

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 874

Aplikacija za praćenje rangiranja sveučilišta prema Šangajskoj listi

Ivan Bilobrk

Zagreb, svibanj 2023.

Zagreb, 10. ožujka 2023.

ZAVRŠNI ZADATAK br. 874

Pristupnik: **Ivan Bilobrk (0036531335)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: prof. dr. sc. Igor Mekterović

Zadatak: **Aplikacija za praćenje rangiranja sveučilišta prema Šangajskoj listi**

Opis zadatka:

Godišnje liste sveučilišta daju poredak svjetska sveučilišta na temelju različitih parametara. Jedna od najpoznatijih i najutjecajnijih listi je ARWU - Academic Ranking of World Universities, poznata i kao Šangajska lista. Izdaje se jednom godišnje od strane Shanghai Ranking Consultancyja i cijenjena je zbog metodologije i objektivnosti, iako postoje i zamjerke da ne uvažava u dovoljnoj mjeri veličinu institucije. Lista donosi poredak sveučilišta u različitim područjima, a u kontekstu ovog rada zanimljiva su nam samo područja Computer Science & Engineering (CSE) i Electrical Engineering (EEE). Potrebno je proučiti i opisati metodologiju odnosno algoritam formiranja poretka u Šangajskoj listi, te napraviti prototip sustava koji će reproducirati izradu liste za spomenuta područja CSE i EEE. Pritom je potrebno pribaviti podatke s javno dostupnih izvora kao što su npr. Incites i Web Of Science baze. Ustrojiti postupak prikupljanja podataka tako da se obavlja periodički i puni lokalnu bazu podataka koja će onda dalje poslužiti za izračune. Pritom obratiti pozornost na robusnost i mogućnost oporavka od pogrešaka tih pozadinskih procesa. Napraviti prototip web-aplikacije koja će prezentirati izračunate podatke i omogućiti praćenje pozicije sveučilišta tokom godine, te potencijalne što-ako (engl. what if) analize. Osim tabličnih i statističkih podataka, napraviti i osnovne grafičke prikaze podataka. Donijeti ocjenu ostvarenog pristupa te smjernice za budući razvoj.

Rok za predaju rada: 9. lipnja 2023.

SADRŽAJ

1. Uvod	1
2. Shanghai Ranking metodologija	3
2.1. Izvori za prikupljanje podataka	3
2.2. Minimalni broj publikacija	3
2.3. Preslikavanje područja istraživanja	3
2.4. Indikatori za računanje rankinga	4
2.5. Računanje ukupnog rezultata sveučilišta	6
2.5.1. Primjer izračuna vrijednosti za ranking	6
2.6. Računanje uspješnosti procjene rankinga	8
3. Funkcionalni zahtjevi	9
3.1. Funkcionalni zahtjevi	9
3.2. Nefunkcionalni zahtjevi	11
4. Korištene tehnologije	12
4.1. Arhitektura sustava	12
4.1.1. Web korisničko sučelje	12
4.1.2. <i>Backend</i> poslužitelj	13
4.1.3. Baza podataka	14
4.2. Docker	15
5. Pregled funkcionalnosti	17
5.1. Početna stranica	17
5.2. Tablični prikaz rankinga sveučilišta	18
5.3. Stranica sveučilišta	23
5.4. Stranica za postavljanje podataka sa Shanghai Ranking stranice	28

6. Opis rješenja i podatkovni model	31
6.1. Baza podataka	31
6.1.1. Relacija <i>award</i>	32
6.1.2. Relacija <i>ranking</i>	32
6.1.3. Relacija <i>rankingreal</i>	33
6.2. <i>Backend</i> poslužitelj	34
6.2.1. Punjenje baze podataka i korištenje alata Puppeteer	35
6.2.2. Rute i kontroleri	40
6.3. React web korisničko sučelje	42
6.4. Docker	48
7. Zaključak	50
Literatura	51

1. Uvod

U svijetu postoji više od 25000 sveučilišta. Svako od njih trudi se imati što bolje predavače, kvalitetniju nastavu, puno znanstvenih radova, sudjelovanja na konferencijama, objava u časopisima te drugih raznih uspjeha. Sveučilištima je teško uskladiti svoju organizaciju bez neke povratne informacije o svojim uspjesima. Upravo zbog toga napravljene su razne rang liste koje svakom sveučilištu pridružuju neku numeričku vrijednost te na osnovu nje ih sortiraju. Na ovaj način svako sveučilište dobiva svoju poziciju koja može služiti kao mjerilo uspjeha, ukazati na potencijalne probleme na sveučilištu te tako omogućiti uvođenje promjena na sveučilištu kako bi na sljedećoj rang listi sveučilište bilo na nekoj višoj poziciji.

Neke od organizacija koje objavljuju rang liste sveučilišta: Times Higher Education, Round University Ranking, U.S. News, i Shanghai Ranking. U kontekstu ovog završnog rada, zanima nas način rangiranja Shanghai Ranking sustava.

Shanghai Ranking objavljuje jednom godišnje dvije rang liste. Prva rangira sveučilišta neovisno o područjima istraživanja i zove se Academic Ranking of World Universities (ARWU). ARWU se objavljuje od 2003. godine i temelji se na 6 indikatora uspjeha. Rangira više od 2000 sveučilišta, a samo najboljih 1000 objavi na službenoj stranici. Nama zanimljivija rang lista je Global Ranking of Academic Subjects (GRAS) koja rangira sveučilišta u nekom području istraživanja. Ovu rang listu Shanghai Ranking objavljuje od 2009. godine. Zadnji ranking, 2022. godine uključivao je više od 1800 sveučilišta u više od 96 zemalja u 54 područja istraživanja. Fakultet elektrotehnike i računarstva u Zagrebu (FER) dio je Sveučilišta u Zagrebu te su nam rang liste u području računarske znanosti i inženjerstva (engl. *Computer Science & Engineering (CSE)*) i elektrotehnike (engl. *Electrical & Electronic Engineering (EEE)*) najzanimljivije. Problem tih rang lista na Shanghai Ranking stranici je taj što objavljuju samo prvih 500 najboljih sveučilišta u tim područjima što znači da se ranking Sveučilišta u Zagrebu u kategorijama CSE i EEE ne može ni provjeriti. Ovaj završni rad bavi se izradom web aplikacije koja će prikupljati podatke o sveučilištima na isti način kao i Shanghai Ranking, ali uz iznimku da nema ograničenja na broj sveučilišta koja se mogu pojaviti

na konačnoj rang listi. Na ovaj način Sveučilište u Zagrebu, a i FER moći će pratiti svoj napredak iz godine u godinu te raditi određene promjene kako bi im se pozicija na rang listi poboljšala.

2. Shanghai Ranking metodologija

Shanghai Ranking Global Ranking of Academic Subjects (GRAS) objavljuje se svake godine i temelji se na podacima kroz četiri godine. Ako promatramo ranking za neku godinu x , donja granica godina za podatke je $x - 6$, a donja granica je $x - 2$. Tako primjerice ako nas zanima ranking za 2022. godinu promatrat ćemo podatke od 2016. do 2020. godine, uključivo.

2.1. Izvori za prikupljanje podataka

Podatke za izračun vrijednosti svih indikatora osim indikatora *Award* prikupljamo sa baza InCites i Web of Science (WoS). Podatke za *Award* indikator prikupljamo sa raznih stranica ovisno o području koje nas zanima. Za *Computer Science & Engineering* (CSE) to je stranica A.M. Turing Award, a za *Electrical & Electronic Engineering* (EEE) IEEE Awards.

2.2. Minimalni broj publikacija

Kako bi sveučilište ušlo na ranking za područja CSE i EEE mora imati minimalno 150 publikacija koje su vidljive na bazama WoS i InCites.

2.3. Preslikavanje područja istraživanja

Kako bi uspješno prikupili podatke s navedenih baza moramo na tim stranicama odabrati ispravno područje istraživanja jer preslikavanja nisu 1 : 1.

U sljedećim tablicama možemo vidjeti kako izgledaju preslikavanja za područja CSE i EEE.

Tablica 2.1: Preslikavanje za područje *Computer Science & Engineering* (CSE)

Područje na Shanghai Ranking stranici	Područje na InCites i Web of Science (WoS) bazama
<i>Computer Science & Engineering</i>	<i>Computer Science, Information Systems</i>
<i>Computer Science & Engineering</i>	<i>Computer Science, Cybernetics</i>
<i>Computer Science & Engineering</i>	<i>Computer Science, Software Engineering</i>
<i>Computer Science & Engineering</i>	<i>Computer Science, Artificial Intelligence</i>
<i>Computer Science & Engineering</i>	<i>Computer Science, Hardware & Architecture</i>
<i>Computer Science & Engineering</i>	<i>Computer Science, Theory & Methods</i>
<i>Computer Science & Engineering</i>	<i>Computer Science, Interdisciplinary Applications</i>

Tablica 2.2: Preslikavanje za područje *Electrical & Electronic Engineering* (EEE)

Područje na Shanghai Ranking stranici	Područje na InCites i Web of Science (WoS) bazama
<i>Electrical & Electronic Engineering</i>	<i>Engineering, Electrical & Electronic</i>
<i>Electrical & Electronic Engineering</i>	<i>Imaging Science & Photographic Technology</i>

2.4. Indikatori za računanje rankinga

Global Ranking of Academic Subjects (GRAS) se računa na temelju pet indikatora, a to su: Q1, CNCI, IC, *Top* i *Award*.

Q1 indikator predstavlja broj publikacija sveučilišta u *top* 25% časopisa koji su izabrani za neko područje putem ankete ShanghaiRanking's Academic Excellence Survey (AES) tijekom relevantnog razdoblja.

CNCI indikator (*Category Normalized Citation Impact*) omjer je citiranosti objavljenih radova i prosječnih citiranosti radova u istoj kategoriji, iste godine i iste vrste publikacije u časopisu, od strane sveučilišta u određenom području tijekom relevantnog razdoblja. CNCI vrijednosti manje od 1 znače da je citiranost sveučilišta manja od prosjeka, vrijednost 1 znači da je citiranost prosječna, a vrijednosti veće od 1 znače da je citiranost sveučilišta veća od prosjeka.

IC indikator (*International Collaboration*) predstavlja koliko međunarodnih suradnja sveučilište u nekom području istraživanja ima. Točna vrijednost indikatora dobije se kao omjer broja publikacija koje su pronađene u najmanje dvije različite zemlje u odnosu na adresu autora prema ukupnom broju publikacija iz odgovarajućeg područja istraživanja za instituciju tijekom relevantnog razdoblja.

Top indikator predstavlja broj radova koje je sveučilište objavilo u *Top* časopisima u nekom području istraživanja tijekom relevantnog razdoblja. *Top* časopise biraju profesori sveučilišta isto kao i Q1 časopise kroz anketu ShanghaiRanking's Academic Excellence Survey (AES). Iznimka ovdje je područje CSE jer se za to područje uzima broj radova predstavljenih na 31 odabranoj konferenciji, također kroz AES.

Award indikator predstavlja broj osoba sveučilišta koje je dobilo značajnu nagradu iz nekog područja istraživanja. Značajne nagrade biraju se također putem AES-a. Značajna nagrada za područje CSE je Turingova nagrada, a za područje EEE je IEEE Medal of Honor. Kako bi dobitak nagrade išao u korist sveučilištu, osoba koje je dobila nagradu morala je u trenutku dobitka nagrade raditi puno radno vrijeme na tom sveučilištu, a ako je osoba u trenutku dobitka nagrade bila povezana s više sveučilišta ili drugih institucija, svakoj ustanovi se pridjeljuje recipročan broj broja sveučilišta. Tako na primjer, ako je osoba bila povezana s 3 ustanove, svakoj od njih se pridjeljuje $1/3$, a ako je bila povezana samo s jednom ustanovom pridjeljuje se toj ustanovi 1. Vrijeme dobitka nagrade također igra ulogu jer u obzir dolaze nagrade dodijeljene unazad 4 desetljeća od gornje granice godine za koju promatramo podatke. Ako nas zanima ranking za 2022. godinu, onda je gornja granica godine za koju promatramo podatke 2020. Svakom desetljeću pridjeljuju se različite težine s kojima se onda množi prethodno dobiveni broj. Desetljeću najbližem sadašnjosti pridjeljuje se težina 1, a svim ostalima smanjuje se za 0.25. Ukupna vrijednost indikatora dobije se zbrajanjem pojedinih vrijednosti za neko sveučilište u nekom području.

Tablica 2.3: Primjer težina za indikator *Award* za ranking 2022. godine

2011.-2020.	2001.-2010.	1991.-2000.	1981.-1990.
1	0.75	0.5	0.25

2.5. Računanje ukupnog rezultata sveučilišta

Uz vrijednosti svih indikatora za neko sveučilište ukupna brojčana vrijednost prema kojoj se sveučilišta rangiraju izračuna se na sljedeći način:

Svaki od indikatora osim CNCI-a podijeli se s najvećom vrijednosti indikatora od svih sveučilišta, iz tog broja izvadi se drugi korijen te dobiveni broj pomnoži se s težinom tog indikatora. Na kraju dobivene vrijednosti se sumiraju.

Izračun vrijednosti koja se pridjeljuje nekom indikatoru:

$$\sqrt{\frac{\text{vrijednost indikatora}}{\max(\text{indikator})}} * \text{težina}(\text{indikator}) \quad (2.1)$$

Indikator CNCI je poseban te se njemu pridjeljuje sljedeća vrijednost:

$$\sqrt{\frac{\text{vrijednost CNCI indikatora}}{\min(2 * \text{average}(\text{CNCI}), \max(\text{CNCI}))}} * \text{težina}(\text{CNCI}) \quad (2.2)$$

Ukoliko je vrijednost CNCI indikatora veća od vrijednosti brojnika u izrazu 2.2 onda se sveučilištu automatski pridjeljuje vrijednost 100.

Tablica 2.4: Težine s kojima se vrijednosti pridružene indikatorima množe

Q1	CNCI	IC	Top	Award
100	100	20	100	100

2.5.1. Primjer izračuna vrijednosti za ranking

U sljedećem dijelu prikazan je primjer izračuna vrijednosti za ranking za sveučilište University of California, Berkeley koje je 2022. godine bilo prvo na ranking u području EEE. Period tijekom kojeg se uzimaju podatci je 2016. godina - 2020. godina.

Vrijednost Q1:

Pretraživanjem InCites baze saznaje se da navedeno sveučilište ima vrijednost indikatora Q1 603. Najveća vrijednost Q1 indikatora u tom razdoblju iznosi 3285. Vrijednost koja se veže za Q1 indikator iznosi; $\sqrt{\frac{603}{3285}} * 100 = 42.8$

Vrijednost CNCI:

Pretraživanjem InCites baze saznaje se da navedeno sveučilište ima vrijednost indikatora CNCI 1.71. Najveća vrijednost CNCI indikatora u tom razdoblju iznosi 4.13. Kako je riječ o indikatoru CNCI u obzir se još mora uzeti i dvostruka prosječna vrijednost svih CNCI vrijednosti, a to je 2.27. Vrijednost koja se veže za CNCI indikator iznosi:

$$\sqrt{\frac{1.71}{\min(2.27, 4.13)}} * 100 = 86.8$$

Vrijednost IC:

Pretraživanjem InCites baze saznaje se da navedeno sveučilište ima vrijednost indikatora IC 57.18. Najveća vrijednost IC indikatora u tom razdoblju iznosi 97.15.

Vrijednost koja se veže za IC indikator iznosi:

$$\sqrt{\frac{57.18}{97.15}} * 20 = 15.3$$

Vrijednost Top:

Pretraživanjem InCites baze saznaje se da navedeno sveučilište ima vrijednost indikatora Top 25. Najveća vrijednost Top indikatora u tom razdoblju iznosi 25. Vrijednost koja se veže za Top indikator iznosi:

$$\sqrt{\frac{25}{25}} * 100 = 100$$

Vrijednost Award:

Pretraživanjem stranice IEEE Awards saznaje se da je navedeno sveučilište osvojilo nagradu IEEE Medal of Honor sljedećih godina: 1985., 1995., 1998., 2020. Kako je svake godine nagradu osvojila uvijek jedna osoba koja nije bila vezana za nijednu drugu ustanovu te je u vrijeme dobitka nagrade radila puno radno vrijeme na tom sveučilištu, vrijednost indikatora Award je $1 * 0.25 + 1 * 0.5 + 1 * 0.5 + 1 * 1 = 2.25$ (u obzir se uzimaju težine koje se vežu za godinu osvajanja nagrade). Najveća vrijednost indikatora Award u tom razdoblju iznosi 3. Vrijednost koja se veže za Award indikator iznosi:

$$\sqrt{\frac{2.25}{3}} * 100 = 86.6$$

Jednom kada se izračunaju sve vrijednosti koje se vežu za pojedine indikatore, ukupan rezultat sveučilišta University of California, Berkeley u 2022. godini dobije se zbrajanjem tih vrijednosti: $42.8 + 86.8 + 15.3 + 100 + 86.6 = 331.5$.

