

Interaktivna Web aplikacija temporalne analitike: praćenje trendova medijskih objava

Pavlaković, Marko

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:195:318956>

Rights / Prava: [Attribution-NonCommercial-ShareAlike 4.0 International/Imenovanje-Nekomercijalno-Dijeli pod istim uvjetima 4.0 međunarodna](#)

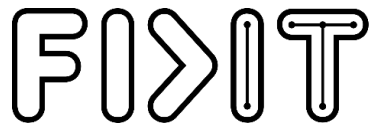
Download date / Datum preuzimanja: **2024-10-19**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)





Sveučilište u Rijeci
**Fakultet informatike
i digitalnih tehnologija**

Sveučilišni prijediplomski studij Informatika

Marko Pavlaković

**Interaktivna Web aplikacija temporalne
analitike: praćenje trendova medijskih objava**

Završni rad

Mentor: prof. dr. sc. Ana Meštrović

Rijeka, 22.6.2024.

Rijeka, 3. lipnja 2024.

Zadatak za završni rad

Pristupnik/ica: Marko Pavlaković

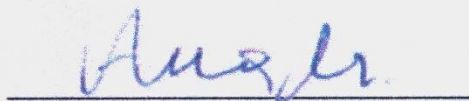
Naziv završnog rada: Interaktivna web aplikacija temporalne analitike: praćenje trendova medijskih objava

Naziv završnog rada na engleskom jeziku: Interactive Temporal Analysis Web Application: Exploring Trends in Online Media

Sadržaj zadatka:

Sustavi za autoamtizirano prikupljanje i analizu medijskih objava omogućavaju bolje informiranje. To je posebno važno u primjerice kriznim situacijama kada se plasiraju velike količine objava u kojima se teško snalaziti. Zadatak završnog rada je razviti interaktivnu web aplikaciju koja omogućava analizu i praćenje trendova u medijskim objavama. U radu je potrebno dati pregled tehnologija koje su korištene te opisati razvoj aplikacije.

Mentor/ica
Prof. dr. sc. Ana Meštrović



Voditelj za završne radove
Izv. prof. dr. sc. Miran Pobar



Zadatak preuzet: 3.6.2024.


(potpis pristupnika/ice)

Sažetak

Ovaj rad predstavlja proces razvoja i navodi dostupne funkcionalnosti interaktivne web aplikacije CroTrends, dizajnirane za vremensku analizu trendova u *online* medijima. Aplikacija je izrađena s Next.js okvirom baziranom na React-u, Tailwind CSS-om (Cascading Style Sheets) i Supabase bazom podataka. Korisnicima nudi jednostavno i ujednačeno sučelje za istraživanje učestalosti i konteksta određenih ključnih riječi u različitim predefiniranim vremenskim razdobljima.

Proces započinje razvojem dizajna u alatu Figma, nakon čega slijedi implementacija u Next.js-u s Tailwind CSS-om i integracija sa Supabase za upravljanje bazom podataka. Cilj je korisnicima pružiti jasne i moderne vizualizacije podataka koje pomažu u razumijevanju trendova digitalnih medija tijekom vremena. Postoji više mogućih razloga korištenja opisane web aplikacije. Neki od njih uključuju: alat za analizu odabira marketing plana kompanija, izvor sažetog stanja medija prikazan ključnim riječima za Hrvatsku, pa čak i znatizelja samog korisnika.

Ključne riječi: temporalna analitika; web aplikacija; Next.js; Tailwind CSS; Supabase; *online* medijski trendovi; analiza *online* medija

Sadržaj

| | |
|--|----|
| 1. Uvod..... | 1 |
| 2. Moderni <i>Full Stack</i> razvoj: alati i tehnologije..... | 3 |
| 2.1. Klijentske tehnologije | 3 |
| 2.2. Pozadinske tehnologije..... | 4 |
| 2.3. Tehnologije baza podataka | 5 |
| 2.4. Korišteni tehnološki stog CroTrends aplikacije | 6 |
| 2.4.1. Next.js | 6 |
| 2.4.2. Tailwind CSS..... | 6 |
| 2.4.3. Supabase..... | 8 |
| 3. Proces dizajna i postavljanja CroTrends projekta | 9 |
| 3.1. Izrada prototipa u Figma | 9 |
| 3.2. Korisničko iskustvo..... | 10 |
| 3.3. Važnost responzivnog dizajna | 10 |
| 3.4. Postavljanje Next.js projekta s Tailwind CSS-om..... | 12 |
| 3.5. Integracija Supabase baze podataka | 13 |
| 4. CroTrends Korisničko sučelje i funkcionalnosti | 15 |
| 4.1. Poglavlja korisničkog sučelja..... | 15 |
| 4.2. Mehanizam pretraživanja | 22 |
| 4.3. Dohvaćanje i vizualizacija podataka | 23 |
| 5. Internetsko udomljavanje CroTrends aplikacije na Vercelu..... | 27 |
| 5.1. GitHub integracija | 27 |
| 5.2. Implementacija na Vercelu | 28 |
| 6. Rezultati i rasprava o CroTrends aplikaciji..... | 29 |
| 6.1. Procjena učinkovitosti | 29 |
| 6.2. Povratne informacije korisnika | 29 |
| 6.3. Usporedba sa drugim alatima za analizu trendova..... | 29 |
| 7. Zaključak..... | 31 |
| 8. Literatura | 32 |
| 9. Popis tablica | 33 |
| 10. Popis slika | 34 |
| 11. Popis priloga..... | 35 |

| | |
|--|----|
| 12. Prilog 1: Supabase funkcija `get_term_count_last_time_interval` | 36 |
| 13. Prilog 2: Supabase funkcija `get_combined_term_count_last_time_interval` | 37 |

1. Uvod

Web stranice su se značajno razvile u posljednjih nekoliko desetljeća, s napretkom u klijentskim i pozadinskim tehnologijama. U početku, 1990-ih godina je web razvoj bio uglavnom statičan, s ograničenom interaktivnošću i funkcionalnošću (Rios, 2022). Kako je vrijeme prolazilo dolazila je potreba za sve složenijim i naprednijim dizajnom. Tako je 2000-ih došlo do uvođenja tehnologija poput JavaScripta, CSS-a i skriptnih jezika na strani poslužitelja (PHP, ASP.NEXT). 2010-ih su se pojavili prvi moderni okviri kao što su Angular, React i Node.js, koji omogućuju interaktivnije i responzivnije web stranice (Craig, n.d.; Prasad, 2024; Tiwari, 2023).

Evolucija web stranica promijenila je način na koji se web-aplikacije izrađuju, omogućujući interaktivnija, responzivnija i skalabilnija rješenja. Moderne web stranice nisu samo vizualno privlačne, već nude i napredne funkcionalnosti koje poboljšavaju korisničko iskustvo (engl. User Experience, UX).

Brzi rast internetskih medija generirao je ogromnu količinu podataka, zbog čega je neophodno imati alate koji mogu učinkovito analizirati i prezentirati te informacije. Značajan broj takvih alata već postoji (Lua, 2023), unatoč tome samo je nekolicina primjenjiva i za područje Hrvatske, kao što su Mention¹, Google Trends², Talkwalker³, itd. Uloga ovog rada na višoj razini je da se ponudi novi alat za analizu *online* medija koji je primjenjiv za područje Hrvatske.

Primarni cilj je razviti web aplikaciju koja je prilagođena korisnicima i omogućuje im istraživanje vremenskih trendova za objave u *online* medijima. Unosom ključne riječi korisnici mogu vidjeti koliko je često ona spominjana u objavama tijekom različitih vremenskih razdoblja i identificirati povezane oznake. Na taj način korisnik može dobiti bolji uvid u trendove objavljivanja u *online* medijima, poboljšavajući svoje razumijevanje konteksta i evolucije ključne riječi.

Ovaj je rad strukturiran u nekoliko dijelova. Započinje retrospekcijom na evoluciju web stranica, nakon čega slijedi istraživanje modernog *Full Stack* razvoja, s detaljima o najpopularnijim tehnologijama za klijentsku (engl. frontend) i pozadinsku (engl. backend) stranu, te tehnologijama baza podataka (engl. database). Zatim se rad bavi tehnološkim skupom korištenim u ovom projektu, kao što je Next.js, Tailwind CSS i Supabase, te objašnjava njihov odabir i prednosti. Poglavlje o procesu dizajna naglašava korištenje alata Figma za izradu prototipa i ključna razmatranja dizajna. Poglavlje implementacije pokriva postavljanje i integraciju odabranih tehnologija. Poglavlje korisničkog sučelja detaljno opisuje dizajn i funkcionalnost dijelova web stranice. Rad zatim opisuje funkcionalnost `Explore` stranice koja sadrži svu logiku za unos ključne riječi, dohvaćanje podataka i njihovu vizualizaciju. Poglavlje o internetskom udomljavanju na Vercelu objašnjava integraciju s platformom GitHub i proces implementacije sa platformom Vercel. Na kraju, rad predstavlja usporedbu s alatom Google

¹ <https://mention.com/en/>

² <https://trends.google.com/trends/>

³ <https://www.talkwalker.com/>

Trends. Poglavlje s rezultatima i raspravom ocjenjuje izvedbu aplikacije i povratne informacije korisnika, a rad završava zaključkom i prijedlozima za buduća istraživanja.

2. Moderni *Full Stack* razvoj: alati i tehnologije

Moderni *Full Stack* razvoj koristi niz alata i tehnologija za izgradnju robusnih web aplikacija. Ovi alati pojednostavljuju razvojne procese i poboljšavaju performanse aplikacija. Svi alati i tehnologije za *Full Stack* development dijele se u 3 glavne grupe: klijentske tehnologije, pozadinske tehnologije i tehnologije baza podataka (Craig, n.d.; Prasad, 2024; Tiwari, 2023).

2.1. Klijentske tehnologije

Klijentski razvoj fokusiran je na korisničko sučelje i korisničko iskustvo web aplikacija. Uključuje stvaranje vizualnih elemenata i interaktivnih značajki s kojima korisnici izravno rukuju. Svakim danom izlazi sve veći broj klijentskih tehnologija s novim funkcionalnostima koje poboljšavaju razvoj web stranica. Među svim tehnologijama koje su dostupne na današnji dan najpopularnije bi bile: React, Angular i Vue.js (Prasad, 2024).

