



ŠOLSKI CENTER CELJE

Srednja šola za kemijo, elektrotehniko in računalništvo

# CONVIBE

Raziskovalna naloga

## **Avtorji:**

Devan Bajrić, Miha Kerznar, Timotej Polenek, R-4.b

## **Mentor:**

Boštjan Lubej, dipl. ing. inf. in tehn. kom.

Celje, april 2025

### IZJAVA\*

Mentor Boštjan Lubej v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom ConVibe, katere avtorji so Devan Bajrić, Miha Kerznar, Timotej Polenek:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 5.4.2025



Podpis mentorja

Podpis odgovorne osebe

\*

### POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.

## **Zahvala**

Iskrena hvala mentorju, ki nas je s svojim znanjem, potrpežljivostjo in neomajno podporo vodil skozi celoten proces nastajanja naše raziskovalne naloge. Njegovi nasveti, spodbude in predanost so bili neprecenljivi in so nam omogočili, da smo nalogo izpeljali na takšni ravni.

Zahvaljujemo se tudi lektorici, ki je z natančnim očesom in skrbnim delom poskrbela, da je bila naša naloga jezikovno dovršena in brezhibno zapisana. Njena pomoč je bistveno pripomogla k boljši kakovosti končnega izdelka.

Vsem zgoraj omenjenim posameznikom se še enkrat zahvaljujemo, saj brez Vas raziskovalna naloga ne bi uspela.

## **POVZETEK**

Raziskovalna naloga predstavlja idejno zasnovo, uporabljena ogrodja in orodja ter končne rezultate naše aplikacije za komunikacijo med uporabniki, ConVibe.

Raziskovalna naloga se osredotoča na uporabljene postopke in principe, ki so potrebni za doseg optimalnega delovanja aplikacije ter zagotavljanje varne in hitre izmenjave sporočil. Na koncu je aplikacija, predstavljena v tej raziskovalni nalogi, primerjana z drugimi priljubljenimi rešitvami na trgu, kot so Discord in Snapchat, pri čemer so ovrednoteni predvsem uporabniška izkušnja, stabilnost, hitrost in varnost aplikacije.

### **Ključne besede:**

komunikacija, aplikacija, ConVibe, varnost, hitrost, uporabniška izkušnja, stabilnost, sporočila, digitalna povezljivost, primerjava platform

## **ABSTRACT**

This research paper presents the conceptual design, frameworks, and tools used in the development of our communication application, ConVibe. The research focuses on the methods and principles necessary to achieve optimal application performance and ensure secure and fast message exchange. At the end of the paper, ConVibe is compared to other popular solutions on the market, such as Discord and Snapchat, evaluating aspects like user experience, stability, speed, and security.

**Key words:** ConVibe, communication, security, speed, user experience, stability, messaging, digital connectivity, platform comparison.

CONVIBE

## Kazalo

UVOD .....	9
1.1 Hipoteze .....	9
1.2 Metode raziskovanja .....	10
1.3 Struktura raziskovalnega dela.....	10
2. Obstoječi trg.....	10
2.1 Discord .....	11
2.2 Snapchat .....	12
2.3 WhatsApp .....	13
3. Uporabljene tehnologije .....	14
3.1 React.js .....	14
3.2 Node.js .....	15
3.3 Socket.IO.....	15
3.4 MongoDB.....	16
3.5 Supabase.....	16
3.6 Redux .....	17
4. Praktični del raziskovalne naloge .....	19
4.1 Idejna zasnova .....	19
4.2 Varnost podatkov .....	19
4.3 Prvi koncepti aplikacije .....	20
4.4 Optimizacija aplikacije.....	21
5. PREDSTAVITVENI DEL .....	22
Login page:.....	22
Register page:.....	22
Verification code page.....	23
Chat page:.....	24
Ob kliku na levo zgornji gumb, se prikaže profil uporabnika: .....	24
Izberes enega izmed uporabnikov .....	25
Odpre se ti chat, in lahko pošiljaš razne emojije, gife ali pa pošlješ datoteke iz računalnika: .....	25
Pošiljaš lahko tudi zvočna sporočila: .....	26

Lahko si tudi urediš profil:.....	26
Lahko si spremeniš svoje geslo: .....	27
Lahko si spremeniš svetlobo, iz temnega načina na svetlega ali obratno, s pritiskom na ta gumb:.....	27
Tukaj je še primer svetlega načina: .....	27
S tem gumbom se odjavis: .....	28
PREDSTAVITEV REZULTATOV RAZISKOVALNE NALOGE.....	28
HIPOTEZE .....	29
1.hipoteza: .....	29
2.hipoteza: .....	29
3.hipoteza: .....	30
7. ZAKLJUČEK.....	30
VIRI IN LITERATURA .....	31

## **Kazalo slik**

Slika 1 Začetna stran .....	22
Slika 2 Ustvarjanje Profila .....	22
Slika 3 Potrditvena koda .....	23
Slika 4 Prva stran aplikacije.....	24
Slika 5 Prikaz profila .....	24
Slika 6 + gumb za začetek pogovora .....	25
Slika 7 Izbira uporabnikov .....	25
Slika 8 Vrstica za klepet .....	25
Slika 9 Zvočno sporočilo .....	26
Slika 10 Urejanje profila .....	26
Slika 11 Spreminjanje gesla .....	27
Slika 12 Spreminjanje teme aplikacije .....	27
Slika 13 Primer druge teme .....	27
Slika 14 Gumb za odjavo.....	28

## UVOD

V sodobnem času si življenja brez pametnih naprav skoraj ne moremo več predstavljati. Njihova razširjenost in dostopnost sta privedli do razvoja številnih digitalnih rešitev, ki so nadomestile mnoge tradicionalne načine komunikacije in organizacije vsakodnevnih nalog. Ideja za naš projekt se je porodila iz potrebe po bolj intuitivni in dinamični platformi za povezovanje uporabnikov, kar smo želeli uresničiti z razvojem ConVibe.

Med delom na projektu smo se soočali z različnimi izzivi, zato smo podrobno raziskali in analizirali sodobne pristope k razvoju komunikacijskih aplikacij. Naš cilj je bil ustvariti hitro, varno in uporabniku prijazno platformo, ki bi omogočala učinkovito izmenjavo besedilnih in večpredstavnostnih vsebin. Uporabili smo napredne tehnologije, ki zagotavljajo zanesljivo delovanje, dobro odzivnost in visoko stopnjo varnosti.