2.6. Računanje uspješnosti procjene rankinga

Na svakoj Shanghai Ranking listi za područja CSE i EEE nalazi se 500 sveučilišta. Samo prvih 50 ima dodijeljenu točnu poziciju, dok se ostale pozicije određuju otprilike te sveučilištima nije dodijeljena egzaktna pozicija nego interval pozicija. Ocjena uspješnosti rankinga računa se kao postotak sveučilišta kojima je pozicija uspješno procijenjena. Za prvih 50 sveučilišta pozicija se smatra uspješno procijenjena ako je odstupanje u odnosu na stvarnu poziciju manje od 3, a za ostala sveučilišta ako je procijenjena pozicija unutar intervala koji je objavljen na Shanghai Ranking stranici. Ukupan broj uspješno procijenjenih pozicija podijeli se s ukupnim brojem sveučilišta te se dobije informacija koliko je procjena dobra.

3. Funkcionalni zahtjevi

3.1. Funkcionalni zahtjevi

Funkcionalni zahtjevi predstavljaju sve usluge koje programski proizvod mora pružiti korisnicima te definiraju kako sustav reagira na određene ulazne poticaje.

Aktori ovog programskog sustava su korisnici, React web grafičko sučelje, Node.js poslužitelj i PostgreSQL baza podataka.

Korisnici mogu:

- a) pregledati bazu procjene rankinga sveučilišta za određenu godinu u područjima CSE i EE
- b) pregledati vrijednosti svih indikatora nekog sveučilišta pomoću kojih se računa procjena rankinga za željeno područje i godinu
- c) usporediti vrijednosti Shanghai Ranking sustava s procijenjenim vrijednostima za sva sveučilišta u nekom području i za neku godinu
- d) pregledati uspješnost procjene rankinga za željeno područje i godinu
- e) pregledati grafički prikaz promjene procjene vrijednosti indikatora i pozicije na rang listi sveučilišta u nekom području tijekom svih godina za koje se računa ranking, kao i pratiti napredak sveučilišta za trenutnu godinu

React web grafičko sučelje može:

- a) omogućiti korisniku odabir područja i godine za pregled procjene rankinga sveučilišta
- b) omogućiti korisniku tablični prikaz procjene rankinga sveučilišta sa svim vrijednostima indikatora i pozicije sveučilišta za željeno područje i godinu
- c) omogućiti korisniku pretragu sveučilišta na rankingu za određeno područje i godinu
- d) omogućiti korisniku grafički prikaz promjene procjene vrijednosti indikatora i pozicije nekog sveučilišta za željenu godinu i područje

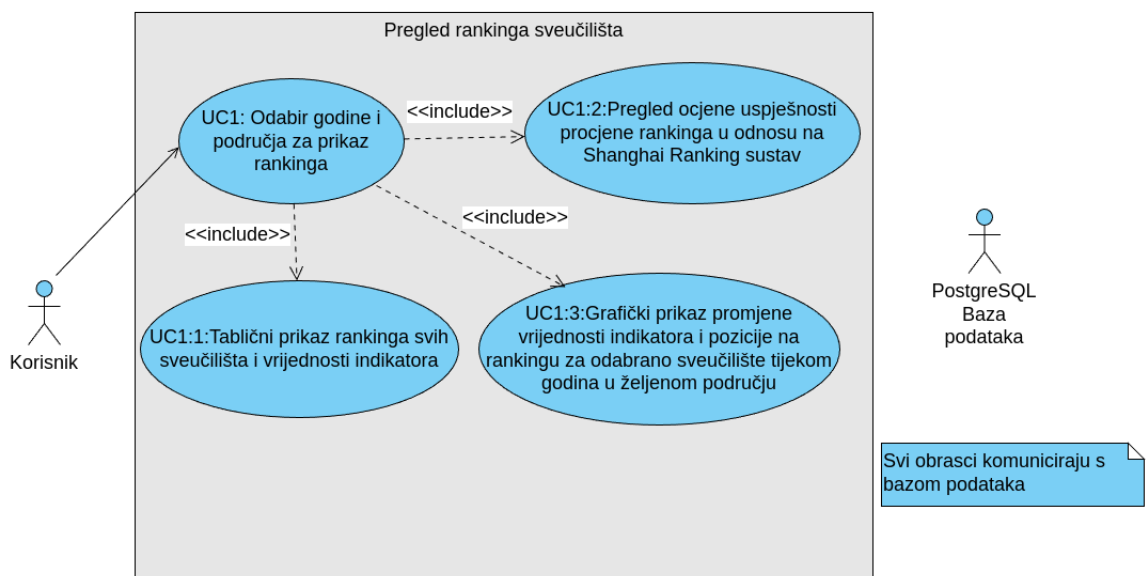
e) omogućiti korisniku prikaz uspješnosti procjene rankinga u odnosu na Shanghai Ranking sustav za željeno područje i godinu

Node.js poslužitelj može:

- a) inicijalno napuniti bazu podataka koristeći baze WoS i InCites s podacima potrebnim za izračun procjene rankinga sveučilišta
- b) svakih dva tjedna prikupiti podatke sa baza WoS i InCites za procjenu rankinga sveučilišta u trenutnoj godini
- c) nudi React web grafičkom sučelju krajnje točke potrebne za prikaz podataka korisniku.

PostgreSQL baza podataka može:

- a) pohranjivati vrijednosti indikatora za izračun procjene rankinga sveučilišta



Slika 3.1: Dijagram obrasca uporabe, korisnička funkcionalnost

3.2. Nefunkcionalni zahtjevi

Nefunkcionalni zahtjevi opisuju koja svojstva sustav mora imati, a ne funkcionalnost koju pruža korisniku. Ova web aplikacija mora:

- a) biti robusna, otporna na pogreške i stabilna
- b) omogućiti brzo i fluidno korisničko sučelje
- c) dati što bolju procjenu rankinga sveučilišta u odnosu na Shanghai Ranking sustav

4. Korištene tehnologije

4.1. Arhitektura sustava

Arhitektura ovog programskog sustava sastoji se kao i mnoge web aplikacije od 3 dijela:

1. web korisničko sučelje
2. *backend* poslužitelj
3. baza podataka

4.1.1. Web korisničko sučelje

Web korisničko sučelje ove aplikacije napravljeno je u JavaScript biblioteci React.

React je biblioteka otvorenog koda, razvijena od strane Facebooka, prva verzija objavljena je 2013. godine te je danas vrlo raširena i često se koristi za izradu interaktivnih i dinamičkih web korisničkih sučelja. React omogućava jednostavniju izradu web aplikacija uz manje programiranja i manju složenost u odnosu na izradu web aplikacije u čistom JavaScriptu. Jedna od prednosti React biblioteke je ta što omogućava izradu jednostraničnih web aplikacija (engl. *Single Page Application (SPA)*) koja radi tako da dinamički surađuje s preglednikom te navigiranje aplikacijom ne uzrokuje dohvat potpuno drugačije web stranice već se trenutna stranica mijenja i prepisuje s podacima dohvaćenih s web poslužitelja. Ovu funkcionalnost omogućava dodatak *react-router-dom* koji nudi komponente kao što su *Link*. Navigiranjem po web aplikaciji korištenjem komponente *Link* mijenja se URL u pregledniku, ali web stranica zapravo ostaje ista uz promijenjen sadržaj. React koristi virtualni DOM (engl. *Document Object Model*) koji prati stanja komponenti web stranice te kada dođe do promjene stanja, u stvarnom preglednikovom DOM-u mijenja samo one elemente koji su se promijenili. Ova funkcionalnost uvelike poboljšava performanse web aplikacije. Jedna stranica u Reactu sastoji se od više manjih komponenti koje se mogu dijeliti

između više stranica i proizvoljno gnijezditi. Ovakvom organizacijom postiže se dobra organizacija koda uz mogućnosti višestrukog korištenja komponenti. Kombiniranjem navedenih funkcionalnosti React biblioteke dobiva se fluidno korisničko sučelje bez puno ponovnih učitavanja stranica (engl. *reload*).

Axios

Axios je biblioteka koja omogućava jednostavno stvaranje HTTP zahtjeva kao što su GET i POST na *backend* poslužitelj i rukovanje odgovorima koje taj poslužitelj vraća.

Material UI

Pisanje vlastitih komponenti u Reactu od početka je korisno ako je potrebna potpuna kontrola nad komponentama te velika prilagodljivost, ali često se koriste komponente s nekom generičkom funkcionalnosti te pisanje takvih komponenti svaki put od nule nije potrebno. Material UI je biblioteka za React koja nudi veliki broj gotovih komponenti koje se mogu prilagođavati i uređivati prema vlastitim potrebama .

Tailwind CSS

Tailwind CSS je radni okvir (engl. *framework*) za CSS (engl. *Cascading Style Sheets*) koji ima niz gotovih CSS razreda koji omogućavaju lagano postizanje željenog izgleda komponenti.

React-chartjs-2

React-chartjs-2 je jedna od najpopularnijih React biblioteka koja nudi funkcionalnost jednostavne izrade grafova. Koristi se za izradu grafova na kojima pratimo promjene ranking podataka tijekom vremena.

4.1.2. Backend poslužitelj

Node.js

Node.js je JavaScript pokretačko okruženje (engl. *runtime environment*) namijenjeno izvođenju na poslužiteljskoj strani. Pokreće se na V8 JavaScript *engineu* te omogućuje izvršavanje JavaScript koda izvan preglednika. Koristi asinkronu arhitekturu zasnovanu na događajima (engl. *asynchronous event-driven architecture*) te nudi mogućnost izrade skalabilnih web aplikacija.

Express.js

Express.js je *framework* Node.js-a te omogućava izradu RESTful API-ja (engl. *Application Programming Interface*). Express.js nudi lagano upravljanje HTTP zahtjevima i izradu krajnjih točaka (engl. *endpoint*) s kojima će React web aplikacija komunicirati.

Puppeteer

Puppeteer je Node.js biblioteka koja nudi bogati API pomoću kojeg se preglednici Chrome i Chromium mogu kontrolirati koristeći DevTools protokol. DevTools protokol omogućuje alatima upravljanje preglednicima kao što su Chrome i Chromium na temelju uputa koje smo dali tim alatima. Puppeteer koristimo za izradu *web scrappera* (alat koji posjećuje web stranice i na njima obavlja neke radnje bez potrebe za intervencijom čovjeka) za baze InCites i WoS. Iako navedene baze nude API pomoću kojeg bi mogli dohvatiti sve podatke potrebne za ranking, on se plaća. Puppeteer alatu moramo zadati niz koraka koje treba obaviti na nekoj stranici (upisati tekst u neko polje, pritisnuti na gumb, otići na drugu stranicu) kako bi postigli željeni rezultat.

Node-postgres

Kako bi *backend* poslužitelj mogao komunicirati s bazom podataka koristi se Node-postgres. To je skup Node.js modula koji nude sučelje prema bazi podataka. Pomoću ovog proširenja s *backend* poslužitelja se mogu raditi sve uobičajene radnje s bazom podataka (stvaranje novih tablica, unos podataka u tablice, dohvaćanje podataka iz tablica, brisanje podataka iz tablica i razne druge radnje).

Node Cron

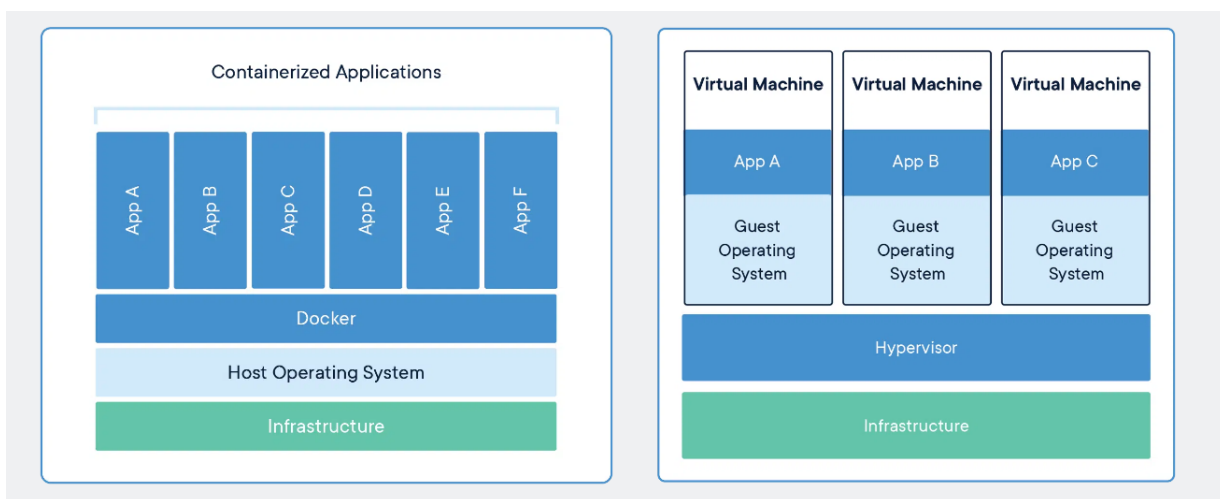
Node Cron je Node.js modul koji omogućava obavljanje nekih radnji na *backend* poslužitelju u željeno vrijeme. Node Cron modul se u ovoj web aplikaciji koristi kako bi se svaka dva tjedna pokrenuo alat Puppeteer koji će prikupiti najnovije podatke za ranking sveučilišta.

4.1.3. Baza podataka

Ova web aplikacija koristi relacijsku bazu podataka otvorenog koda PostgreSQL. PostgreSQL baza podataka ima ACID (engl. *Atomicity, Consistency, Isolation, Durability*) svojstva koja su bitna kako bi se osigurala robusnost i stabilnost web aplikacije.

4.2. Docker

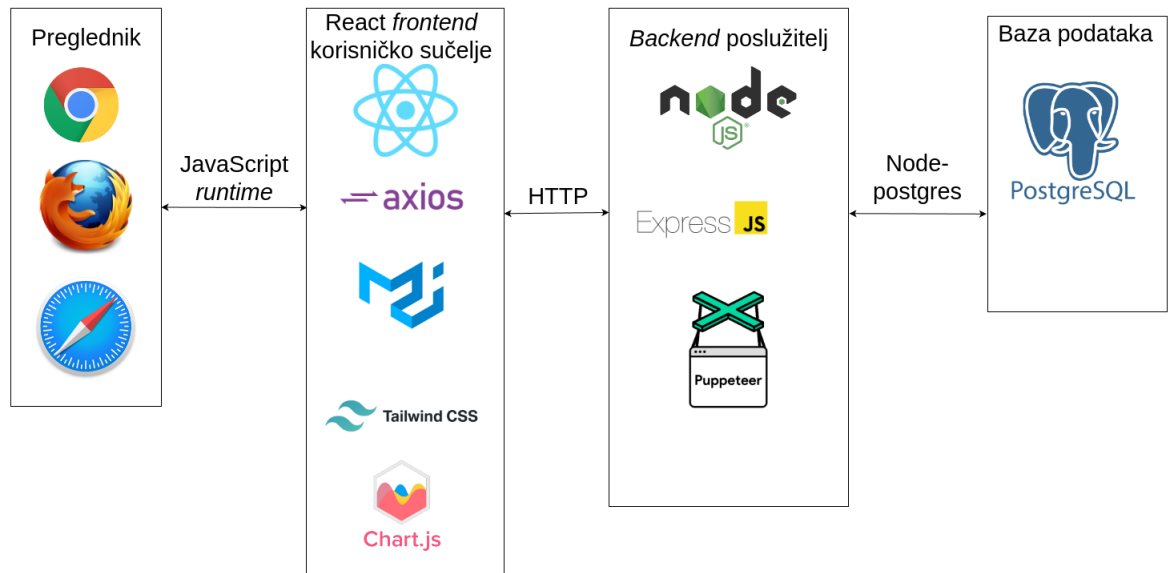
Kako bi se ova web aplikaciju mogla lako postaviti (engl. *deploy*) na neki poslužitelj te tako omogućiti svim korisnicima pristup aplikaciji, koristi se platforma Docker. Docker je platforma otvorenog koda koja se koristi za razvoj, isporuku i pokretanje aplikacija. Docker pakira dijelove web aplikacije u odvojene dijelove koji se zovu kontejneri (engl. *containers*). Kontejneri su zapravo Docker slike (engl. *images*) u izvođenju. Docker slika je lagana (engl. *lightweight*) komponenta koja sadrži sve što je aplikaciji potrebno za izvođenje (izvorni kod, pokretačko okruženje (engl. *runtime*), razne alate i biblioteke). Način stvaranja docker slike se definira datotekom Dockerfile. Jednom kad je stvorena Docker slika, može se pokrenuti. Time se dobije kontejner koji se izvršava na Docker Engineu. Velika prednost Dockera je ta što je podržan na puno operacijskih sustava (Windows, Linux, MacOS i ostali) te uz pomoć samo jedne naredbe i datoteke Dockerfile dobije se pokrenuta i funkcionalna aplikacija. Aplikacije koje se pokreću kao kontejneri rade na jednak način na svim operacijskim sustavima zbog ugrađene virtualizacije. Docker virtualizira operacijski sustav, a ne sklopovlje. Ovo je velika prednost u odnosu na virtualizaciju koju rade virtualni strojevi. Virtualni strojevi emuliraju sklopovlje i upravljanjem pomoću hipervizora omogućuju da se na istom sklopovlju izvršava više operacijskih sustava. Docker kontejneri dijele isti operacijski sustav te svaki predstavlja poseban proces. Docker kontejneri zauzimaju manje resursa, lakši su i brži.



