response.Message = "Prijava uspješna";
    return Ok(response);
}
```

Slika 2.2 Krajnja točka za prijavu korisnika u sustav

Kako bi se zaštitili određeni osjetljivi resursi dostupni na kontrolerima, na razini API sloja implementirana je autorizacija temeljena na ulogama. Sustav je zamišljen da podržava dvije različite uloge korisnika: uloga „Teacher“ odnosno uloga učitelja i uloga „Student“ odnosno uloga učenika. Informacije o ulozi konkretnog identificiranog korisnika spremljene su unutar kolačića. Prilikom svakog zahtjeva koji klijentska aplikacija šalje prema poslužitelju, web preglednik prilaže kolačić. Poslužiteljska aplikacija koristi taj kolačić za autentifikaciju i autorizaciju korisnika, određujući time ima li korisnik pravo pristupa traženim resursima. Primjer zaštite resursa prikazan je na slici 2.3.

```

[Authorize(Roles = "Teacher")]
[HttpPut]
[Route("Update")]
0 references
public async Task<ActionResult<TestDto>> UpdateTest(CreateTestDto test)
{
    var _test = await _testService.UpdateTest(test);
    return Ok(_test);
}

```

Slika 2.3 Zaštićen resurs kojem može pristupiti isključivo korisnik s ulogom učitelja

### 2.2.2. Poslovni sloj

Poslovni sloj poslužiteljske aplikacije predstavlja srednji sloj implementirane troslojne arhitekture. Sastoji se od servisa i odgovoran je za obradu podataka primljenih iz API sloja prema definiranim poslovnim pravilima te njihovo slanje podatkovnom sloju. Također, u suprotnom smjeru, poslovni sloj prilagođava „sirove“ podatke dobivene iz podatkovnog sloja u željenu strukturu i prosljeđuje ih API sloju.

Jedan od zahtjeva implementiranog sustava bio je automatsko ispravljanje zadataka. Prema poslovnom pravilu, zadatak tipa „Označi na karti“ smatra se točnim ako je presjek površina točno označenog poligona i poligona označenog od strane učenika veći ili jednak od 70 %. Navedena provjera implementirana je unutar metode **ProcessMarkPolygonTask** servisa **TestInstanceService** na poslovnom sloju aplikacije, uz pomoć **NetTopologySuite** NuGet paketa. Programski kod metode prikazan je na slici 2.4.

```

1 reference
private async Task<TestTaskInstance> ProcessMarkPolygonTask(TestTask testTask, TestTaskInstance testTaskInstance)
{
    if (testTaskInstance.StudentAnswer == null)
    {
        testTaskInstance.Correct = false;
        testTaskInstance.Checked = true;
        return await Task.FromResult(testTaskInstance);
    }

    // Parse the JSON strings into arrays of points
    List<Coordinate> answerCoords = ParseCoordinates(testTask.Answer);
    List<Coordinate> studentAnswerCoords = ParseCoordinates(testTaskInstance.StudentAnswer);

    // Ensure closed linestring for answerCoords
    if (answerCoords.First() != answerCoords.Last())
    {
        // Add the first coordinate to the end to close the linestring
        answerCoords.Add(answerCoords.First());
    }

    // Ensure closed linestring for studentAnswerCoords
    if (studentAnswerCoords.First() != studentAnswerCoords.Last())
    {
        // Add the first coordinate to the end to close the linestring
        studentAnswerCoords.Add(studentAnswerCoords.First());
    }

    // Create Polygon objects from the arrays of points
    Polygon answerPolygon = new Polygon(new LinearRing(answerCoords.ToArray()));
    Polygon studentAnswerPolygon = new Polygon(new LinearRing(studentAnswerCoords.ToArray()));

    // Calculate the intersection of the two polygons
    Geometry intersection = OverlayNG.Overlay(answerPolygon, studentAnswerPolygon, OverlayNG.INTERSECTION);

    // Calculate the area of the intersection polygon
    double intersectionArea = intersection.Area;

    // Calculate the area of each input polygon
    double answerArea = answerPolygon.Area;
    double studentAnswerArea = studentAnswerPolygon.Area;

    // Determine the minimum area of the input polygons
    double minArea = Math.Min(answerArea, studentAnswerArea);

    // Calculate the percentage of overlap
    double overlapPercentage = (intersectionArea / minArea) * 100;

    if (overlapPercentage >= 70)
    {
        testTaskInstance.Correct = true;
    }
    else
    {
        testTaskInstance.Correct = false;
    }
    testTaskInstance.Checked = true;