ConVibe je zasnovan kot orodje, ki omogoča preprosto povezovanje med uporabniki in podpira različne oblike komunikacije, kot so glasovna sporočila. Pri razvoju smo posebno pozornost namenili dostopnosti in intuitivni uporabi, saj želimo, da aplikacija služi širokemu spektru uporabnikov. V nadaljevanju naloge podrobneje predstavljamo idejno in tehnično zasnovo aplikacije ter njene ključne funkcionalnosti.

### 1.1 Hipoteze

Glavni namen raziskovalne naloge je bil razviti funkcionalno aplikacijo, ki bi omogočala različnim skupinam uporabnikov dostop do ključnih funkcij za nemoteno in varno komunikacijo. Pri implementaciji smo se osredotočili na poenotenost sistema in zagotavljanje visoke stopnje zaščite podatkov.

Na podlagi tega smo oblikovali naslednje hipoteze:

1. Aplikacija bo v enakih pogojih za vsaj 40 % hitrejša v primerjavi s podobnimi platformami.
2. Uporaba ogrodja Node.js v kombinaciji z React bo zagotavljala boljšo odzivnost kot tradicionalne monolitne arhitekture.
3. Naša aplikacija bo bolj zavarovana z »verification code« kot so aplikacije whatsapp, viber....

## 1.2 Metode raziskovanja

Pri pripravi raziskovalne naloge smo se soočili s pomanjkanjem strokovnih virov in literature, ki bi pokrivala to specifično tematiko. Zato smo se pri raziskovanju in razvoju večinoma opirali na analizo obstoječih aplikacij, ki ponujajo podobne funkcionalnosti. Velik del raziskave in implementacije smo izvedli neodvisno, saj na tem področju še ne obstaja univerzalna rešitev, ki bi v celoti ustrezala naši idejni zasnovi.

## 1.3 Struktura raziskovalnega dela

Raziskovalna naloga je strukturirana tako, da najprej predstavi osrednjo temo in hipoteze. V drugem poglavju smo analizirali obstoječe rešitve na trgu, ki ponujajo podobne funkcionalnosti. V tretjem delu smo opisali uporabljena orodja in tehnologije, ki so bile ključne za razvoj aplikacije. Četrto poglavje je namenjeno opisu samega procesa razvoja in izvedbe projekta. V predzadnjem poglavju smo predstavili rezultate analize ter ključne ugotovitve. Zaključno, v petem poglavju, pa smo ovrednotili zastavljene hipoteze in podali končne sklepe.

## 2. Obstoječi trg

V začetni fazi raziskave smo podrobno raziskali obstoječi trg aplikacij za komunikacijo in socialna omrežja. Ugotovili smo, da na trgu obstaja veliko že uveljavljenih rešitev, kot so Discord, Snapchat, WhatsApp in druge, vendar se na tem področju še vedno pojavljajo potrebe po novih funkcionalnostih in izboljšavah. Kljub široki uporabi teh aplikacij za komunikacijo obstaja vedno večja potreba po bolj personaliziranih, varnih in enostavnih rešitvah, ki uporabnikom omogočajo hitrejšo in boljše izkušnjo.

Med najbolj priljubljenimi rešitvami za komunikacijo danes so Discord, ki je osredotočen na igre in skupinske pogovore, Snapchat, ki omogoča pošiljanje kratkih videoposnetkov in sporočil, ter WhatsApp, ki je široko uporabljena platforma za vsakodnevno besedilno sporočanje. Kljub temu, da te aplikacije ponujajo različne prednosti, pa še vedno obstajajo vrzeli, ki lahko povzročijo težave pri uporabniški izkušnji, varnosti in dostopnosti.

Na trgu trenutno ni aplikacije, ki bi združevala najboljše lastnosti teh obstoječih rešitev in obenem ponujala edinstvene funkcionalnosti, kot so boljša integracija multimedijskih vsebin, naprednejša orodja za upravljanje skupin in varno shranjevanje osebnih podatkov. Tako se zdi, da je prostor za nove inovacije in izboljšave, kar je tudi motivacija za razvoj aplikacije ConVibe. Ta aplikacija je zasnovana z namenom, da

uporabnikom omogoči preprosto in varno komunikacijo, ki bo hkrati odgovarjala na vse zgoraj omenjene izzive in priložnosti.

### **2.1 Discord**

Discord je brezplačna aplikacija za sporočanje in glasovne klepete, ki je prvotno namenjena igralcem video iger, a je kmalu postala priljubljena tudi med širšo uporabniško bazo. Discord omogoča uporabnikom enostavno pošiljanje besedilnih sporočil, klicanje z videom, glasovno klepetanje in ustvarjanje skupin (strežnikov), kjer se lahko povežejo z drugimi uporabniki glede na interese ali dejavnosti. Aplikacija nudi bogat nabor funkcionalnosti, kot so nalaganje datotek, integracije z drugimi platformami, kot so YouTube in Twitch, ter možnost ustvarjanja različnih kanalov znotraj strežnikov za boljšo organizacijo.

Med ključnimi prednostmi Discorda je njegova visoka prilagodljivost. Uporabniki lahko ustvarijo javne ali zasebne strežnike, določijo posebne nastavitve za vsak kanal, dodajo botov za avtomatizacijo nalog in prilagodijo vsebino v skladu z lastnimi potrebami. Poleg tega je aplikacija na voljo za vse glavne platforme, vključno z Windows, macOS, Linux, iOS in Android.

Discord se ponaša tudi z visokimi varnostnimi standardi, kot so šifriranje komunikacije, avtentikacija v dveh korakih in nadzor nad dovoljenji za vsak kanal in strežnik, kar zagotavlja zaščito osebnih podatkov in preprečuje zlorabe.

Vendar pa Discord ni brez pomanjkljivosti. Kljub temu, da je odlična rešitev za bolj sproščene in igralne skupine, ni najbolj primerna za profesionalno rabo ali manjše podjetje, ki išče enostavno in intuitivno komunikacijsko orodje brez prekompleksnih funkcij. Poleg tega je bila aplikacija kritizirana zaradi morebitnih težav z zasebnostjo, saj je bila v preteklosti tarča različnih napadov in zlorab podatkov.

## 2.2 Snapchat

Snapchat je priljubljena mobilna aplikacija za hitro pošiljanje kratkih videoposnetkov, slik in besedilnih sporočil, ki se samodejno izbrišejo po nekaj sekundah. Ponuja edinstveno izkušnjo, saj se vsebina, ki jo uporabniki pošljejo, samodejno odstrani po ogledu, kar omogoča večjo stopnjo zasebnosti in spontanosti v komunikaciji. Snapchat je priljubljen predvsem med mlajšimi uporabniki, saj omogoča kreativno izražanje z različnimi filtri, nalepkami, besedilnimi napisi in drugimi interaktivnimi elementi.