Slika 4.1: Usporedba virtualnih strojeva i Docker načina virtualizacije

U slučaju ove aplikacije postoje 3 Docker slike koje će postati Docker kontejneri prilikom izvođenja. Jedna slika je za bazu podataka, druga za *backend* poslužitelj, a treća za web korisničko sučelje. Jednom kada je napisan Dockerfile za sve navedene

dijelove aplikacije, dijelovi se povezuju datotekom docker-compose. U toj datoteci stoje upute od kojih se sve kontejnera aplikacija sastoji, kako kontejneri međusobno surađuju te kako se pokreću. Zahvaljujući ovoj datoteci, umjesto da se svaki kontejner posebno inicijalizira i pokrene, s jednom naredbom se pokrene cijela aplikacija.



Slika 4.2: Prikaz raspodjele korištenih tehnologija po arhitekturnim slojevima

5. Pregled funkcionalnosti

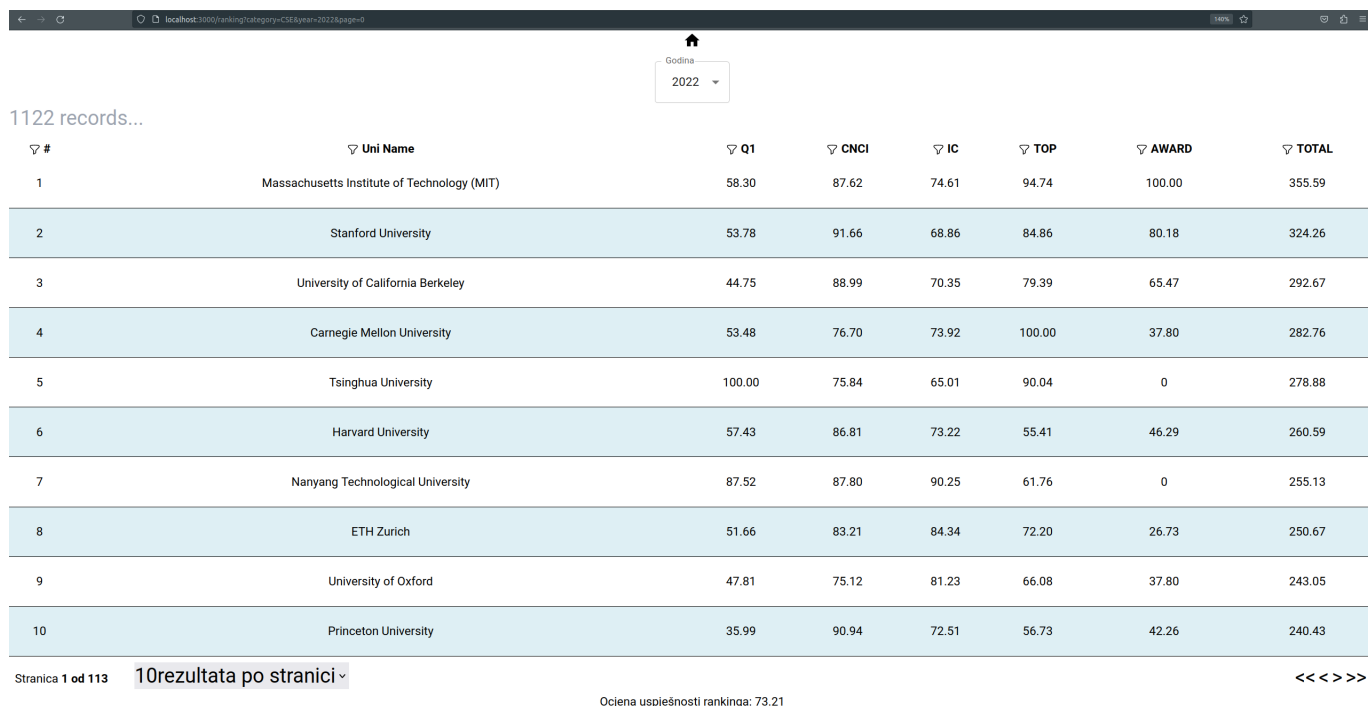
5.1. Početna stranica



Slika 5.1: Prikaz početne stranice

Početna stranica nalazi se na putanji "/". Na početnoj stranici korisnik može birati između tri opcije. Pritiskom na neki od gornja dva gumba *CSE* ili *EEE* korisnika će se preusmjeriti na stranicu za pregled procjene Shanghai Rankinga i svih vrijednosti indikatora za područje CSE, odnosno EEE. Pritiskom na donji gumb *UPLOAD STVARNIH RANKING PODATAKA* korisnik će biti preusmjeren na stranicu na kojoj može objaviti .xlsx datoteku koja ima identične vrijednosti indikatora i pozicija za pojedina sveučilišta kao i Shanghai Ranking stranica. Na osnovu te datoteke popunit će se baza podataka, a objavljeni podatci služiti će za usporedbu procjene rankinga i Shanghai Rankinga.

5.2. Tablični prikaz rankinga sveučilišta



1122 records...

▼ #	▼ Uni Name	▼ Q1	▼ CNCI	▼ IC	▼ TOP	▼ AWARD	▼ TOTAL
1	Massachusetts Institute of Technology (MIT)	58.30	87.62	74.61	94.74	100.00	355.59
2	Stanford University	53.78	91.66	68.86	84.86	80.18	324.26
3	University of California Berkeley	44.75	88.99	70.35	79.39	65.47	292.67
4	Carnegie Mellon University	53.48	76.70	73.92	100.00	37.80	282.76
5	Tsinghua University	100.00	75.84	65.01	90.04	0	278.88
6	Harvard University	57.43	86.81	73.22	55.41	46.29	260.59
7	Nanyang Technological University	87.52	87.80	90.25	61.76	0	255.13
8	ETH Zurich	51.66	83.21	84.34	72.20	26.73	250.67
9	University of Oxford	47.81	75.12	81.23	66.08	37.80	243.05
10	Princeton University	35.99	90.94	72.51	56.73	42.26	240.43

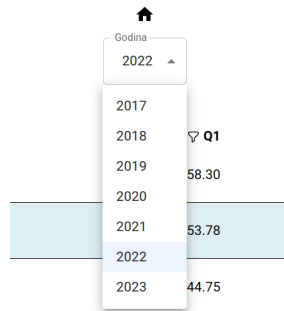
Stranica 1 od 113 10 rezultata po stranici > Ocjena uspešnosti rankinga: 73.21 <<<>>>

Slika 5.2: Tablični prikaz rankinga sveučilišta

Pritiskom na neki od gumba *CSE* ili *EEE* korisniku se prikaže stranica na slici 5.2. Putanja koja vodi do ove stranice je `/ranking?category=CSE\&year=2022\&page=0`. URL parametar *category* specificira za koje područje se prikazuje procjena rankinga, parametar *year* specificira za koju godinu se gleda procjena rankinga, a parametar *page* je pomoćni parametar za ostvarenje paginacije. U ovom konkretnom primjeru iz URL-a se vidi da korisnik trenutno gleda prvu stranicu (indeksiranje stranica kreće od 0) procjene rankinga sveučilišta u području CSE za 2022. godinu.

Stranica na vrhu ima ikonu u obliku kućice s kojom se korisnik vraća na početnu stranicu.

Odmah ispod ikone kućice nalazi se komponenta za odabir godine za koju korisnik želi pogledati procjenu rankinga sveučilišta. Pritiskom na tu komponentu prikazuje se padajući izbornik s popisom godina od 2017. godine do trenutne godine. Početna godina je 2017. jer od te godine su dostupne rang liste na stranici Shanghai Ranking. Kada korisnik odabere jednu od ponuđenih godina u URL-u se mijenja parametar *year*, šalje se HTTP GET zahtjev na *backend* poslužitelj te se ažurira tablica procjene rankinga s podacima za odabranu godinu koje *backend* poslužitelj pošalje u HTTP odgovoru (engl. *response*).



Ispod komponente za odabir godine, a prije tablice rankinga nalazi se polje za pretragu sveučilišta po njihovom imenu. Prije nego što korisnik krene upisivati slova u to polje, u njemu kao zamjenski tekst (engl. *placeholder*) piše koliko sveučilišta se određene godine u nekom području našlo na procijenjenoj rang listi. Na slici 5.2 se vidi da je za 2022. godinu u području CSE na procijenjenoj rang listi 1122 sveučilišta. Upisom imena sveučilišta u tablici za procjenu rankinga pojavit će se podatci samo za ona sveučilišta koja u imenu sadrže podniz koji je korisnik upisao u polje za pretragu. Upisom podniza na *backend* poslužitelj HTTP GET zahtjevom šalje se taj podniz te se pretražuje baza podataka procjene rankinga za određenu godinu u nekom području po imenu sveučilišta. Dolaskom HTTP odgovora ažuriraju se podatci u tablici.

Slika 5.4: Primjer pretrage sveučilišta po imenu

Na slici 5.4 se vidi primjer pretrage sveučilišta po imenu. Korisnik je upisao niz slova "zag" te se u tablici vide podatci za dva sveučilišta koja u svom imenu sadrže navedeni

podniz.

Tablica procjene rankinga sastoji se od 8 stupaca. Zaglavlje tablice sadrži nazive vrijednosti koje se nalaze u pojedinim stupcima. Prvi stupac pokazuje poziciju sveučilišta na rang listi za određenu godinu te u nekom području, drugi pokazuje ime sveučilišta, sljedećih pet stupaca sadrže vrijednosti koje se vežu uz indikatore Q1, CNCI, IC, *Top* i *Award*, a zadnji stupac sadrži ukupan rezultat sveučilišta u određenoj godini za neko područje prema kojem su sveučilišta sortirana i zove se Total. U zaglavlju tablice prije samog naziva vrijednosti stupca nalazi se ikona koja simbolizira sortiranje tablice rankinga. Ova funkcionalnost omogućena je za sve stupce osim prva dva. Ako korisnik pritisne na tu ikonu ili naziv stupca cijela tablica procjene rankinga sortirat će se prema vrijednosti tog stupca silazno odnosno uzlazno.

1122 records...

▼ #	▼ Uni Name	↑ Q1	▼ CNCI	▼ IC	▼ TOP	▼ AWARD	▼ TOTAL
1	Institute of Mathematical Sciences (IMSc) India	3.02	45.20	87.34	0	0	65.69
2	Hassan II University of Casablanca	7.70	43.33	54.68	0	0	61.97
3	Jaypee University of Information Technology	7.70	75.99	37.99	0	0	91.29
4	Amrita Vishwa Vidyapeetham Coimbatore	9.06	56.93	39.32	0	0	73.85
5	Bulgarian Academy of Sciences	9.55	39.37	62.59	0	0	61.45
6	Birla Institute of Technology Mesra	10.02	48.90	36.61	0	0	66.24
7	Qassim University	10.02	48.45	74.61	0	0	73.39
8	University Teknikal Malaysia Melaka	10.02	69.44	66.58	0	0	92.78
9	Cadi Ayyad University of Marrakech	10.24	47.99	60.07	0	0	70.25
10	Technical University Kosice	10.24	42.47	52.76	0	0	63.26

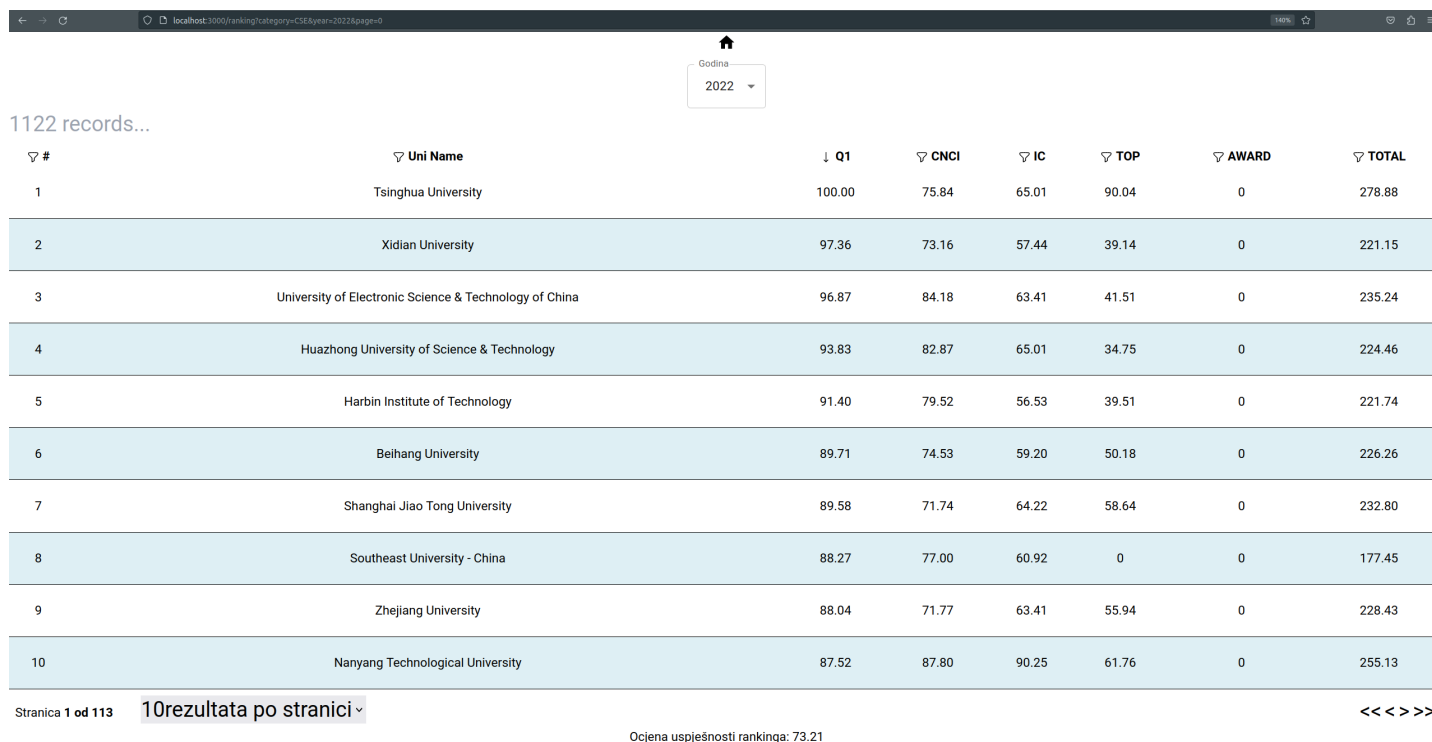
Stranica 1 od 113 10 rezultata po stranici

Ocjena uspjehnosti rankinga: 73.21

Slika 5.5: Primjer uzlaznog sortiranja tablice procjene rankinga prema vrijednosti indikatora Q1

Slika 5.5 prikazuje stanje tablice nakon što je korisnik pritisnuo u zaglavlju tablice treći stupac koji ima naziv Q1. Sada se ikona koja simbolizira sortiranje zamijenila s ikonom strelice koja pokazuje prema gore. Strelica prema gore simbolizira da se cijela tablica procjene rankinga sortira uzlazno prema vrijednosti indikatora uz koji se nalazi strelica, u ovom slučaju to je indikator Q1. Sada su se podaci u tablici ažurirali te su sveučilišta sortirana uzlazno po vrijednosti indikatora Q1 što se i vidi u samoj tablici

na slici 5.5 jer vrijednost indikatora Q1 za prvo sveučilište je 3.02, a za svako sljedeće sveučilište ta vrijednost je veća u odnosu na prethodno sveučilište. Sortiranje se može raditi i silazno. Tada se ikona promjeni u strelicu koja je okrenuta prema dolje. Promjene vrste sortiranja između silaznog, uzlaznog ili zadanog (engl. *default*) se radi uzastopnim pritiskanjem jednog od zadnjih 6 stupaca u zaglavlju tablice.