    return await Task.FromResult(testTaskInstance);
}

```

Slika 2.4 Metoda za automatsko ocjenjivanje zadatka tipa „Označi na karti“

### 2.2.3. Podatkovni sloj

Podatkovni sloj poslužiteljske aplikacije predstavlja donji sloj implementirane troslojne arhitekture. Podatkovni sloj odgovoran je za slanje i primanje podataka u bazu podataka. Za potrebe objektno-relacijskog preslikavanja podataka unutar aplikacije i podataka pohranjenih u bazi podataka, korišten je Entity Framework Core alat. Entity

Framework Core je moderni alat namijenjen za objektno-relacijska preslikavanja u .NET-u, podržava LINQ upite te se može koristiti u kombinaciji s različitim bazama podataka [35]. Entity Framework model kreiran je iz postojeće baze podataka (tzv. database first pristup) postupkom reverznog inženjerstva.

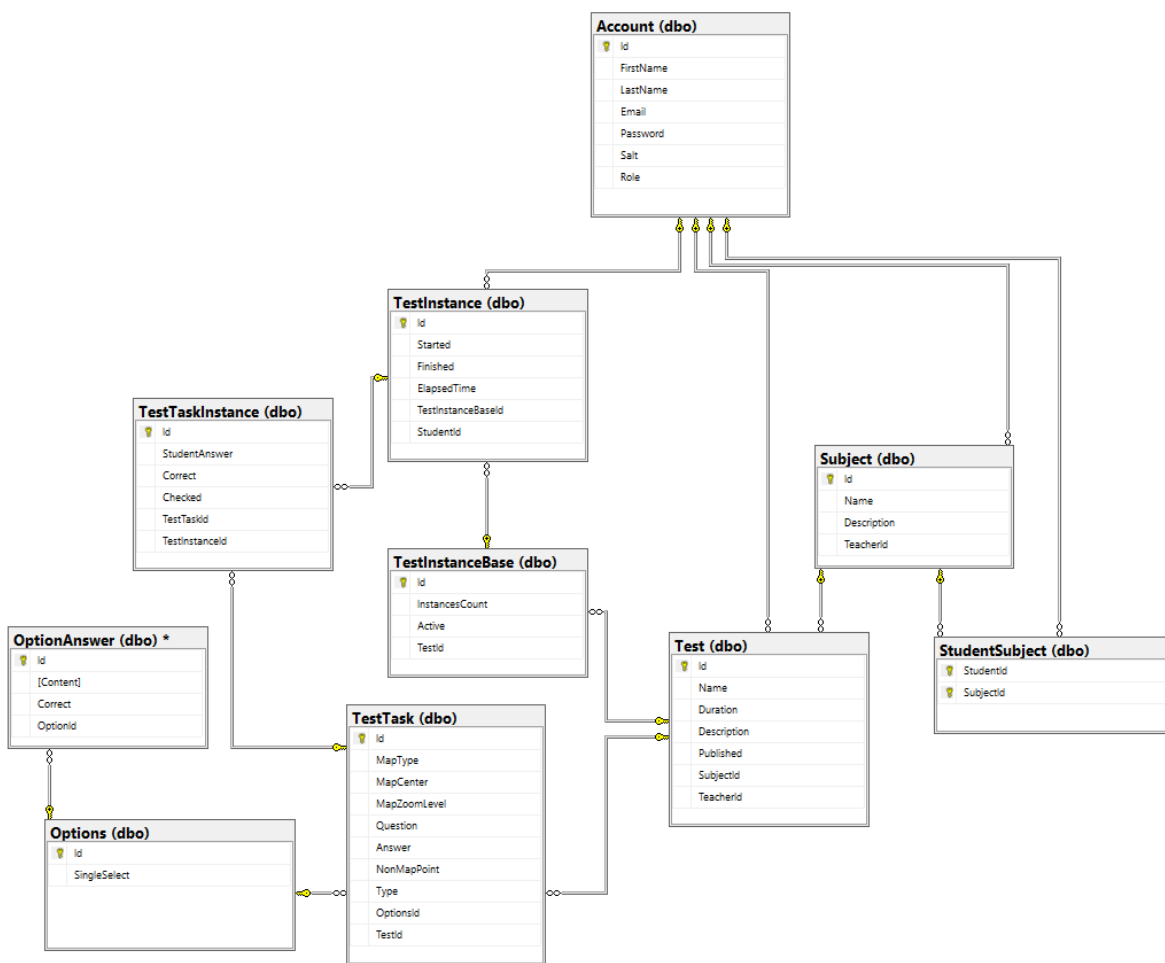
## **2.3. Baza podataka**

Kako bi se podaci trajno pohranjivali, moraju biti spremljeni u bazu podataka. U sklopu ovog diplomskog rada, za pohranu podataka odabran je SQL Server, sustav za upravljanje relacijskim bazama podataka koji je razvila tvrtka Microsoft. Više o SQL Serveru nalazi se u poglavlju 4.3. Kako bi rad sa SQL Server bazom podataka bio lakši i učinkovitiji, korišten je alat SQL Server Management Studio. Ovaj alat nudi grafičko korisničko sučelje koje omogućuje jednostavno upravljanje bazom podataka i njenim entitetima, olakšavajući tako administraciju i manipulaciju podacima [36]. S obzirom na to da je SQL Server razvijen od strane Microsofta, prirodno je da se za postavljanje baze podataka koristi Microsoft Azure, njihova platforma za usluge računalstva u oblaku.



### 3. Model podataka

Za pravilno funkcioniranje sustava koji uključuje korisničke račune, predmete, skice ispita s pripadajućim zadacima, ispite, instance ispita i zadatke za svakog pojedinog učenika, potrebno je osmisliti i implementirati odgovarajući model za pohranu svih potrebnih podataka u bazu podataka. Aplikacija koristi relacijsku bazu podataka. Svi potrebni entiteti pohranjeni su kao relacije (tablice) s odgovarajućim atributima (stupcima). Za ispravno funkcioniranje sustava, u relacijskoj bazi podataka implementirano je deset tablica. Veze između različitih tablica ostvarene su s pomoću primarnih i stranih ključeva. Na slici 3.1 prikazan je dijagram relacijskog modela implementirane baze podataka.



Slika 3.1 Relacijski model baze podataka

### 3.1. Opis relacija

U sklopu ovog poglavlja detaljno je opisano svih deset relacija baze podataka s pripadnim atributima. Zbog jednostavnijeg uočavanja atributa relacije, one su prikazane u tabličnom obliku, uz popratni opis značajnijih atributa relacije. Prvi stupac tablice predstavlja naziv atributa. Svi atributi imaju svoje nazive na engleskom jeziku. Drugi stupac predstavlja podatkovni tip atributa. Ako je atribut opcionalan, tj. može poprimiti vrijednost NULL, uz podatkovni tip atributa stajat će i oznaka „(NULL)“. Za sve ostale atribute smatra se da moraju poprimiti konkretnu vrijednost, odnosno da imaju postavljeno ograničenje NOT NULL. Treći stupac predstavlja kratak opis promatranog atributa. Uz naziv atributa u prvom stupcu, ako je atribut ujedno i primarni ključ relacije, stajat će oznaka „(PK)“. U slučaju kad je atribut strani ključ relacije, uz naziv atributa stajat će oznaka „(FK)“.

- **Account**

Relacija koja predstavlja račun korisnika u sustavu. Svi atributi relacije su obavezni. Primarni ključ relacije je **Id**. Spomenuti primarni ključ je samoinkrementirajući jedinstveni identifikator pojedinog korisničkog računa u sustavu. Atribut **Email** predstavlja email adresu korisnika koja također mora biti jedinstvena na razini čitavog sustava. Kriptirana lozinka korisničkog računa pohranjena je pod atributom **Password**, dok je niz bajtova korišten za kriptiranje svake pojedine lozinke pohranjen pod atributom **Salt**. Atribut **Role** označava ulogu pojedinog korisnika unutar sustava. Vrijednost „0“ označava ulogu učitelja, dok vrijednost „1“ označava ulogu učenika. Opisana relacija prikazana je tablicom (Tablica 3.1).

**Tablica 3.1** Tablični prikaz relacije Account

|           |              |                                                                   |
|-----------|--------------|-------------------------------------------------------------------|
| Id (PK)   | INT          | Jedinstveni samoinkrementirajući identifikator korisničkog računa |
| FirstName | VARCHAR(50)  | Ime korisnika                                                     |
| LastName  | VARCHAR(50)  | Prezime korisnika                                                 |
| Email     | VARCHAR(100) | Email adresa korisnika                                            |

|          |              |                                                        |
|----------|--------------|--------------------------------------------------------|
| Password | VARCHAR(MAX) | Kriptirana korisnička lozinka                          |
| Salt     | VARCHAR(MAX) | Niz bajtova korišten za kriptiranje korisničke lozinke |
| Role     | INT          | Zastavica uloge korisničkog računa                     |

- **Subject**

Relacija koja predstavlja predmet. Primarni ključ relacije je **Id**. Spomenuti primarni ključ je samoinkrementirajući jedinstveni identifikator svakog predmeta u sustavu. Opis predmeta pod atributom **Description** je opcionalan. Atribut **TeacherId** je strani ključ relacije koji povezuje učitelja s njegovim predmetom. Opisana relacija prikazana je tablicom (Tablica 3.2).

**Tablica 3.2** Tablični prikaz relacije Subject

|                |                      |                                                                                     |
|----------------|----------------------|-------------------------------------------------------------------------------------|
| Id (PK)        | INT                  | Jedinstveni samoinkrementirajući identifikator predmeta                             |
| Name           | VARCHAR(100)         | Naziv predmeta                                                                      |
| Description    | VARCHAR(1000) (NULL) | Opis predmeta                                                                       |
| TeacherId (FK) | INT                  | Jedinstveni samoinkrementirajući identifikator korisničkog računa s ulogom učitelja |

- **StudentSubject**

Relacija koja povezuje učenike s predmetima na kojima se oni nalaze. Relacija je nužna zbog veze učenika i predmeta koja je tipa više naprema više (eng. many-to-many). Jedan učenik se istovremeno može nalaziti na više različitih predmeta. S druge strane, na jednom predmetu se istovremeno može nalaziti više različitih učenika. Oba atributa relacije,

**StudentId** i **SubjectId**, su ujedno primarni i strani ključevi. Opisana relacija prikazana je tablicom (Tablica 3.3).

**Tablica 3.3** Tablični prikaz relacije StudentSubject

|                     |     |                                                                                    |
|---------------------|-----|------------------------------------------------------------------------------------|
| StudentId (PK) (FK) | INT | Jedinstveni samoinkrementirajući identifikator korisničkog računa s ulogom učenika |
| SubjectId (PK) (FK) | INT | Jedinstveni samoinkrementirajući identifikator predmeta                            |

- **Test**

Relacija koja predstavlja skicu ispita. Svi atributi relacije, osim **Description** atributa koji predstavlja opis skice ispita, su obavezni. Primarni ključ relacije je **Id**. Spomenuti primarni ključ je samoinkrementirajući jedinstveni identifikator pojedine skice ispita u sustavu. Atribut **Description** označava maksimalno dopušteno trajanje rješavanja ispita koji će nastati objavom pripadne skice ispita. Budući da se skica ispita radi za jedan konkretan predmet, u relaciji se nalazi atribut **SubjectId** koji je strani ključ prema relaciji predmeta. Skicu ispita kreira točno jedan učitelj, stoga se u relaciji nalazi i strani ključ **TeacherId** prema relaciji korisničkih računa. Opisana relacija prikazana je tablicom (Tablica 3.4).

**Tablica 3.4** Tablični prikaz relacije Test

|             |                     |                                                                  |
|-------------|---------------------|------------------------------------------------------------------|
| Id (PK)     | INT                 | Jedinstveni samoinkrementirajući identifikator skice ispita      |
| Name        | VARCHAR(100)        | Naziv skice ispita                                               |
| Duration    | TIME                | Maksimalno dopušteno trajanje rješavanja objavljene skice ispita |
| Description | VARCHAR(500) (NULL) | Opis skice ispita                                                |

|                |     |                                                                                     |