Ena izmed glavnih značilnosti Snapchata je njegov "Story" funkcionalnost, ki omogoča uporabnikom, da delijo svoje trenutke s prijatelji v obliki zaporedja slik in videov, ki so na voljo 24 ur. Ta funkcionalnost je bila kasneje kopirana tudi s strani drugih platform, kot sta Instagram in Facebook. Poleg tega Snapchat ponuja različne načine interakcije, vključno z video klici, klepeti v skupinah in celo igrami znotraj aplikacije.

Snapchat je tudi zelo prilagodljiv, saj omogoča uporabnikom, da ustvarijo svojo osebno izkušnjo z izbiro različnih nastavitvenih možnosti, kot so obvestila, varnostne nastavitve in možnosti zasebnosti. Aplikacija ponuja visoko stopnjo zasebnosti, saj so sporočila in slike po pregledu samodejno izbrisane, kar uporabnikom omogoča večjo kontrola nad tem, kaj delijo.

Kljub temu pa Snapchat ni brez svojih omejitev. Ena izmed pomanjkljivosti je ta, da aplikacija ni idealna za bolj resne in profesionalne vrste komunikacije, saj njen glavni poudarek ostaja na zabavi in neformalni interakciji. Poleg tega je občasno precej težko slediti večjemu številu pogovorov, saj aplikacija nima tako razvitega sistema za organizacijo sporočil kot nekatere druge platforme. Prav tako je uporabniška izkušnja lahko nekoliko zmedena za tiste, ki se z aplikacijo srečujejo prvič, saj vmesnik ni najbolj intuitiven.

## 2.3 WhatsApp

WhatsApp je ena najbolj priljubljenih aplikacij za sporočanje na svetu, ki omogoča hitro in enostavno pošiljanje besedilnih sporočil, slik, videoposnetkov, zvokovnih posnetkov in dokumentov. Aplikacija, ki je del družbe Meta (prej Facebook), omogoča tudi glasovne in video klice ter omogoča enostavno ustvarjanje skupinskih klepetov, kjer lahko sodeluje več ljudi hkrati. WhatsApp omogoča neposredno povezovanje z ljudmi po vsem svetu, brez stroškov, ki jih običajno nalaga mobilna telefonija, saj uporablja internetno povezavo za prenos podatkov.

WhatsApp omogoča širok spekter funkcionalnosti, kot so pošiljanje sporočil v realnem času, enostavno pošiljanje medijskih datotek, deljenje lokacij, klicanje in videoklicanje, nalaganje zgodovine klepetov v oblak ter varno šifriranje vseh sporočil. Z aplikacijo so uporabniki sposobni takoj komunicirati z družino, prijatelji ali sodelavci ter ustvarjati skupinske klepete za različne namene, bodisi za poslovno komunikacijo bodisi za družabne pogovore.

Ena izmed glavnih prednosti WhatsApp-a je njegova preprostost in široka dostopnost, saj je na voljo za vse glavne mobilne platforme, vključno z Androidom, iOS-om, pa tudi za osebne računalnike prek spletnega vmesnika ali aplikacije za namizje. Poleg tega WhatsApp uporablja šifriranje končnih točk, kar pomeni, da so vsa sporočila med uporabniki zaščitena pred zlorabami in prestrezanjem.

Kljub temu pa WhatsApp ni brez svojih pomanjkljivosti. Ena izmed glavnih omejitev je ta, da aplikacija zahteva telefonsko številko za registracijo, kar pomeni, da uporabniki, ki želijo ostati anonimni, ne morejo uporabljati aplikacije brez razkritja svoje identitete. Prav tako, ker je aplikacija zasnovana predvsem za osebno rabo, morda ni najbolj primerna za bolj formalne ali poslovne namene, kljub dodatnim funkcionalnostim, kot so poslovni računi. Nekateri uporabniki so tudi poročali o težavah s shranjevanjem velikih količin podatkov, kot so medijske datoteke, zaradi omejitev pomnilnika v napravi.

### 3. Uporabljene tehnologije

Pri razvoju aplikacije ConVibe smo uporabili številne sodobne tehnologije, ki so omogočile, da je naša aplikacija postala funkcionalna in uporabniku prijazna. Vsaka od teh tehnologij je bila izbrana z namenom optimizacije delovanja aplikacije, izboljšanja uporabniške izkušnje ter zagotavljanja varnosti in zanesljivosti. V tem razdelku bomo na kratko predstavili ključne tehnologije, ki smo jih uporabili pri razvoju aplikacije ConVibe.

#### 3.1 React.js

Za razvoj uporabniškega vmesnika aplikacije ConVibe smo uporabili React.js, priljubljeno odprtokodno JavaScript knjižnico, ki omogoča gradnjo dinamičnih, interaktivnih in učinkovitih enostranskih spletnih aplikacij. React.js je bil razvit s strani Facebooka in je danes ena izmed najbolj uveljavljenih tehnologij za razvoj sodobnih spletnih vmesnikov.

Ena izmed glavnih prednosti React.js je njegova komponentna arhitektura, ki omogoča ponovno uporabo kode, boljšo organizacijo projekta ter hitrejši razvoj. Z uporabo komponent smo lahko ustvarili modularen in pregleden vmesnik, kjer vsak del aplikacije, kot so prijavni obrazci, klepetalna okna in navigacija, deluje kot samostojna enota, kar močno poenostavi vzdrževanje in nadgradnje sistema.

React.js uporablja virtualni DOM, ki omogoča hitro in učinkovito posodabljanje uporabniškega vmesnika brez nepotrebnega osveževanja celotne strani. Ta tehnologija prispeva k boljši odzivnosti aplikacije, kar je ključno za nemoteno uporabniško izkušnjo pri uporabi funkcij, kot so sprotno pošiljanje sporočil ali preklapljanje med pogovori.

Poleg tega React.js omogoča enostavno integracijo z drugimi tehnologijami in knjižnicami, kot so Redux za upravljanje stanja, Axios za HTTP zahteve ter različne oblikovne knjižnice, kar nam omogoča hitrejšo implementacijo kompleksnih funkcionalnosti. Zaradi široke skupnosti in odlične dokumentacije je React.js odlična izbira za razvoj robustnih in sodobnih spletnih aplikacij, kot je ConVibe.

### 3.2 Node.js

Za strežniški del aplikacije ConVibe smo uporabili Node.js, odprtokodno okolje za izvajanje JavaScript kode na strežniški strani. Node.js temelji na pogonu V8, ki ga uporablja tudi brskalnik Google Chrome, in omogoča hitro, učinkovito ter asinhrono

obdelavo zahtevkov, kar je ključno za sodobne spletne aplikacije, ki zahtevajo visoko zmogljivost in odzivnost.