1122 records...

▼ #	▼ Uni Name	↓ Q1	▼ CNCI	▼ IC	▼ TOP	▼ AWARD	▼ TOTAL
1	Tsinghua University	100.00	75.84	65.01	90.04	0	278.88
2	Xidian University	97.36	73.16	57.44	39.14	0	221.15
3	University of Electronic Science & Technology of China	96.87	84.18	63.41	41.51	0	235.24
4	Huazhong University of Science & Technology	93.83	82.87	65.01	34.75	0	224.46
5	Harbin Institute of Technology	91.40	79.52	56.53	39.51	0	221.74
6	Beihang University	89.71	74.53	59.20	50.18	0	226.26
7	Shanghai Jiao Tong University	89.58	71.74	64.22	58.64	0	232.80
8	Southeast University - China	88.27	77.00	60.92	0	0	177.45
9	Zhejiang University	88.04	71.77	63.41	55.94	0	228.43
10	Nanyang Technological University	87.52	87.80	90.25	61.76	0	255.13

Stranica 1 od 113 10 rezultata po stranici ▼ Ocjena uspjehnosti rankinga: 73.21 << >>>

Slika 5.6: Primjer silaznog sortiranja tablice procjene rankinga prema vrijednosti indikatora Q1

Default sortiranje je silazno prema vrijednosti stupca Total jer se tako dobije rang lista sveučilišta od najboljih prema najgorim uzimajući u obzir sve vrijednosti koje se vežu uz pojedine indikatore. Istodobno sortiranje sveučilišta prema više vrijednosti stupaca nije moguće. Tako na primjer ako korisnik prvo odabere uzlazno sortiranje prema vrijednosti stupca Q1 te zatim odabere sortiranje za stupac IC, sortiranje prema vrijednosti stupca Q1 će se poništiti te će se primijeniti željeno sortiranje prema vrijednosti stupca IC. Sortiranje sveučilišta se također odvija na *backend* poslužitelju, a parametri se prenose u sklopu HTTP GET zahtjeva. Dolaskom HTTP odgovora ažuriraju se podatci u tablici.

Zbog potencijalno velikog broja sveučilišta na rang listama, korisniku se ne prikazu sva sveučilišta u obliku jedne velike tablice nego je napravljena paginacija tako da se na jednoj stranici prikaže samo određeni broj zapisa iz rang liste sveučilišta. Informacija o tome na kojoj je trenutno stranici i koliko ukupno stranica rang liste ima korisniku je

vidljiva ispod tablice uz lijevi rub.

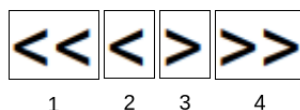
Slika 5.7: Prikaz trenutne stranice i ukupnog broja stranica rang liste

Korisnik također može birati količinu redaka tablice po jednoj stranici. Pritiskom na komponentu ispod tablice otvara se izbornik gdje korisnik može birati između 10, 20, 30, 40 ili 50 redaka tablice po jednoj stranici. *Default* vrijednost je 10. Ova vrijednost zajedno s trenutnim brojem stranice prenose se kao parametri prilikom HTTP GET zahtjeva kako bi *backend* poslužitelj znao koliko zapisa i s kojim pomakom mora vratiti u HTTP odgovoru klijentu. Na ovaj način na klijentskoj strani imamo pohranjeno uvijek najviše 50 zapisa što pridonosi brzini korisničkog sučelja.



Slika 5.8: Odabir broja redaka tablice po jednoj stranici rang liste

Ispod tablice rankinga uz desni rub nalazi se područje za navigiranje stranicama rang liste. Postoje četiri mogućnosti navigacije kao što se i vidi na slici 5.9. Pritiskom na



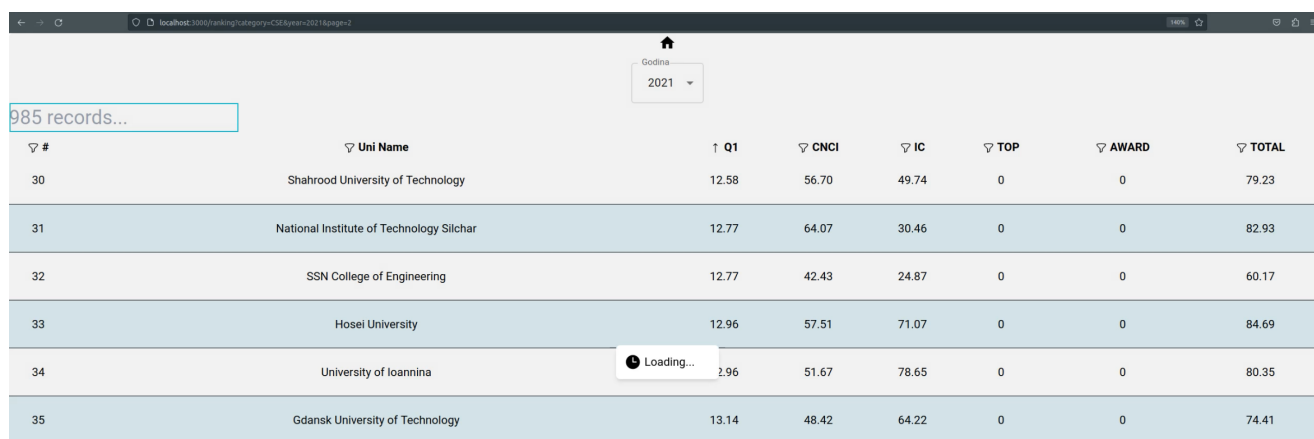
Slika 5.9: Područje za navigiranje stranicama rang liste

ikonu označenu brojem 1 korisniku će se prikazati prva stranica rang liste. Pritiskom na ikonu označenu brojem 2 korisnik će se pomaknuti za jednu stranicu unazad, a pritiskom na ikonu označenu brojem 3 prikazat će se sljedeća stranica. Pritisci na ikone 2 i 3 funkcioniraju ako uistinu postoji prethodna odnosno sljedeća stranica, u suprotnom se ništa ne događa. Pritiskom na zadnju ikonu označenu rednim brojem 4 korisniku će se prikazati zadnja stranica rang liste.

Ako je korisnik recimo navigirao na petu stranicu rang liste te u tom trenutku krenuo pretraživati sveučilišta, automatski ga se navigira na prvu stranicu rang liste.

Prilikom ažuriranja prikaza tablice rankinga ne dolazi do ponovnog učitavanja stranice što je jedna od opisanih koristi jednostraničnih web aplikacija.

Za vrijeme čekanja HTTP odgovora od *backend* poslužitelja korisniku se prikaže odgovarajuća ikona koja simbolizira učitavanje podataka te se pozadina posivi.



▼ #	▼ Uni Name	↑ Q1	▼ CNCI	▼ IC	▼ TOP	▼ AWARD	▼ TOTAL
30	Shahrood University of Technology	12.58	56.70	49.74	0	0	79.23
31	National Institute of Technology Silchar	12.77	64.07	30.46	0	0	82.93
32	SSN College of Engineering	12.77	42.43	24.87	0	0	60.17
33	Hosei University	12.96	57.51	71.07	0	0	84.69
34	University of Ioannina	Loading...	51.67	78.65	0	0	80.35
35	Gdansk University of Technology	13.14	48.42	64.22	0	0	74.41

Slika 5.10: Primjer čekanja HTTP odgovora *backend* poslužitelja

Na samom dnu stranice nalazi se ocjena uspješnosti procjene rankinga za neku godinu u određenom području u odnosu na stvarni ranking objavljen na stranici Shanghai Ranking. To je decimalni broj u rasponu od 0 do 100. Što je vrijednost tog broja veća znači da se procjena rankinga to bolje podudara sa stvarnim ranking podacima sa Shanghai Ranking stranice.

Ocjena uspješnosti rankinga: 73.21

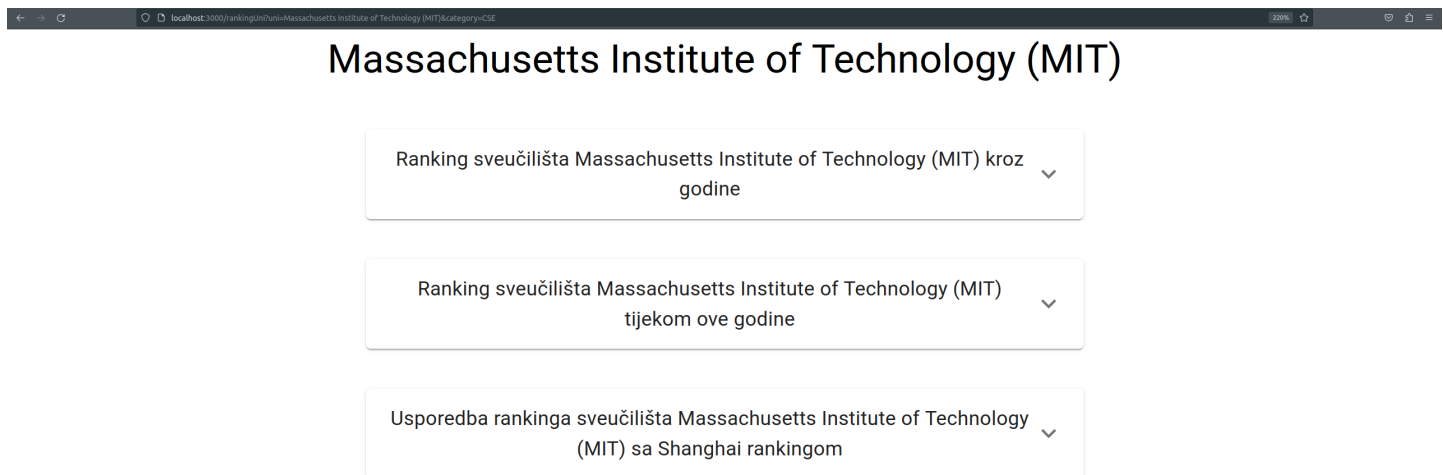
Slika 5.11: Labela s ocjenom uspješnosti procjene rankinga

Na svako ime sveučilišta u tablici rankinga korisnik može pritisnuti mišem te će ga se preusmjeriti na stranicu s detaljnijim podacima za to konkretno sveučilište.

5.3. Stranica sveučilišta

Temeljni funkcionalni zahtjevi nalažu da korisnik može pratiti ranking sveučilišta i promjene vrijednosti indikatora tijekom godina, pogledati procjenu rankinga sveučilišta u tekućoj godini te usporediti procjenu rankinga sa stvarnim ranking podacima na stranici Shanghai Ranking. Ova stranica nudi te funkcionalnosti korisniku u obliku

grafičkog prikaza.

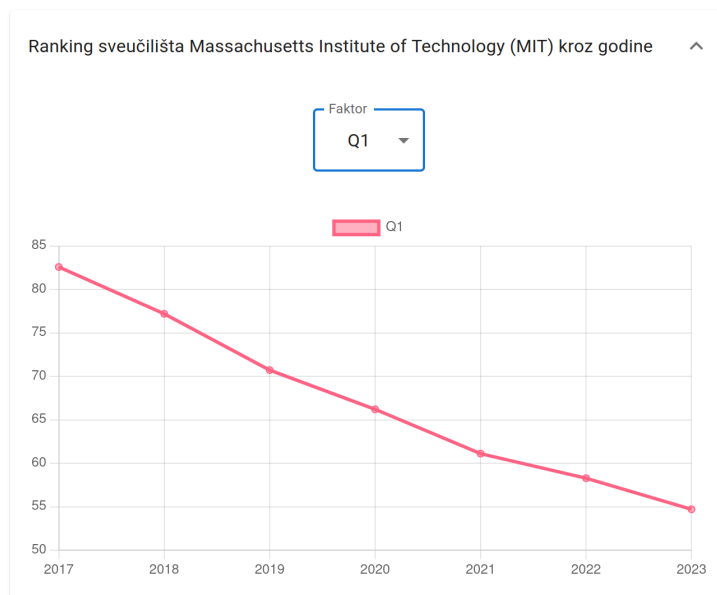


Slika 5.12: Stranica sveučilišta

Nakon što je korisnik pritisnuo ime nekog sveučilišta prikazat će mu se stranica na slici 5.12. Stranica sa slike nalazi se na putanji `"/rankingUni?uni=Massachusetts Institute of Technology (MIT)&category=CSE"`. Parametar *uni* specificira za koje sveučilište će se prikazivati podatci, a parametar *category* specificira za koje područje se prikazuju podatci. U primjeru sa slike 5.12 gledamo stranicu koja se odnosi na sveučilište Massachusetts Institute of Technology (MIT) za područje CSE. Područje za koje promatramo podatke ovisi za koje područje je korisnik prethodno gledao rang listu te se ono prenosi na ovu stranicu.

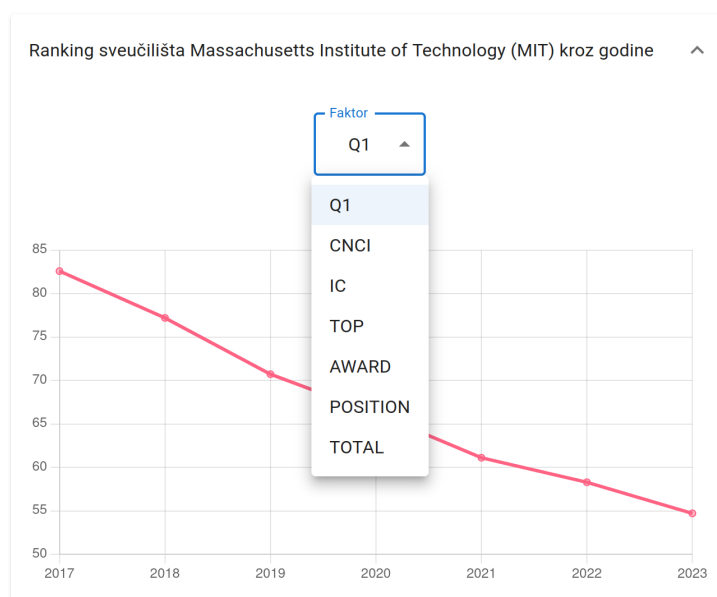
Na ovoj stranici korisnik može birati između 3 ponuđene opcije. Pritiskom na prvi padajući izbornik prikazat će se grafički prikaz promjene podataka za ranking od 2017. godine, što se vidi na slici 5.13. Ako korisnika zanima vrijednost nekog drugog podatka za ranking, na raspolaganju mu je izbornik koji je prikazan na slici 5.14. Postoji mogućnost pregleda promjene vrijednosti svih indikatora koji služe za računanje rankinga: Q1, CNCI, IC, *Top*, *Award*. Osim vrijednosti indikatora može se promatrati kako se pozicija (engl. *position*) sveučilišta te ukupna vrijednost (engl. *total*) prema kojoj se sveučilišta sortiraju na rang listi mijenjala kroz godine. Iz grafičkog prikaza se ne mogu potpuno točno pročitati vrijednosti podataka za neku godinu. Točan iznos nekog podatka za određenu godinu dobije se zadržavanjem strelice miša na krivulji iznad određene godine, što se vidi na slici 5.15.