Razvijen od strane Facebooka, React je JavaScript biblioteka za izgradnju korisničkog sučelja. Programerima omogućuje stvaranje komponenti korisničkog sučelja za višekratnu uporabu čime razvojni proces postaje učinkovitiji. React-ov virtualni DOM (Document Object Model, hrv. dokumentni objektni model) poboljšava performanse minimiziranjem broja ažuriranja prema stvarnom DOM-u, što rezultira bržim renderiranjem (Prasad, 2024).

Održavan od strane Google-a, Angular je sveobuhvatna platforma za izradu mobilnih i desktop web aplikacija. Koristi TypeScript, baziran na JavaScriptu, i pruža robusne alate za usmjeravanje, upravljanje obrascima i klijentsko-poslužiteljsku komunikaciju. Angularovo uvođenje ovisnosti i dvosmjerno povezivanje podataka pojednostavljuju razvoj i poboljšavaju performanse aplikacije (Prasad, 2024).

Vue.js je progresivni okvir za izgradnju korisničkih sučelja. Dizajniran za postupnu prilagodbu, Vue može funkcionirati kao biblioteka za integraciju u postojeće projekte ili kao cjeloviti okvir za izradu web aplikacija. Njegov reaktivni sustav povezivanja podataka osigurava da se promjene u stanju aplikacije odmah odražavaju na korisničkom sučelju (Prasad, 2024).

2.2. Pozadinske tehnologije

Pozadinski razvoj usredotočen je na poslužiteljsku stranu web aplikacija, uključujući poslužiteljsku logiku, interakcije s bazom podataka i autentifikaciju. Poslužiteljska logika odnosi se na implementaciju funkcija koje vrše pozive slanja i primanja podataka iz neke baze podataka. Baza podataka koja nudi te informacije može biti pod vlasništvu samog programera/tvrtke ili otvorena baza od strane nekog drugog vlasnika. Neke od najpoznatijih pozadinskih tehnologija su: Node.js, Django i Ruby on Rails (GeeksForGeeks, 2023).

Node.js je JavaScript runtime izgrađen na Chrome-ovom V8 motoru (engl. engine-u), poznatom po I/O modelu koji se ne blokira. To čini Node.js laganim i učinkovitim, posebno prikladnim za izgradnju skalabilnih mrežnih aplikacija. Njegov opsežni ekosustav, uključujući npm (Node Package Manager, hrv. Node upravitelj paketa), pruža brojne module i biblioteke za poboljšavanje pozadinskog razvoja (GeeksForGeeks, 2023).

Django je Python web okvir visoke razine koji promovira brzi razvoj i čist, pragmatičan dizajn. Uključuje ugrađene značajke kao što su ORM (Object-Relational Mapping, hrv. objektno-relacijsko mapiranje), provjera autentičnosti i mehanizam za izradu predložaka. Django slijedi filozofiju „baterije uključene“ (engl. batteries-included), nudeći široku paletu alata za pojednostavljenje pozadinskog razvoja koji su dostupni odmah nakon inicijalizacije projekta (GeeksForGeeks, 2023).

Često nazivan i Rails, Ruby on Rails je okvir za izradu web aplikacija na poslužiteljskoj strani (engl. server-side) napisan u Ruby-ju. Slijedi MVC (Model-View-Controller, hrv. model-prikaz-kontroler) arhitekturu i naklašava konvenciju umjesto konfiguracije, smanjujući količinu odluka koju programeri moraju donijeti. Rails uključuje sve što je potrebno za stvaranje web aplikacije vođene bazom podataka, kao što su usmjeravanje, migracija baze podataka i skele (engl. scaffolding) (GeeksForGeeks, 2023).

2.3. Tehnologije baza podataka

Tehnologije baze podataka ključne su za pohranjivanje, dohvaćanje i upravljanje podacima u web aplikacijama. Mogu biti relacijske ili ne relacijske (NoSQL), a svaki je prikladan za različite vrste podataka i slučajeve upotreba. Najpoznatije bazne tehnologije ubrajaju: MongoDB, PostgreSQL i Firebase (Ostrowska, 2024).

MongoDB je NoSQL baza podataka poznata po svojoj fleksibilnosti i skalabilnosti. Pohranjuje podatke u dokumente slične JSON-u, što olakšava upravljanje i postavljanje upita velikim skupovima podataka. MongoDB je posebno koristan za aplikacije koje zahtijevaju pohranu i dohvaćanje podataka velikih razmjera, kao što su platforme za e-trgovinu i sustavi za upravljanje sadržajem (Ostrowska, 2024).

PostgreSQL je relacijska baza podataka otvorenog koda koja nudi napredne značajke kao što su ACID (Atomicity Consistency Isolation Durability, hrv. atomnost konzistentnost izolacija durabilnost) usklađenost, pretraživanje cijelog teksta i podrška za JSON vrste podataka. Njegova proširivost i podrška za složene upite čine ga robusnim izborom za aplikacije koje zahtijevaju pouzdano i učinkovito upravljanje podacima (Ostrowska, 2024).

Firebase, koji je razvio Google, pruža NoSQL bazu podataka u stvarnom vremenu, autentifikaciju, *hosting* i funkcije u oblaku. Firebase-ove mogućnosti sinkronizacije u stvarnom vremenu čine ga idealnim za aplikacije kojima su potrebna trenutna ažuriranja i besprijekorna korisnička iskustva, kao što su aplikacije za chat i alati za suradnju (Ostrowska, 2024).

2.4. Korišteni tehnološki stog CroTrends aplikacije

U ovom poglavlju raspravlja se o alatima i tehnologijama korištenim u razvoju CroTrends web aplikacije. Obuhvatit će se specifičnosti Next.js okvira, Tailwind CSS okvira i Supabase baze podataka, ističući njihove uloge, prednosti i razloge za njihov odabir. Svaki alat igra ključnu ulogu i izgradnji robusne, učinkovite i skalabilne web aplikacije.

2.4.1. Next.js

Next.js je popularan okvir baziran na React-u koji pruža:

- **SSR** (Server-Side Rendering, hrv. iscrtavanje na strani poslužitelja) – Osigurava brže učitavanje stranica renderiranjem (hrv. iscrtavanjem) na poslužitelju
- **SST** (Static Site Generation, hrv. generiranje statične web stranice) – Unaprijed renderira stranice za vrijeme builda (hrv. izrade), poboljšavajući performanse i SEO
- **API** (Application Programming Interface, hrv. sučelje za programiranje aplikacija) rute – Pojednostavljuje stvaranje pozadinskih endpointa (hrv. krajnjih točaka) unutar istog projekta

Pored iznad spomenutih značajki tu je i optimizirano iskustvo za programere. Pojednostavljuje razvoj složenih web aplikacija nudeći značajke kao što su usmjeravanje (engl. routing) i dijeljenje koda (engl. code-splitting) i bez potrebe za dodatnim alatima, odmah nakon postavljanja projekta. Ove funkcionalnosti olakšavaju izvedbu aplikacije i njeno skaliranje u slučaju većeg broja korisnika od početno planiranog. Upravo iz tih razloga je Next.js prikladan izbor za izgradnju CroTrends interaktivne aplikacije s velikim brojem podataka (Vercel, 2024).

2.4.2. Tailwind CSS

Tailwind CSS je korisnost-prvo (engl. utility-first), mobilno-prvo (engl. mobile-first) CSS okvir koji omogućuje brzo oblikovanje web stranice. Promovira konzistentan sustav dizajna i omogućuje programerima stvaranje prilagođenih dizajna bez pisanja opsežnog CSS-a. Tailwind-ove utility klase olakšavaju primjenu stilova izravno u HTML-u, osiguravajući da je dizajn i fleksibilan i održiv. Uklanja potrebu za osmišljavanjem imena CSS klasa i olakšava kasnije promjene u dizajnu bez brige o mogućim kvarovima na drugim elementima (Tailwind Labs Inc., 2024).

Ključne prednosti Tailwind CSS-a:

- Konzistentnost
- Fleksibilnost
- Efikasnost

2.4.3. Supabase

Supabase je pozadinski SAAS (Software As A Service, hrv. softver kao usluga) otvorenog koda (engl. open-source) koji pruža bazu podataka u stvarnom vremenu, autentifikaciju korisnika i pohranu datoteka. Temeljen na relacijskoj bazi podataka PostgreSQL, Supabase nudi robusna i skalabilna rješenja za upravljanje podacima sa besplatnim planom. Pošto je ovo open-source usluga moguće je i samostalno postaviti *hosting* za bazu, ili koristiti neke od njihovih plaćenih planova kao što je vidljivo u Tablici 1, u slučaju većeg broja korisnika ili širine pojasa (engl. bandwidth) (Supabase Inc., 2024).

Tablica 1 - Planovi Supabase usluge (Supabase Inc., 2024)

| Dostupni planovi | Cijena | Broj aktivnih korisnika | Veličina baze | Bandwidth | Veličina pohrane |
|-------------------------|--------------|-------------------------|---------------|-----------|------------------|
| FREE | \$0/mjesec | 50 000 | 500 MB | 5 GB | 1 GB |
| PRO (najpopularniji) | \$25/mjesec | 100 000 | 8 GB | 250 GB | 100 GB |
| TEAM | \$599/mjesec | 100 000 | 8 GB | 250 GB | 100 GB |
| ENTERPRISE | Po dogovoru | - | - | - | - |

PRO i TEAM planovi su po gore navedenim vrijednostima isti, ali TEAM plan nudi dodatne značajke poput:

- SOC2
- HIPAA available as paid add-on
- Read only and Biling member roles
- SSO for Supabase Dashboard
- Priority email support & SLAs
- Daily backups stored for 14 days (u odnosu na 7 u PRO planu)
- 28-day log retention (u odnosu na 7 u PRO planu)

Sve ovo ga čini odličnim izborom za aplikacije koje zahtijevaju ažuriranje podataka u stvarnom vremenu, ali nemaju velik broj korisnika ili budžet (Supabase Inc., 2024).