Ena izmed glavnih prednosti Node.js je njegov dogodekno usmerjen model, ki omogoča neblokirajoče delovanje in s tem obdelavo velikega števila sočasnih zahtevkov brez upočasnjevanja sistema. Ta lastnost je posebej pomembna za aplikacije, kot je ConVibe, kjer mora strežnik istočasno upravljati s številnimi uporabniki, ki pošiljajo in prejemajo sporočila v realnem času.

Node.js omogoča uporabo enotnega programskega jezika (JavaScript) tako na odjemalski kot na strežniški strani, kar poenostavi razvojno okolje in komunikacijo med različnimi deli aplikacije. Poleg tega ima Node.js bogat ekosistem paketov preko upravitelja npm, kar omogoča hitro vključevanje dodatnih funkcionalnosti, kot so avtentikacija uporabnikov, obdelava datotek in povezava z bazami podatkov.

Zaradi svoje razširljivosti in zmogljivosti je Node.js odlična izbira za razvoj realnočasovnih aplikacij, kjer je zanesljivost in hitrost ključnega pomena. V kombinaciji z drugimi tehnologijami, ki jih uporabljamo v ConVibe, kot sta React.js in MySQL, Node.js zagotavlja stabilno in učinkovito delovanje celotnega sistema.

### 3.3 Socket.IO

Za implementacijo komunikacije v realnem času med uporabniki v aplikaciji ConVibe smo uporabili knjižnico Socket.IO. Gre za zmogljivo JavaScript knjižnico, ki omogoča dvosmerno, dogodkovno usmerjeno komunikacijo med strežnikom in odjemalcem preko spletnih vtičnic (WebSockets) ali alternativnih transportnih protokolov, kadar WebSocket ni na voljo.

Socket.IO smo uporabili predvsem za funkcionalnosti, povezane s takojšnjim pošiljanjem in prejemanjem sporočil, kar je bistvenega pomena za delovanje sodobne klepetalne aplikacije. S pomočjo Socket.IO lahko vsak uporabnik prejme nova sporočila v trenutku, ko so poslana, brez potrebe po osveževanju strani ali ročnih zahtevah do strežnika.

Ena izmed največjih prednosti Socket.IO je njegova preprostost uporabe in dobra integracija z Node.js okoljem. Strežnik lahko posluša in oddaja različne dogodke, ki jih odjemalci nato obravnavajo v realnem času. To omogoča gladko in hitro izmenjavo

informacij, kot so besedilna sporočila, obvestila, statusi uporabnikov (npr. "uporabnik je online") in drugo.

Poleg tega Socket.IO vključuje različne varnostne mehanizme, kot so preverjanje pristnosti in nadzor nad povezavami, kar omogoča zanesljivo in varno komunikacijo med uporabniki. Zaradi svoje prilagodljivosti in visoke zmogljivosti je Socket.IO idealna izbira za vse funkcionalnosti, ki zahtevajo interaktivno in nemoteno izkušnjo v realnem času, kot je to primer v ConVibe.

### **3.4 MongoDB**

Za shranjevanje nekaterih podatkov v aplikaciji ConVibe smo poleg relacijske baze uporabili tudi MongoDB, priljubljen sistem za upravljanje nerelacijskih (NoSQL) podatkovnih baz. MongoDB temelji na dokumentno usmerjenem modelu, kjer se podatki shranjujejo v obliki JSON-podobnih dokumentov (BSON), kar omogoča večjo fleksibilnost pri strukturi podatkov in lažje prilagajanje spreminjajočim se zahtevam aplikacije.

Ena izmed ključnih prednosti MongoDB je njena sposobnost učinkovitega upravljanja s kompleksnimi in dinamičnimi podatkovnimi strukturami, brez potrebe po togi shemi, kot jo poznamo pri relacijskih bazah. To nam omogoča hitro in preprosto razširjanje funkcionalnosti aplikacije brez večjih sprememb v strukturi baze.

MongoDB se odlično povezuje z okoljem Node.js prek knjižnic, kot je Mongoose, kar omogoča enostavno definiranje modelov, preverjanje pravilnosti podatkov in izvajanje naprednih poizvedb. V aplikaciji ConVibe smo MongoDB uporabljali predvsem za shranjevanje podatkov, ki so bolj nepredvidljive narave ali zahtevajo večjo prilagodljivost, kot so zgodovina pogovorov, obvestila ali nastavitve uporabnikov.

Poleg tega MongoDB omogoča enostavno skaliranje in visoko razpoložljivost podatkov prek replikacij in porazdeljenih zbirk (sharding), kar zagotavlja stabilno delovanje tudi pri večjem številu uporabnikov. Zaradi svoje prilagodljivosti, zmogljivosti in enostavne integracije z ostalimi tehnologijami je MongoDB pomemben gradnik aplikacije ConVibe.

### **3.5 Supabase**

V aplikaciji ConVibe smo za hitro postavitve in upravljanje podatkovne baze ter avtentikacije uporabnikov preizkusili tudi Supabase, odprtokodno alternativo storitvi Firebase, ki temelji na tehnologijah, kot so PostgreSQL, real-time zmogljivosti in RESTful API-ji. Supabase ponuja celovito rešitev za backend, ki vključuje podatkovno bazo, avtentikacijo, shrambo datotek in funkcionalnosti v realnem času, kar omogoča hitrejši razvoj brez potrebe po kompleksni infrastrukturi.

## CONVIBE

Ena izmed glavnih prednosti Supabase je njegova enostavna uporaba in integracija s front-end tehnologijami, kot sta React.js in Next.js. S pomočjo pripravljenih SDK-jev smo lahko hitro implementirali prijavo in registracijo uporabnikov, varno shranjevanje podatkov ter spremljanje sprememb v realnem času, kar bistveno olajša razvoj dinamičnih spletnih aplikacij, kot je ConVibe.

Supabase uporablja PostgreSQL, kar pomeni, da združuje prednosti relacijske baze podatkov z dodatnimi funkcionalnostmi, kot so vgrajena podpora za JSON, real-time naročnine in avtomatsko generiranje API-jev za vsako tabelo. To nam omogoča večjo kontrolo nad podatki in hkrati enostaven dostop do njih iz aplikacije.