Massachusetts Institute of Technology (MIT)

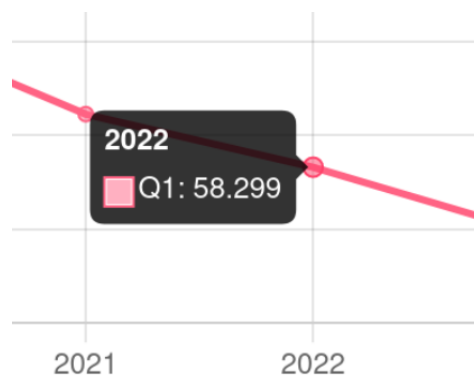


Slika 5.13: Grafički prikaz promjene vrijednosti indikatora

Massachusetts Institute of Technology (MIT)

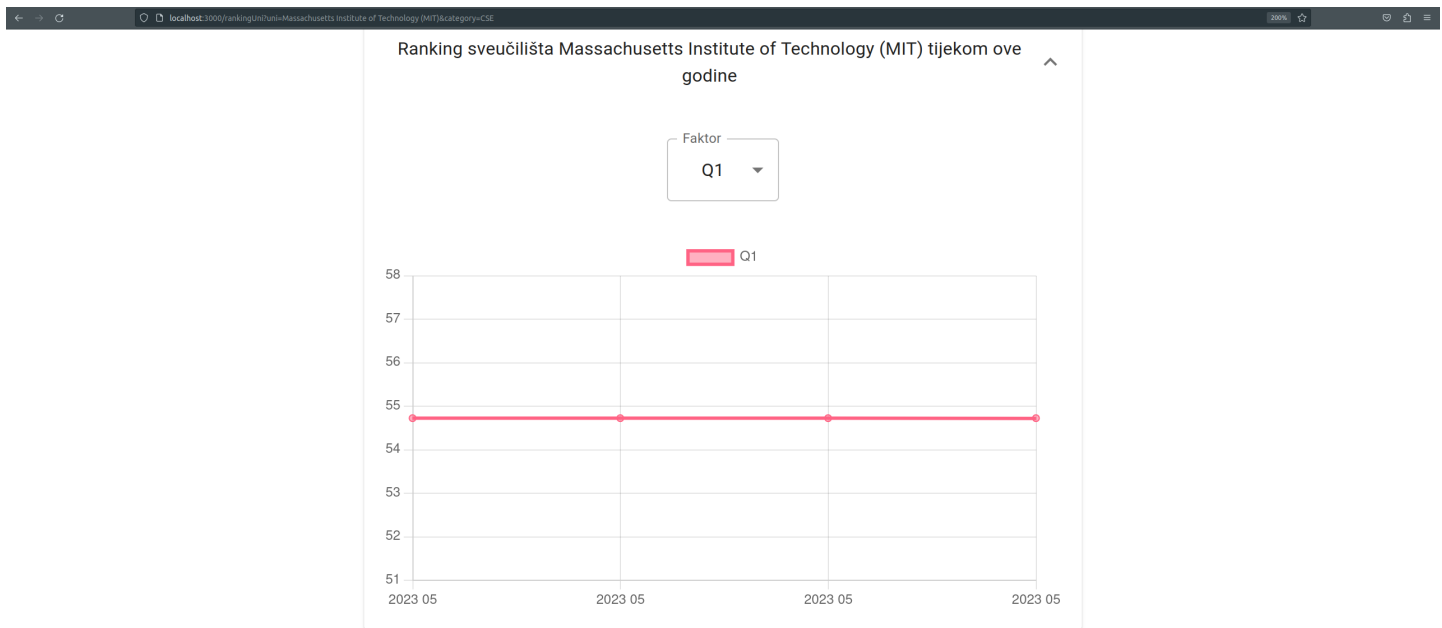


Slika 5.14: Izbornik podataka čije će se vrijednosti prikazati na grafu



Slika 5.15: Detaljniji prikaz vrijednosti podataka na grafu

Pritiskom na drugi padajući izbornik korisniku se prikaže isti graf s opcijom pregleda podataka za ranking kao i u prethodnom slučaju. Jedina razlika je što ovaj graf prikazan na slici 5.16 prikazuje promjenu procjene ranking podataka tijekom mjeseci tekuće godine. Primjer sa slike 5.16 nije reprezentativan jer su podatci prikupljeni u kratkom vremenskom razdoblju, no kada bi se aplikacija držala pokrenutom tijekom jedne godine vidjeli bismo promjene na grafu. Pregledom ovog grafa može se procijeniti kako će izgledati Shanghai Ranking rang lista za trenutnu godinu prije službene objave.

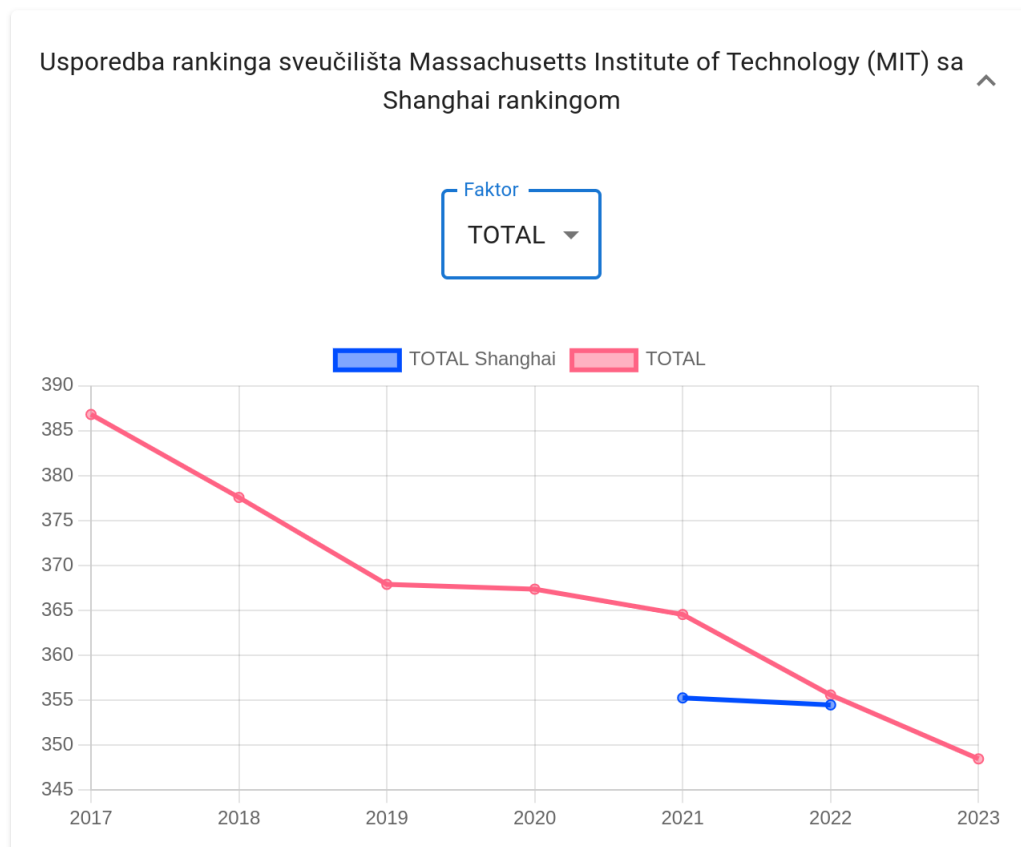


Slika 5.16: Prikaz ranking podataka za trenutnu godinu

Pritiskom na treći padajući izbornik prikazuje se grafički prikaz na slici 5.17. Na ovom grafu postoje dvije krivulje, ona označena plavom bojom predstavlja stvarne podatke rankinga sa stranice Shanghai Ranking, dok ona označena crvenom bojom predstavlja procijenjene vrijednosti podataka za ranking. Na ovaj način može se lakše vizualno usporediti koliko je procjena rankinga dobra. Na ovom konkretnom primjeru vidi se odnos procijenjene i stvarne vrijednosti Total. Kao i na prethodnim grafovima ukoliko se zadrži strelica miša na dijelu krivulje iznad godine dobit će se točniji iznos vrijednosti podatka za tu godinu. Podatke sa Shanghai Ranking stranice korisnik unosi postavljanjem (engl. *upload*) odgovarajuće .xlsx datoteke, što je opisano u sljedećem dijelu.

Svi podatci potrebni za prikaz grafova dohvaćaju se pomoću HTTP GET zahtjeva na

backend poslužitelj. Na ovoj stranici, kao i prethodno opisanoj, vidimo prednosti jednostraničnih web aplikacija jer promjenom grafa ili vrijednosti koja se prikazuje na grafu ne dolazi do ponovnog učitavanja web stranice.



Slika 5.17: Grafički prikaz usporedbe procjene rankinga i stvarnih Shanghai Ranking podataka

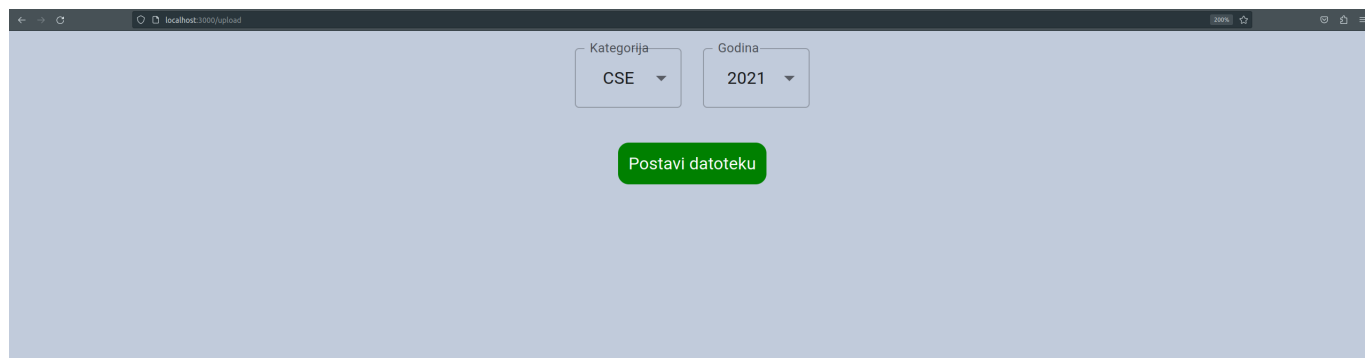
5.4. Stranica za postavljanje podataka sa Shanghai Ranking stranice

Usporedba procijenjenih podataka rankinga i stvarnih podataka sa Shanghai Ranking stranice moguća je postavljanjem .xlsx datoteke koja sadrži podatke sa Shanghai Ranking stranice. Kako korisnik ne bi morao direktno unositi podatke u bazu podataka, na ovoj stranici može kroz interaktivno web sučelje postaviti datoteku sa Shanghai Ranking podacima koji će se automatski upisati u odgovarajuću tablicu u bazi podataka.

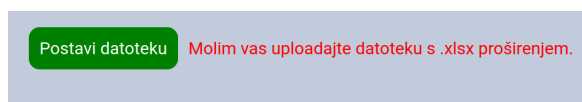
Ova stranica prikazana na slici 5.18 ima URL `"/upload"` te se na nju dolazi s početne stranice prikazane na slici 5.1 pritiskom na gumb s tekstom "UPLOAD STVARNIH

RANKING PODATAKA".

Na stranici u gornjem dijelu postoje dvije komponente za odabir područja, koje može biti CSE ili EEE, i godine, koja se kreće od 2017. do trenutne. Nakon što se odaberu te dvije vrijednosti korisnik može pritisnuti gumb s tekстом "Postavi datoteku" nakon čega mu se otvara preglednik datoteka osobnog računala u kojem može izabrati koju datoteku želi postaviti. Ako korisnik postavi datoteku koja nema .xlsx proširenje, ispisat će mu se poruka pogreške kao na slici 5.19.



Slika 5.18: Stranica za postavljanje stvarnih Shanghai Ranking podataka



Slika 5.19: Poruka greške u slučaju postavljanja neispravne datoteke

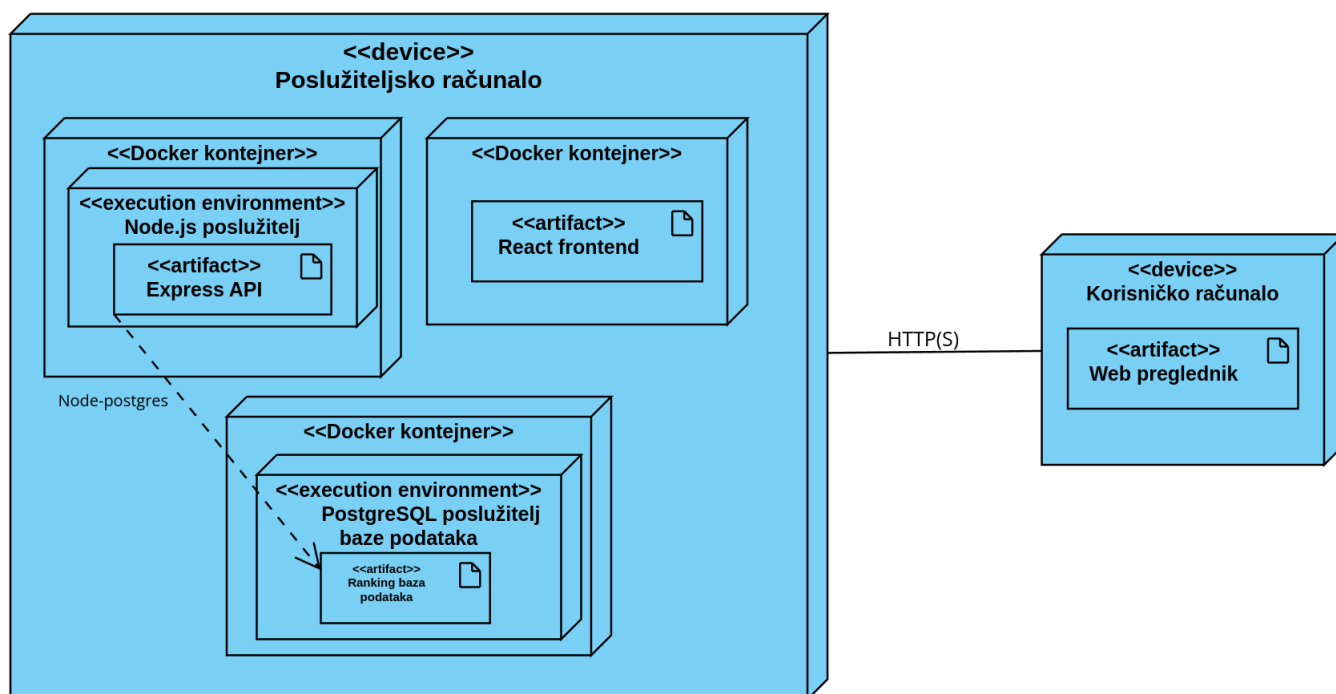
Zapisi u postavljenoj datoteci trebaju biti formatirani na ispravan način kako bi se podaci uspješno unijeli u bazu podataka. Na slici 5.20 vidimo nekoliko zapisa ispravno formatirane .xlsx datoteke. Svaki redak zapisa treba imati sedam stupaca čiji su nazivi redom: ime sveučilišta, vrijednost Q1, CNCI, IC, *Top* i *Award* indikatora te pozicija. Vrijednost podatka *Total* prema kojem se sortiraju sveučilišta nije potrebno upisati jer se on lako izračuna iz zadanih vrijednosti indikatora. Pozicija sveučilišta se također mogla izostaviti uz pretpostavku da će svaka postavljena datoteka imati zapise za sva sveučilišta za neku godinu u određenom području, ali na ovaj način korisnik može unijeti samo nekoliko zapisa za sveučilišta koja su mu od interesa te dobiti mogućnost grafičke usporedbe podataka na stranici opisanoj u odjeljku 5.3.

Massachusetts Institute of Technology (MIT)	58.3	81.1	75.4	100	100	1
Stanford University	53.8	87.2	69	85.5	80.2	2
University of California Berkeley	44.7	83.5	71	82.9	65.5	3
Carnegie Mellon University	53.8	73.7	74.3	99.8	37.8	4
Tsinghua University	100	74.7	65.5	81.3	0	5
Harvard University	57.3	81.6	73.1	54	46.3	6
ETH Zurich	51.6	79.9	84.8	72.8	26.7	7
Nanyang Technological University	87.5	85.5	90.3	54.4	0	8
University of Oxford	47.8	72.2	81.5	63.9	37.8	9

Slika 5.20: Prikaz zapisa datoteke sa stvarnim Shanghai Ranking podatcima

6. Opis rješenja i podatkovni model

Kao što je već rečeno u odjeljku 4.1, arhitektura ovog sustava ima tri dijela. U odjeljku koji slijedi opisano je kako je svaki od njih izveden te kako međusobno surađuju.



Slika 6.1: Dijagram razmještaja

6.1. Baza podataka

Ova web aplikacija temelji se na tri relacije u PostgreSQL bazi podataka. Nazivi relacija su: *award*, *ranking* i *rankingreal*.

6.1.1. Relacija *award*

Tablica 6.1: Atributi relacije *award*

Naziv atributa	Tip podatka	Opis
<i>id</i>	<i>integer</i>	jedinstveni identifikator zapisa i primarni ključ relacije
<i>uni</i>	<i>text</i>	naziv sveučilišta
<i>year</i>	<i>integer</i>	godina osvajanja nagrade
<i>share</i>	<i>double precision</i>	udio bodova sveučilišta
<i>category</i>	<i>text</i>	naziv područja za koje je nagrada osvojena

Relacija *award* sadrži podatke o osvojenim nagradama sveučilišta. Svaki zapis sadrži atribut *share* koji označava koliko bodova za ranking će sveučilište dobiti u ovisnosti o tome s koliko institucija je osvajač nagrade za vrijeme dobitka nagrade surađivao. Postupak izračuna udjela opisan je u odjeljku 2.4, a predstavlja recipročan broj broja institucija s kojima je osvajač nagrade surađivao. Svaki zapis također ima i atribut *category* koji označava područje u kojem je nagrada osvojena te može biti CSE ili EEE. Atribut *year*, koji predstavlja godinu osvajanja nagrade, bitan je kod izračuna težine s kojom množimo vrijednost indikatora.