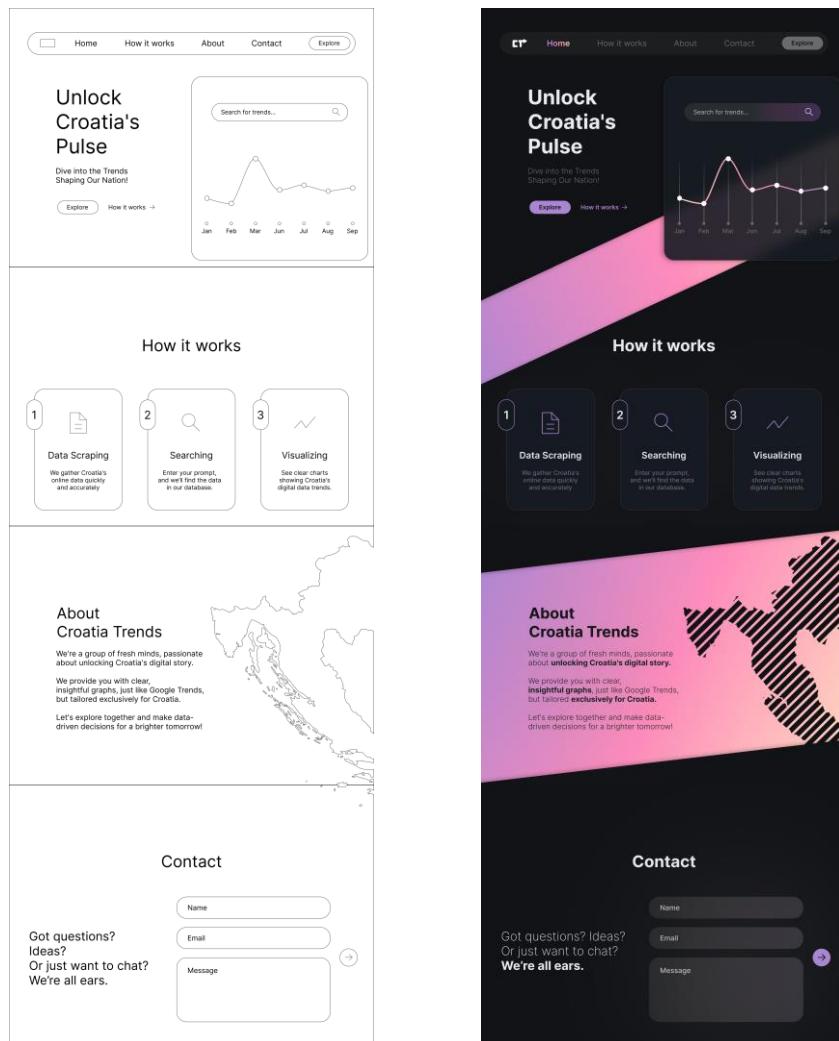
3. Proces dizajna i postavljanja CroTrends projekta

U ovom poglavlju izložit će se postupak dizajniranja i postavljanja projekta. Pokrivat će se različite faze, počevši od stvaranja prototipa u Figma, osiguravajući ugodno i responzivno korisničko iskustvo. Nakon faze projektiranja detaljno se opisuje inicijalizacija Next.js projekta sa Tailwindom i Supabase bazom podataka.

3.1. Izrada prototipa u Figma

Proces dizajna započet je izradom prototipa u Figma, alatu za kolaborativni dizajn. Figma prototip uključuje žičane okvire (engl. wireframe) i visoko-precizne (engl. high-fidelity) dizajne za svako poglavlje web aplikacije. Prototip je poslužio kao nacrt za fazu razvoja što je omogućilo manje zastoja u fazi razvoja/implementacije same aplikacije.

CroTrends jezik dizajna bio je usredotočen na jednostavnost i upotrebljivost, s minimalističkim pristupom kako bi se osiguralo da korisnici mogu jednostavno i brzo upravljati aplikacijom. Za inspiraciju su korišteni moderni i trenutno aktualni trendovi, što je inspiriralo odabranu tamnu paletu sa pastelnim bojama i dizajn elementa, te odjeljka na stranici.



Slika 1 - Wireframe i Mockup CroTrends web aplikacije

3.2. Korisničko iskustvo

Ključna zapažanja tijekom faze dizajna uključivala su jednostavnost navigacije, intuitivne tokove interakcije i responzivni dizajn za prilagodbu različitim uređajima. Korisničko sučelje dizajnirano je da bude čisto i minimalističko, s fokusom na pružanje jasnih i korisnih uvida na podatke korisnicima.

3.3. Važnost responzivnog dizajna

Responzivni dizajn odnosi se na pristup stvaranja web aplikacija koje dobro rade na različitim uređajima i veličinama zaslona. Osigurava da je korisničko iskustvo cjelovito i dostupno, bez obzira na uređaj koji se koristi (Marcotte, 2010; Learning Web Design, 2018).

Neke od prednosti responzivnog web dizajna uključuju: poboljšano korisničko iskustvo, prednosti SEO-a jer tražilice daju prioritet responzivnim web lokacijama i povećani doseg jer korisnici web stranici mogu pristupiti s različitih uređaja (Learning Web Design, 2018).

Dostupno je više tehnika kojima se može postići responzivni dizajn. Trenutno najkorištenija je medijski upiti (engl. Media Queries). Pomoću njih je moguće primijeniti različite vrijednosti na stilove ovisno o karakteristikama zaslona uređaja kao što su širina, visina i orijentacija. Druge tehnike uključuju fluidne mreže i fleksibilne slike koje koriste relativne jedinice poput postotaka za definiranje dimenzija izgleda (Marcotte, 2010).

Primjer za Media Query:

```
@media (min-width: 768px) {  
  .card_number::before {  
    font-size: 1.875rem; /* 3x1 */  
  }  
}
```

3.4. Postavljanje Next.js projekta s Tailwind CSS-om

Faza implementacije započela je postavljanjem novog Next.js projekta i integracijom Tailwind CSS-a za stiliziranje elemenata.

Koraci za postavljanje:

- **Stvaranje Next.js projekta** – Inicijalizacija novog Next.js projekta pomoću naredbe:
``npx create-next-app``
- **Instalacija Next.js paketa** – Pri prvoj inicijalizaciji Next.js projekta potrebno je upisati ``y`` nakon što program pita za instalaciju `create-next-app@x.x.x` paketa
- **Odabir konfiguracije Next.js projekta** – Nakon instalacije `create-next-app` paketa nude se brojne opcije za konfiguraciju projekta, odabrane su sljedeće:
 - **What is your project named?** `cro-trends`
 - **Would you like to use TypeScript?** `No / Yes`
 - **Would you like to use ESLint?** `No / Yes`
 - **Would you like to use Tailwind CSS?** `No / Yes`
 - **Would you like to use `src/` directory?`** `No / Yes`
 - **Would you like to use App Router?** `No / Yes`
 - **Would you like to customize the default import alias?** `No / Yes`
- **Otvaranje i pokretanje projekta** – Zadnji korak je ulazak u mapu samog projekta i njegovo pokretanje što se postiže sa sljedećim naredbama:
``code .\cro-trends\`` - u slučaju ovog projekta ako se koristi Visual Studio Code
``npm run dev``

3.5. Integracija Supabase baze podataka

Supabase je korišten za upravljanje podacima aplikacije. Integracija je uključivala postavljanje PostgreSQL baze podataka unutar Supabase sučelja. To se postiglo definiranjem strukture tablice koja sadrži sve podatke i dohvaćanjem podataka u stvarnom vremenu pomoću Supabase API-ja (Supabase Inc., 2024).

Koraci za postavljanje (Supabase stranica):

- **Kreiranje novog profila i odlazak na Nadzornu ploču (engl. dashboard)** – Nadzorna ploča se može pronaći na sljedećem linku: <https://supabase.com/dashboard/projects>
- **Kreiranje novog projekta** – Klikom na gumb `New project` prikazuje se forma koja se popunila na sljedeći način:
 - **Organization:** MarkoPvlkvc's Org
 - **Project name:** cro-trends
 - **Database Password:** *****
 - **Region:** Central EU (Frankfurt)
- **Spremanje API ključa i URL-a** – Projektu će biti potrebno nekoliko trenutaka da dovrši sa postavljanjem, nakon toga klikom na `Connect` gumb na `Home` stranici projekta možemo pronaći API ključ i URL pod `App Frameworks` poglavlju

Koraci za postavljanje (Next.js projekt):

- **Dodavanje API ključa i URL-a u projekt** – U root direktoriju projekta kreirana je `.env.local` datoteka koja će sadržati API ključ i URL, u datoteci je to zapisano na sljedeći način:

```
NEXT_PUBLIC_SUPABASE_URL=<url>  
NEXT_PUBLIC_SUPABASE_ANON_KEY=<ključ>
```

- **Instalacija Supabase paketa** – Potrebno je dodati Supabase paket projektu kako bismo mogli dovršiti implementaciju, to se postiže sljedećom naredbom:

```
`npm install @supabase/ssr @supabase/supabase-js`
```

- **Definiranje potrebnih funkcija** – Potrebno je napraviti datoteku `client.ts` unutar `utils\supabase\` koji je napravljen u root direktoriju; unutar nje napisan je sljedeći kod:

```
`import { createBrowserClient } from „@supabase/ssr“;  
  
export const createClient = () =>  
  createBroswerClient(  
    process.env.NEXT_PUBLIC_SUPABASE_URL!,  
    process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!,  
  );`
```

- **Import funkcije u `page.tsx`** – Funkciju je potrebno uključiti u datoteku u kojoj želimo napraviti poziv na bazu, to je moguće sa sljedećom linijom:

```
`import { createClient } from „@/utils/supabase/client“;`
```

- **Kreiranje instance i primjer poziva na bazu** – Kreiramo instancu funkcije koju spremimo u konstantnu varijablu `supabase` i napravimo probni poziv na bazu (glavna funkcija komponente mora biti asinkrona kako bi bilo moguće napraviti `await` poziv prema tablici baze):

```
`const supabase = createClient();`
```

```
const { data } = await supabase.from(„<ime tablice>“).select(„*“)`
```

4. CroTrends Korisničko sučelje i funkcionalnosti

U ovom poglavlju zaronit će se u programski kod sekcija korisničkog sučelja i funkcionalnosti aplikacije. Osim toga, raspravljat će se o mehanizmu pretraživanja, navodeći kako korisnici mogu komunicirati s aplikacijom kako bi dohvatili relevantne podatke. Konačno, obuhvatit ću metode koje se koriste za vizualizaciju podataka.

