Interaktivna edukativna igra za razvojni alat Devtools

Tehnička dokumentacija

Verzija 1.0

Studentski tim: <Ime i prezime>

<Ime i prezime>

<Ime i prezime>

**....**

Nastavnik: <Ime i prezime>

Sadržaj

1. Opis razvijenog proizvoda 4

2. Tehničke značajke 5

3. Upute za korištenje 6

4. Literatura 7

Tehnička dokumentacija

***Na koji način koristiti predložak?***

Dokument se po potrebi može prilagoditi potrebama pojedinog projekta promjenom predloženih naslova predloženih poglavlja, kao i eventualnim dodavanjem novih poglavlja i potpoglavlja.

Cilj dokumenta je opisati rezultat rada studentskog tima, problem koji je riješen u okviru projekta, korištenu tehnologiju, mogućnosti i značajke dobivenog proizvoda i sl. Razinu detalja opisanu u ovom dokumentu studentski tim treba dogovoriti s nastavnikom.

***Literatura:***

U tekstu rada treba biti navedena literatura svugdje gdje je tekst, slika ili grafički prikaz preuzet ili se temelji na nekom pisanom predlošku. Literatura se navodi iza zaključka. U tekstu se literatura navodi unutar zagrada s navođenjem prvog autora i godine izdanja, npr. (Martinis, 1998).

***Primjer citiranja knjige:***

Prezime, inicijal(i) imena autora. Naslov: podnaslov. Podatak o izdanju. Mjesto izdavanja: Nakladnik, godina izdavanja.

***Primjer citiranja članka u časopisu:***

Prezime, inicijal(i) imena autora. Naslov članka: podnaslov. Naziv časopisa. Oznaka sveska/godišta, broj(godina), str. početna-završna.

***Primjer citiranja rada sa konferencije:***

Prezime, inicijal(i) imena autora. Naslov rada: podnaslov. Naslov zbornika, mjesto održavanja konferencije, (godina), str. početna-završna.

***Primjer citiranja doktorskog, magistarskog ili diplomskog rada:***

Prezime, inicijal(i) imena autora. Naslov. Vrsta rada. Ustanova na kojoj je rad obranjen, godina.

***Primjer citiranja www izvora:***

Ime(na) autora (ako je/su poznata), naslov dokumenta, datum nastanka (ako se razlikuje od datuma pristupa izvoru), naslov potpunog djela (italic), potpuna http adresa, datum pristupa dokumentu.

***Ostale upute***

U svim dokumentima obvezno primjenjivati SI jedinice. Slike, formule i tablice potrebno je numerirati. Opis tablice stavlja se iznad, a opis slike ispod nje. U opisu slike ili tablice pišu se samo podaci neophodni za njeno razumijevanje (npr. Slika 6. Pojačalo s promjenljivim pojačanjem). Dodatna objašnjenja daju se u tekstu uz povezivanje sa slikom ili tablicom. Osi i parametri na slikama i grafičkim prikazima trebaju biti obilježeni. Daljnji opis tog grafičkog prikaza treba se nalaziti u tekstu rada. Formule se obilježavaju brojevima u zagradi, uz desni rub stranice, a u tekstu se poziva na broj formule.

# Opis razvijenog proizvoda

U okviru ovog projekta razvijen je prototip interaktivne edukativne igre DevTools Dash, koja korisnicima omogućuje učenje i vježbanje rada s razvojnim alatom *DevTools* kroz igrificirani pristup. Prototip uključuje glavnu arhitekturu za autentifikaciju korisnika, praćenje napretka kroz nivoe te fleksibilan sustav za jednostavno dodavanje i nadogradnju novih nivoa. Autentifikacija je izvedena pomoću biblioteke *Auth0*, a korisnici se prijavljuju putem nadimka i lozinke. Implementiran je i sustav za bilježenje metrika napretka, uključujući informaciju o tome je li nivo započet ili riješen, kao i broj pokušaja i vrijeme potrebno za rješavanje.

A screenshot of a computer

Description automatically generated

Slika 1.1 primjer jednog nivoa

Za lakšu interakciju razvijeno je korisničko sučelje za navigaciju po kategorijama i nivoima, omogućujući igračima intuitivan pregled i odabir izazova. Implementirane su četiri glavne kategorije koje obuhvaćaju ključne alate iz razvojnog alata *DevTools*:

1. **Elements**:

* Prvi nivo testira sposobnost čitanja HTML elemenata.
* Drugi nivo fokusira se na manipulaciju stanjima elemenata (npr. hover efekti).
* Treći nivo provjerava znanje o pristupačnosti kroz upotrebu ARIA oznaka (aria-label, aria-labelledby).