Indikator *award* ima posebnu relaciju, za razliku od ostalih indikatora, zbog bolje organizacije relacija te jer se vrijednost indikatora *award* za jednu godinu koristi za izračun više rankinga sveučilišta. Ako bi indikator *award* bio u istoj relaciji kao i ostali indikatori dobila bi se relacija koja ima puno ponavljanja istih zapisa.

6.1.2. Relacija *ranking*

Relacija *ranking* sadrži vrijednosti koje se vežu za neko sveučilište u nekoj godini te u području CSE ili EEE. Nakon što se podatci dohvate s baze InCites, vrijednosti koje se vežu uz indikatore se izračunaju kako je opisano u odjeljku 2.5.1 te se unesu u ovu relaciju. Atribut *readingyear* pomoćni je atribut koji se koristi kod grafičkog prikaza napretka procjene rankinga nekog sveučilišta koji omogućava sortiranje zapisa od najstarijeg prema najmlađem.

Tablica 6.2: Atributi relacije *ranking*

Naziv atributa	Tip podatka	Opis
<i>id</i>	<i>integer</i>	jedinstveni identifikator zapisa i primarni ključ relacije
<i>uni</i>	<i>text</i>	naziv sveučilišta
<i>year</i>	<i>integer</i>	godina vrijednosti indikatora
<i>q1</i>	<i>double precision</i>	vrijednost uz indikator Q1
<i>cnci</i>	<i>double precision</i>	vrijednost uz indikator CNCI
<i>ic</i>	<i>double precision</i>	vrijednost uz indikator IC
<i>top</i>	<i>double precision</i>	vrijednost uz indikator <i>Top</i>
<i>category</i>	<i>text</i>	naziv područja za koje vrijede podatci
<i>readingyear</i>	<i>double precision</i>	vrijeme unosa zapisa u relaciju

6.1.3. Relacija *rankingreal*

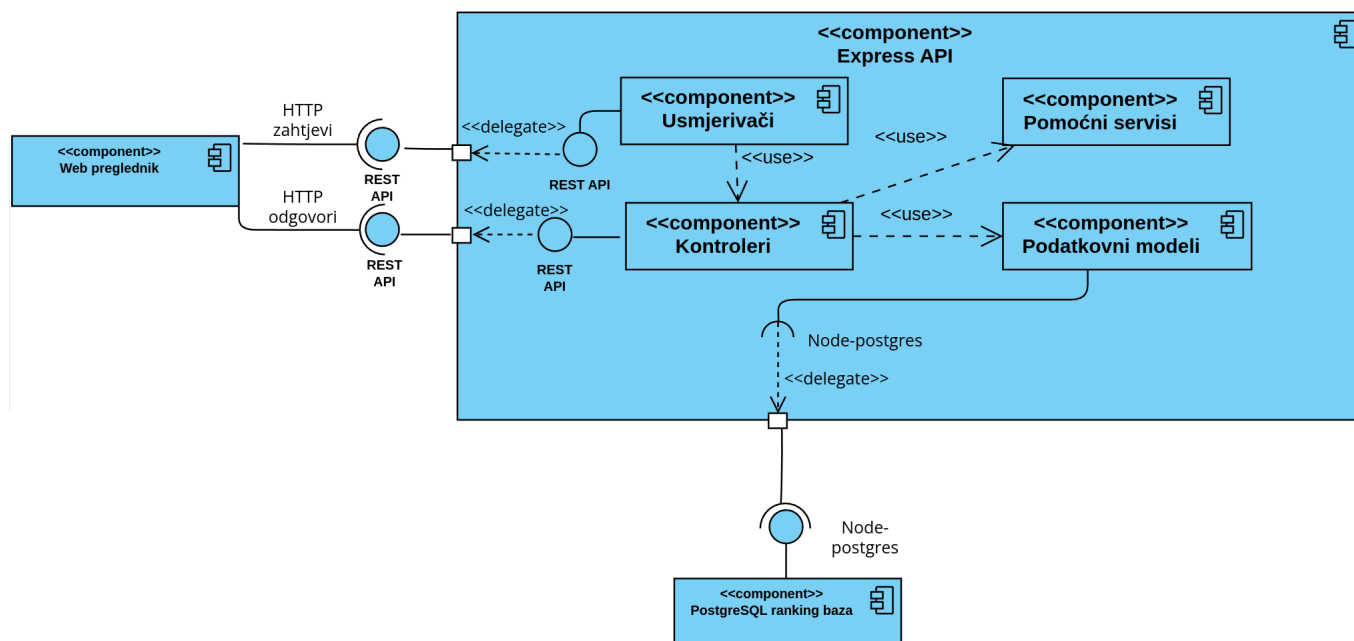
Tablica 6.3: Atributi relacije *rankingreal*

Naziv atributa	Tip podatka	Opis
<i>id</i>	<i>integer</i>	jedinstveni identifikator zapisa i primarni ključ relacije
<i>uni</i>	<i>text</i>	naziv sveučilišta
<i>year</i>	<i>integer</i>	godina vrijednosti indikatora
<i>q1</i>	<i>double precision</i>	vrijednost uz indikator Q1
<i>cnci</i>	<i>double precision</i>	vrijednost uz indikator CNCI
<i>ic</i>	<i>double precision</i>	vrijednost uz indikator IC
<i>top</i>	<i>double precision</i>	vrijednost uz indikator <i>Top</i>
<i>award</i>	<i>double precision</i>	vrijednost uz indikator <i>Award</i>
<i>category</i>	<i>text</i>	naziv područja za koje vrijede podatci

Relacija *rankingreal* sadrži ranking podatke prikupljene sa Shanghai Ranking stranice. Ova relacija sadrži sve vrijednosti koje se vežu uz indikatore za neko sveučilište u nekom području te za određenu godinu. Svaki put kad korisnik postavi datoteku na stranici s putom `/upload` ažurirat će se i upisati novi zapisi u ovu relaciju. Podatci iz ove relacije koriste se za usporedbu procjene rankinga te stvarnog Shanghai Ranking rankinga.

6.2. Backend poslužitelj

Poslužiteljska aplikacija napisana je koristeći Node.js JavaScript pokretačko okruženje koje omogućava jednostavnu izradu poslužiteljskih strana web aplikacija. Node.js u kombinaciji s Express radnim okvirom omogućava jednostavnu izradu RESTful API-ja ove web aplikacije čiji je dijagram komponenti prikazan na slici 6.2.



Slika 6.2: Dijagram komponenti Express API-ja, korisničkog preglednika te baze podataka

Poslužiteljska strana nudi korisniku RESTful API, a komunikacija se odvija putem HTTP protokola. Ova web aplikacija je jednostranična što znači da će se u korisnički preglednik pristupom aplikaciji prvo dostaviti sav potreban HTML, CSS i JavaScript kod za cijelu aplikaciju, a podatci za ažuriranje sadržaja stranice dohvaćat će se s poslužitelja koristeći RESTful API napisan pomoću Expressa kako je prikazano na slici 6.2.

Komunikacija Express API-ja i PostgreSQL baze podataka omogućena je Node-postgres modulom. Node-postgres nudi jednostavno sučelje za obavljanje operacija nad bazom podataka te stvaranje bazena otvorenih veza (engl. *connection pool*). Koristeći *connection pool* jednom otvorena veza s bazom podataka može se koristiti više puta te se ne mora stvarati nova veza s bazom podataka za svaki novi korisnički upit čime se štedi vrijeme i smanjuje mogućnost pogreške. Novostvoreni *connection pool* inicijalno je prazan te se nove veze dodaju lijeno, po potrebi.

Primjer stvaranja *connection pool*a:

```
const pool = new Pool({
  user: env.DB_USER,
  host: env.DB_HOST,
  database: env.DB_NAME,
  password: env.DB_PASSWORD
});
```

Vrijednosti potrebne za spajanje na bazu nalaze se u datoteci `.env` u koju se spremaju varijable okoline. Osim navedenih podataka potrebna je još vrijednost pristupa (engl. *port*) koja se konfigurira u datoteci `Dockerfile`. Primjer SQL upita koristeći stvoreni *connection pool*:

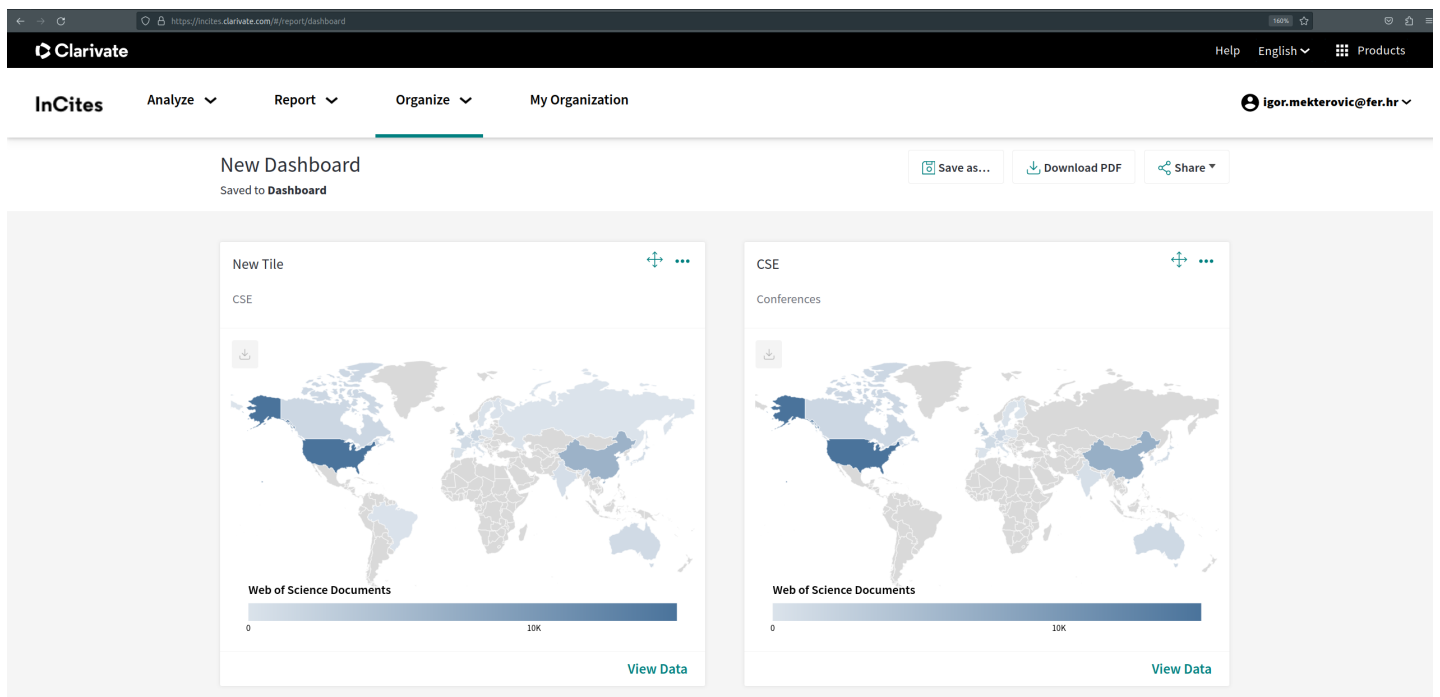
```
await pool.query(`delete from rankingreal
where year = $1 and category = $2`, [i, category]);
```

6.2.1. Punjenje baze podataka i korištenje alata Puppeteer

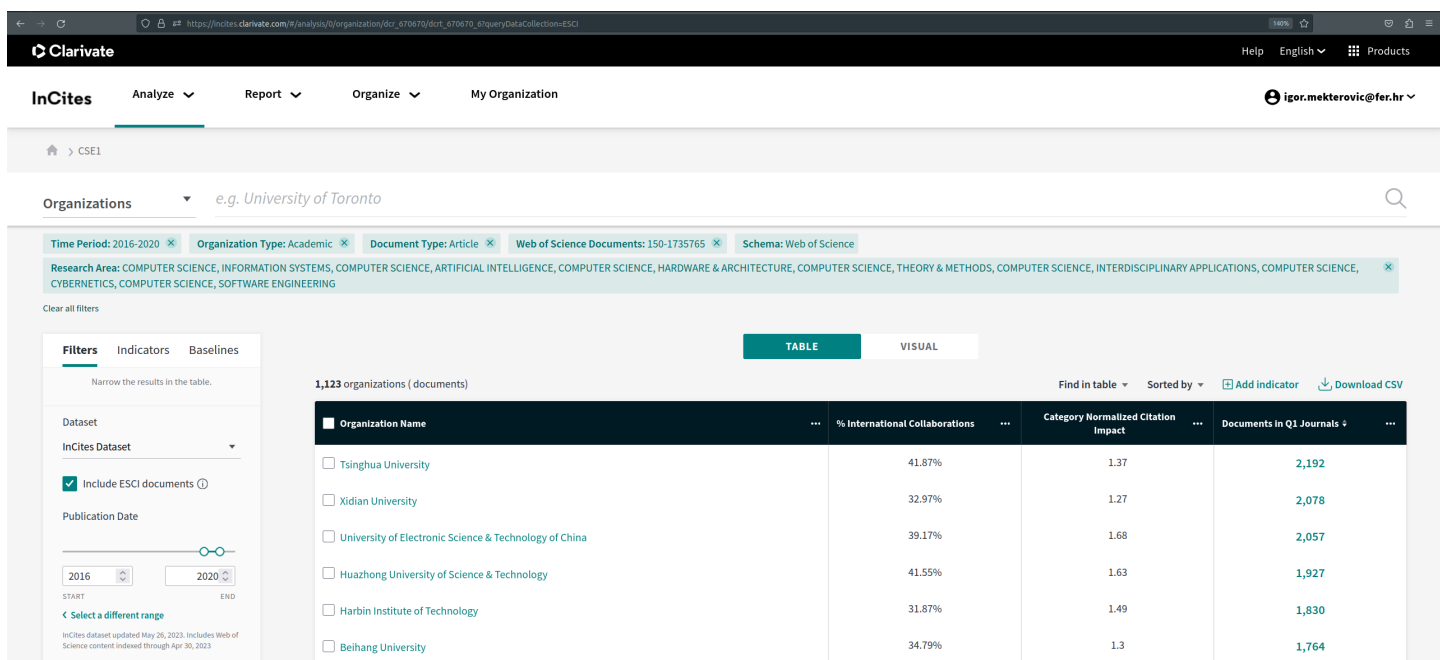
Prilikom prvog pokretanja poslužiteljske aplikacije potrebno je stvoriti tablice u bazi podataka i napuniti ih ranking podatcima od prijašnjih godina. Vrijednosti svih indikatora, osim indikatora *Award*, mogu se dobiti s InCites stranice. Vrijednosti indikatora *Award* teže je automatski prikupljati s web stranica zbog potrebne analize institucija s kojima je dobitnik nagrade u vrijeme dobitka bio povezan. Iz tog razloga ti podatci su ručno prikupljeni te kasnije uneseni u bazu podataka.

Dio programskog koda koji inicijalno puni bazu podataka nalazi se prema slici 6.2 u komponenti podatkovni modeli, a implementiran je pomoću funkcijskog izraza koji se odmah poziva (engl. *Immediately Invoked Function Expression*) prilikom pokretanja poslužiteljske aplikacije.