4.1. Poglavlja korisničkog sučelja

`Home` sekcija služi kao odredišno područje web aplikacije, s istaknutim pozivom na radnju (engl. Call to action), sloganom aplikacije i prikazom izgleda grafikona koji će predstavljati podatke. Kod za sekciju napisan je u nastavku.

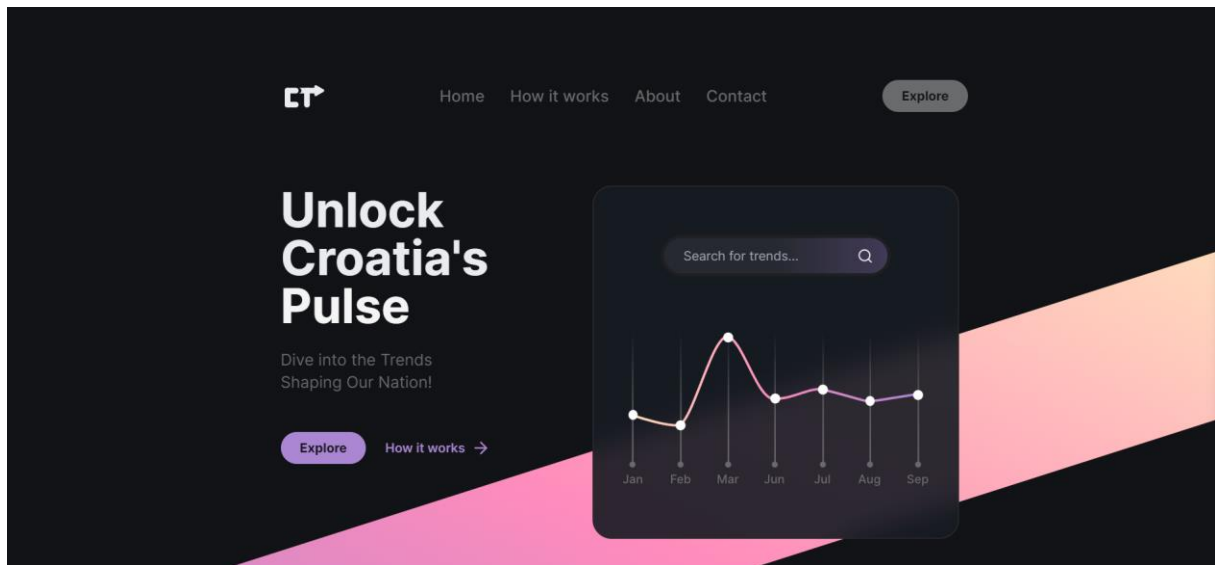
Next.js dio koda za CroTrends `Home` sekciju:

```
<section className="relative h-fit overflow-hidden">
  <div className="flex h-full w-full flex-col justify-center gap-9 px-6 pt-14 md:flex-row
md:gap-16 md:px-16 md:pt-16 lg:gap-32 lg:pt-20">
    <div className="flex w-min flex-col justify-self-end">
      <div className="text-4xl font-bold md:text-5xl lg:text-6xl">
        Unlock Croatia&apos;s{ " "}
        <FlipWords
          words={words}
          className="hidden text-white lg:inline-block"
        />
        <span className="inline-block lg:hidden">Pulse</span>
      </div>
    </div>
    <p className="mt-2 text-base text-gray md:mt-4 md:text-lg lg:mt-6 lg:text-xl">
      Dive into the Trends <br /> Shaping Our Nation!
    </p>
    <div className="mt-8 flex items-center gap-6 md:mt-10 lg:mt-12">
      <Link href="/explore">
        <Button text="Explore" className="bg-purple" />
      </Link>
      <ButtonSecondary text="How it works" href="#howItWorks" />
    </div>
  </div>

  <div className="mt-6 flex h-fit flex-col items-center justify-center rounded-3xl
border-2 border-containerBorder bg-container px-9 py-16 backdrop-blur-lg md:mt-0">
    <Input
      text="Search for trends..."
      className="bg-gradient-to-r from-gray/20 via-gray/20 to-purple/30"
      disabled={true}
    />

    
  </div>
</div>
```

</section>



Slika 2 - `Home` sekcija CroTrends web aplikacije

`How it works` sekcija objašnjava proces od tri koraka koji se koristi da bi se podaci o određenom upisanom ključnom pojmu prikazali korisnicima:

1. **Struganje podataka (engl. Data Scraping)** – Dohvaćanje, spremanje i oblikovanje svih podataka koji su skupljeni sa stranica poput 24sata⁴, n1⁵, itd.
2. **Pretraživanje (engl. Searching)** – Korisnik unosi ključnu riječ ili više riječi za koje želi prikaz u grafikonu, taj unos se zatim pretražuje u bazi podataka
3. **Vizualizacija (engl. Visualizing)** – Ako postoje podaci za traženi pojam oni se vizualiziraju i korisnik ih može proučavati

Kod za sekciju napisan je u nastavku.

Next.js dio koda za CroTrends `How it works` sekciju:

```
<section
  id="howItWorks"
  className="relative flex flex-col items-center overflow-hidden"
>
  <div className="mt-24 px-6 md:mt-32 lg:mt-40">
    <p className="text-center text-2xl font-bold md:text-3xl lg:text-4xl">
      How it works
    </p>
    <div className="mt-9 flex flex-wrap justify-center gap-x-16 gap-y-9 px-6 md:mt-12
lg:mt-16">
      <div className="card_number relative flex w-[250px] flex-col items-center rounded-3xl
border-[2px] border-white/10 bg-container px-6 py-9 md:px-9">
        <FileText className="mt-6 size-14 stroke-[0.5] text-purple md:size-16" />
        <p className="mt-10 text-nowrap text-xl font-semibold md:text-2xl">
          Data Scraping
```

⁴ <https://www.24sata.hr/>

⁵ <https://n1info.hr/>

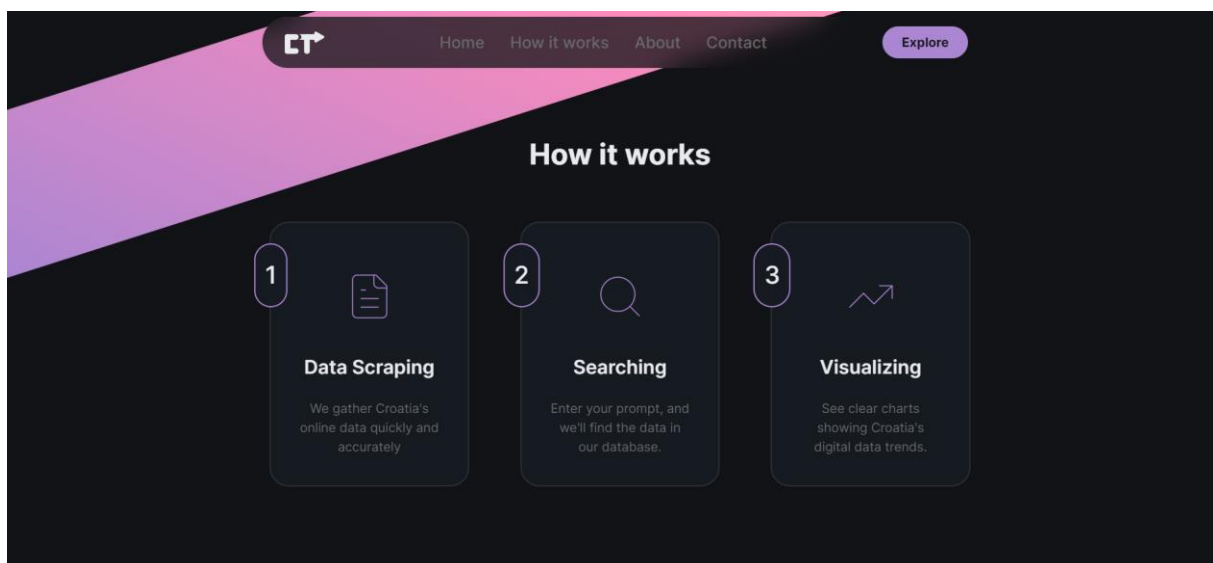
```

</p>
<p className="mt-6 max-w-44 text-center text-xs text-gray md:text-base">
  We gather Croatia&apos;s online data quickly and accurately
</p>
</div>

<div className="card_number relative flex w-[250px] flex-col items-center rounded-3xl
border-[2px] border-white/10 bg-container px-6 py-9 before:content-['2'] md:p-9 md:px-9">
  <Search className="mt-6 size-14 stroke-[0.5] text-purple md:size-16" />
  <p className="mt-10 text-nowrap text-xl font-semibold md:text-2xl">
    Searching
  </p>
  <p className="mt-6 max-w-44 text-center text-xs text-gray md:text-base">
    Enter your prompt, and we&apos;ll find the data in our database.
  </p>
</div>

<div className="card_number relative flex w-[250px] flex-col items-center rounded-3xl
border-[2px] border-white/10 bg-container px-6 py-9 before:content-['3'] md:p-9 md:px-9">
  <TrendingUp className="mt-6 size-14 stroke-[0.5] text-purple md:size-16" />
  <p className="mt-10 text-nowrap text-xl font-semibold md:text-2xl">
    Visualizing
  </p>
  <p className="mt-6 max-w-44 text-center text-xs text-gray md:text-base">
    See clear charts showing Croatia&apos;s digital data trends.
  </p>
</div>
</div>
</div>
</section>