|----------------|-----|-------------------------------------------------------------------------------------|
| Published      | BIT | Zastavica je li skica objavljena                                                    |
| SubjectId (FK) | INT | Jedinstveni samoinkrementirajući identifikator predmeta                             |
| TeacherId (FK) | INT | Jedinstveni samoinkrementirajući identifikator korisničkog računa s ulogom učitelja |

- **TestTask**

Relacija koja predstavlja zadatak. Primarni ključ relacije je **Id**. Spomenuti primarni ključ je samoinkrementirajući jedinstveni identifikator pojedinog zadatka u sustavu. Atributi koji su vezani uz sami prikaz interaktivne geografske karte u sklopu zadatka su **MapType**, **MapCenter** i **MapZoomLevel**. Atribut **MapType** je zastavica koja predstavlja odabrani tip karte (vrijednost „1“ predstavlja normalnu kartu, vrijednost „2“ predstavlja slijepu kartu, dok vrijednost „3“ predstavlja satelitsku kartu). Atribut **MapCenter** sadrži koordinate točke središta prikazane geografske karte, dok atribut **MapZoomLevel** označava početnu razinu zumiranja karte.

Atribut **Type** je zastavica koja predstavlja tip zadatka (vrijednost „1“ predstavlja zadatak tipa „Označi točkom na karti“, vrijednost „2“ predstavlja zadatak tipa „Označi na karti“, vrijednost „3“ predstavlja zadatak tipa „Odaberi točku na karti“, vrijednost „4“ predstavlja zadatak tipa „Odaberi na karti“, dok vrijednost „5“ predstavlja zadatak tipa „Obično pitanje“).

Ako je riječ o zadatku tipa „Obično pitanje“, atribut **Answer** ne poprima vrijednost, dok atribut **NonMapPoint** poprima vrijednost. Za sve ostale tipove zadataka vrijedi obratno. Kada atribut **Answer** poprima vrijednost, ona je jednaka koordinatama točke ili poligona, u slučaju zadataka s označavanjem, odnosno jednaka je identifikatoru odgovora opcije u slučaju zadatka s ponuđenim odgovorima. Strani ključ **OptionsId** poprima vrijednost ako je riječ o zadacima s odabirom između ponuđenih odgovora opcije, inače ne poprima vrijednost. Strani ključ **TestId** uvijek je postojan budući da svaki zadatak pripada točno jednoj skici ispita. Opisana relacija prikazana je tablicom (Tablica 3.5).

**Tablica 3.5** Tablični prikaz relacije TestTask

|                |                        |                                                             |
|----------------|------------------------|-------------------------------------------------------------|
| Id (PK)        | INT                    | Jedinstveni samoinkrementirajući identifikator zadatka      |
| MapType        | INT                    | Zastavica tipa karte                                        |
| MapCenter      | VARCHAR(MAX)           | Koordinate za centriranje karte                             |
| MapZoomLevel   | INT                    | Razina zumiranja karte                                      |
| Question       | VARCHAR(MAX)           | Tekst zadatka                                               |
| Answer         | VARCHAR(MAX)<br>(NULL) | Točan odgovor                                               |
| NonMapPoint    | VARCHAR(MAX)<br>(NULL) | Koordinate točke kod pitanja tipa „Obično pitanje“          |
| Type           | INT                    | Zastavica tipa zadatka                                      |
| OptionsId (FK) | INT (NULL)             | Jedinstveni samoinkrementirajući identifikator opcije       |
| TestId (FK)    | INT                    | Jedinstveni samoinkrementirajući identifikator skice ispita |

- **Options**

Relacija koja predstavlja opcije. Svi atributi relacije su obavezni. Primarni ključ relacije je **Id**. Spomenuti primarni ključ je samoinkrementirajući jedinstveni identifikator pojedine opcije. Atribut **SingleSelect** predstavlja zastavicu tipa opcije. Ako je zastavica postavljena na „1“, riječ je o opciji s jednostrukim odabirom, dok je u slučaju da je zastavica postavljena na „0“, riječ o opciji s višestrukim odabirom. Trenutno je sustav implementiran tako da su sve opcije s jednostrukim odabirom. Učitelj označava jedan odgovor kao točan,

dok učenik može odabrati samo jedan od ponuđenih odgovora koji smatra točnim. Opisana relacija prikazana je tablicom (Tablica 3.6).

**Tablica 3.6** Tablični prikaz relacije Options

|              |     |                                                                   |
|--------------|-----|-------------------------------------------------------------------|
| Id (PK)      | INT | Jedinstveni samoinkrementirajući identifikator opcije             |
| SingleSelect | BIT | Zastavica je li riječ o jednostrukom ili višestrukome tipu opcije |

- **OptionAnswer**

Relacija koja predstavlja odgovore pojedine opcije. Svi atributi relacije su obavezni. Primarni ključ relacije je **Id**. Spomenuti primarni ključ je samoinkrementirajući jedinstveni identifikator odgovora pojedine opcije. Atribut **Content** sadrži koordinate točke ili poligona, ovisno o tipu zadatka za koji je opcija s pripadnim joj odgovorima namijenjena. Atribut **Correct** predstavlja zastavicu točnosti odgovora opcije. U sklopu svake opcije, točno jedan odgovor opcije mora biti označen kao točan, tj. imati vrijednost atributa **Correct** postavljenu na „1“, dok svi ostali odgovori imaju vrijednost tog atributa postavljenu na „0“. Strani ključ **OptionsId** definira kojoj opciji pripada pojedini odgovor. Opisana relacija prikazana je tablicom (Tablica 3.7).

**Tablica 3.7** Tablični prikaz relacije OptionAnswer

|         |              |                                                                |
|---------|--------------|----------------------------------------------------------------|
| Id (PK) | INT          | Jedinstveni samoinkrementirajući identifikator odgovora opcije |
| Content | VARCHAR(MAX) | Koordinate odgovora opcije                                     |
| Correct | BIT          | Zastavica je li odgovor točan                                  |

|           |     |                                                             |
|-----------|-----|-------------------------------------------------------------|
| OptionsId | INT | Jedinstveni<br>samoinkrementirajući<br>identifikator opcije |
|-----------|-----|-------------------------------------------------------------|

- **TestInstance**

Relacija koja predstavlja instancu objavljenog ispita. Primarni ključ relacije je **Id**. Spomenuti primarni ključ je samoinkrementirajući jedinstveni identifikator pojedine instance ispita. Strani ključ relacije je **StudentId**, budući da svaki učenik dobiva vlastitu instancu objavljenog ispita na rješavanje. Atribut **Started** je zastavica koja označava je li učenik započeo s rješavanjem vlastite instance („0“ označava da učenik nije započeo s rješavanjem, dok „1“ označava da je započeo). Atribut **Finished** je zastavica koja označava je li učenik završio s rješavanjem vlastite instance („0“ označava da učenik nije završio s rješavanjem, dok „1“ označava da je završio). Jedini opcionalan atribut je **ElapsedTime**. On ne poprima vrijednost sve do kada učenik nije započeo s rješavanjem vlastite instance ispita. Nakon što je započeo s rješavanjem, **ElapsedTime** prati proteklo vrijeme. Instance ispita vezane su uz točno jedan objavljen ispit, a ta je veza postignuta uz strani ključ **TestInstanceBaseId**. Opisana relacija prikazana je tablicom (Tablica 3.8).

**Tablica 3.8** Tablični prikaz relacije TestInstance

|                         |             |                                                                            |
|-------------------------|-------------|----------------------------------------------------------------------------|
| Id (PK)                 | INT         | Jedinstveni<br>samoinkrementirajući<br>identifikator instance ispita       |
| Started                 | BIT         | Zastavica je li učenik<br>započeo s rješavanjem<br>svoje instance ispita   |
| Finished                | BIT         | Zastavica je li rješavanje<br>instance ispita završeno                     |
| ElapsedTime             | TIME (NULL) | Proteklo vrijeme rješavanja<br>instance ispita                             |
| TestInstanceBaseId (FK) | INT         | Jedinstveni<br>samoinkrementirajući<br>identifikator objavljenog<br>ispita |



|                |     |                                                                                             |
|----------------|-----|---------------------------------------------------------------------------------------------|
| StudentId (FK) | INT | Jedinstveni<br>samoinkrementirajući<br>identifikator korisničkog<br>računa s ulogom učenika |
|----------------|-----|---------------------------------------------------------------------------------------------|

- **TestInstanceBase**