1. **Network**: Implementiran je nivo u kojem korisnik analizira HTTP zahtjeve, uključujući statusne kodove i odgovore servera.
2. **Sources**: Uključuje nivo u kojem igrači koriste zaustavne točke(eng. breakpoint) za očitavanje povratnih vrijednosti iz funkcija u skripti.
3. **Console**:

* Prvi nivo zahtijeva čitanje informacija iz konzole.
* Drugi nivo fokusira se na ispisivanje podataka u konzoli.

Dodatno, implementirana je obfuskacijakoda gdje je to bilo potrebno kako bi se spriječilo varanje igrača, čime se osigurava autentično iskustvo rješavanja izazova. Cijela arhitektura igre razvijena je modularno i prilagodljivo, omogućujući jednostavno dodavanje novih nivoa i proširenje funkcionalnosti.

Ovim prototipom postavljen je snažan temelj za daljnji razvoj igre kroz Diplomski rad.

# Tehničke značajke

Korisničko sučelje aplikacije izgrađen je korištenjem *React* biblioteke uz alat *Vite*, koji poslužuje aplikaciju za lokalni razvoj. Korisničko sučelje obuhvaća forme za prijavu i registraciju, navigacijski sustav za kretanje kroz nivoe organizirane u kategorije (poput "Elements" ili "Console"), te vizualne komponente za prikaz statistike o korisničkim postignućima. Navigacija je implementirana pomoću biblioteke *react-router-dom*, koja osigurava jednostranično ponašanje (SPA) bez ponovnog učitavanja stranice prilikom promjene ruta.

Komunikacija s poslužiteljskom stanom odvija se putem HTTP klijenta *Axios*, konfiguriranog za automatsko upravljanje autorizacijskim tokenima i centralizirano hvatanje grešaka. Dinamička generacija HTTP zahtjeva i TypeScript tipova modela podataka ostvarena je kroz *open-api-client* paket, koji automatski generira kod iz *OpenAPI* specifikacije koju poslužiteljski dio aplikacije generira. Ovo je omogućilo direktnu poveznicu između klijentskog i poslužiteljskom dijela aplikacije te je time izbjegnuto dupliciranje koda i osigurana ujednačenost tipova podataka.