Poleg tega Supabase vključuje tudi upravljanje z dovoljenji in varnostnimi pravilniki (Row Level Security), kar omogoča natančen nadzor nad tem, kateri uporabniki lahko dostopajo do posameznih podatkov. Zaradi svoje odprtosti, enostavne uporabe in bogatega nabora funkcionalnosti je Supabase odlična izbira za hitrejši razvoj in prototipiranje aplikacij, kot je ConVibe.

### 3.6 Redux

Za učinkovito upravljanje stanja v aplikaciji ConVibe smo uporabili knjižnico Redux, ki omogoča centralizirano in predvidljivo upravljanje s podatki v aplikacijah z več komponentami. Redux je posebej uporaben v kompleksnejših aplikacijah, kjer je potrebno dosledno in pregledno prenašanje podatkov med različnimi deli uporabniškega vmesnika.

Glavna prednost Reduxa je njegova enotna shramba stanja (store), ki omogoča, da so vsi podatki dostopni na enem mestu, kar zmanjšuje potrebo po prenašanju podatkov preko več komponent in poenostavi odpravljanje napak. V ConVibe smo Redux uporabili za shranjevanje informacij o prijavljenem uporabniku, seznamu pogovorov, aktivnih stikov in drugih ključnih podatkov, ki jih uporabljajo različne komponente v aplikaciji.

Redux deluje na osnovi dejanj (actions) in reduktorjev (reducers), kar zagotavlja jasen potek sprememb stanja in olajša testiranje ter nadzor nad delovanjem aplikacije. Vsaka sprememba stanja je rezultat eksplicitno definiranega dejanja, kar povečuje transparentnost in stabilnost sistema.

V kombinaciji z orodji, kot je Redux DevTools, smo imeli možnost pregledovanja zgodovine stanja in spremljanja vseh sprememb v realnem času, kar je pripomoglo k hitrejšemu razvoju in odpravljanju morebitnih napak. Redux se odlično povezuje z

## CONVIBE

React.js, zato je bil logična izbira za naše potrebe po strukturiranem in odzivnem upravljanju stanja v aplikaciji ConVibe.

## **4. Praktični del raziskovalne naloge**

### **4.1 Idejna zasnova**

Naša aplikacija, ConVibe, se osredotoča na izboljšanje komunikacije med uporabniki na varen in intuitiven način. Na začetku smo želeli ustvariti platformo, ki omogoča enostavno in hitro komunikacijo med ljudmi, pri čemer smo se navdihovali v priljubljenih aplikacijah, kot sta Discord in Snapchat. Naša ideja je bila razviti aplikacijo, ki bi združevala različne funkcionalnosti družbenih omrežij, hkrati pa ponujala visoko stopnjo varnosti in zasebnosti za svoje uporabnike.

Zasnova aplikacije je vključevala tri glavne tipe uporabnikov: osnovni uporabnik, moderator in administrator. Osnovni uporabnik je imel možnost komunikacije z drugimi uporabniki, pošiljanja sporočil, oblikovanja skupin, deljenja vsebin ter sodelovanja v različnih pogovorih. Moderator je imel dodatne pravice za nadzorovanje in moderiranje vsebin, medtem ko je administrator imel popoln nadzor nad aplikacijo, ustvarjanjem in upravljanjem uporabniških računov ter urejanjem nastavitve platforme.

Na začetku je bila ideja osredotočena na možnost uporabnikov, da se povežejo in komunicirajo z drugimi, pri tem pa so imeli na voljo različne vrste sporočil, kot so besedilna sporočila, slike, video klici in še več. Kasneje smo vključili tudi dodatne možnosti za organizacije, ki so omogočale ustvarjanje javnih in zasebnih skupin, v katerih so se lahko člani organizacij povezovali in sodelovali.

Da bi zagotavljali varnost uporabnikov, smo vključili več varnostnih ukrepov, kot so šifriranje vseh komunikacijskih kanalov in zaščita uporabniških podatkov. Poudarek smo dali tudi zaščiti pred zlorabami in nepooblaščenimi dostopi, zato smo za vsakega uporabnika uporabili enkratne prijavnne ključe in izvedli strog nadzor nad tem, kdo ima dostop do katere vsebine.

### **4.2 Varnost podatkov**

Vse aplikacije, ki obdelujejo osebne podatke uporabnikov, morajo zagotoviti ustrezno varnost teh podatkov in zaščito pred morebitnimi zlorabami. Pri razvoju aplikacije ConVibe smo se osredotočili na zagotavljanje varnosti in zasebnosti uporabniških podatkov, da bi uporabniki brezskrbno komunicirali z drugimi. Naša aplikacija tako vključuje številne varnostne ukrepe za zaščito podatkov in komunikacije.

Ob registraciji uporabniki prejmejo jasne informacije o tem, zakaj in kako so njihovi podatki obdelovani ter za koliko časa bodo shranjeni. Za uporabo aplikacije mora

uporabnik sprejeti pogoje glede zbiranja in obdelave svojih podatkov. Prav tako je uporabniku omogočeno, da kadarkoli izbriše svoj račun, kar pomeni, da se bodo vsi podatki, povezani z njegovim profilom, trajno izbrisali iz naših strežnikov. Ta možnost je na voljo pod zavihkom za urejanje profila, kjer lahko uporabnik enostavno izbriše vse svoje podatke.

Za varno shranjevanje podatkov smo uporabili šifriranje povezave med aplikacijo in strežnikom, kar omogočajo protokoli SSL in CORS. Vsaka zahteva, ki jo aplikacija pošlje strežniku, mora vsebovati avtorizacijski žeton, kar zagotavlja, da le pooblašeni uporabniki dostopajo do svojih podatkov. Poleg tega so gesla uporabnikov zaščitena z algoritmom za zgoščevanje SHA256, ki zagotavlja, da gesla niso shranjena v svoji izvorni obliki. Ta algoritem je enosmeren, kar pomeni, da se gesla ne dajo rekonstruirati, kar zagotavlja dodatno varnost.

Za dodatno zaščito baze podatkov smo uporabili naključno generirano geslo, ki ustreza najvišjim varnostnim standardom. Naša podatkovna baza je prav tako shranjena znotraj Docker virtualnega okolja, kar pomeni, da je dostop do nje omejen le na znotraj tega okolja. Ta metoda zagotavlja dodatno zaščito pred morebitnimi zunanjimi vdori, saj so dostop do podatkovne baze in strežnika močno omejeni.

### **4.3 Prvi koncepti aplikacije**

Pred začetkom razvoja aplikacije smo najprej zasnovali osnovno idejo, kako bi najin projekt izgledal. To smo storili z uporabo orodja Figma, ki nam je omogočilo vizualizacijo uporabniškega vmesnika in funkcionalnosti aplikacije. Zasnova v Figma je služila kot smernica pri nadaljnjem razvoju.