Relacija koja predstavlja objavljeni ispit. Svi atributi relacije su obavezni. Primarni ključ relacije je **Id**. Spomenuti primarni ključ je samoinkrementirajući jedinstveni identifikator objavljenog ispita. Atribut **TestId** je strani ključ relacije koji povezuje objavljeni ispit sa skicom ispita na temelju koje je on nastao. Atribut **Active** služi kao zastavica koja označava je li ispit otvoren za rješavanje. Vrijednost „0“ označava da je ispit zatvoren za rješavanje, dok vrijednost „1“ označava da je otvoren za rješavanje. Opisana relacija prikazana je tablicom (Tablica 3.9).

**Tablica 3.9** Tablični prikaz relacije TestInstanceBase

|                |     |                                                                            |
|----------------|-----|----------------------------------------------------------------------------|
| Id (PK)        | INT | Jedinstveni<br>samoinkrementirajući<br>identifikator objavljenog<br>ispita |
| InstancesCount | INT | Broj pripadnih stvorenih<br>instanci ispita                                |
| Active         | BIT | Zastavica je li ispit otvoren<br>za rješavanje                             |
| TestId (FK)    | INT | Jedinstveni<br>samoinkrementirajući<br>identifikator skice ispita          |

- **TestTaskInstance**

Relacija koja predstavlja instancu zadatka. Primarni ključ relacije je **Id**. Spomenuti primarni ključ je samoinkrementirajući jedinstveni identifikator instance zadatka. Atribut **TestTaskId** je strani ključ relacije koji povezuje instancu zadatka sa zadatkom na temelju kojeg je instanca nastala. Još jedan strani ključ je atribut **TestInstanceId** koji povezuje instancu zadatka s instancom ispita u sklopu koje se on nalazi. Budući da učenik može ne odgovoriti na instancu zadatka, atribut **StudentAnswer** je opcionalan, dok su svi ostali

atributi obavezni. Atribut **Correct** služi kao zastavica koja označava je li učenikov odgovor točan. Vrijednost „0“ označava da je odgovor netočan, dok vrijednost „1“ označava da je točan. Atribut **Checked** također služi kao zastavica. On označava je li instanca zadatka ispravljena. Vrijednost „0“ označava da instanca nije ispravljena, dok vrijednost „1“ označava da je ispravljena. Opisana relacija prikazana je tablicom (Tablica 3.10).

**Tablica 3.10** Tablični prikaz relacije TestTaskInstance

|                     |                        |                                                                 |
|---------------------|------------------------|-----------------------------------------------------------------|
| Id (PK)             | INT                    | Jedinstveni samoinkrementirajući identifikator instance zadatka |
| StudentAnswer       | VARCHAR(MAX)<br>(NULL) | Odgovor učenika na instancu zadatka                             |
| Correct             | BIT                    | Zastavica je li odgovor na instancu zadatka točan               |
| Checked             | BIT                    | Zastavica je li instanca zadatka ocjenjena                      |
| TestTaskId (FK)     | INT                    | Jedinstveni samoinkrementirajući identifikator zadatka          |
| TestInstanceId (FK) | INT                    | Jedinstveni samoinkrementirajući identifikator instance ispita  |

## 4. Korištene tehnologije

U ovom poglavlju dan je pregled tehnologija, biblioteka, radnih okvira i platformi korištenih u razvoju sustava za učenje s pomoću interaktivnih geografskih karata. O JavaScript biblioteci Leaflet pisano je u poglavlju 1.2.1, u sklopu pregleda dostupnih biblioteka za stvaranje interaktivnih geografskih karata. Također, pregled GeoJSON tehnologije dan je u poglavlju 1.3.

### 4.1. Angular

Angular je radni okvir otvorenog koda kojeg održavaju Google i zajednica programera [37]. Dizajniran je za učinkovitu izgradnju dinamičkih i interaktivnih jednostraničnih aplikacija (eng. single-page application, SPA). Aplikacija s jednom stranicom je web aplikacija koja dinamički prepisuje trenutnu web stranicu s novim podacima dohvaćenim s web poslužitelja, umjesto da web preglednik učitava čitavu novu stranicu [38]. Google je 2010. godine objavio prvu verziju Angular radnog okvira pod nazivom Angular 1, danas poznat kao AngularJS. Tijekom godina, Angular je doživio značajne transformacije. Današnje verzije Angular radnog okvira podržavaju JavaScript, ali TypeScript je postao primarni jezik zbog potrebe za objektno orijentiranim pristupom. Angular [39] i drugi klijentski radni okviri poput React.js-a temelje se na konceptu komponenti. Zahvaljujući svojoj arhitekturi temeljenoj na komponentama i robusnim performansama, Angular je prikladan za izgradnju velikih aplikacija. Najnovija verzija radnog okvira je Angular 18 iz svibnja 2024. godine [40].

### 4.2. .NET

.NET je platforma otvorenog koda za izradu web, mobilnih i dekstop aplikacija koje mogu raditi nativno na bilo kojem operativnom sustavu. Platforma .NET uključuje alate, biblioteke i jezike koji podržavaju moderno, skalabilno i visokoučinkovito razvijanje softvera [41]. Platforma .NET razvijena je i održavana od strane Microsoft-a. Podržava različite programske jezike, tako da se .NET aplikacije mogu pisati u C#, F#, Visual C++ ili Visual Basic programskom jeziku [42]. Prva verzija .NET-a izlazi 2002. godine pod nazivom

.NET Framework, s integriranim razvojnim okruženjem Visual Studio<sup>11</sup>, kao dio Microsoftove inicijative za pojednostavljenje razvoja aplikacija za operacijski sustav Windows [43]. Trenutno najnovija verzija platforme je .NET 8, objavljena u studenom 2023. godine. Platforma .NET 8 svestrana je platforma koja podržava različite vrste razvoja aplikacija, što je čini popularnom među programerima zbog povećane produktivnosti, performansi i fleksibilnosti. Microsoft nastavlja razvijati platformu kako bi zadovoljio potrebe suvremenog razvoja aplikacija. .NET 8 trenutno je široko korišten u sljedećim područjima industrije: aplikacije namijenjene poduzećima (eng. enterprise application), Internet stvari (eng. Internet of Things, IoT), aplikacije u oblaku (eng. cloud-based application), mobilne aplikacije, razvoj igara, web aplikacije i desktop aplikacije [43].

### 4.3. SQL Server

SQL Server je sustav upravljanja relacijskim bazama podataka (eng. Relational Database Management System, RDBMS) razvijen od strane Microsofta. U relacijskim bazama podataka podaci su organizirani u tablicama (relacijama), koje su međusobno povezane. Svaka tablica sadrži retke i stupce (atribute) [44]. Podržava standardizirani upitni jezik SQL (eng. Structured Query Language). Područje primjene SQL-a uključuje upit za dohvat podataka, manipulaciju podataka, definiciju podataka i kontrolu pristupa podacima. Uz standardizirani SQL, SQL Server ima i vlastito proširenje SQL-a poznato kao Transact-SQL, odnosno T-SQL. T-SQL proširuje mogućnosti SQL-a pružanjem proceduralnog programiranja, lokalnih varijabli, raznovrsnih funkcija, kao i poboljšanih DELETE i UPDATE naredbi [45].

### 4.4. Microsoft Azure

Azure je javna platforma za računalstvo u oblaku, razvijena od strane Microsofta, s više od 200 proizvoda i usluga dostupnih putem javnog interneta [46]. Poput drugih cloud platformi, Azure upravlja i održava hardver, infrastrukturu i resurse kojima se može pristupiti na neki od sljedećih načina: besplatno, plaćanjem po korištenju (eng. pay-per-use)

---

<sup>11</sup> **Visual Studio** - moćan alat koji prati čitav ciklus razvoja softvera. On je sveobuhvatno integrirano razvojno okruženje (eng. Integrated Development Environment, IDE) koje se koristi za pisanje, uređivanje, debugiranje i izgradnju koda te naposljetku postavljanje aplikacija na poslužitelje [47].

ili plaćanjem po zahtjevu (eng. on-demand). Azure nudi upravljanje, pristup i razvoj aplikacija i usluga putem svoje globalne infrastrukture. Također pruža različite mogućnosti, uključujući softver kao uslugu (eng. Software as a Service, SaaS), platformu kao uslugu (eng. Platform as a Service, PaaS) i infrastrukturu kao uslugu (eng. Infrastructure as a Service, IaaS) [46]. Microsoft Azure podržava mnoge programske jezike, alate i okvire, uključujući Microsoft-specifične i sustave/softvere trećih strana.

## **5. Opis rada sustava**

Ostvaren sustav za učenje s pomoću interaktivnih geografskih karata predviđen je za korištenje od strane dvaju tipa korisnika, učitelja i njihovih učenika. Sustav nije nužno vezan uz učenje jednog predmeta, već je pogodan za različite predmete poput povijesti i geografije. Konkretno, u sklopu predmeta Povijest, sustav povijesnom događaju daje geografski kontekst s pomoću karte, omogućujući učenicima bolje razumijevanje gdje su se ključni događaji odvijali i kako su geografski čimbenici utjecali na povijesni razvoj.

Sustav je koristan učiteljima i profesorima svih razreda osnovnoškolskog i srednjoškolskog obrazovanja koji imaju za cilj jednostavnije i brže sastavljanje ispita, provjere znanja učenika i ispravljanja njihovih odgovora. Pružajući učiteljima alate za kreiranje interaktivnih ispita, sustav također štedi vrijeme i povećava efikasnost u nastavi.