```

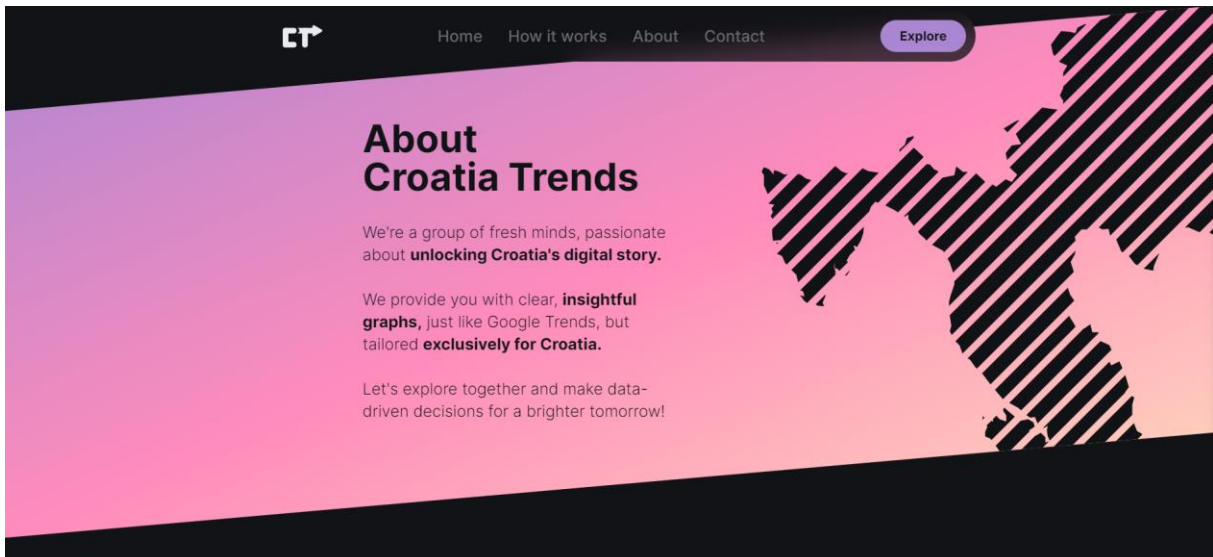


Slika 3 - 'How it works' sekcija CroTrends web aplikacije

'About' sekcija pruža pozadinske informacije o timu koji stoji iza projekta. Naglašava prilagođenu prirodu analize podataka, što je čini specifičnom za hrvatske trendove. Kod za sekciju napisan je u nastavku.

Next.js dio koda za CroTrends `About` sekciju:

```
<section
  id="about"
  className="relative mt-24 flex w-full flex-col items-center justify-center overflow-
  hidden bg-gradient-to-br from-purple via-pink to-yellow py-24 text-black clip-about md:mt-
  32 md:py-32 md:clip-about-md lg:mt-40 lg:py-36 lg:clip-about-lg"
>
  <div className="flex w-full max-w-screen-md justify-center px-6 sm:justify-start sm:px-
  16">
    <div className="relative flex max-w-96 flex-col">
      <p className="text-3xl font-bold md:text-4xl lg:text-5xl">
        About <br /> Croatia Trends
      </p>
      <p className="mt-8 text-base font-light md:text-lg lg:text-xl">
        We&apos;re a group of fresh minds, passionate about{" "}
        <span className="font-bold">
          unlocking Croatia&apos;s digital story. <br></br>
        <br></br>
        </span>{" "}
        We provide you with clear,{" "}
        <span className="font-bold">insightful graphs,</span> just like
        Google Trends, but tailored{" "}
        <span className="font-bold">exclusively for Croatia.</span>{" "}
        <br></br>
        <br></br>
        Let&apos;s explore together and make data-driven
        decisions for a brighter tomorrow!
      </p>
      
    </div>
  </div>
</section>
```



Slika 4 - `About` sekcija CroTrends web aplikacije

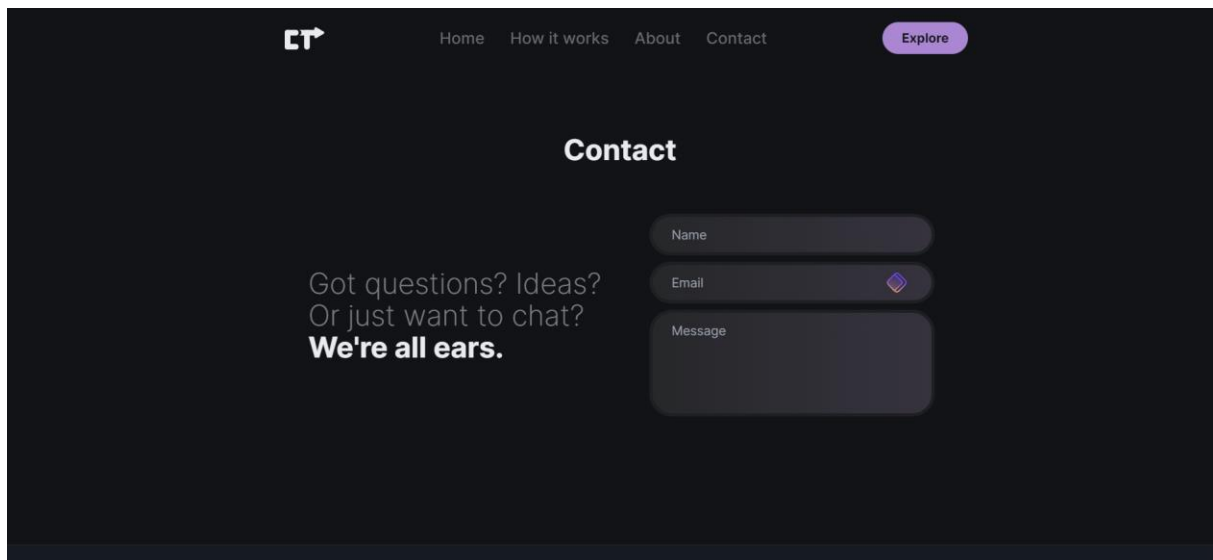
`Contact` sekcija uključuje obrazac za korisnike koji korisnici mogu koristiti kao način uspostave sa timom i postavljanje pitanja. Mogu također dobiti povratne informacije, stupiti u kontakt ako imaju ideje za suradnju, ili ako samo žele razgovarati sa timom. Osmišljen je kako bi olakšao komunikaciju i angažman s korisnicima. Kod za sekciju napisan je u nastavku.

Next.js dio koda za CroTrends `Contact` sekciju:

```
<section id="contact" className="relative w-full overflow-hidden pb-1">
  <p className="mt-24 text-center text-2xl font-bold md:mt-32 md:text-3xl lg:mt-40
lg:text-4xl">
    Contact
  </p>

  <div className="mt-9 flex flex-col items-center justify-center gap-16 px-6 md:mt-12
md:flex-row lg:mt-16">
    <p className="max-w-sm text-2xl font-extralight text-white/50 md:text-3xl lg:text-4xl">
      Got questions? Ideas? <br /> Or just want to chat? <br />
      <span className="font-bold text-white">We&apos;re all ears.</span>
    </p>

    <div className="flex w-full max-w-[350px] flex-col gap-4">
      <input
        hasIcon={false}
        text="Name"
        className="w-full bg-gradient-to-r from-[#2a2b2d]/85 to-[#36333e]"
      />
      <input
        hasIcon={false}
        text="Email"
        className="w-full bg-gradient-to-r from-[#2a2b2d]/85 to-[#36333e]"
      />
      <textarea
        placeholder="Message"
        style={{ resize: "none" }}
        className="h-[125px] w-full rounded-3xl bg-transparent bg-gradient-to-r from-
[#2a2b2d]/85 to-[#36333e] px-6 py-2.5 text-white ring-4 ring-containerElevated transition-
all focus-within:ring-2 focus-within:ring-purple focus:outline-none"
      />
    </div>
  </div>
</section>
```



Slika 5 - `Contact` sekcija CroTrends web aplikacije

`Explore` sekcija sadrži mjesto za unos ključnih riječi od strane korisnika zajedno sa dijelovima gdje će se prikazati vizualizirani i svi ostali dostupni podatci u slučaju da je odabrana ključna riječ prisutna u bazi podataka. Kod za sekciju napisan je u nastavku.

Next.js dio koda za CroTrends `Explore` sekciju:

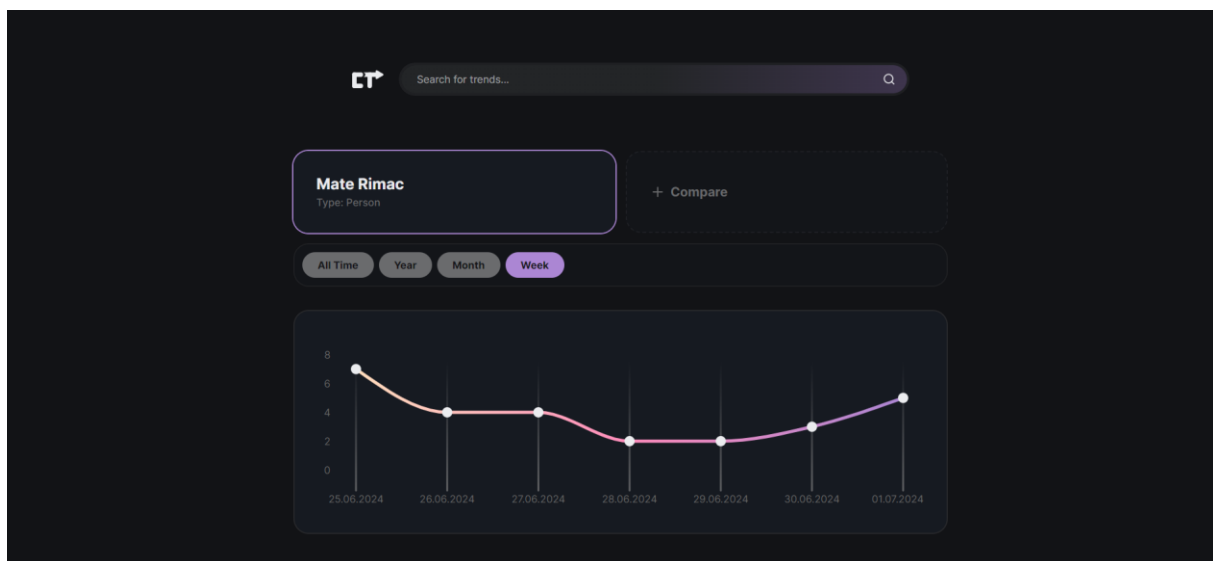
```
<section className="flex w-full flex-col items-center px-6">
  <div
    className={` ${activeLines > 1 ? "sm:grid-cols-3" : "sm:grid-cols-2"} mt-9 grid w-full
max-w-screen-lg grid-cols-1 gap-4 md:mt-12 lg:mt-20` }
  >
    {searchTerms.slice(0, activeLines).map((search_term, index) => (
      <SearchTermButton
        key={search_term}
        search_term={search_term}
        colorIndex={index}
        disabled={loading}
        isFocused={focusedLine[index]}
        isCompareActive={activeLines > 1}
        onClick={() => handleFocusedLineButtonClick(index)}
        onRemoveClick={() => deleteTerm("search_term" + (index + 1))}
      />
    ))}

    {activeLines < 3 && (
      <button
        onClick={() => handleCompareButtonClick()}
        disabled={loading}
        className={` ${focusedLine.indexOf(true) == activeLines ? "border-purple text-
purple" : "border-white/5 text-gray hover:text-[hsl(269,7%,68%)] enabled:hover:border-
[hsl(269,7%,68%)]"} group flex w-full flex-col justify-center rounded-3xl border-2 border-
dashed bg-container/25 p-5 text-start transition-all md:p-7 lg:p-9` }
      >
        <p className="flex items-center font-bold group-disabled:text-gray/50 md:text-lg
lg:text-xl">
          <Plus className="mr-2" />
          Compare
        </p>
      </button>
    )}
  </div>
</section>
```

```

    </button>
  )}
  <div
    className={` ${activeLines} > 1 ? "sm:col-span-3" : "sm:col-span-2"} flex w-full flex-
wrap justify-start gap-2 rounded-3xl border-2 border-containerElevated bg-container/25 p-3
font-bold`}
  >
    {Object.keys(activeTimeInterval).map((interval) => (
      <Button
        key={interval}
        text={interval}
        className={` ${activeTimeInterval[interval as TimeInterval]} ? "bg-purple
disabled:bg-purple/50" : "enabled: hover: bg-[hsl(269,7%,68%)] disabled: bg-gray/50"} w-fit
disabled: transition-colors`}
        onClick={() => handleTimeIntervalButtonClick(interval)}
        disabled={loading}
      />
    ))}
  </div>
  <div className="mt-9 flex h-[350px] w-full max-w-screen-lg items-center justify-center
rounded-3xl border-2 border-containerBorder bg-container p-5 md:p-7 lg:p-9">
    <Graph
      terms={terms}
      loading={loading}
      error={error}
      activeLines={activeLines}
    />
  </div>
</section>

```

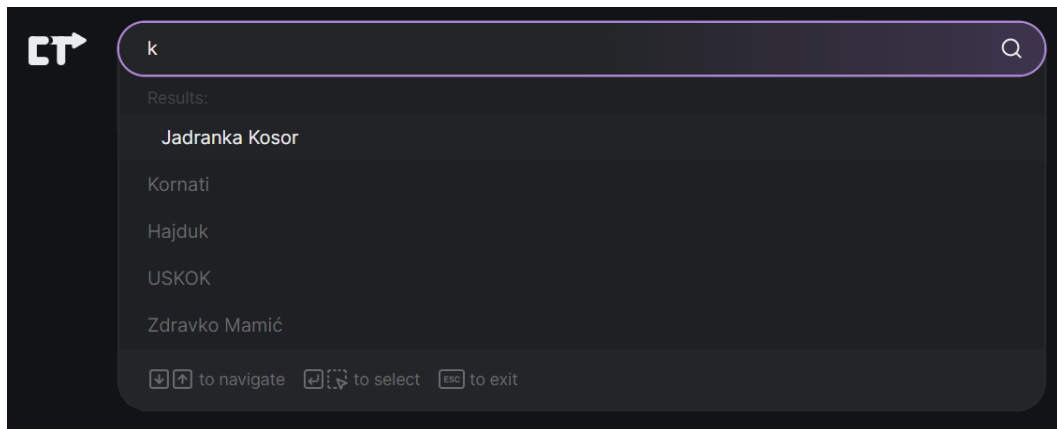


Slika 6 - `Explore` sekcija CroTrends web aplikacije

4.2. Mehanizam pretraživanja

Mehanizam pretraživanja korisniku nudi niz opcija za samo popunjavanje pri unosu dijela ključne riječi, prikazanih u listi kao dodatak ispod prostora za pretraživanje. Kako se od korisnika ne očekuje da zna kojim ključnim riječima web aplikacija raspolaže, koristi se ovaj pristup. Zatim se u odnosu na korisnikov odabir iz liste trenutnih ključnih riječi radi poziv prema bazi.

Kako bi se detaljnijim opisivanjem ključnih riječi i načinom dolaska do njih previše izlazilo izvan aspekta ovog projekta, biti će spomenuto samo kako su ključne riječi spremljene u bazi web aplikacije i da je riječ o entitetima spomenutim u sastruganom sadržaju. Sadržaj same baze je rezultat projekta od strane druge osobe.



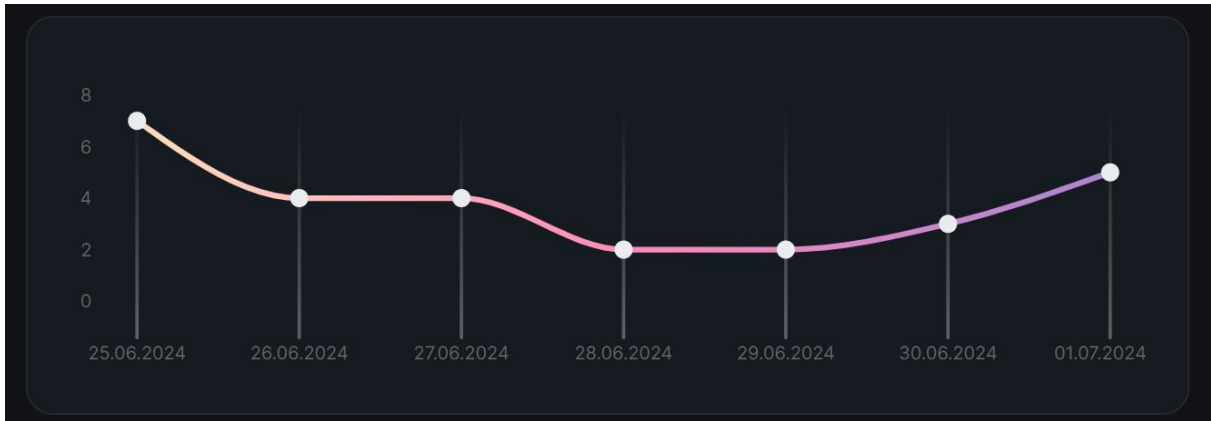
Slika 7 - Mehanizam pretraživanja CroTrends web aplikacije

Osnovna logika mehanizma pretraživanja:

```
useEffect(() => {
  if (userInput) {
    const unLinked = suggestions
      .filter(
        (suggestion) =>
          suggestion.toLowerCase().indexOf(userInput.toLowerCase()) > -1,
      )
      .slice(0, 5);
    setFilteredSuggestions(unLinked);
    setActiveSuggestionIndex(0);
  } else {
    setFilteredSuggestions([]);
  }
}, [userInput, suggestions]);
```

4.3. Dohvaćanje i vizualizacija podataka

Kako bi pozivi prema bazi bili što optimiziraniji, odabran je način rukovanja sa podacima gdje baza napravi `select` nad elementima sa uvjetima da bi podaci bili upotrebljivi za vizualizaciju i onda samo filtriranje podatke šalje klijentskoj strani. Manje optimalan način za prijenos podataka u ovom slučaju bio bi povrat svih vrijednosti iz baze nakon poziva nad njom i filtriranje podataka lokalno.



Slika 8 – Primjer vizualizacije podataka CroTrends web aplikacije za pojam „Mate Rimac“

Dohvaćanje samih podataka izvršava se sa `supabase.rpc()` funkcijom koja poziva predefiniranu funkciju na strani Supabase baze. Funkcija `get_term_count_last_time_interval` vidljiva u Prilogu 1 je osnovna funkcija koja za odabrani ključni pojam i vremenski interval vraća podatke u sljedećem JSON formatu sa ključevima `period` i `article_count`:

Primjer vraćenih podataka funkcije `get_term_count_last_time_interval` za pojam „Mate Rimac“:

```
[{"period": "2024-06-25", "article_count": 7}, {"period": "2024-06-26", "article_count": 4}, {"period": "2024-06-27", "article_count": 4}, {"period": "2024-06-28", "article_count": 2}, {"period": "2024-06-29", "article_count": 2}, {"period": "2024-06-30", "article_count": 3}, {"period": "2024-07-01", "article_count": 5}]
```

Funkcija `get_combined_term_count_last_time_interval` vidljiva u Prilogu 2 je ona funkcija koja se poziva unutar `supabase.rpc()` funkcije. Razlog zašto je spomenuta funkcija iz Priloga 1 je zato jer se ona poziva 3 puta u funkciji iz Priloga 2. Na taj način kao povratni sadržaj poziva dobijemo JSON koji ima kombinirane vrijednosti za 3 pojma zbog mogućnosti usporedbe ključnih riječi koje se nude korisniku aplikacije. JSON podaci vraćeni su u formatu sa ključevima `period`, `Series A`, `Series B` i `Series C`.

Primjer vraćenih podataka funkcije `get_combined_term_count_last_time_interval` za pojam „Mate Rimac“:

```
[{"period": "2024-06-25", "Series A": 7, "Series B": 0, "Series C": 0}, {"period": "2024-06-26", "Series A": 4, "Series B": 0, "Series C": 0}, {"period": "2024-06-27", "Series A": 4, "Series B": 0, "Series C": 0}, {"period": "2024-06-28", "Series A": 2, "Series B": 0, "Series C": 0}, {"period": "2024-06-29", "Series A": 2, "Series B": 0, "Series C": 0}, {"period": "2024-06-30", "Series A": 3, "Series B": 0, "Series C": 0}, {"period": "2024-07-01", "Series A": 5, "Series B": 0, "Series C": 0}]
```

U gore prikazanom primjeru ključevi `Series B` i `Series C` ne sadrže nikakve vrijednosti. To je zato jer je pozivanje funkcije izvršeno samo sa jednom ključnom riječi. Da je funkcija pozvana sa dvije ili tri ključevi B i C bi imali vrijednosti.