Razvoj smo začeli sistematično. Najprej smo se osredotočili na oblikovanje podatkovne baze in razvoj zalednega sistema. Sprva smo se odločili za kombinacijo Laravel-a in Vue.js, vendar smo kasneje ugotovili, da bo uporaba Laravel Blade boljša izbira za naš projekt. Z uporabo Blade-a smo poenostavili razvoj celotne aplikacije, saj smo imeli boljšo integracijo med front-endom in back-endom, kar je pripomoglo k večji hitrosti in boljši preglednosti aplikacije.

Z Laravelom smo ustvarili migracije, ki so nam omogočile enostavno ustvarjanje in upravljanje podatkovne baze. Migracije so v bistvu PHP ukazi, s katerimi smo lahko avtomatsko ustvarili strukturo baze, brez potrebe po zapletenih SQL poizvedbah. Ko smo imeli podatkovno bazo pripravljeno, smo prešli na razvoj modelov. Ti so predstavljali entitete v podatkovni bazi, saj smo uporabili pristop ORM (Object-Relational Mapping), ki nam je omogočil enostavno delo z bazo ter povezovanje entitet.

Ko smo končali z zalednim delom, smo se osredotočili na oblikovanje uporabniškega vmesnika. Začeli smo z uporabo CSS ogrodja Bootstrap, saj je omogočilo hitro preverjanje funkcionalnosti in izgled aplikacije. Ta korak nam je pomagal zagotoviti, da vse funkcije delujejo pravilno, še preden smo zaključili z razvojem končnega oblikovanja aplikacije. S pomočjo Bootstrap-a smo hitro zasnovali osnovno strukturo uporabniškega vmesnika, kar je pripomoglo k nastanku prve funkcionalne različice naše aplikacije.

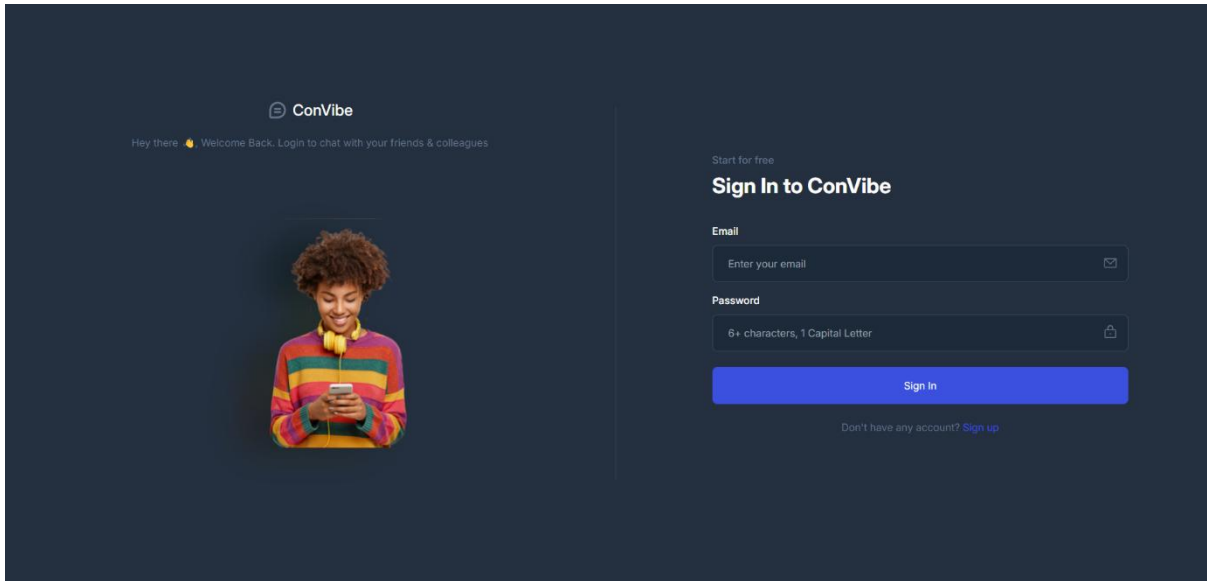
### 4.4 Optimizacija aplikacije

Po tem, ko smo uspešno implementirali aplikacijo, smo se osredotočili na optimizacijo njene hitrosti in zmožljivosti. Naš cilj ni bil le zagotoviti, da aplikacija uspešno deluje, temveč da bo kos obremenitvam večjega števila uporabnikov in da bo zagotavljala hitro odzivnost. Ena izmed prvih težav, na katero smo naleteli, je bila počasnost naše podatkovne baze, še posebej pri obdelavi kompleksnih poizvedb, ki so se ponavljale.

Za začetek smo raziskovali različne rešitve, vključno z morebitno migracijo na NOSQL baze podatkov ali uporabo Firebase. Vendar smo hitro ugotovili, da to ne bi bistveno izboljšalo naše stanje. Po podrobnem raziskovanju smo naleteli na rešitev, ki nam je ustrezala – ogrodje **Redis**. To je omogočalo optimizacijo brez potrebe po migraciji naše obstoječe podatkovne baze. Redis smo enostavno implementirali v našo aplikacijo in ga uporabili za predpomnjenje ponavljajočih se poizvedb, kar je bistveno izboljšalo čas odziva.