Sustavu se može pristupiti putem bilo kojeg uređaja s mogućnošću spajanja na internet. Kako bi započeli korištenje obrazovnog sustava, učenici i učitelj najprije moraju kreirati vlastiti korisnički račun te se prijaviti u njega. Nakon prijave, učitelj stvara predmete s pripadajućim učenicima, kreira skice ispita te ih objavljuje. Nakon objave skice ispita, učitelj nadzire proces rješavanja istog.

Učenici, nakon što su dodijeljeni svom predmetu, po trenutku objave ispita zaprimaju svoj primjerak te mogu započeti s njegovim rješavanjem. Također, učenici imaju pristup svim svojim prethodno riješenim ispitima, što im omogućuje praćenje vlastitog napretka i dodatno učenje iz prethodnih grešaka.

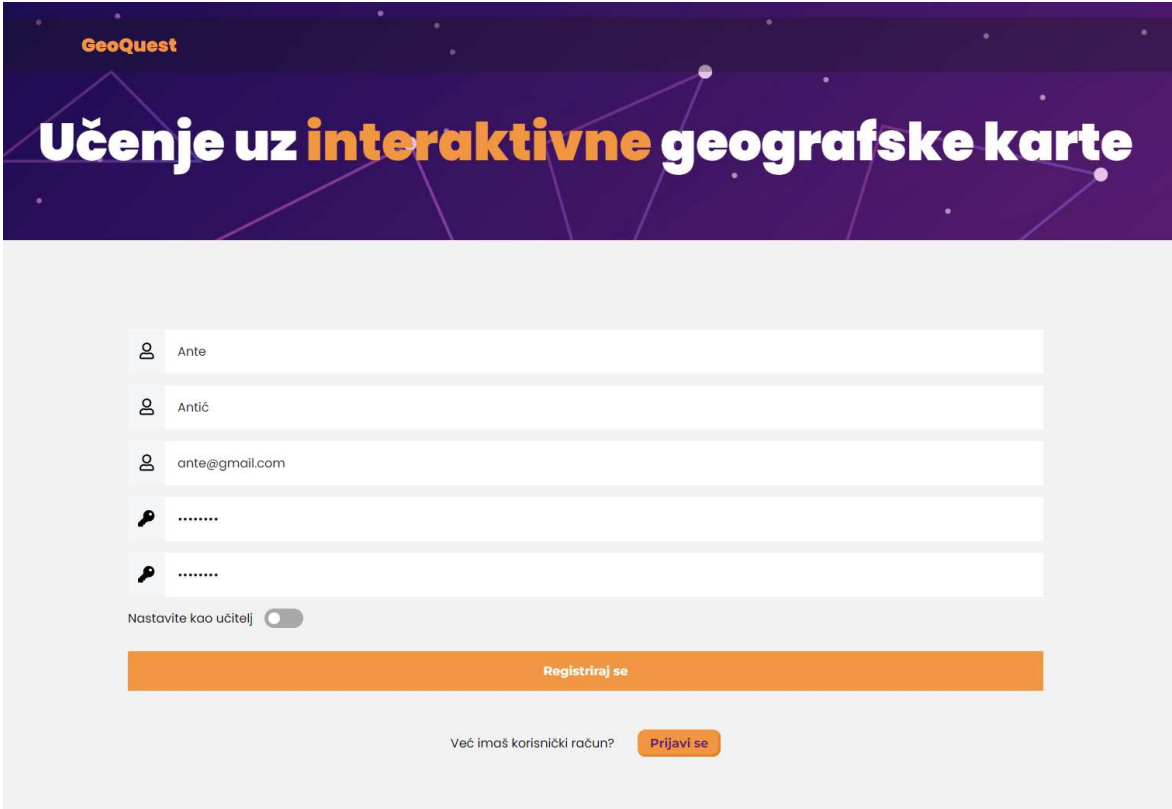
U nastavku su opisani opći procesi vezani uz sam korisnički račun, jednaki za obje uloge korisnika: registracija novog korisničkog računa (u ulozi učitelja ili učenika), prijava u sustav te odjava. Zatim su detaljnije opisane radnje specifične za pojedinu ulogu korisnika.

### **5.1. Korisnički procesi**

#### **5.1.1. Registracija**

Kako bi koristio sustav za učenje s pomoću interaktivnih geografskih karata, korisnik mora najprije stvoriti vlastiti korisnički račun. Korisnički račun stvara se ispravnim

ispunjavanjem registracijskog formulara (Slika 5.1) te njegovom potvrdom i slanjem. Korisnik treba unijeti svoje ime i prezime. Ovi podaci su važni učitelju kako bi mogao identificirati učenike te učenicima kako bi znali koji učitelj upravlja predmetom na kojem se nalaze. Zatim korisnik unosi email adresu koja mora biti jedinstvena na razini čitavog sustava. Svaki korisnički račun zaštićen je lozinkom. Korisnik upisuje željenu lozinku te je potvrđuje njenim ponovnim upisom. Iz sigurnosnih razloga, postavljen je zahtjev na duljinu lozinke od najmanje 8 znakova. Naposljetku, korisnik odabire željeni tip korisničkog računa, korisnički račun učitelja ili učenika. Ako su zadovoljena sva validacijska ograničenja, potvrdom se formular šalje na obradu, a korisnik se preusmjerava na stranicu za prijavu s porukom o uspješnoj registraciji.



GeoQuest

## Učenje uz **interaktivne** geografske karte

Nastavite kao učitelj

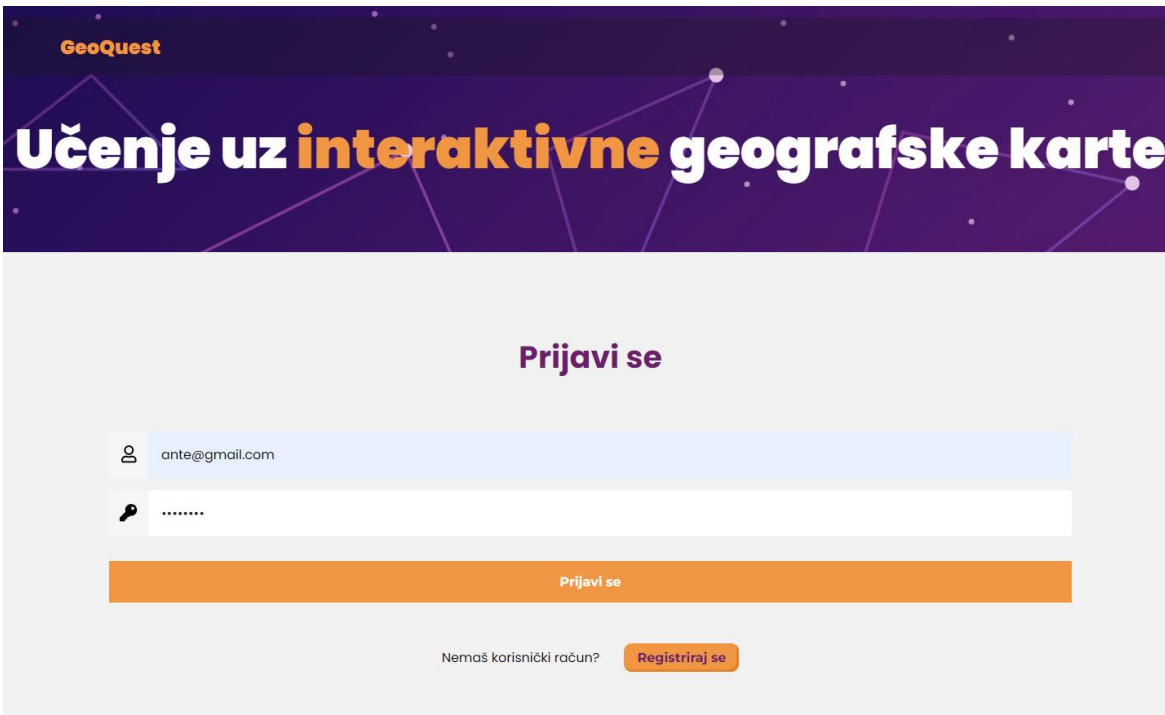
[Registriraj se](#)

Već imas korisnički račun? [Prijavi se](#)

**Slika 5.1** Registracija korisničkog računa

### 5.1.2. Prijava

Korisnik se, nakon uspješne registracije korisničkog računa, prijavljuje u sustav unosom email adrese i lozinke iskoristene u postupku registracije (Slika 5.2). Ako email i lozinka korisničkog računa nisu ispravni, sustav dojavljuje generičku poruku o grešci, ne otkrivajući tako potencijalnom napadaču nikakve informacije iskoristive za napad. Nakon uspješne prijave, sustav učitelja preusmjerava na stranicu s njegovim predmetima, dok učenika preusmjerava na stranicu s aktivnim neriješenim ispitima. Korisnik ostaje prijavljen u sustavu 120 minuta (2 sata), osim ako se eksplicitno ne odjavi. Nakon tog vremena, mora ponovno proći kroz postupak prijave u sustav.



The image shows a login interface for GeoQuest. At the top, the logo 'GeoQuest' is in orange. Below it, the main title 'Učenje uz interaktivne geografske karte' is displayed in white and orange on a dark purple background. The central heading 'Prijavi se' is in purple. The login form consists of two input fields: an email field with the text 'ante@gmail.com' and a password field with dots. Below these fields is a large orange button labeled 'Prijavi se'. At the bottom of the form, there is a link 'Nemaš korisnički račun?' and a smaller orange button labeled 'Registriraj se'.

Slika 5.2 Prijava korisnika u sustav

### 5.1.3. Odjava

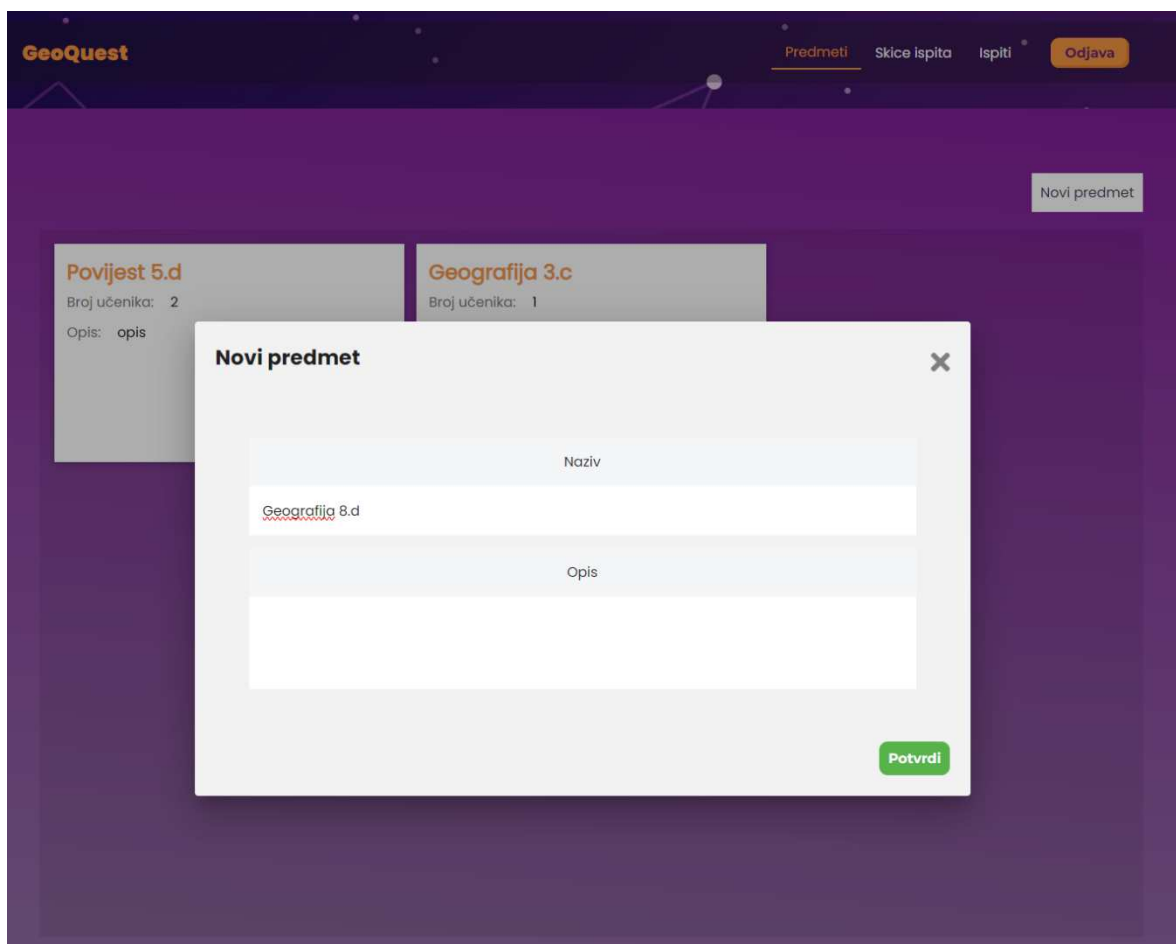
Odjavom prijavljenog korisnika iz sustava, klikom na gumb „Odjavi se“ u desnom kutu navigacijske trake, preusmjerava ga se na početnu stranicu s formom za prijavu u sustav.



## 5.2. Učitelj

### 5.2.1. Upravljanje predmetima

Na stranici „Predmeti“, učitelj pregledava popis već stvorenih vlastitih predmeta te ima mogućnost dodavanja novog predmeta. Glavna svrha predmeta je okupiti učenike na njemu te omogućiti objavljivanje ispita za taj predmet. Na popisu predmeta, za svaki je predmet prikazan naziv, opis i trenutni broj učenika na njemu. Klikom na gumb „Novi predmet“, otvara se skočni prozor (Slika 5.3) s formom za unos naziva i opcionalnog opisa predmeta. Naziv predmeta obično sadrži naziv predmeta i konkretni razred. Kada učitelj klikne na određeni predmet, otvara se nova stranica s detaljnim informacijama o tom predmetu.



Slika 5.3 Dodavanje novog predmeta

Na stranici predmeta (Slika 5.4) ponovo se nalaze opće informacije o predmetu kao što su naziv, opis i broj učenika na njemu, ali i dvije tablice:

- Popis učenika na predmetu: Ova tablica sadrži popis svih učenika koji su upisani na taj predmet. Za svakog učenika prikazane su osnovne informacije kao što su ime, prezime i email adresa.
- Popis ispita za predmet: Ova tablica prikazuje sve ispite koji su objavljeni za taj predmet. Za svaki ispit prikazani su podaci kao što su naziv, trajanje ispita, napredak rješavanja ispita i njegov status. Učitelj ima mogućnost pregleda ispita klikom na njegov naziv te zatvaranja ispita ako je njegov status „U tijeku“.

The screenshot shows the 'GeoQuest' application interface. At the top, there is a navigation bar with the logo 'GeoQuest' on the left and menu items 'Predmeti', 'Skice ispita', 'Ispiti', and 'Odjava' on the right. The main content area is divided into two columns. The left column, titled 'Predmet', contains the following information: Naziv: Povijest 5.d, Broj učenika: 2, and Opis: opis. The right column, titled 'Učenici', contains a table with the following data:

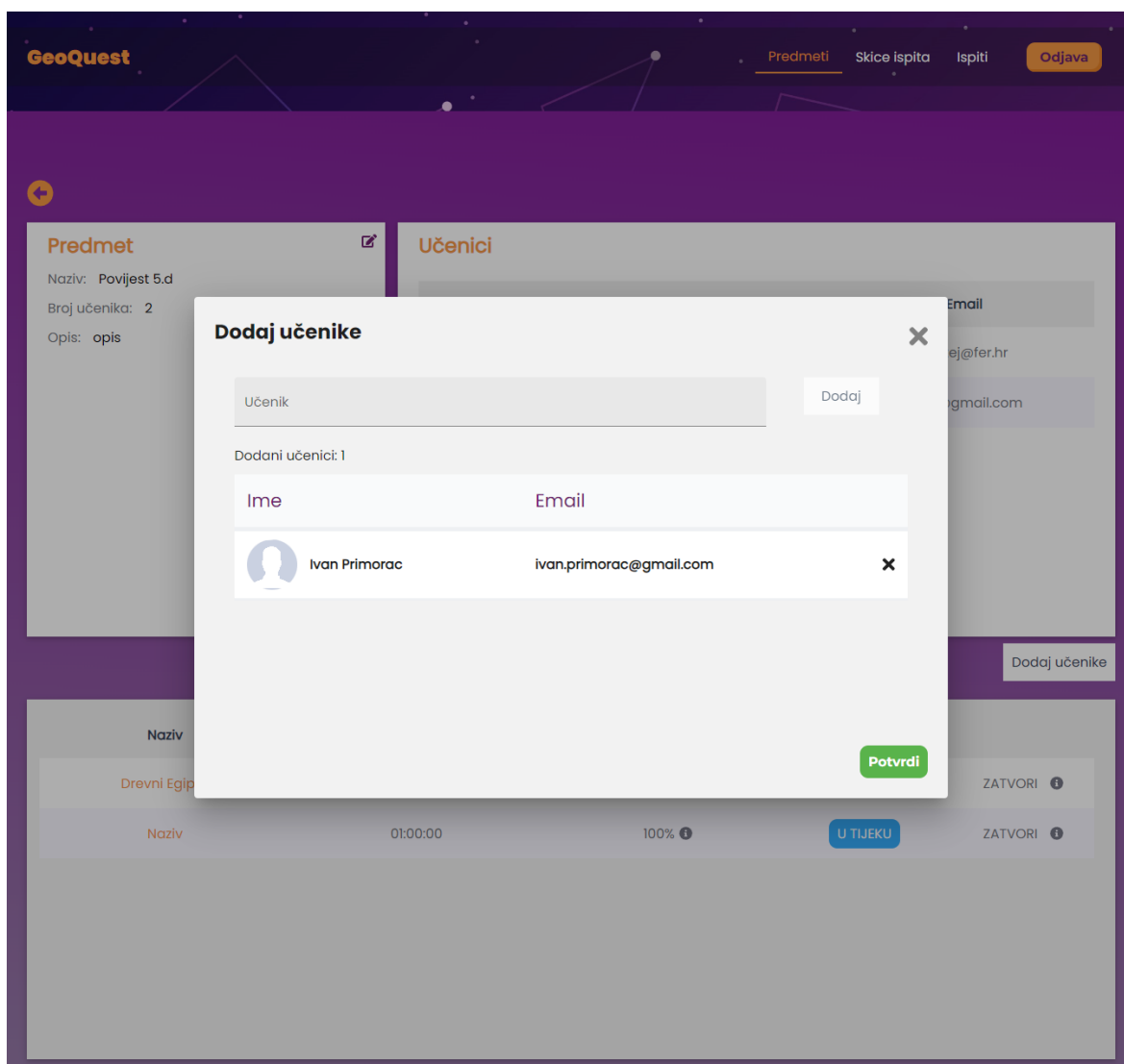
| Ime   | Prezime | Email          |
|-------|---------|----------------|
| Matej | Đurin   | matej@fer.hr   |
| Ante  | Antić   | ante@gmail.com |

Below the 'Učenici' table is a button labeled 'Dodaj učenike'. At the bottom of the interface is a table listing exams with the following data:

| Naziv          | Trajanje | Napredak | Status             |
|----------------|----------|----------|--------------------|
| Drevni Egipat  | 00:10:00 | 100% ⓘ   | U TIJEKU ZATVORI ⓘ |
| Hrvatski otoci | 01:00:00 | 100% ⓘ   | U TIJEKU ZATVORI ⓘ |

Slika 5.4 Stranica predmeta s popisom učenika na predmetu i ispita za predmet

Uz uređivanje općih informacija o predmetu, učitelj ima i mogućnost dodavanja novih učenika na predmet. Klikom na gumb „Dodaj učenike“, otvara se skočni prozor (Slika 5.5) s poljem za pretraživanje korisničkih računa učenika. Nakon pronalaska željenih učenika i potvrđivanjem radnje, odabrani učenici se dodaju na predmet.

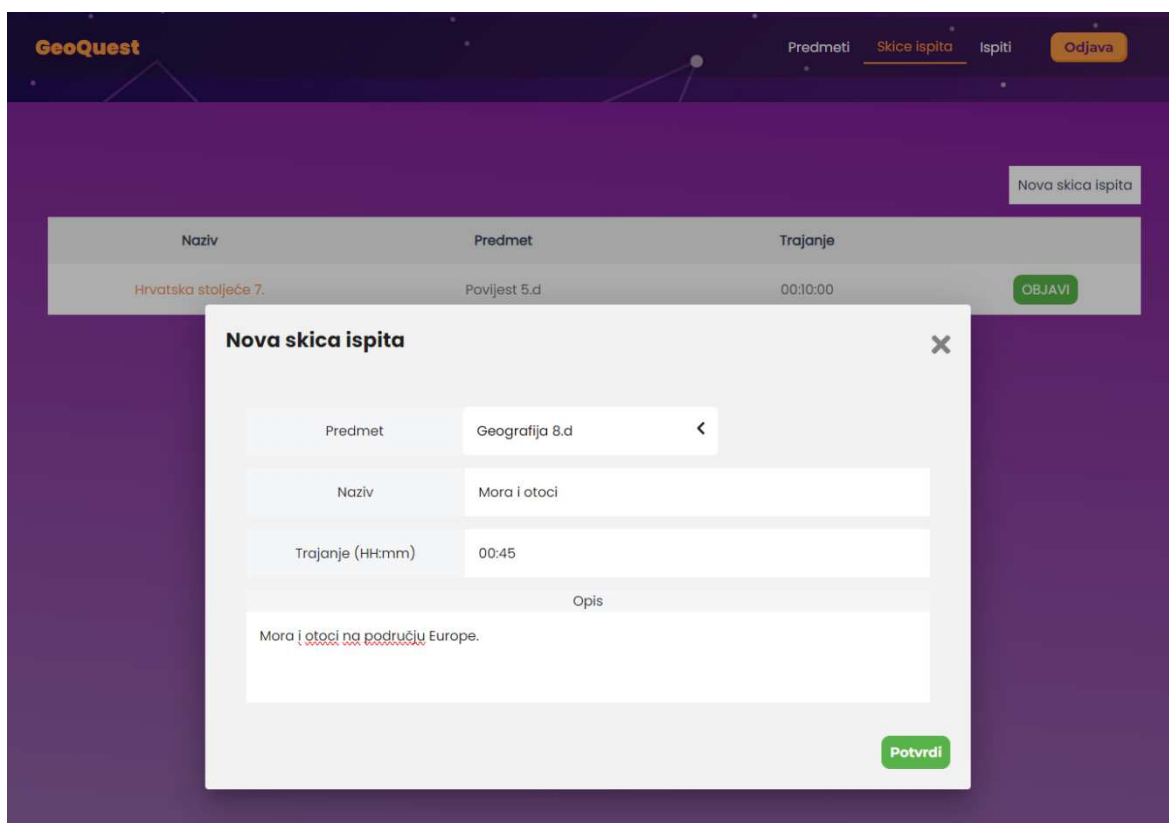


**Slika 5.5** Dodavanje učenika na predmet

### 5.2.2. Upravljanje skicama ispita

Popis postojećih skica ispita nalazi se na stranici „Skice ispita“. Klikom na gumb „Nova skica ispita“ otvara se skočni prozor (Slika 5.6) s formularom za stvaranje nove skice ispita. Učitelj je obavezan odabrati predmet za koji je skica ispita namijenjena, upisati naziv

skice ispita, trajanje rješavanja ispita (kad skica bude objavljena, u formatu „HH:mm“, gdje „HH“ predstavlja sate, a „mm“ minute ) te opcionalno dodati opis. Klikom na naziv pojedine skice ispita u tablici, učitelja se preusmjerava na stranicu s detaljima skice.



**Slika 5.6** Stvaranje nove skice ispita

Na stranici skice ispita (Slika 5.7), učitelj ima mogućnost pregledavanja i uređivanja podataka koje je unio prilikom stvaranja skice ispita. Također, učitelj vidi popis zadataka te može dodati nove zadatke u skicu ispita klikom na gumb „Stvori zadatak“. Ako je učitelj zadovoljan s pripremljenom skicom ispita i pripadnim zadacima, može je objaviti klikom na gumb „Objavi“. Skica ispita se može objaviti pod uvjetom da sadrži barem jedan zadatak i da na predmetu za koji je skica napravljena postoji barem jedan učenik. Objavom skice ispita, skica se trajno arhivira i više nije dostupna za naknadnu izmjenu ni uređivanje. Od objavljene skice ispita automatski se stvara ekvivalentan ispit koji se distribuira svim učenicima koji pohađaju taj predmet na rješavanje. Po uspješnoj objavi skice ispita, sustav učitelja preusmjerava na stranicu s detaljima i analitikom upravo objavljenog ispita.

The screenshot shows the 'GeoQuest' interface for creating a sketch exam. At the top, there is a navigation bar with 'Predmeti', 'Skice ispita', 'Ispiti', and 'Odjava'. Below this, a 'Stvori zadatak' button is present. The main content area is titled 'Mora i otoci' and includes the following details:
 

- Predmet:** Geografija 8.d
- Opis:** Mora i otoci na području Europe.
- Trajanje:** 00:45:00

 A green 'OBJAVI' button is located to the right of these details. Below the text is a map of Europe with the Black Sea highlighted in blue. To the right of the map, there is a form for creating a task:
 

- A text input field labeled '1. zadatak'.
- Two dropdown menus: 'Tip zadatka' and 'Označi na karti'.
- A text input field labeled 'Tekst zadatka'.
- A button labeled 'Označite na karti Crno more!'.