Next.js supabase.rpc funkcija:

```
useEffect(() => {
  async function fetchData() {
    setLoading(true);
    const term1 = params.get("search_term1");
    const term2 = params.get("search_term2");
    const term3 = params.get("search_term3");

    const { data, error } = await supabase.rpc(
      "get_combined_term_count_last_time_interval",
      {
        search_term1: term1,
        search_term2: term2,
        search_term3: term3,
        time_interval: params.get("time_interval"),
      },
    );
    if (error) {
      setError(error);
    } else {
      setTerms(data);
    }
    setLoading(false);
  }

  fetchData();

  setActiveLines(nonNullTermsCount);

  if (!isAddingNew.current) {
    setFocusedLine(
      focusedLine.map((_, index) => index === nonNullTermsCount - 1),
    );
  }
}, [params]);
```

Dohvaćeni podaci vizualiziraju se pomoću linearnog grafikona, pružajući korisnicima jasan pregled učestalosti ključnih riječi tijekom vremena. Aplikacija nudi mogućnost uspoređivanja do 3 ključna pojma odjednom na jednom grafikonu. Pri tome moguće je u svakom trenu mijenjati vremenski interval grafikona. Nudi se 4 vremenska intervala: „All Time“, „Year“, „Month“ i „Week“. Način na koji vremenski intervali rade je da se prikazuju podaci iz zadnjih X dana ovisno o odabranoj opciji. Vrijednosti su za svaki interval grupirane na drugi način. U slučaju „All Time“ opcije biti će prisutna jedna točka na grafikonu za godinu dana. Kod „Year“ opcije biti će prisutna jedna točka za mjesec dana. „Month“ opcija ima podatke grupirane po tjednima u mjesecu i „Week“ opcija po danima u tjednu.

Next.js Graph komponenta:

```

<ResponsiveContainer className="h-full w-full">
  <LineChart
    width={500}
    height={300}
    data={terms}
    margin={{ top: 0, right: 0, left: -32, bottom: 0 }}
  >
    <defs>
      <linearGradient
        id="verticalLines"
        x1="0%"
        y1="0%"
        x2="0%"
        y2="100%"
        gradientUnits="userSpaceOnUse"
      >
        <stop offset="0%" stopColor="rgba(106, 107, 109, 0)" />
        <stop offset="15%" stopColor="rgba(106, 107, 109, 0)" />
        <stop offset="100%" stopColor="rgba(106, 107, 109, 1)" />
      </linearGradient>
      <linearGradient id="lineGradient" x1="0%" y1="0" x2="100%" y2="0">
        <stop offset="0%" stopColor="rgba(255,222,189,1)" />
        <stop offset="50%" stopColor="rgba(255,140,188,1)" />
        <stop offset="100%" stopColor="rgba(171,134,211,1)" />
      </linearGradient>
    </defs>
    <CartesianGrid
      strokeDasharray="0"
      stroke="url(#verticalLines)"
      strokeWidth={3}
      strokeLinecap="round"
      horizontal={false}
    />
    <XAxis
      dataKey="period"
      axisLine={false}
      tickLine={false}
      tickFormatter={dateFormatter}
      padding={{ left: 32, right: 32 }}
    />
    <YAxis
      axisLine={false}
      tickLine={false}
      padding={{ top: 32, bottom: 32 }}
    />
    <Tooltip
      contentStyle={{

```

```

        backgroundColor: "#1f2023",
        border: "solid 2px rgba(235, 236, 239, 0.1)",
        borderRadius: "24px",
        padding: "16px",
    }}
    labelFormatter={dateFormatter}
    animationEasing="ease-out"
  />
  <Line
    connectNulls
    type="monotone"
    dataKey="Series A"
    stroke={ `${activeLines > 1 ? "#FFDEBD" : "url(#lineGradient)"} ` }
    strokeWidth={lineStrokeWidth}
    fill="#EBECEF"
    dot={{ r: dotRadius, strokeWidth: 0 }}
    activeDot={{
      fill: "#1f2023",
      stroke: "#EBECEF",
      r: dotRadius,
      strokeWidth: activeDotStrokeWidth,
    }}
  />

  {activeLines > 1 && (
    <Line
      connectNulls
      type="monotone"
      dataKey="Series B"
      stroke="#FF8CBC"
      strokeWidth={lineStrokeWidth}
      fill="#EBECEF"
      dot={{ r: dotRadius, strokeWidth: 0 }}
      activeDot={{
        fill: "#1f2023",
        stroke: "#EBECEF",
        r: dotRadius,
        strokeWidth: activeDotStrokeWidth,
      }}
    />
  )}

  {activeLines > 2 && (
    <Line
      connectNulls
      type="monotone"
      dataKey="Series C"
      stroke="#AB86D3"
      strokeWidth={lineStrokeWidth}
      fill="#EBECEF"
      dot={{ r: dotRadius, strokeWidth: 0 }}
      activeDot={{
        fill: "#1f2023",
        stroke: "#EBECEF",
        r: dotRadius,
        strokeWidth: activeDotStrokeWidth,
      }}
    />
  )}
</LineChart>
</ResponsiveContainer>

```


5. Internetsko udomljavanje CroTrends aplikacije na Vercelu

Hosting web stranice kritičan je korak u njezinoj dostupnosti korisnicima na internetu. Vercel je popularna platforma za *hosting* modernih web aplikacija, nudi vrhunsku integraciju s GitHubom i pruža robusne značajke za implementaciju i skaliranje (Vercel, 2024).

5.1. GitHub integracija

Prije implementacije web stranice na Vercelu, kod projekta mora biti smješten na platformi za kontrolu verzija (engl. version control) kao što je GitHub. GitHub je web platforma koja se koristi za kontrolu verzija i kolaborativni razvoj. Programerima omogućuje lako upravljanje sa više projekata, praćenje promjena i suradnju sa drugim programerima. U sljedećim natuknicama biti će objašnjen postupak kreiranja novog repozitorija i prijenos projekta na njega (HTG Staff, 2016).

Koraci za prijenos (GitHub stranica):

- **Kreiranje novog profila i odlazak na prikaz svih repozitorija** – Prikaz svih repozitorija u slučaju profila `MarkoPv1kvc` može se pronaći na sljedećem linku: <https://github.com/MarkoPv1kvc?tab=repositories>
- **Kreiranje novog repozitorija** – klikom na gumb `New` prikazuje se forma koja se ispunila na sljedeći način:
 - **Repository name:** cro-trends
 - **Visibility:** Public
 - Ostale mogućnosti nisu dirane, nema deskripcije kako se ona lako doda kasnije i nije nam potreban README file kako je on automatski generiran u Next.js projektu
- **Komande za prijenos projekta** – nakon što je kreiran novi repozitorij korisnik je preusmjeren na stranicu istoga, gdje su vidljive komande za prijenos projekta na GitHub, te komande je potrebno zapaziti jer su korištene u sljedećem koraku

Koraci za prijenos (Next.js projekt):

- **Git Init** – nakon inicijalizacije Next.js projekta on automatski inicijalizira lokalni git repozitorij pa nam komanda ``git init`` neće biti potrebna, ako kojim slučajem to nije slučaj slobodno se iskoristi komanda (za vrijeme ove i sljedećih komanda locirani smo u mapi od projekta u terminalu)
- **Prvi commit** – prvo je potrebno pozvati komandu:
``git add .``
kako bi se osiguralo da sve datoteke uđu u commit (commit-evi su lokalni „save point-ovi“ koji nam omogućavaju da se vratimo na to stanje projekta kada god želimo), kako bismo napravili prvi commit potrebno je koristiti sljedeću naredbu:
``git commit -m „<komentar>“``
- **Spajanje lokalnog repozitorija sa GitHub repozitorijem** – da bi lokalni repozitorij znao kamo slati datoteke potrebno je dodati link na repozitorij na webu, to se postiže komandom:
``git remote add origin https://github.com/MarkoPvlkvc/cro-trends.git``
(link za repozitorij ``cro-trends`` korisnika ``MarkoPvlkvc``)
- **Prijenos projekta na GitHubu** – Nakon prvog commita možemo isti taj commit prenijeti na GitHub komandom:
``git push origin master``

5.2. Implementacija na Vercelu

Vercel pojednostavljuje proces postavljanja web aplikacija izravnom integracijom s GitHub repozitorijima. Nakon što se projekt nađe na GitHubu, može se uvesti u Vercel za implementaciju.

Koraci za implementaciju:

- **Prijava sa GitHubom i odlazak na prikaz svih projekata** – svi projekti korisnika MarkoPvlkvc vidljivi su na linku:
<https://vercel.com/markopvlkvcs-projects>
- **Kreiranje novog projekta** – klikom na gumb ``Add New`` izabire se opcija ``Project``, korisnik je preusmjeren na drugu stranicu gdje stiče gumb ``Import`` pored imena repozitorija sa GitHuba
- **Dovršavanje implementacije** – korisnik je preusmjeren na stranicu sa formom, sva mjesta za unos bi se trebala sama popuniti, ime poprima po GitHub repozitoriju, okvir prepoznaje samostalno (ako je projekt dobro postavljen), jedino što je potrebno dodati su varijable okoline (engl. environment variables) u slučaju korištenja nekih API ključeva (Supabase u našem slučaju); klikom na gumb ``Deploy`` Vercel će pokušati hostati stranicu (postoji mogućnost da će biti potrebno ispraviti neke greške u projektu)

6. Rezultati i rasprava o CroTrends aplikaciji

6.1. Procjena učinkovitosti

Učinkovitost web aplikacije procijenjena je na temelju vremena odziva i zadovoljstva probnih korisnika. Aplikacija je uspješno rukovala sa podacima kojima je imala pristup i pružila vrijedne i jednostavne uvide na vizualizaciju istih.