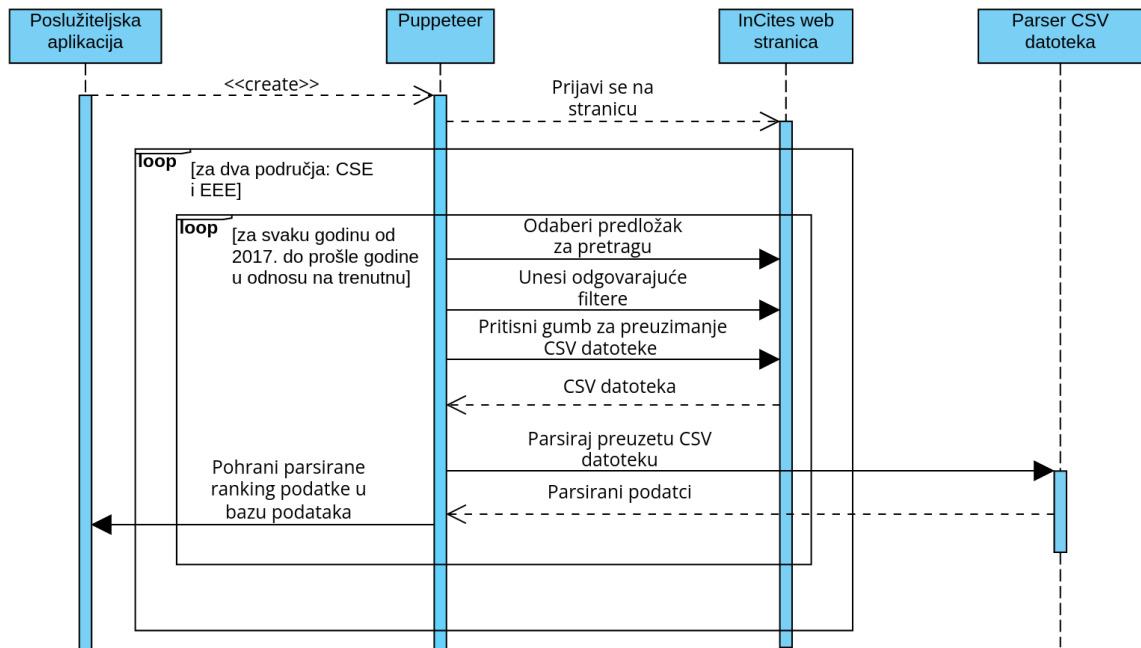
Lakše prikupljanje podataka sa stranice InCites omogućava izrada predložaka za pretragu koji se mogu pohraniti na korisničkom profilu u odjeljku *Dashboard*. Ukupno je pohranjeno 4 predložka. Za svaku od kategorija CSE i EEE pohranjen je predložak za dohvat indikatora Q1, CNCI, IC te još jedan predložak za indikator *Top*. U svakom predlošku unaprijed su odabrani filteri kao što su područje istraživanja, vrsta organizacije, vrsta dokumenta i vremenski period. Indikator *Top* ima vlastiti predložak jer zahtjeva unos filtera prema časopisima za područje EEE te prema konferencijama za područje CSE.



Slika 6.3: Stranica *Dashboard* na InCites web aplikaciji



Slika 6.4: Primjer izgleda predloška za dohvat ranking podataka



Slika 6.5: Sekvencijski dijagram inicijalnog punjenja baze podataka

Prema sekvencijskom dijagramu sa slike 6.5 funkcija koja inicijalno puni bazu podataka mora za svako područje i godinu od 2017. do prošle u odnosu na trenutnu prikupiti sve vrijednosti indikatora potrebnih za računanje rankinga.

Programski kod funkcije koja inicijalno puni bazu podataka:

```

(async () => {
    let number = 0;
    try {
        let rows = (await pool.query(`select count (*)
                                     from ranking`, [])).rows
        number = rows[0].count
    } catch (exception) {
    }
    if (number == 0) {
        await createTables();
        await seedAward("EEE")
        await seedAward("CSE")
        await seedRealRanking();

        dataSelector1 = `[aria-label='View more
                        data for EEE1']`
    }
})
  
```

```

dataSelector2 = `[aria-label='View more
                    data for EEE2']`
await seedIndicators("EEE", dataSelector1,
                    dataSelector2, 2017,
                    new Date().getFullYear()-1,
                    6, 2, 'ranking')

dataSelector1 = `[aria-label='View more
                    data for CSE1']`
dataSelector1 = `[aria-label='View more
                    data for CSE2']`
await seedIndicators("CSE", dataSelector1,
                    dataSelector2, 2017,
                    new Date().getFullYear()-1,
                    6, 2, 'ranking')
}
}) ()

```

Funkcija provjerava je li relacija za izračun rankinga već popunjena podacima te nastavlja s izvršavanjem samo ako je prazna. Za oba područja CSE i EEE puni bazu podataka vrijednostima indikatora *Award* te zatim priprema CSS selektore koji će omogućiti alatu Puppeteer pritisak na odgovarajući predložak. Funkcija poziva drugu funkciju *seedIndicators* koja kao argumente redom prima:

- a) područje istraživanja
- b) CSS selektor prvog predloška
- c) CSS selektor drugog predloška
- d) donju i gornju granicu godina za koje će se prikupiti podatci
- e) dva broja pomoću kojih se izračuna donja i gornja granica vremenskog perioda tijekom kojeg se uzimaju podatci za ranking određene godine (prema Shanghai Ranking algoritmu taj period je 4 godine, a postupak izračuna granica objašnjen je u dijelu 2.)
- f) ime relacije u koju će se unijeti dohvaćeni podatci.

Inicijalizacija alata Puppeteer radi se sljedećim programskim kodom:

```

const browser = await puppeteer.launch({
    executablePath: '/usr/bin/google-chrome',

```

```

    headless: true,
    args: [
      "--disable-gpu",
      "--disable-dev-shm-usage",
      "--disable-setuid-sandbox",
      "--no-sandbox",
    ]
  });

```

Kako bi alat ispravno radio unutar Docker kontejnera mora se postaviti parametar *executablePath* koji predstavlja putanju do preglednika kojeg Puppeteer koristi, isključiti grafičko sučelje preglednika postavljanjem parametra *headless* na *true* te predati polje argumenata koji omogućavaju dodatnu konfiguraciju, a preporučeni su u dokumentaciji alata za upotrebu unutar kontejnera. Nakon što je alat instanciran putem dobivene reference može se navigirati na željenu stranicu koristeći funkciju *goto("url/stranice")*.

Jednom kad je stranica učitana moguće je obavljati razne akcije na stranici kao što su:

- a) pritisak na gumb s funkcijom *click("CSS selektor")*
- b) upisati tekst u polje za unos teksta s funkcijom *type("CSS selektor", "tekst za unos")*
- c) pričekati obavljanje daljnjih akcija dok se željeni HTML element nije učitao koristeći funkciju *waitForSelector("CSS selektor")*.

Kombinacijom navedenih funkcija moguće je preuzeti podatke u obliku CSV datoteke koji su nakon parsiranja spremni za unos u bazu podataka. Programski kod navigiranja stranicom koristeći alat Puppeteer te parsiranje preuzetih CSV datoteka nalazi se prema 6.2 dijagramu u sklopu pomoćnih servisa.

Procjena rankinga trenutne godine te periodičko punjenje baze podataka s vrijednostima indikatora kako bi se mogao pratiti napredak sveučilišta kroz tekuću godinu ostvareno je pozivanjem funkcije *seedIndicators* svaka dva tjedna. Pozivanje funkcije svaka dva tjedna ostvareno je Node.js modulom Node cron.

Primjer korištenja modula Node cron:

```

const job = new cron.CronJob(`0 0 */14 * *`,
  async function() {
    await scheduledFunction();
  },

```

```

        null, false, 'UTC');
    job.start();

```

Node cron modul nalaže da se inicijalizira jedan *posao*. Za inicijaliziranje posla potrebno je predati sljedeće argumente:

- a) *string* koji označava koliko često i kada će se funkcija predana u drugom argumentu pozivati (u ovom primjeru to je svakih 14 dana)
- b) funkciju koja će se pozvati kad za to dođe vrijeme
- c) funkcija koja će se pozvati nakon što se posao zadan drugim argumentom obavio
- d) *boolean* parametar koji označava hoće li se posao pokrenuti odmah prilikom izlaska iz konstruktora
- e) *string* koji označava vremensku zonu

Funkcija *scheduledFunction* poziva funkcije *seedAward* i *seedIndicators*. Argumenti funkcije *seedIndicators* koji označavaju donji i gornju granicu godina za koje će se prikupiti podatci su isti te imaju vrijednost trenutne godine.

6.2.2. Rute i kontroleri

Krajnje točke (engl. *endpoint*) RESTful API-ja ostvarene su upotrebom usmjerivača (engl. *router*) i kontrolera u sklopu Express radnog okvira. Prije nego što Express aplikacija može primiti HTTP zahtjeve, potrebno je registrirati rute i povezati ih s odgovarajućim *routerima*. Primjer registriranja rute */rankingReal* i povezivanja na *router* *getRealRanking* gdje varijabla *app* predstavlja instancu Express aplikacije:

```
app.use('/rankingReal', require('./routes/getRealRanking'))
```

Dolaskom HTTP zahtjeva na poslužitelj Express aplikacija provjerava postoji li ruta koja se podudara s registriranim rutama te ako postoji zahtjev preusmjerava na definirani *router*.

Routeri su dio Express radnog okvira koji omogućavaju definiranje raznih *middleware* funkcija koje će se izvršavati na nekoj ruti. *Middleware* funkcije se izvršavaju između prijema HTTP zahtjeva i slanja HTTP odgovora. Primjer definiranja *middleware* funkcija za HTTP GET zahtjev na ruti */rankingReal*:

```

const express = require('express');
const router = express.Router();
const getRealRanking = require(
    './controllers/getRealRankingUni')

```

```
router.get('/',
  getRealRanking.getRealRankingForYearUniCategory)
module.exports = router
```

Iz priloženog programskog koda vidi se da je *middleware* funkcija definirana u kontroleru *getRealRankingUni*:

```
const getRealRankingForYearUniCategory = async (req, res)=>{
  const{ year, category, uni } = req.query
  let results = await getRealRanking(uni, year, category)
  if(results.length > 0){
    results[0].total = results[0].q1+results[0].cnci+
    results[0].ic*0.2+results[0].top+results[0].award
  }
  return res.json(results[0])
}
```

Parametri iz URL-a kod HTTP GET zahtjeva dostupni su unutar objekta *req.query*, a JSON tijelo poruke koje se šalje kod POST zahtjeva u objektu *req.body*. *Middleware* funkcije u kontrolerima obavljaju neku poslovnu logiku, komuniciraju s bazom podataka te vraćaju rezultat HTTP zahtjeva u JSON obliku.

Tablica 6.4: Krajnje točke *backend* poslužitelja

Redni broj	Ruta	Vrsta zahtjeva	Dodatni parametri
1.	/rankingsYear	GET	<i>page, limit, search, orderid, orderkey</i> <i>year, category</i>
2.	/rankingUni	GET	<i>year, category, uni</i>
3.	/uniCurrentYear	GET	<i>uni, category</i>
4.	/rankingReal	GET	<i>year, category, uni</i>
5.	/difference	GET	<i>year, category</i>
6.	/upload	POST	-

Krajnja točka rednog broja 1 iz tablice 6.4 koristi se za ostvarenje paginacije tablice rankinga za neku godinu u nekom području. Pomak od prvog zapisa rang liste računa se pomoću parametara *page*, a broj zapisa koje treba vratiti pomoću parametra *limit*. Sortiranje rang liste prema željenoj vrijednosti indikatora omogućeno je parametrima *orderkey* i *orderid*. *Orderkey* je *boolean* vrijednost koja označava silazno odnosno uzlazno sortiranje, a *orderid* je oznaka indikatora prema kojem se rang lista sortira.

Pretraga sveučilišta po imenu omogućena je parametrom *search* koji predstavlja podniz koji je korisnik unio kroz grafičko sučelje.

Krajnje točke rednog broja 2, 3, 4 iz tablice 6.4 služe za dohvat podataka s kojima se crtaju grafovi na stranici objašnjennoj u dijelu 5.3. Sve tri krajnje točke primaju parametre *year*, *category*, *uni* osim one rednog broja 3 jer se za parametar *year* podrazumijeva da je riječ o trenutnoj godini.

Krajnja točka rednog broja 5 iz tablice 6.4 služi za dohvat uspješnosti procjene rankinga neke godine u određenom području.

Krajnja točka rednog broja 6 iz tablice 6.4 služi za postavljanje .xlsx datoteka u kojoj su zapisani stvarni Shanghai Ranking podatci koji se kasnije koriste za usporedbu s procijenjenim ranking podatcima.

Request URL:	http://127.0.0.1:8080/rankingUni/?year=2019&category=CSE&uni=Massachusetts%20Institute%20of%20Technology%20(MIT)
Request Method:	GET
Status Code:	● 200 OK

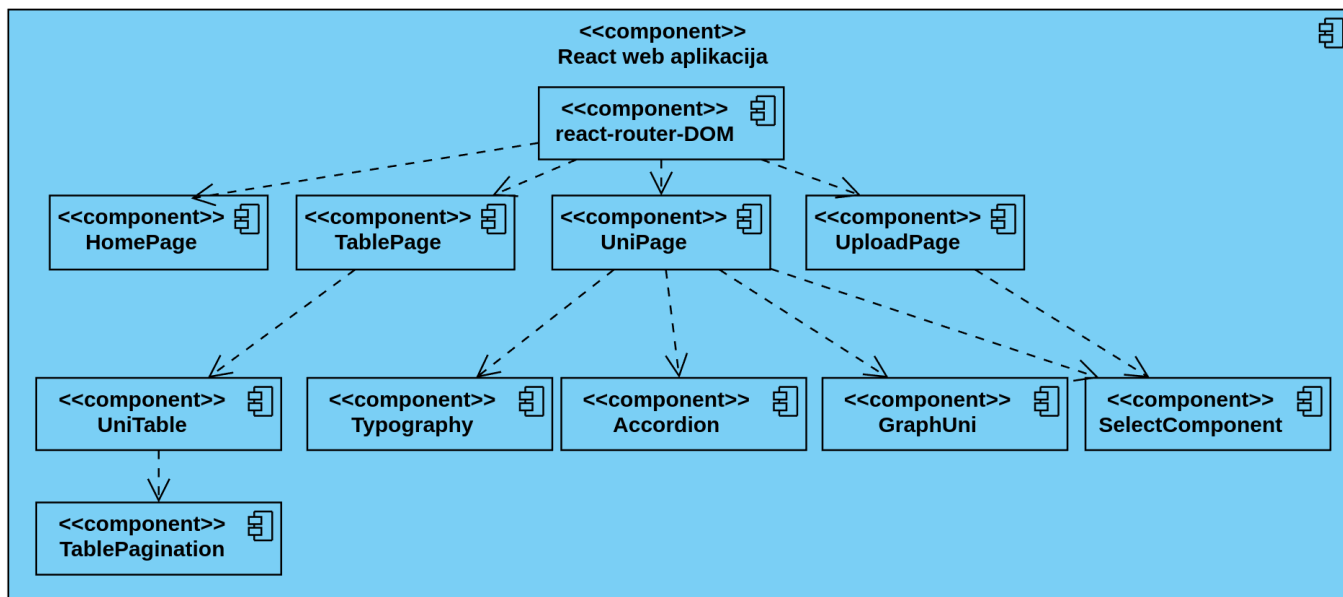
Slika 6.6: Primjer HTTP GET zahtjeva za dohvat ranking podataka sveučilišta Massachusetts Institute of Technology (MIT) u području CSE za 2019. godinu

```
award: 100
category: "CSE"
cnci: 93.02833433369373
ic: 70.3452458112659
id: 6047
position: 1
ql: 70.73404196469747
readingyear: "2023-05-14T10:32:28.856Z"
top: 90.06827570504586
total: 367.8997011656902
uni: "Massachusetts Institute of Technology (MIT)"
year: 2019
```

Slika 6.7: HTTP odgovor na zahtjev sa slike 6.6

6.3. React web korisničko sučelje

React biblioteka omogućava jednostavnu izradu web aplikacija. Programski kod ove web aplikacije organiziran je po stranicama (engl. *pages*) od kojih se svaka sastoji od više komponenti (engl. *components*).



Slika 6.8: Dijagram komponenti React web aplikacije

Tablica 6.5: Stranice iz dijagrama 6.8 i njihov opis

Naziv stranice	Opis
<i>HomePage</i>	početna stranica aplikacije
<i>TablePage</i>	ranking tablica
<i>UniPage</i>	stranica s grafovima za praćenje napretka sveučilišta
<i>UploadPage</i>	stranica za objavu Shanghai Ranking podataka

Prema slici 6.8 vidi se da je organizacija komponenti hijerarhijska jer svaka komponenta može gnijezditi proizvoljan broj drugih komponenti. Dijagram 6.8 sadrži navedene najbitnije komponente koje čine stranice ove web aplikacije, osim njih postoje druge jednostavne komponente i HTML elementi koji čine konačnu stranicu. Komponente ove web aplikacije su obične JavaScript funkcije koje vraćaju React elemente napisane JSX (*JavaScript XML*) sintaksom koja omogućava kombiniranje HTML elemenata i JavaScript koda. Svaka komponenta može, kao i obična funkcija, primiti argumente koji se u Reactu zovu *props*. Drugi način oblikovanja React komponenti je pisanjem razreda, no verzijom Reacta 16.8 uvedena je funkcionalnost React *useState hook* koja omogućava praćenje stanja komponente i u funkcijskim komponentama. Funkcijske komponente zahtijevaju pisanje manje programskog koda i jednostavnije su te se iz tog razloga danas koriste više od razrednih komponenti.