 A trash icon is visible in the top right corner of the map area.

Slika 5.7 Stranica skice ispita

### 5.2.3. Upravljanje zadacima

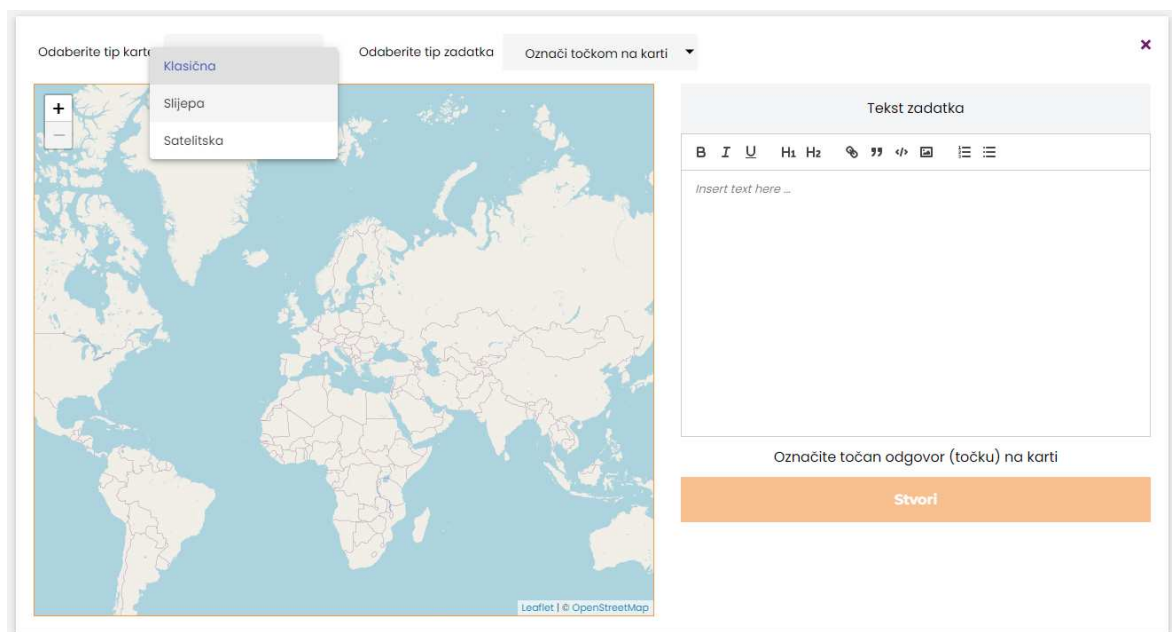
Na stranici skice ispita, učitelj uz osnovne informacije o skici ispita vidi i trenutni popis zadataka. Učitelj ima mogućnost brisanja postojećih zadataka te stvaranja novih. Klikom na gumb „Stvori zadatak“ otvara se sučelje za stvaranje novog zadatka (Slika 5.8). Sučelje sadrži geografsku kartu, polje za unos teksta zadatka te dva padajuća izbornika za odabir tipa karte i tipa zadatka. Učitelj se može slobodno kretati po karti, uvećavati je ili umanjivati te proizvoljno centrirati. Podaci o uvećanju karte i njenom centru spremaju se kao dio zadatka kako bi se geografska karta na isti način mogla prikazati učenicima u trenutku rješavanja ispita. U gornjem desnom kutu sučelja nalazi se gumb za odustajanje od stvaranja novog zadatka, dok se klikom na gumb „Stvori“ stvara novi zadatak.

Iz padajućeg izbornika za odabir tipa karte, učitelj može birati između tri različita tipa:

- klasična
- slijepa
- satelitska

Također, učitelj bira jedan od sljedećih pet mogućih tipova zadatka:

- Označi točkom na karti
- Označi na karti
- Odaberi točku na karti
- Odaberi na karti
- Obično pitanje



**Slika 5.8** Sučelje za stvaranje novog zadatka

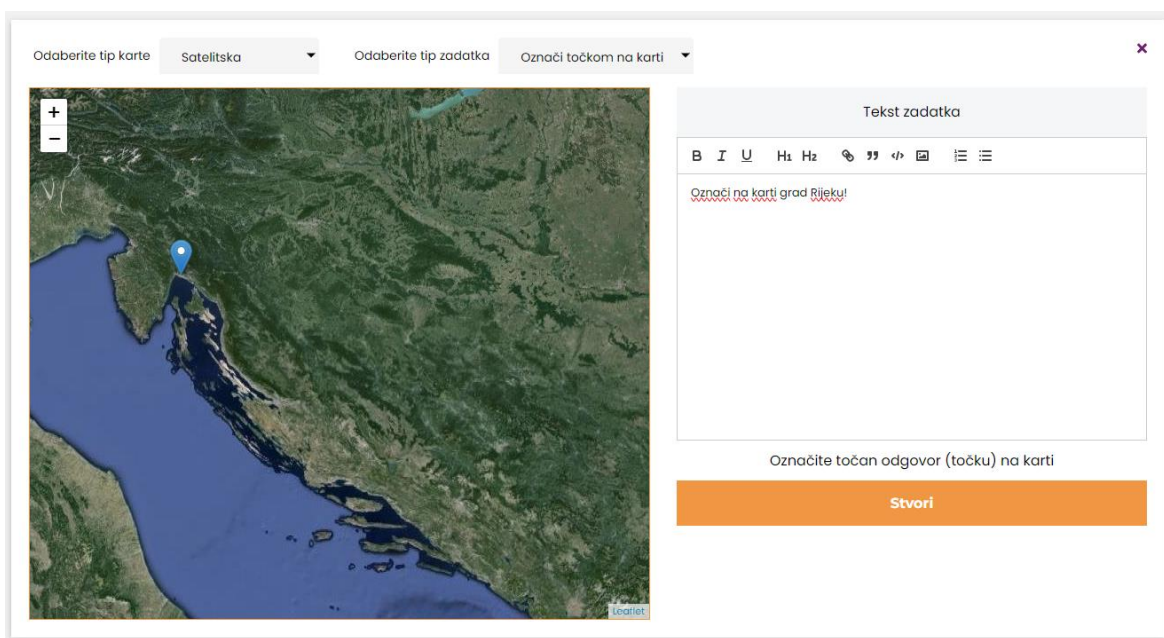
Za stvaranje novog zadatka, učitelj mora unijeti tekst zadatka, odabrati željeni tip karte i tip zadatka te označiti ili unijeti odgovore na karti, ovisno o odabranom tipu zadatka.

## 5.2.4. Tipovi zadataka

Prilikom stvaranja novog zadatka, ovisno o odabranom tipu zadatka, učitelj mora na karti precizno označiti točan odgovor. Ova informacija omogućava sustavu automatsko ispravljanje zadataka koje je riješio učenik (osim za tip zadatka „Obično pitanje“), što značajno olakšava posao učitelju prilikom ispravljanja ispita. Zadatak bilo kojeg tipa, ako je točno odgovoren, donosi jedan bod, dok netočno odgovoreni ili neodgovoreni zadaci donose nula bodova.

- **Označi točkom na karti**

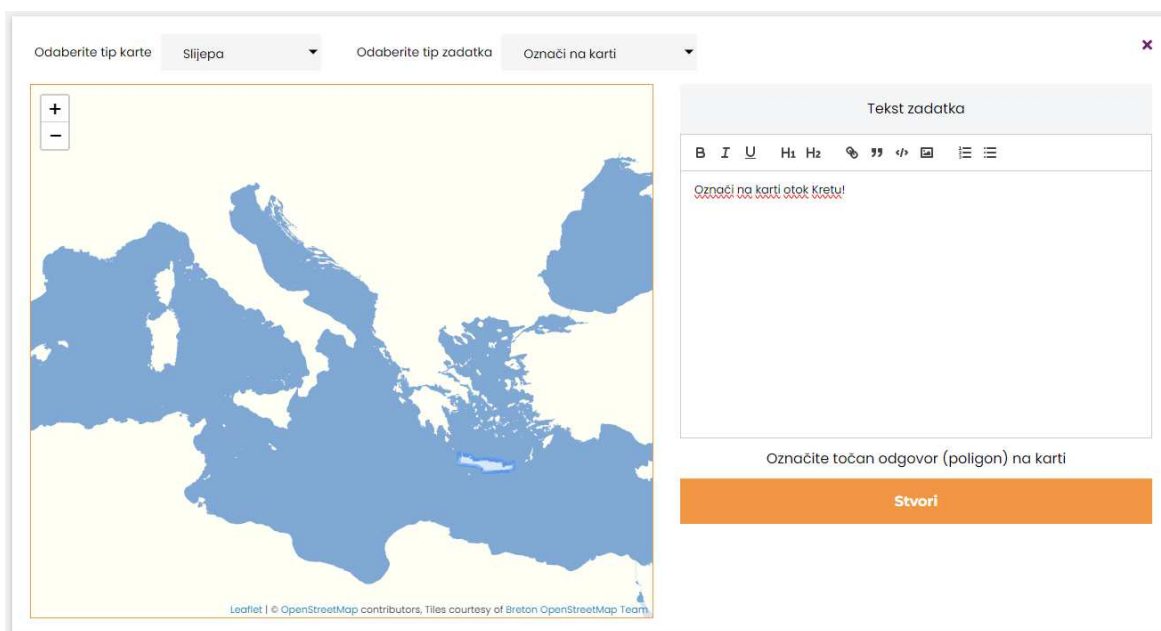
Tip zadatka koji zahtijeva precizan odgovor točkom na karti. Prilikom stvaranja zadatka, učitelj mora na karti točkom označiti točan odgovor (Slika 5.9). Sustav automatski ispravlja ovaj tip zadatka tako da izračunava zračnu udaljenost između točno označene točke i učenikovog odgovora na karti. Ako je ta udaljenost manja od 11.1 kilometara, učenikov odgovor se smatra točnim.



Slika 5.9 Sučelje za stvaranje novog zadatka tipa „Označi točkom na karti“

- **Označi na karti**

Tip zadatka koji zahtjeva odgovor poligonom na karti. Prilikom stvaranja zadatka, učitelj mora na karti poligonom označiti točan odgovor (Slika 5.10). Sustav automatski ispravlja ovaj tip zadatka tako da izračunava postotak preklapanja površina točno označenog poligona i učenikovog odgovora na karti. Ako je postotak veći od 70 %, učenikov odgovor se smatra točnim.

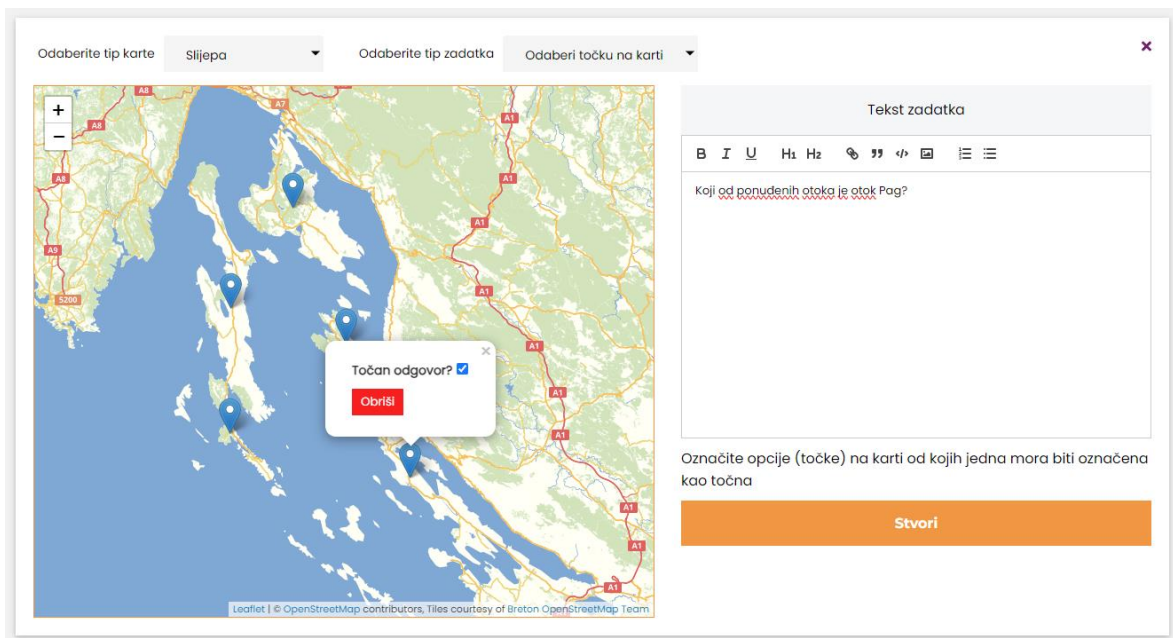


**Slika 5.10** Sučelje za stvaranje novog zadatka tipa „Označi na karti“

- **Odaberi točku na karti**

Tip zadatka koji zahtjeva odabir jedne od ponuđenih točki na karti. Prilikom stvaranja zadatka, učitelj mora na karti označiti dvije ili više točaka, od kojih isključivo jednu mora označiti kao točnu (Slika 5.11). Sustav automatski ispravlja ovaj tip zadatka.

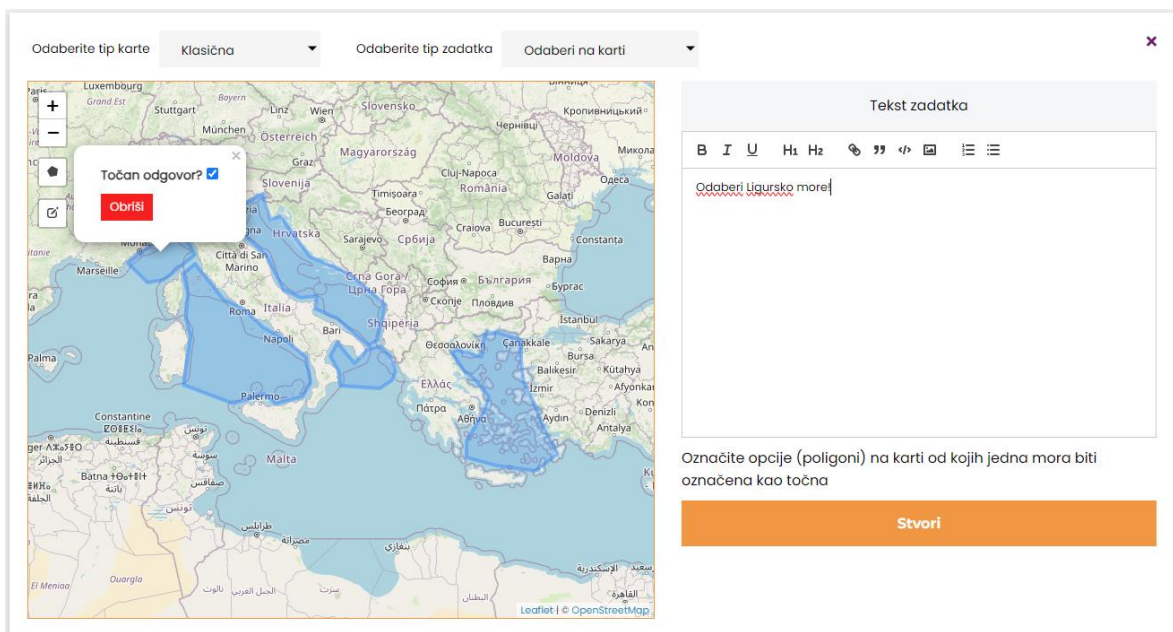




Slika 5.11 Sučelje za stvaranje novog zadatka tipa „Odaberi točku na karti“

- **Odaberi na karti**

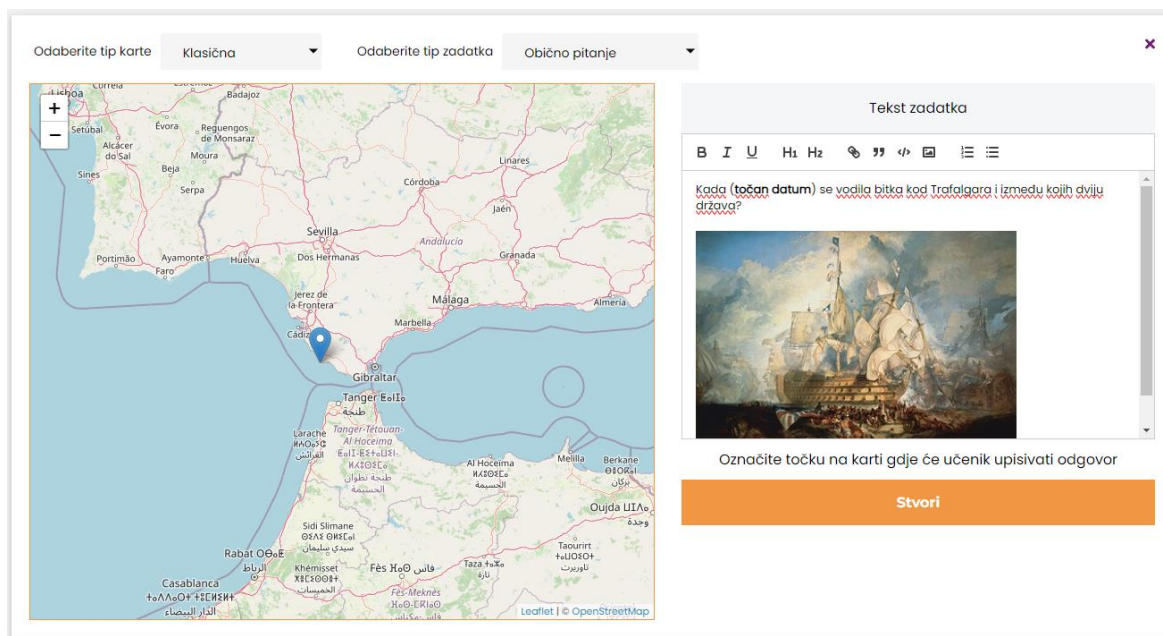
Tip zadatka koji zahtjeva odabir jednog od ponuđenih poligona na karti. Prilikom stvaranja zadatka, učitelj mora na karti označiti dva ili više poligona, od kojih isključivo jedan mora označiti kao točan (Slika 5.12). Sustav automatski ispravlja ovaj tip zadatka.



Slika 5.12 Sučelje za stvaranje novog zadatka tipa „Odaberi na karti“

- **Obično pitanje**

Tip zadatka koji zahtjeva tekstualni odgovor. Prilikom stvaranja zadatka, učitelj mora na karti označiti točku (Slika 5.13). Označena točka služi kao referentno mjesto gdje će učenik upisivati svoj odgovor. U ovom tipu zadatka, odgovor nije direktno na karti, već je povezan s kartom koja pruža kontekst. Sustav ne može automatski ispraviti ovaj tip zadatka, stoga je potrebna intervencija učitelja za evaluaciju učenikovog odgovora.



Slika 5.13 Sučelje za stvaranje novog zadatka tipa „Obično pitanje“

## 5.2.5. Upravljanje ispitima

Na stranici „Ispiti“ nalazi se tablica sa svim objavljenim ispitima (Slika 5.14). Za svaki ispit navedene su ključne informacije: naziv, predmet, trajanje, napredak rješavanja i status. Ako je ispit još uvijek otvoren, odnosno ako je status ispita „U tijeku“, učitelj ima mogućnost zatvaranja ispita. Zatvaranjem ispita, njegov status prelazi u „Zatvoren“, čime učenici koji nisu ni započeli ispit više neće moći pristupiti istom.

| Naziv         | Predmet        | Trajanje | Napredak | Status   |
|---------------|----------------|----------|----------|----------|
| Drevni Egipat | Povijest 5.d   | 00:10:00 | 100.00%  | U TIJEKU |
| Naziv         | Povijest 5.d   | 01:00:00 | 100.00%  | ZATVOREN |
| Mora i otoci  | Geografija 8.d | 00:45:00 | 0.00%    | U TIJEKU |

Slika 5.14 Stranica objavljenih ispita

Klikom na naziv ispita, učitelja se preusmjerava na detaljan pregled objavljenog ispita. Detaljan pregled (Slika 5.15) sadrži opće informacije o ispitu, kao što su naziv, opis, predmet, trajanje i status, te analitiku koja uključuje postotak riješenosti ispita, prosječan broj bodova, prosječno trajanje rješavanja ispita, kao i napredak rješavanja i ispravljanja ispita.

**Mora i otoci** U TIJEKU

Opis ispita  
Mora i otoci na području Europe.

Predmet  
Geografija 8.d

Trajanje  
00:45:00

Prosjek trajanja ispita  
00:05:10

Ukupan broj bodova  
6

ZATVORI

**66.67%**  
Riješenost ispita

Prosječan broj bodova je 4.00

Napredak rješavanja ispita

Napredak ispravljanja ispita

| Učenik        | Proteklo vrijeme | Bodovi | Riješeno | Završeno | Ispravljeno |
|---------------|------------------|--------|----------|----------|-------------|
| Matej Đurin   | 00:05:10         | 4      | Da       | Da       | Da          |
| Ivan Primorac | 00:00:00         | 0      | Ne       | Ne       | Ne          |

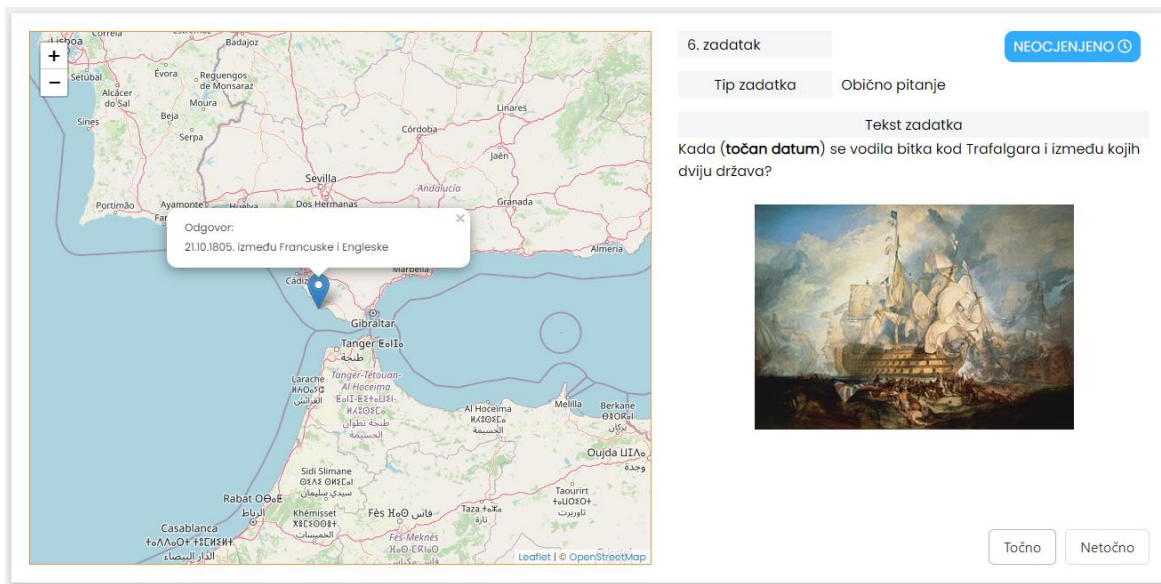
Slika 5.15 Stranica objavljenog ispita

U dodatnoj tablici prikazan je popis inačica ispita od svakog pojedinog učenika. Za svaku inačicu ispita prikazano je ime učenika kojemu pripada, te status rješavanja. Ako je učenik riješio ispit, prikazano je proteklo vrijeme rješavanja i ostvareni bodovi. Također je navedeno je li ispit završen te je li u potpunosti ispravljen.

### 5.2.6. Pregled i ispravljanje ispita

Na stranici detaljnog pregleda objavljenog ispita, u dodatnoj tablici s popisom inačica ispita od svakog učenika, učitelj može klikom na ime učenika pristupiti pregledu riješenog ispita tog učenika, pod uvjetom da je inačica ispita riješena i završena.

Na vrhu stranice za pregled i ispravljanje ispita prikazani su ostvareni broj bodova, postotak riješenosti te status ispravljenosti ispita. Učitelj također ima uvid u sve zadatke i učenikove odgovore te legendu koja pojašnjava oznake na karti (razlikovanje točnog odgovora od učenikovog odgovora). Svaki zadatak je označen kao točan, netočan ili neocijenjen u slučaju zadataka tipa „Obično pitanje“. Učitelj mora ručno utvrditi točnost neocijenjenih zadataka (Slika 5.16). Nakon što su svi zadaci tipa „Obično pitanje“ ručno ocijenjeni, status ispravljenosti ispita prelazi iz „Djelomično ispravljeno“ u „Ispravljeno“.

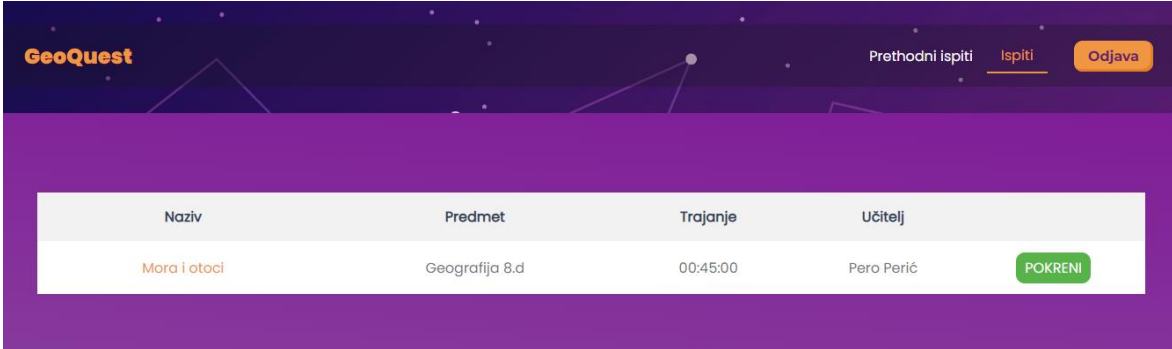


The screenshot displays a quiz interface. On the left, a map of the Mediterranean region is shown, with a blue pin marking the location of Gibraltar. A white text box over the map contains the answer: "Odgovor: 21.10.1805. između Francuske i Engleske". On the right, the quiz question is displayed: "6. zadatak" (6. question) with a status of "NEOCIJENJENO" (Not graded). The question type is "Obično pitanje" (Regular question). The question text asks: "Kada (točan datum) se vodila bitka kod Trafalgara i između kojih dviju država?" (When (correct date) was the Battle of Trafalgar fought and between which two countries?). Below the question is a small image of a sailing ship at sea. At the bottom right, there are two buttons: "Točno" (Correct) and "Netočno" (Incorrect).

Slika 5.16 Primjer neocijenjenog zadatka

## 5.3. Učenik

Početna stranica u sustavu za učenika je stranica „Ispiti“. Na njoj se nalazi tablica (Slika 5.17) sa svim učeniku dostupnim ispitima za rješavanje. Klikom na naziv ispita učenik se preusmjerava na stranicu s nazivom i opisom ispita te ostalim informacijama poput naziva predmeta, maksimalnog dopuštenog trajanja rješavanja ispita i imena učitelja. Klikom na gumb „Pokreni“, učenik započinje s rješavanjem ispita.