A diagram of a diagram

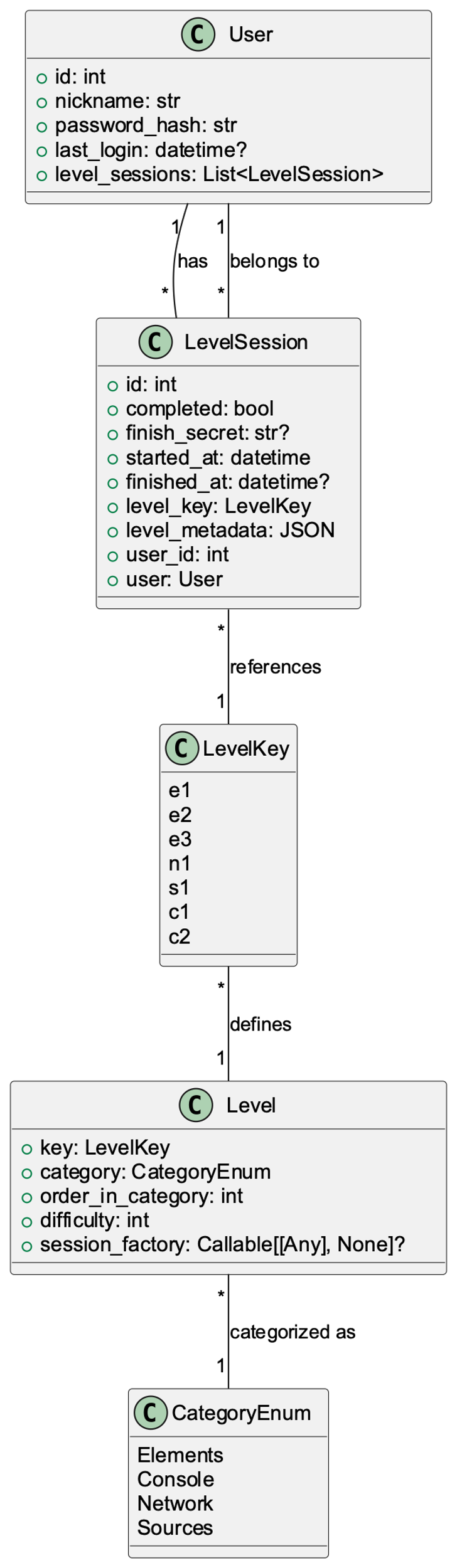
Description automatically generated

Slika 2.1 dijagram arhitekture

Poslužiteljski dio aplikacije razvijen je u razvojnom okviru *FastAPI*  za jezik *Python*. Ima dvojnu ulogu: programsko sučelje za opskrbu podataka (*REST API*) te prikaz nivoa generiranih na poslužitelju (eng. *server-side rendering*). *FastAPI* nudi asinkronu podršku za brzu obradu zahtjeva, automatsku dokumentaciju u specifikaciji *OpenAPI* dostupnu na */docs* ruti, te integraciju sa sustavom *Jinja* za generiranje *HTML*-a za nivoe (eng. *templating engine*). On omogućuje dinamičko generiranje nivoa.

Pristup bazi podataka omogućuje ORM *SQLAlchemy*, koji apstrahira *SQL* operacije u *Python* objekte, olakšavajući transakcijsko upravljanje i definiciju modela Korištenjem asinkronog *asyncpg* drivera, backend neblokirajuće komunicira s *PostgreSQL* bazom podataka.

PostgreSQL baza služi kao središnji repozitorij za pohranu korisničkih računa, podataka o sesijama nivoa i metapodataka. Shema uključuje *User* tablicu s osnovnim korisničkim informacijama (npr. hash lozinke, nadimak) i *LevelSession* tablicu koja bilježi detalje o korisnikovom napretku (vrijeme početka, status dovršenosti, specifični metapodaci). Relacija 1:n između *User* i *LevelSession* omogućuje praćenje više istovremenih sesija po korisniku.



Slika 2.2 UML dijagram klasa

Svaki nivo u sustavu može imati prilagođenu inicijalizacijsku logiku zahvaljujući session\_factory parametru u modelu Level, što omogućuje dinamičko generiranje jedinstvenih uvjeta i izazova. Primjenom obrasca tvornice, session\_factory djeluje kao apstraktna tvornica koja proizvodi specifičnu konfiguraciju sesije ovisno o nivou. Na primjer, za nivo s1 (kategorija *Sources*) session\_factory poziva funkciju s1\_intialize\_level koja generira nasumičnu vrijednost koja se s hash funkcijom pretvara u šifru. Ta hash funkcija je specifična za nivo s1 te ju je bilo potrebno implementirati i u JavaScriptu za klijentski dio aplikacije i u Pythonu. Ova fleksibilnost omogućuje da svaki nivo ima vlastitu logiku postavljanja (npr. postavljanje početnih vrijednosti, generiranje dinamičkih meta-podataka ili priprema konteksta za igru), dok Level model ostaje generičan. Zahvaljujući factory pristupu, dodavanje novih nivoa ne zahtijeva promjenu postojećeg koda – dovoljno je definirati novu tvornicu (factory) i pridružiti je ključu nivoa. Time se osigurava modularnost, lakše održavanje i mogućnost beskonačnog proširivanja igre novim mehanikama.

Kako bi se spriječilo varanje putem analize izvornog koda u određenim nivoima, implementirana je selektivna obfuskacija koda, odnosno namjerno zamućivanje koda radi zaštite logike. Za ovu svrhu iskorištena je jednostavna funkcija na javno dostupnom GitHub Gist rješenju, koji automatski pretvara čitljivi JavaScript kod u teže razumljiv oblik (npr. zamjenom imena varijabli, uklanjanjem komentara). Ključna karakteristika sustava je dinamička obfuskacija tijekom izvođenja: za razliku od tradicionalnog pristupa gdje se kod jednokratno obfuscira prije posluživanja, ovdje se proces obfuskacije pokreće prilikom inicijalizacije nivoa. Ovo je nužno jer obfuscirani kod mora sadržavati dinamički generiranu šifru za završetak nivoa, koja se stvara u trenutku pokretanja sesije. Tek nakon što backend generira šifru i ubaci je u kontekst nivoa (npr. kroz varijable u JavaScriptu), započinje postupak obfuskacije – time se osigurava da šifra nije vidljiva ni u izvornom kodu ni u statički pripremljenim datotekama.