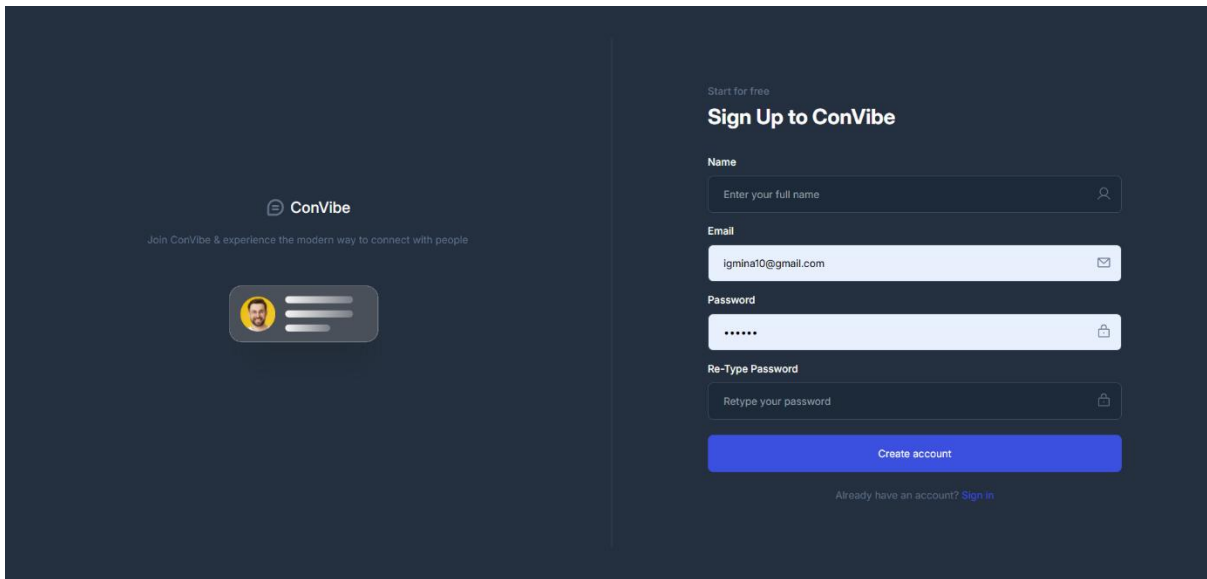
## 5. PREDSTAVITVENI DEL

### Login page:



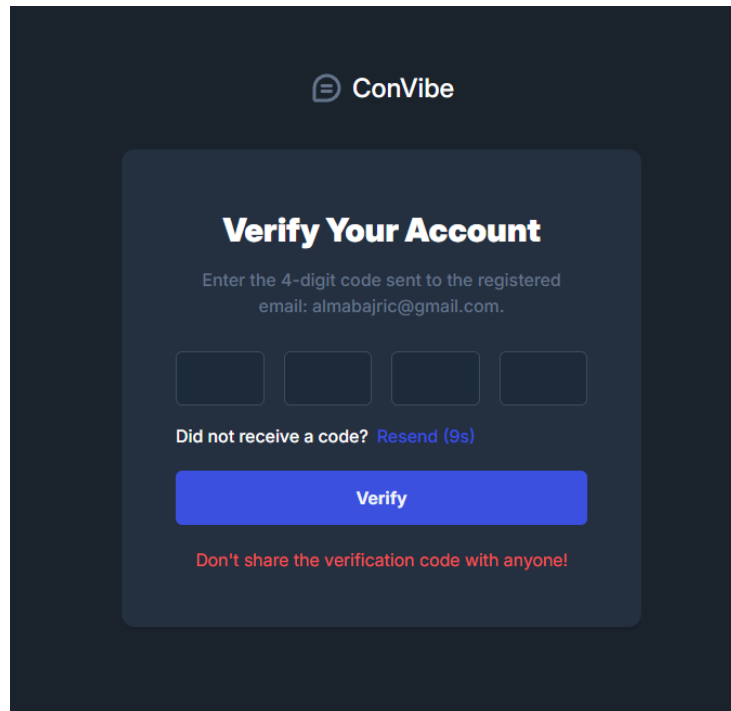
Slika 1 Začetna stran

### Register page:



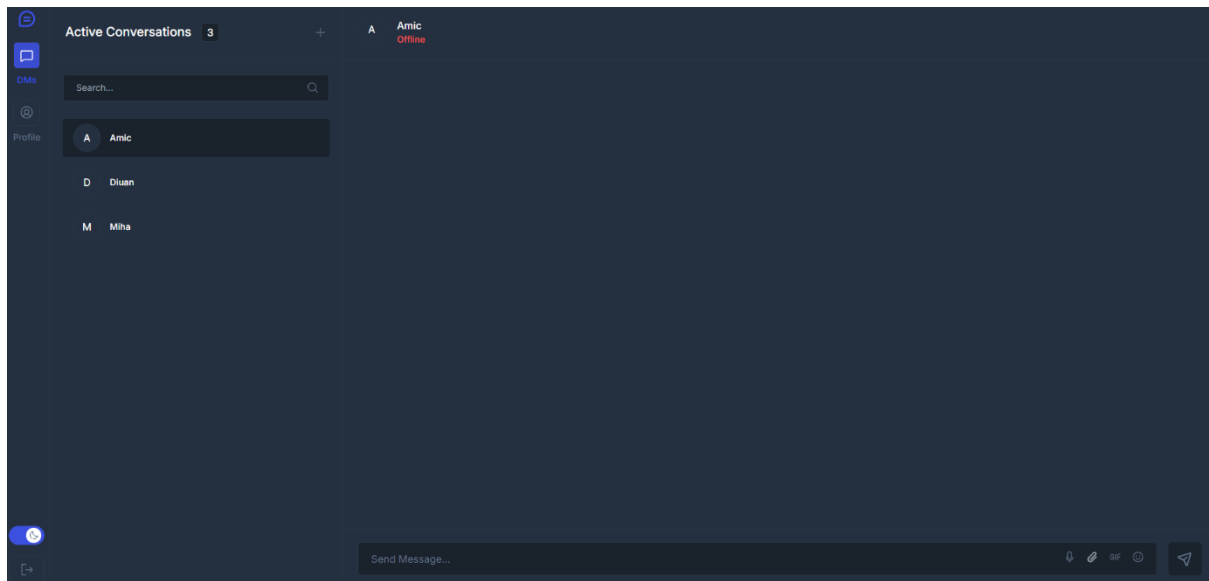
Slika 2 Ustvarjanje Profila

**Verification code page:**



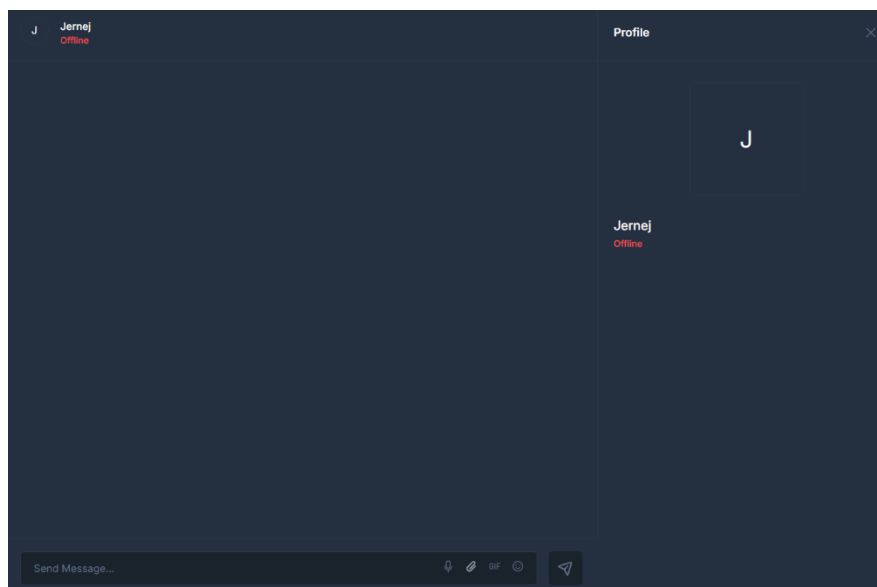
*Slika 3 Potrditvena koda*

**Chat page:**

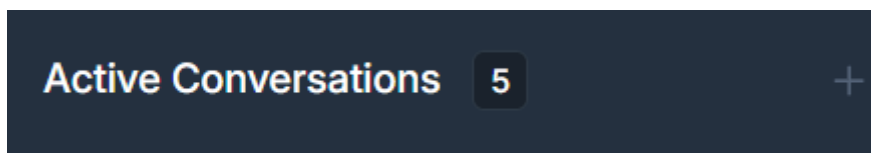


*Slika 4 Prva stran aplikacije*