Tablica 3 - Procjena učinkovitosti CroTrends web aplikacije

| <i>Metrika</i> | <i>Rezultat</i> |
|---------------------------------|----------------------|
| <i>Prosječno vrijeme odziva</i> | 368ms |
| <i>Zadovoljstvo korisnika</i> | 4.8/5 sa 5 korisnika |

6.2. Povratne informacije korisnika

Povratne informacije korisnika prikupljene su putem izravnih interakcija. Općenito, korisnici su cijenili moderno i fluidno sučelje i količinu informacija koju pruža aplikacija. Prijedlozi za poboljšanja uključivali su dodatne izvore podataka i preciznije vremenske okvire.

Tablica 4 - Povratne informacije korisnika CroTrends web aplikacije

| <i>Segment povratne informacije</i> | <i>Pozitivne povratne informacije (% / broj korisnika)</i> | <i>Prijedlozi za poboljšanje (% / broj korisnika)</i> |
|-------------------------------------|--|---|
| <i>Korisničko sučelje</i> | 80% / 4 | 20% / 1 |
| <i>Točnost podataka</i> | 100% / 5 | 0% / 0 |
| <i>Kompletnost značajki</i> | 80% / 4 | 20% / 1 |

6.3. Usporedba sa drugim alatima za analizu trendova

Ovo poglavlje uspoređuje razvijenu web aplikaciju s drugim popularnim alatima za analizu trendova. Jedan od takvih najpopularnijih alata je Google Trends, pošto nema drugog alata koji je na razini poput njega, ova usporedba je bazirana na tom alatu. Google Trends je

široko korišten alat koji pruža uvid u popularnost ključnih riječi u pretraživanju tijekom vremena. Nudi niz značajki, uključujući geografske podatke i povezane upite (Google, 2024).

Tablica 2 - Usporedba Google Trends i CroTrends

| <i>Značajka</i> | <i>Google Trends</i> | <i>CroTrends</i> |
|--------------------------------------|----------------------|----------------------------|
| <i>Vremenski intervali</i> | Da | Da |
| <i>Geografski podaci</i> | Da | Ne (fokusrano na Hrvatsku) |
| <i>Unos ključnih riječi</i> | Da | Da |
| <i>Ažuriranje u stvarnom vremenu</i> | Ne | Da |
| <i>Prilagođeni izvori podataka</i> | Ne | Da |

Iako Google Trends ima pristup većoj bazi podataka i geografskim podacima cijelog svijeta, prednost u CroTrends-u je ta da su prikupljeni i obrađeni podatci upravo oni koji se u određenom vremenskom intervalu spominju na web stranicama. Na podatke Google Trends platforme utječu korisnički upiti sa njihove platforme i tako mogu promaknuti neke stvari koje su doista bitne zbog korisničke fokusiranosti na druge.

7. Zaključak

U radu je prikazan razvoj interaktivne web aplikacije za vremensku analizu *online* medijskih trendova. Koristeći Next.js, Tailwind CSS i Supabase tehnološki skup, aplikacija korisnicima nudi moćan alat za istraživanje i razumijevanje digitalnih trendova za područje Hrvatske.

Testiranjem, procjenom učinkovitosti i povratnim informacijama korisnika dolazi se do zaključka da je web aplikacija uspješno implementirana. Alat se može koristiti za pregled stanja u medijima Hrvatske, te postoji velika mogućnost da će nakon marketinga inspirirati druge lokalne programere da započnu izrađivanje web aplikacija ili da drugi veći programeri krenu uključivati potporu za naše područje.

Buduća poboljšanja mogla bi uključivati proširenje izvora podataka na više regija i jezika, implementaciju naprednih opcija filtriranja, više vremenskih intervala i integraciju algoritama strojnog učenja za prediktivnu analizu.

8. Literatura

- (2018). U J. N. Robbins, *Learning Web Design*. O'REILLY.
- Craig, W. (n.d.). *The Evolution of Web Design*. Dohvaćeno iz WebFX: <https://www.webfx.com/blog/web-design/the-evolution-of-web-design/>
- GeeksForGeeks. (24. 6 2023). *Top 10 Backend Technologies You Must Know*. Dohvaćeno iz GeeksForGeeks: <https://www.geeksforgeeks.org/backend-technologies/>
- Google. (2024). Dohvaćeno iz Google Trends: <https://trends.google.com/trends/>
- Marcotte, E. (25. 5 2010). *Responsive Web Design - A List Apart*. Dohvaćeno iz A List Apart: <https://alistapart.com/article/responsive-web-design/>
- Ostrowska, K. (6. 4 2024). *The Most Popular Databases in 2024*. Dohvaćeno iz LearnSQL: <https://learnsql.com/blog/most-popular-databases-2024/>
- Prasad, T. (4. 4 2024). *The Ultimate Guid to Full-Stack Development: Technologies, Frameworks, and Stacks Explained*. Dohvaćeno iz Kwantify: <https://www.kwantify.com/the-ultimate-guide-to-full-stack-development-technologies-frameworks-and-stacks-explained/>
- Supabase Inc. (22. 6 2024). *Supabase Docs*. Dohvaćeno iz Supabase: <https://supabase.com/docs>
- Tailwind Labs Inc. (22. 6 2024). *Installation*. Dohvaćeno iz Tailwindcss: <https://tailwindcss.com/docs/installation>
- Tiwari, A. (19. 8 2023). *The History and Evolution of Web Development: From HTML to the Modern Web*. Dohvaćeno iz Medium: <https://medium.com/@Abhishek-Tiwari/the-history-and-evolution-of-web-development-from-html-to-the-modern-web-982e3f90e891>
- Vercel. (22. 6 2024). *Docs*. Dohvaćeno iz Next.js: <https://nextjs.org/docs>

9. Popis tablica

Tablica 1 - Planovi Supabase usluge

Tablica 2 - Usporedba Google Trends i CroTrends

Tablica 3 - Procjena učinkovitosti CroTrends web aplikacije

Tablica 4 - Povratne informacije korisnika CroTrends web aplikacije

10. Popis slika

Slika 1 - Wireframe i Mockup CroTrends web aplikacije

Slika 2 - `Home` sekcija CroTrends web aplikacije

Slika 3 - `How it works` sekcija CroTrends web aplikacije

Slika 4 - `About` sekcija CroTrends web aplikacije

Slika 5 - `Contact` sekcija CroTrends web aplikacije

Slika 6 - `Explore` sekcija CroTrends web aplikacije

Slika 7 - Mehanizam pretraživanja CroTrends web aplikacije

Slika 8 – Primjer vizualizacije podataka CroTrends web aplikacije za pojam „Mate Rimac“

11. Popis priloga

Prilog 1: Supabase funkcija `get_term_count_last_time_interval`

Prilog 2: Supabase funkcija `get_combined_term_count_last_time_interval`

12. Prilog 1: Supabase funkcija `get_term_count_last_time_interval`

```
DROP FUNCTION IF EXISTS get_term_count_last_time_interval;

CREATE OR REPLACE FUNCTION get_term_count_last_time_interval(search_term TEXT,
time_interval TEXT)
RETURNS TABLE (
  period DATE,
  article_count BIGINT
) AS $$
BEGIN
  RETURN QUERY
  SELECT
    CASE
      WHEN time_interval = 'All Time' THEN date_trunc('year', t.publication_date)::date
      WHEN time_interval = 'Week' THEN t.publication_date
      WHEN time_interval = 'Month' THEN date_trunc('week', t.publication_date)::date
      WHEN time_interval = 'Year' THEN date_trunc('month', t.publication_date)::date
    END AS period,
    COUNT(t.article_id) AS article_count
  FROM
    "Terms" t
  WHERE
    t.term = search_term
    AND (
      time_interval = 'All Time' OR
      t.publication_date >= (
        SELECT max(t2.publication_date) -
          CASE
            WHEN time_interval = 'Week' THEN INTERVAL '7 days' - INTERVAL '1 day'
            WHEN time_interval = 'Month' THEN INTERVAL '1 month' - INTERVAL '2 week'
            WHEN time_interval = 'Year' THEN INTERVAL '1 year' - INTERVAL '2 month'
          END
        FROM "Terms" t2
      )
    )
  GROUP BY
    period
  ORDER BY
    period ASC;
END;
$$ LANGUAGE plpgsql;
```

13. Prilog 2: Supabase funkcija

`get_combined_term_count_last_time_interval`

```
DROP FUNCTION IF EXISTS get_combined_term_count_last_time_interval;

CREATE OR REPLACE FUNCTION get_combined_term_count_last_time_interval(
    search_term1 TEXT,
    search_term2 TEXT DEFAULT NULL,
    search_term3 TEXT DEFAULT NULL,
    time_interval TEXT DEFAULT 'Week'
)
RETURNS TABLE (
    period DATE,
    "Series A" BIGINT,
    "Series B" BIGINT,
    "Series C" BIGINT
) AS $$
BEGIN
    RETURN QUERY
    WITH series_a AS (
        SELECT a.period, a.article_count AS "Series A"
        FROM get_term_count_last_time_interval(search_term1, time_interval) as a
        WHERE search_term1 IS NOT NULL
    ),
    series_b AS (
        SELECT b.period, b.article_count AS "Series B"
        FROM get_term_count_last_time_interval(search_term2, time_interval) as b
        WHERE search_term2 IS NOT NULL
    ),
    series_c AS (
        SELECT c.period, c.article_count AS "Series C"
        FROM get_term_count_last_time_interval(search_term3, time_interval) as c
        WHERE search_term3 IS NOT NULL
    )
    SELECT
        COALESCE(series_a.period, series_b.period, series_c.period) AS period,
        COALESCE(series_a."Series A", 0) AS "Series A",
        COALESCE(series_b."Series B", 0) AS "Series B",
        COALESCE(series_c."Series C", 0) AS "Series C"
    FROM
        series_a
        FULL OUTER JOIN series_b ON series_a.period = series_b.period
        FULL OUTER JOIN series_c ON COALESCE(series_a.period, series_b.period) =
series_c.period
    ORDER BY
        period ASC;
END;
$$ LANGUAGE plpgsql;
```