The screenshot shows the 'Ispiti' (Exams) page in the GeoQuest system. The page has a dark purple header with the 'GeoQuest' logo on the left and navigation links for 'Prethodni ispiti', 'Ispiti', and 'Odjava' on the right. Below the header is a table with the following columns: 'Naziv', 'Predmet', 'Trajanje', and 'Učitelj'. A single row is visible in the table, representing an exam titled 'Mora i otoci' in the subject 'Geografija 8.d', with a duration of '00:45:00' and the teacher 'Pero Perić'. To the right of this row is a green button labeled 'POKRENI'.

| Naziv        | Predmet        | Trajanje | Učitelj    |
|--------------|----------------|----------|------------|
| Mora i otoci | Geografija 8.d | 00:45:00 | Pero Perić |

Slika 5.17 Stranica sa učeniku dostupnim ispitima


### 5.3.1. Rješavanje ispita

Prilikom rješavanja ispita (Slika 5.18), navigacijska traka sakriva ostale stranice sustava koje su inače dostupne učeniku. Na vrhu stranice prikazuje se odbrojavanje preostalog vremena za rješavanje ispita. Zadaci su poredani jedan ispod drugoga, a uz svaki je naveden njegov redni broj, tip i tekst zadatka. Ovisno o tipu zadatka, od učenika se očekuje određeni odgovor na karti. Nakon što učenik unese odgovor, isti se automatski sprema u sustav.



**GeoQuest**

Preostalo vrijeme: 42:02




1. zadatak

Tip zadatka    Označi na karti

Tekst zadatka

Označite na karti Crno more!



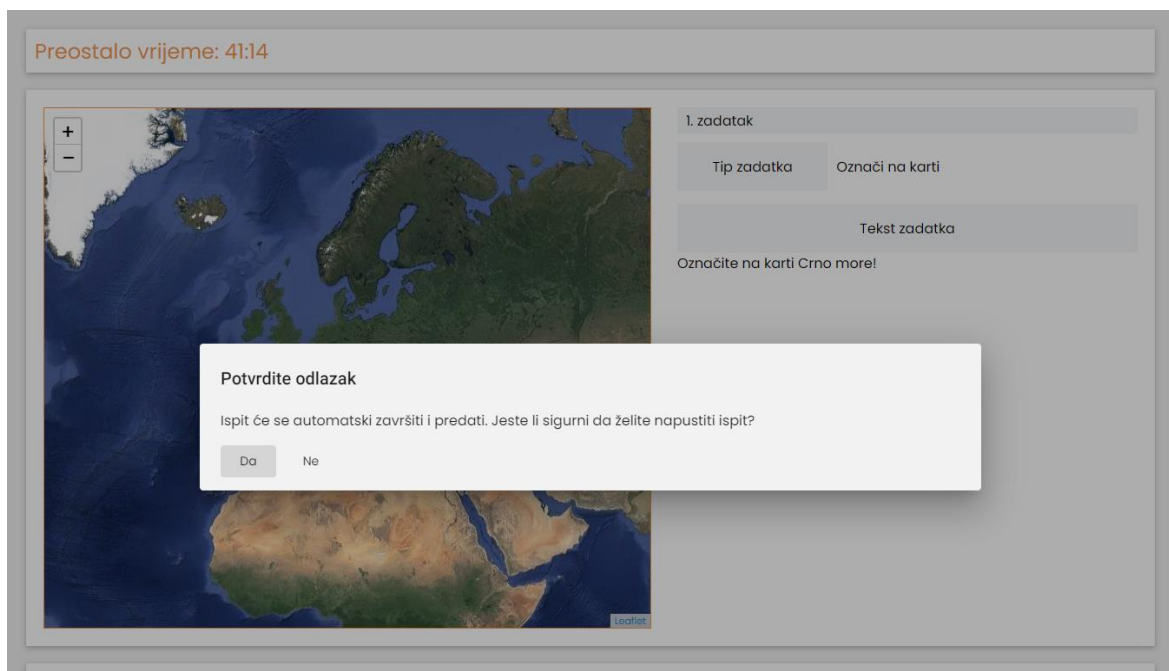
2. zadatak

Tip zadatka    Označi točkom na karti

Tekst zadatka

**Slika 5.18** Rješavanje ispita

Ako učenik pokuša napustiti ispit (navigiranjem u pregledniku ili klikom na „GeoQuest“), otvara se skočni prozor (Slika 5.19) s porukom upozorenja. Ako učenik potvrdi svoj odlazak klikom na „Da“ u skočnom prozoru, rješavanje ispita će se automatski završiti i ispit će se predati. Istekne li preostalo vrijeme za rješavanje ispita, ispit se također automatski završava i predaje. Na dnu stranice, ispod posljednjeg zadatka, nalazi se gumb „Predaj ispit“. Klikom na taj gumb, ispit se završava i predaje, a učenika se preusmjerava na stranicu za pregled prethodno riješenih ispita.



Slika 5.19 Skočni prozor za potvrdu prekida rješavanja ispita

### 5.3.2. Pregled rezultata ispita

Učenik na stranici „Prethodni ispiti“ može pregledati sve svoje prethodno riješene ispite. Klikom na naziv ispita, učenika se preusmjerava na stranicu za pregled rezultata odabranog ispita.

Na vrhu stranice (Slika 5.20) prikazani su ostvareni broj bodova i postotak riješenosti ispita, dok je u gornjem desnom kutu vidljiv status ispravljenosti ispita. Za svaki zadatak prikazani su redni broj, tip, točnost, legenda i tekst zadatka. Legenda objašnjava oznake na karti, omogućujući razlikovanje točnog odgovora od učenikovog odgovora. Zadaci tipa „Obično pitanje“ imaju status „Neocijenjeno“ dokle god učitelj ručno ne odredi točnost zadatka.

**GeoQuest** Prethodni ispiti Ispiti [Odjava](#)

---


**Mora i otoci - Matej Đurin** DJELOMIČNO ISPRAVLJENO

Opis  
Mora i otoci na području Europe.

Ostvaren broj bodova  
3/6

Postotak riješenosti  
60.00%

---



Leaflet

1. zadatak TOČNO ✓


Tip zadatka Označi na karti

Tekst zadatka

Označite na karti Crno more!

■ - Točan odgovor    ■ - Vaš odgovor

---



2. zadatak NETOČNO ☹

Slika 5.20 Pregled riješenog ispita



## 6. Eksperiment i rezultati

U lipnju 2024. godine proveden je eksperiment s ciljem testiranja implementiranog sustava za učenje s pomoću interaktivnih geografskih karata. Eksperiment je uključivao deset dobrovoljnih sudionika, za koje su unutar sustava stvoreni pripadni korisnički računi učenika. Sudionici su rješavali po dva ispita, pri čemu je svaki sadržavao deset zadataka, pokrivajući svih pet mogućih tipova zadataka. Prvi ispit, pod nazivom „Feničani i Greci“, fokusirao se na ispitivanje povijesnog znanja sudionika. Pitanja su bila vezana uz grčke i feničke kolonije, polise, lokacije važnih bitki i slično. Drugi ispit, nazvan „Reljef, rijeke i jezera Hrvatske“, provjeravao je poznavanje geografskih obilježja Hrvatske. Ovaj ispit uključivao je zadatke vezane uz prepoznavanje i označavanje glavnih reljefnih oblika, rijeka i jezera unutar Hrvatske. Sudionici su imali na raspolaganju 30 minuta za rješavanje pojedinog ispita.

Prosječan broj ostvarenih bodova na prvom ispitu pod nazivom „Feničani i Greci“ bio je 7 od maksimalnih 10 bodova. Najniži postignut rezultat sudionika eksperimenta na prvom ispitu bio je 4 boda, dok je najviši bio 9 bodova. Prosjek vremena potrebnog za rješavanje prvog ispita iznosio je 4 minute i 53 sekunde. Na drugom ispitu pod nazivom „Reljef, rijeke i jezera Hrvatske“ prosječan broj ostvarenih bodova bio je 6.86 od maksimalnih 10 bodova. Najniži postignuti rezultat sudionika eksperimenta na drugom ispitu bio je 5 boda, dok je najviši bio 9 bodova. Prosjek vremena potrebnog za rješavanje drugog ispita iznosio je 4 minute i 38 sekundi.

Po završetku rješavanja ispita, sudionicima eksperimenta je dana anonimna anketa (Slika 6.1). Svrha ankete bila je prikupiti povratne informacije sudionika o njihovom iskustvu korištenja sustava za učenje s pomoću interaktivnih geografskih karata. Anketa je imala za cilj utvrditi koliko je implementirani sustav jednostavan za korištenje te koliko su jednostavni i intuitivni zadaci na interaktivnoj geografskoj karti. Anketa je sadržavala pitanja s kratkim tekstualnim odgovorima te tvrdnje s pripadajućim ljestvicama slaganja od 1 do 5, gdje je 1 značilo „u potpunosti se ne slažem“, a 5 „u potpunosti se slažem“. Većina sudionika (90 %) složila se s tvrdnjom da je rješavanje zadataka na karti jednostavno i intuitivno te su u anketi dali ocjene 4 ili 5.

# Sustav za učenje pomoću interaktivnih geografskih karata - anketa

Prikupljanje povratnih informacija o iskustvima u korištenju sustava za učenje s pomoću interaktivnih geografskih karata razvijenog u okviru diplomskog rada.

[Sign in to Google](#) to save your progress. [Learn more](#)

\* Indicates required question

Sustav je jednostavan za korištenje \*

1 2 3 4 5  
U potpunosti se ne slažem      U potpunosti se slažem

Rješavanje zadataka na karti je jednostavno i intuitivno \*

1 2 3 4 5  
U potpunosti se ne slažem      U potpunosti se slažem

Označavanje na interaktivnoj karti jednostavnije je od označavanja na karti na papiru \*

1 2 3 4 5  
U potpunosti se ne slažem      U potpunosti se slažem

Što vam se posebno svidjelo u sustavu?

Your answer

Što vam se nije svidjelo u sustavu?

Your answer

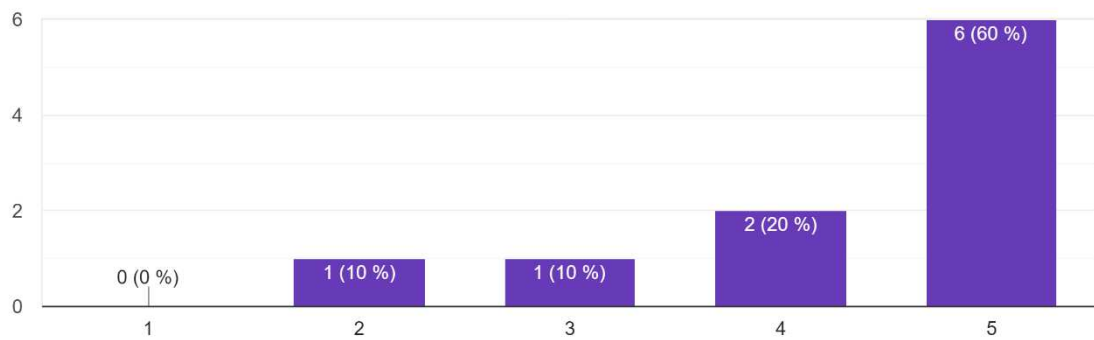
Submit

Clear form

Slika 6.1 Anonimna anketa o iskustvu korištenja implementiranog sustava

S tvrdnjom da je označavanje na karti unutar sustava jednostavnije od klasičnog označavanja na papirnoj karti složilo se 80 % sudionika eksperimenta (Slika 6.2). Sudionicima se sviđelo to što su rezultati ispita bili dostupni odmah po završetku rješavanja. Međutim, nisu bili zadovoljni time što su odgovori na interaktivnoj karti morali biti označeni s velikom preciznošću.

Označavanje na interaktivnoj karti jednostavnije je od označavanja na karti na papiru  
10 odgovora



**Slika 6.2** Graf raspodjele odgovora (1 - U potpunosti se ne slažem, 5 - U potpunosti se slažem)

## Zaključak

Današnje generacije učenika rođene su i odrasle uz stalnu prisutnost digitalnih tehnologija te iste koriste gotovo svakodnevno s velikom lakoćom. Iz tog se razloga primjena digitalnih tehnologija nameće kao logičan izbor pri osmišljavanju što učinkovitijih obrazovnih strategija za današnje generacije učenika, uz održavanje ravnoteže između tradicionalnog i novog digitalnog pristupa učenju. Integracija digitalnih tehnologija u obrazovne procese donosi prednosti poput adaptivnog/prilagodljivog učenja te poticanja angažiranosti i interaktivnosti učenika. Prilikom upotrebe digitalnih tehnologija u obrazovnom procesu, izrazito je važno obratiti pažnju na pedagošku vrijednost upotrebe tehnologije, s ciljem izbjegavanja potencijalnih negativnih utjecaja poput smanjene društvene interakcije učenika ili pojave ovisnosti o digitalnim uređajima.

Nakon pregleda stručne literature iz područja tehnologijom podržanog obrazovanja te detaljne analize postojećih aplikacija za učenje s pomoću interaktivnih geografskih karti, prikupljene informacije iskorištene su za dizajn i razvoj vlastitog sustava za učenje koristeći interaktivne geografske karte. Obrazovni sustav razvijen u okviru ovog diplomskog rada testiran je u praksi, no izvan uobičajenog nastavnog okruženja. Nakon provedenog testiranja, uočena su sljedeća moguća poboljšanja sustava. Osim funkcionalnosti za provođenje ispita, sustav bi mogao omogućiti učiteljima kreiranje vježbi za učenike. Time bi sustav postao pogodan ne samo za provjeru znanja, već i za učenje i praktične vježbe. Dodatno, uvođenje novih tipova zadataka, poput višestrukih izbora (ABCD) i pitanja točno-netočno, obogatilo bi sustav i proširilo mogućnosti njegove primjene. Trenutno, svaki zadatak u sustavu nosi jedan bod ako je točno riješen. Uvođenjem mogućnosti dodjeljivanja različitih bodovnih vrijednosti zadacima, teži zadaci mogli bi nositi više bodova. Tako bi ispiti mogli biti još više prilagođeni potrebama i željama učitelja. Osim toga, uvođenje negativnih bodova za netočno odgovorene zadatke smanjilo bi broj učenika koji pogađaju odgovore.