Primjer jednostavne komponente kojom je oblikovan padajući izbornik za odabir

vrijednosti poput godine rankinga i indikatora koji će se prikazivati na grafu:

```
export default function SelectComponent ({value,
                                         setValue, values, desc}) {
  const handleChange = (event) => {
    setValue(event.target.value);
  };
  return (
    <>
      <Box sx={{ minWidth: 120, mt:2}}>
        <FormControl sx={{minWidth: 100}}>
          <InputLabel
            id="demo-simple-select-label">
            {desc}
          </InputLabel>
          <Select
            labelId="demo-simple-select-label"
            id="demo-simple-select"
            value={value}
            label="Faktor"
            onChange={handleChange}
          >
            {values.map(value => (
              <MenuItem
                value={value}>
                {value}
              </MenuItem>
            ))}
          </Select>
        </FormControl>
      </Box>
    </>
  );
}
export { SelectComponent };
```

Praćenje stanja funkcijske komponente omogućava React *useState hook* koji

inicijalizacijom vraća polje veličine 2. Prvi element polja je varijabla u kojoj je pohranjena trenutna vrijednost stanja, a drugi element je referenca na funkciju s kojom se stanje može ažurirati. Svaka komponenta može imati proizvoljan broj stanja. Primjer inicijalizacije stanja koje pamti ocjenu uspješnosti rankinga, inicijalna vrijednost je 0:

```
const [index, setIndex] = React.useState(0)
```

Navedeno stanje koristi se unutar komponente *TablePagination* i pomoću JSX sintakse vrijednost stanja može se prikazati na stranici koju korisnik vidi sljedećim programskim kodom:

```
<div>
  Ocjena uspješnosti rankinga:
  {parseFloat((correct*100).toFixed(2)) || 0}
</div>
```

Svaki puta kada se vrijednost stanja *correct* promjeni, ažurirat će se prvo Reactov virtualni DOM, a kasnije samo onaj dio koji se treba promijeniti u preglednikovom DOM-u. Obavljanje radnji kada se vrijednost nekog stanja promjeni omogućava React hook *useEffect* koji prima polje ovisnosti (engl. *dependency array*). Hook *useEffect* pokreće se svaki put kad se komponenta prvi put rendera te kada vrijednosti stanja predanih u polju ovisnosti nemaju istu vrijednost koju su imali na prošlom renderu. Kombinacija funkcionalnosti *useEffect* i *useState* koristi se često u ovoj web aplikaciji. Jedan primjer je dohvat podataka s *backend* poslužitelja kada se vrijednost nekog stanja ili više njih promjeni. Programski kod koji dohvaća podatke smješta se unutar *useEffect* React hooka. Prilikom izrade stranice na kojoj je prikazana tablica rankinga *useEffect* je korišten za dohvat podataka o uspješnosti procjene rankinga:

```
React.useEffect(() => {
  (async () => {
    const request = await axios.get
      ('http://localhost:8080/difference
       /?year=${year}&category=${category}',
      {
        headers: {'Content-Type': 'application/json'},
        withCredentials: true
      });
    const response = request.data
    setCorrect(response.correct || 0)
```

```
    }) ()  
  }, [year])
```

Prikazani *useEffect* ima predano stanje *year* u polju ovisnosti te svaki put kada korisnik putem komponente za odabir godine promjeni godinu za koju želi gledati ranking podatke, izvest će se funkcija unutar *useEffect hooka* koja će dohvatiti ocjenu uspješnosti ranking podataka te ažurirati stanje *correct* čime će se automatski ažurirati i DOM te će korisnik imati prikaz uspješnosti rankinga za odabranu godinu. U prethodno prikazanom programskom kodu za stvaranje HTTP GET zahtjeva na *backend* poslužitelj korištena je biblioteka *Axios*.

Navigiranje aplikacijom omogućava vršna komponenta *react-router-dom*.

React-router-dom sastoji se od *BrowserRouter*, *Routes*, *Route* i *Link* komponenti.

BrowserRouter komponenta brine se za sinkroniziranje URL-a i stranice koja se prikazuje korisniku. Ona je roditeljska komponenta koja umata sve ostale komponente za navigiranje aplikacijom.

Routes komponenta grupira pojedine *Route* komponente te ih veže za određenu putanju.

Route komponenta registrira pojedinu stranicu na određenu putanju iz URL-a.

Link komponenta služi za umatanje HTML elemenata ili vlastitih *React* komponenti tako da pritisak miša na tu komponentu korisnika odvede na novu stranicu.

Pritisak miša na recimo komponentu *Link* ne uzrokuje HTTP zahtjev za novom HTML stranicom jer su svi elementi stranica već preuzeti i nalaze se u korisnikovom pregledniku.

React-router-dom radi interno usmjeravanje i brine se da pritiskom na komponentu *Link* se promjeni URL preglednika i ažurira prikaz stranice koju korisnik vidi. Upravo ova karakteristika je razlog zašto inicijalno učitavanje web aplikacije može potrajati, dok je kasnije navigiranje po stranicama kod jednostraničnih aplikacija brzo.

Primjer korištenja *react-router-dom* i registriranja stranice *HomePage* na putanju `/`:

```
<BrowserRouter>  
  <Routes>  
    <Route path="/" element = {<HomePage/>}></Route>  
  </Routes>  
</BrowserRouter>
```

Material UI biblioteka gotovih komponenti u ovoj web aplikaciji korištena je za iz-

radu dijelova pojedinih stranica. Primjer su komponente za odabir godine rankinga na stranici opisanoj u dijelu 5.2 te Accordion komponenta koja skriva odnosno prikazuje jedan od tri vrste grafa na stranici opisanoj u dijelu 5.3. Material UI komponente koriste se kao obične React komponente kojima možemo predati razne parametre za prilagođavanje vlastitim potrebama. Primjer upotrebe Material UI komponenti na stranici opisanoj u dijelu 5.3:

```
<Accordion sx={{width:'50%'}} expanded={isAccordion1Expanded}
  onChange={() =>
    setIsAccordion1Expanded(!isAccordion1Expanded)} >
  <AccordionSummary
    expandIcon={<ExpandMoreIcon />}
    aria-controls="panella-content"
    id="panella-header"
  >
    <Typography>
      Ranking sveučilišta {uni} kroz godine
    </Typography>
  </AccordionSummary>
  <AccordionDetails>
    <SelectComponent value={factor}
      setValue={setFactor}
      values={factors}
      desc='Faktor' >
    </SelectComponent>
    <GraphUni uni={uni} factor={factor} labels={labels1}
      dataFetch={getData1} isExpanded={isAccordion1Expanded}
      dataSets={dataSets1} />
  </AccordionDetails>
</Accordion>
```

U prethodnom primjeru komponenta Accordion gnijezdi druge komponente, a unutar komponente AccordionDetails navodi se što će biti prikazano kada korisnik proširi Accordion komponentu, a u ovom slučaju to je graf za praćenje ranking podataka tijekom godina.

Graf komponenta ostvarena je React-chartjs-2 bibliotekom koja nudi brojne vrse grafova, a u ovoj aplikaciji korišten je linijski graf:

```
<Line options={options} data={graphData}/>
```

Kako bi se graf ispravno nacrtao potrebno mu je predati vrijednosti oznaka na x i y osima.

Tailwind CSS radni okvir korišten je za izradu tablice rankinga prikazane na stranici 5.2. Tailwind CSS radi tako da skenira React datoteke te traži korištena imena CSS razreda te generira CSS datoteku sa željenim stilovima. Popis razreda koji se mogu koristiti nalazi se na službenoj Tailwind CSS stranici.

6.4. Docker

Kako bi se aplikacija mogla lako postaviti na bilo koje računalo s operacijskim sustavom koje podržava Docker virtualizaciju napisane su dvije datoteke Dockerfile te jedna docker-compose. Po jedna Dockerfile datoteka nalazi se u korijenskom direktoriju *backend* poslužitelja i React web aplikacije. Unutar Dockerfile datoteke potrebno je navesti sve korake koji se moraju obaviti kako bi se kontejner uspješno pokrenuo te konfigurirati određene parametre.

Za React web aplikaciju i Express API potrebno je konfigurirati korijenski direktorij unutar kojeg će kontejner raditi, instalirati sve dodane module naredbom *npm install*, podesiti pristup (engl. *port*) kojeg će aplikacija koristiti, kopirati izvorni kod aplikacije u radni direktorij kontejnera te napisati naredbu kojom se aplikacija pokreće.

Prethodnim razvijanjem aplikacije svi potrebni moduli već su instalirani te kako bi se izbjeglo njihovo dupliciranje u direktoriju kontejnera potrebno je dodati *.dockerignore* datoteku unutar koje se navodi imena direktorija i datoteka koje se ne kopiraju u korijenski direktorij kontejnera. Dockerfile Express API-ja razlikuje se malo od onog za React web aplikaciju jer je potrebno instalirati odgovarajući preglednik kojeg alat Puppeteer može koristiti. Alat Puppeteer bi sam instalirao preglednik Chromium no on zahtjeva dodatna podešavanja kako bi radio, iz tog razloga instaliran je preglednik Google Chrome koji dolazi automatski podešen.

Kada su napisane datoteke Dockerfile potrebno ih je povezati s datotekom *docker-compose*. Datoteka *docker-compose* sadrži putanje do svih Dockerfile datoteka koje čine web aplikaciju, imena koja će se dodijeliti pojedinim kontejnerima, vrijednosti varijabla okoline, mapiranje pristupa kontejnera na pristupe računala na

kojem se izvršava kontejner te upute za stvaranje kontejnera unutar kojeg se pokreće PostgreSQL baza podataka.

```
FROM node:17

ENV PUPPETEER_SKIP_CHROMIUM_DOWNLOAD true

RUN apt-get update && apt-get install gnupg wget -y && \
  wget --quiet --output-document=- https://dl-ssl.google.com/linux/linux_signing_key.pub | gpg --dearmor > /etc/apt/trusted.gpg.d/google-archive.gpg && \
  sh -c 'echo "deb [arch=amd64] http://dl.google.com/linux/chrome/deb/ stable main" >> /etc/apt/sources.list.d/google.list' && \
  apt-get update && \
  apt-get install google-chrome-stable -y --no-install-recommends && \
  rm -rf /var/lib/apt/lists/*

WORKDIR /usr/src/app

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 8080

CMD ["node", "API/server.js"]
```

Slika 6.9: Dockerfile datoteka za Express API

```
version: "3.9"
services:
  app:
    restart: on-failure
    build: .
    depends_on:
      - db
    environment:
      INCITES_USER: igor.mekterovic@fer.hr
      INCITES_PWD:
      PORT: 8080
    ports:
      - "8080:8080"

  db:
    image: postgres:14.7
    ports:
      - "4321:5432"
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: baze
      POSTGRES_DB: ShanghaiRanking2

  client:
    build: ./Client/ranking
    command: npm start
    depends_on:
      - app
    ports:
      - "3000:3000"
```

Slika 6.10: Docker-compose datoteka

Pokretanje baze podataka, Express API-ja te React web aplikacije sada je moguće naredbom: *docker-compose up*. Kada se Express API prvi put pokrene, krenut će se puniti baza podataka tako da svi ranking podatci neće biti dostupni odmah već nakon nešto više od 15 minuta.

7. Zaključak

Procjena rankinga sveučilišta omogućava uvid u kvalitetu nastave i profesora koji predaju na tom sveučilištu. Bez ikakve povratne informacije teško je zaposlenicima sveučilišta odrediti rade li dobar posao i što moraju popraviti kako bi im se reputacija poboljšala. Shanghai Ranking lista djelomično rješava taj problem za neka sveučilišta. Problem Shanghai Ranking liste je što za određena područja ne nudi informacije o rankingu sveučilišta ako nije u prvih 500 u svijetu. Ova web aplikacija rješava taj problem te se na procijenjenoj rang listi nalaze sva sveučilišta za koja postoje podatci u InCites i WoS bazama. Dobro razumijevanje računanja ranking podataka bilo je ključno za ovu web aplikaciju. Najveća odstupanja rankinga uočena su u području CSE i to kod indikatore *Top* zbog velikog broja konferencija na kojima sveučilišta mogu sudjelovati, a imena im nisu u potpunosti standardizirana te se ne podudaraju ona sa stranice Shanghai Ranking s onima u InCites bazi podataka.

Odabir tehnologija Node.js, Express, React, Puppeteer PostgreSQL te ostalih bio je ispravan zbog velike količine dokumentacije i dostupne podrške. Ključni dijelovi ove web aplikacije su bili ostvarenje periodičkog punjenja baze podataka s novim ranking podacima te izrada tablice i grafova ranking podataka sveučilišta. Dodavanjem Docker podrške aplikacija je postala široko primjenjiva zbog jednostavnog postavljanja na bilo koji poslužitelj koji podržava Docker.

Aplikacija ima raznih mogućnosti napretka. Neki od mogućih nadogradnji su: dodavanje rankinga za druga područja istraživanja, dodavanje novog algoritma i indikatora za izračun rankinga te detaljnija statistička analiza vrijednosti indikatora pojedinog sveučilišta.

LITERATURA

Axios dokumentacija i upute. URL <https://axios-http.com/docs/intro>.

React-chartjs-2 linijski dijagram. URL <https://react-chartjs-2.js.org/components/line>.

Express.js dokumentacija. URL <https://expressjs.com/>.

Node.js dokumentacija, a. URL <https://nodejs.org/en/docs>.

Node-cron upute, b. URL <https://www.npmjs.com/package/node-cron>.

Docker dokumentacija i upute, a. URL <https://docs.docker.com/>.

Predlošci za izradu datoteka Dockerfile i docker-compose, b. URL <https://docs.docker.com/samples/>.

Dobitnici IEEE Medal of Honor nagrade. URL <https://corporate-awards.ieee.org/>.

InCites baza podataka. URL <https://clarivate.com/>.

Material UI komponente. URL <https://mui.com/material-ui/>.

Node-postgres dokumentacija i upute. URL <https://node-postgres.com/guides/project-structure>.

React dokumentacija i tutorijali. URL <https://react.dev/learn>.

React router dokumentacija i tutorijali. URL <https://reactrouter.com/en/main/start/tutorial>.

Shanghai Ranking. URL <https://www.shanghairanking.com/>.

Stack Overflow forum s odgovorima za implementacijske probleme. URL <https://stackoverflow.com/>.

Tailwind CSS upute za instalaciju, a. URL <https://tailwindcss.com/docs/installation>.

Popis Tailwind CSS razreda, b. URL <https://tailwind.build/classes>.

Dobitnici Turingove nagrade. URL <https://amturing.acm.org/>.

Web of Science baza podataka. URL <https://www.webofscience.com/wos/woscc/basic-search>.

Aplikacija za praćenje rangiranja sveučilišta prema Šangajskoj listi

Sažetak

Završni rad razmatra izradu web aplikacije za praćenje rankinga sveučilišta u područjima *Computer Science & Engineering (CSE)* i *Electrical & Electronic Engineering (EEE)* prema algoritmu Shanghai Ranking sustava. Web aplikacija rješava problem nedostupnosti ranking podataka za niže rangirana sveučilišta te nudi praćenje procjene rankinga za trenutnu godinu prije nego što izađe službena Shanghai Ranking rang lista. Korisniku nudi tablični prikaz procijenjene rang liste s vrijednostima indikatora koji se koriste za izračun rankinga, grafički prikaz promjene ranking podataka odabranog sveučilišta, usporedbu procjene rankinga sa Shanghai Ranking podacima te pretragu baze podataka prema imenu sveučilišta. Glavne tehnologije korištene za izradu aplikacije su: React, Node.js, Express.js, PostgreSQL, Puppeteer i Docker.

Ključne riječi: Shanghai Ranking, *Computer Science & Engineering (CSE)*, *Electrical & Electronic Engineering (EEE)*, Puppeteer, Node.js, Express, React, PostgreSQL, procjena rankinga, Docker, grafički prikaz ranking podataka

An application for monitoring the ranking of universities according to the Shanghai List

Abstract

Final work considers the creation of a web application for monitoring university rankings in the fields of Computer Science & Engineering (CSE) and Electrical & Electronic Engineering (EEE) according to the Shanghai Ranking system algorithm. The web application solves the problem of the unavailability of ranking data for lower-ranked universities and offers monitoring of the ranking assessment for the current year before the official Shanghai Ranking ranking list is out. It offers the user a tabular view of the estimated ranking along with the values of the indicators used to calculate the ranking, graphical representation of changes in the ranking data of the selected university, comparison of the ranking assessment with the Shanghai Ranking data, and database search by university name. The main technologies used to create the application are: React, Node.js, Express.js, PostgreSQL, Puppeteer and Docker.

Keywords: Shanghai Ranking, Computer Science & Engineering (CSE), Electrical & Electronic Engineering (EEE), Puppeteer, Node.js, Express, React, PostgreSQL, ranking assessment, Docker, graphic display of ranking data