Trenutna implementacija koristi jednostavan Python skriptni isječak za obfuskaciju, koji se izvršava na poslužiteljskoj strani prije slanja koda korisniku. Arhitektura je dizajnirana da omogući nadogradnju bez restrukturiranja. Dodatno, sustav koristi različite nazive skripti za obfuscirane i čiste datoteke (npr. [id]\_obfuscated.js i [id].js), što sprječava slučajno izlaganje osjetljivih dijelova koda u nivoima gdje je čitljivost namjerno omogućena (npr. u edukativnim *Sources* izazovima).

# Upute za korištenje

## Postavljanje lokalnog okruženja

Projekt je potrebno prvo preuzeti s GitHub repozitorija putem naredbe *git clone*. Nakon preuzimanja, potrebno je otvoriti terminal u korijenskom direktoriju projekta. Za klijentski dio (smješten u direktoriju *app*) instalirajte ovisnosti naredbom npm install, što će preuzeti sve potrebne pakete navedene u package.json (uključujući *React*, *Vite* i druge biblioteke). Za poslužiteljski dio, u direktoriju *backend* pokrenite pip install -r requirements.txt kako biste instalirali Python pakete poput *FastAPI*, *SQLAlchemy* i *Jinja2*. Prije instalacije preporučuje se stvaranje virtualnog okruženja kako biste izbjegli konflikte s drugim projektima.

## Pokretanje aplikacije

Nakon instalacije ovisnosti, pokrenite oba dijela sustava. U direktoriju *app* izvršite naredbu npm run dev koja će pokrenuti *Vite* razvojni poslužitelj na adresi *http://localhost:5173*. Za backend, u direktoriju *backend* koristite uvicorn main:app --reload () kako biste aktivirali poslužitelj na *http://localhost:8000*. Oba poslužitelja moraju biti pokrenuta istovremeno kako bi aplikacija ispravno funkcionirala.

## Pristup aplikaciji

Aplikaciji pristupite otvaranjem preglednika i upisivanjem adrese *http://localhost:5173*. Po prvoj posjeti, potrebno je registrirati se unosom željenog nadimka i lozinke. Ako već imate račun, prijavite se putem Login forme s istim podacima. Lozinka se pohranjuje u bazu u hashiranom obliku radi sigurnosti.

A screenshot of a login screen

Description automatically generated

Slika 3.1 sučelje za prijavu

## Navigacija kroz igru

Nakon prijave, korisniku se prikazuje popis kategorija (npr. *Elements*, *Console*). Svaka kategorija sadrži listu nivoa poredanih prema težini. Klikom na kategoriju otvara se stranica s dostupnim nivoima, a klikom na pojedini nivo otvara se novi tab preglednika s igrom.

A screenshot of a black background with white text

Description automatically generated

Slika 3.2 Sučelje za izbor kategorija

## Rješavanje nivoa

Svaki nivo zahtijeva pronalaženje tajne šifre od 6 znakova, skrivene u strukturi stranice. Preporuča se cijelo vrijeme imati otvoren *DevTools*. Na primjer, u kategoriji *Elements* šifra može biti skrivena u HTML atributima, dok u *Console* kategoriji može biti dostupna putem JavaScript varijabli ili ispisa u konzoli. Za *Network* nivoe potrebno je analizirati mrežne zahtjeve, a za *Sources* nivoe pregledavati izvorni kod stranice.

## Završetak nivoa

Pronađenu šifru unesite u polje za unos na stranici nivoa i potvrdite je. Ako je šifra točna, sustav će prikazati poruku o uspjehu, ažurirati statistiku na glavnoj stranici i omogućiti pristup sljedećim nivoima.

A white rectangle with black lines

Description automatically generated

Slika 3.3 forma za unos šifre

# Literatura

Chrome Devtools dokumentacija, <https://developer.chrome.com/docs/devtools>, 25.11.2024

FastAPI dokumentacija, <https://fastapi.tiangolo.com/>, 13.1.2025

React dokumentacija, <https://react.dev/reference/react>, 11.12.2024.

Developer Mozilla, <https://developer.mozilla.org/en-US/>, 15.1.2025.