## Literatura

- [1] P. Pant. "How Does Technology Support Teaching and Learning?" Suraasa. <https://www.suraasa.com/blog/technology-in-teaching?var=as1> (pristupljeno 25. svibnja 2024.).
- [2] "What is 'E-learning'." TheEconomicTimes. <https://economictimes.indiatimes.com/definition/e-learning> (pristupljeno 25. svibnja 2024.).
- [3] "The history of e-learning." Talentlms. <https://www.talentlms.com/elearning/history-of-elearning> (pristupljeno 25. svibnja 2024.).
- [4] "Explained: What is Zoom?" WebWise.ie. <https://www.webwise.ie/parents/explainers/explained-what-is-zoom/> (pristupljeno 3. lipnja 2024.).
- [5] "What is Microsoft Teams?" Microsoft. <https://support.microsoft.com/en-us/topic/what-is-microsoft-teams-3de4d369-0167-8def-b93b-0eb5286d7a29> (pristupljeno 3. lipnja 2024.).
- [6] Gulati, S. (2008). Technology-enhanced learning in developing nations: A review. *The International Review of Research in Open and Distributed Learning*, 9(1).
- [7] Maphalala, M. C., & Adigun, O. T. (2021). Academics' Experience of Implementing E-Learning in a South African Higher Education Institution. *International Journal of Higher Education*, 10(1), 1-13.
- [8] Adu, K. M., Plata, L. D. F., Ratilla, M., Novak, P., & Zlamal, L. (2022). Extending the UTAUT model to understand the barriers towards SME digitalization. *Serbian Journal of Management*, 17(2), 403-424.
- [9] "Understanding the Unified Theory of Acceptance and Use of Technology." Enablers of Change. <https://www.enablersofchange.com.au/understanding-the-unified-theory-of-acceptance-and-use-of-technology/> (pristupljeno 24. lipnja 2024.).
- [10] "About Moodle." Moodle. [https://docs.moodle.org/404/en/About\\_Moodle](https://docs.moodle.org/404/en/About_Moodle) (pristupljeno 25. svibnja 2024.).
- [11] "Leaflet." Leaflet. <https://leafletjs.com/> (pristupljeno 25. svibnja 2024.).
- [12] A. Ivanovs. "Top 8 JavaScript Libraries for Creating Dynamic Maps." Colorlib. <https://colorlib.com/wp/javascript-libraries-for-creating-dynamic-maps/> (pristupljeno 25. svibnja 2024.).
- [13] M. Rouse. "Web Map Service." Techopedia. <https://www.techopedia.com/definition/876/web-map-service-wms> (pristupljeno 26. svibnja 2024.).
- [14] "GeoJSON." GeoJSON.org. <https://geojson.org/> (pristupljeno 26. svibnja 2024.).
- [15] "Vector graphics." Wikipedia. [https://en.wikipedia.org/wiki/Vector\\_graphics](https://en.wikipedia.org/wiki/Vector_graphics) (pristupljeno 26. svibnja 2024.).
- [16] "Leaflet Quick Start Guide." Leaflet. <https://leafletjs.com/examples/quick-start/> (pristupljeno 24. lipnja 2024.).

- [17] “What is OpenStreetMap?” OpenStreetMap. <https://welcome.openstreetmap.org/what-is-openstreetmap/> (pristupljeno 28. svibnja 2024.).
- [18] “OpenLayers.” GitHub. <https://github.com/openlayers/openlayers> (pristupljeno 28. svibnja 2024.).
- [19] K. Roy Tirtho. “Choosing Open Source License Wisely.” Medium. <https://medium.com/@krtirtho/choosing-open-source-license-wisely-25b0a11998a9> (pristupljeno 28. svibnja 2024.).
- [20] “OpenLayers.” OpenLayers. <https://openlayers.org/> (pristupljeno 28. svibnja 2024.).
- [21] “Using GeoJSON with Leaflet.” Leaflet. <https://leafletjs.com/examples/geojson/> (pristupljeno 28. svibnja 2024.).
- [22] Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S., & Schaub, T. (2016). The geojson format.
- [23] “What is JSON?” w3schools. [https://www.w3schools.com/whatis/whatis\\_json.asp](https://www.w3schools.com/whatis/whatis_json.asp) (pristupljeno 28. svibnja 2024.).
- [24] “Draw and Edit with GeoJson.io.” handsondataviz.org. <https://handsondataviz.org/geojsonio.html> (pristupljeno 28. svibnja 2024.).
- [25] “Seterra.” Seterra. <https://www.seterra.com/> (pristupljeno 28. svibnja 2024.).
- [26] “World Geography Games.” World Geography Games. <https://world-geography-games.com/> (pristupljeno 28. svibnja 2024.).
- [27] J. Terra. “What is Client-Server Architecture? Everything You Should Know.” Simplilearn. <https://www.simplilearn.com/what-is-client-server-architecture-article> (pristupljeno 24. svibnja 2024.).
- [28] “Client–server architecture.” OpenLearn. <https://www.open.edu/openlearn/science-maths-technology/an-introduction-web-applications-architecture/content-section-1.1> (pristupljeno 24. svibnja 2024.).
- [29] “What Is a Database?” Oracle. <https://www.oracle.com/in/database/what-is-database/> (pristupljeno 24. svibnja 2024.).
- [30] “What is HTTPS?” Cloudflare. <https://www.cloudflare.com/learning/ssl/what-is-https/> (pristupljeno 24. svibnja 2024.).
- [31] “An introduction to the Microsoft Azure portal.” CloudDirect. <https://www.clouddirect.net/knowledge-base/KB0011450/an-introduction-to-the-microsoft-azure-portal> (pristupljeno 24. svibnja 2024.).
- [32] “Introduction to .NET.” Microsoft. <https://learn.microsoft.com/en-us/dotnet/core/introduction> (pristupljeno 24. svibnja 2024.).
- [33] “What is an API (Application Programming Interface)?” AWS. <https://aws.amazon.com/what-is/api/> (pristupljeno 24. svibnja 2024.).
- [34] “What is a REST API?” RedHat. <https://www.redhat.com/en/topics/api/what-is-a-rest-api> (pristupljeno 24. svibnja 2024.).
- [35] “Entity Framework documentation hub.” Microsoft. <https://learn.microsoft.com/en-us/ef/> (pristupljeno 24. svibnja 2024.).

- [36] "Download SQL Server Management Studio (SSMS)." Microsoft. <https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16> (pristupljeno 3. lipnja 2024.).
- [37] "What is Angular?" GeeksForGeeks. <https://www.geeksforgeeks.org/what-is-angular/> (pristupljeno 30. svibnja 2024.).
- [38] K. Lawson. "What Are Single Page Applications and Why Do People Like Them So Much?" Bloomreach. <https://www.bloomreach.com/en/blog/2018/what-is-a-single-page-application> (pristupljeno 30. svibnja 2024.).
- [39] "Difference between TypeScript and JavaScript." GeeksForGeeks. <https://www.geeksforgeeks.org/difference-between-typescript-and-javascript/> (pristupljeno 30. svibnja 2024.).
- [40] "Angular." GitHub. <https://github.com/angular/angular/releases> (pristupljeno 30. svibnja 2024.).
- [41] "What is .Net?" AWS. <https://aws.amazon.com/what-is/net/> (pristupljeno 30. svibnja 2024.).
- [42] "What is .NET?" Codecademy. <https://www.codecademy.com/article/what-is-net> (pristupljeno 1. lipnja 2024.).
- [43] "The History of the .NET Framework." Bocasay. <https://www.bocasay.com/history-net-framework/> (pristupljeno 1. lipnja 2024.).
- [44] "What is MSSQL Server?" Hasura. <https://hasura.io/learn/database/microsoft-sql-server/what-is-mssql/> (pristupljeno 1. lipnja 2024.).
- [45] "What is T-SQL?" Devart. <https://www.pluralsight.com/resources/blog/cloud/what-is-microsoft-azure> (pristupljeno 3. lipnja 2024.).
- [46] "Microsoft Azure." Wikipedia. [https://en.wikipedia.org/wiki/Microsoft\\_Azure](https://en.wikipedia.org/wiki/Microsoft_Azure) (pristupljeno 3. lipnja 2024.).
- [47] "What is Visual Studio?" Microsoft. <https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022> (pristupljeno 1. lipnja 2024.).

# Sažetak

## Naslov:

Sustav za učenje pomoću interaktivnih geografskih karata

## Sažetak:

U sklopu diplomskog rada razvijen je sustav za učenje s pomoću interaktivnih geografskih karata. Sustav je implementiran kao web aplikacija koristeći klijent-poslužitelj arhitekturu. Učitelji unutar sustava jednostavno kreiraju ispite koji se sastoje od pet različitih tipova zadataka povezanih s geografskim kartama. Ti ispiti su namijenjeni učenicima koji su prethodno raspoređeni u razrede. U diplomskom radu pružen je pregled područja tehnologijom podržanog učenja te su istražena i opisana postojeća rješenja slična razvijenom sustavu. Detaljno su opisani arhitektura sustava, korištene tehnologije i način korištenja sustava.

## Ključne riječi:

obrazovni sustav, web-aplikacija, e-učenje, interaktivne geografske karte, Angular, .NET, SQL Server



# Summary

**Title:**

Learning system using interactive geographical maps

**Abstract:**

As part of the thesis, a learning system using interactive geographical maps was developed. The system is implemented as a web application using client-server architecture. Teachers within the system can easily create exams consisting of five different types of tasks related to geographical maps. These exams are intended for students who have been previously assigned to classes. The thesis provides an overview of the field of technology-supported learning and explores and describes existing solutions similar to the developed system. The architecture of the system, the technologies used, and the usage of the system are described in detail.

**Keywords:**

educational system, web application, e-learning, interactive geographical maps, Angular, .NET, SQL Server