**Ob kliku na levo zgornji gumb, se prikaže profil uporabnika:**

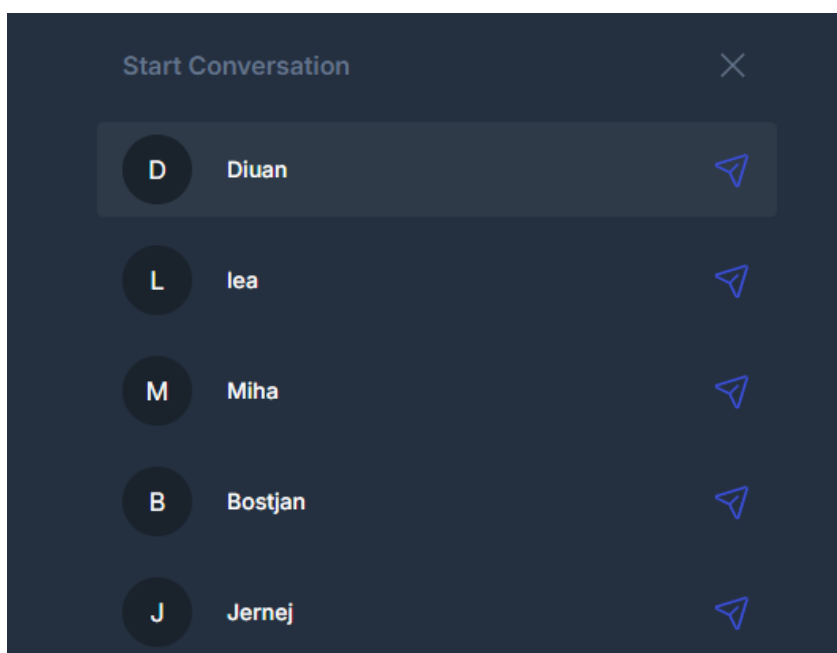


*Slika 5 Prikaz profila*



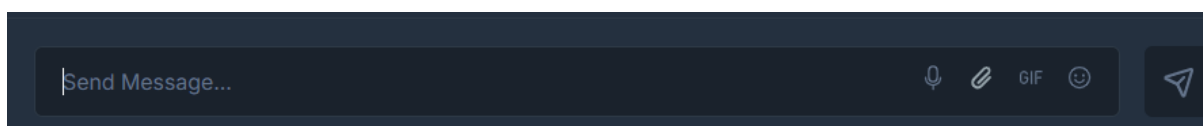
Slika 6 + gumb za začetek pogovora

### Izberes enega izmed uporabnikov



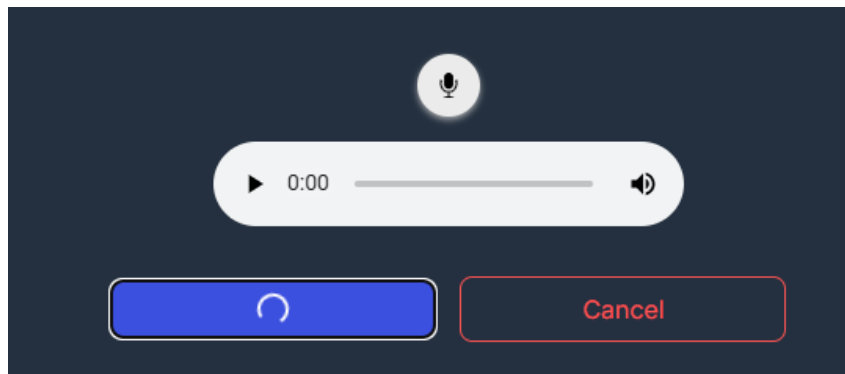
Slika 7 Izbira uporabnikov

**Odpre se ti chat, in lahko pošiljaš razne emojije, gife ali pa pošlješ datoteke iz računalnika:**



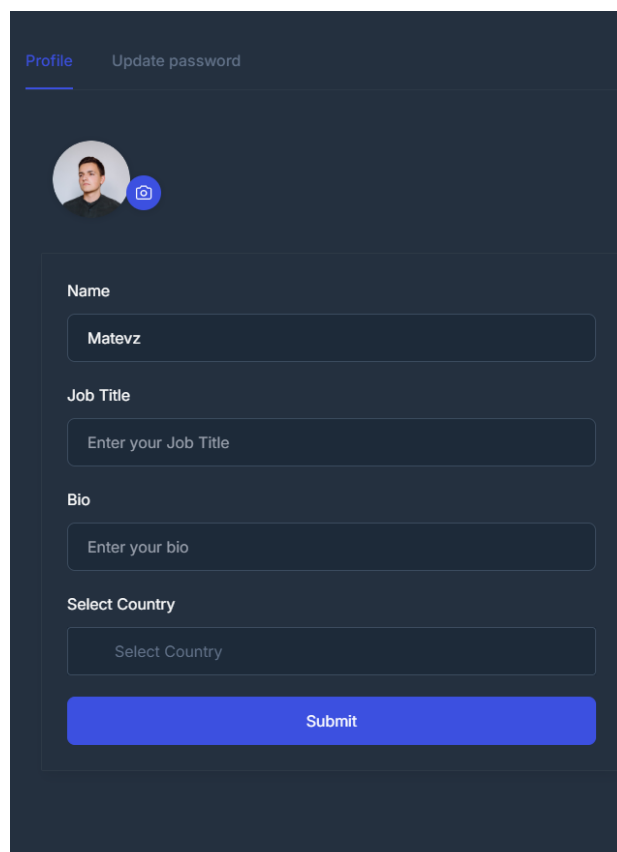
Slika 8 Vrstica za klepet

Pošiljaš lahko tudi zvočna sporočila:



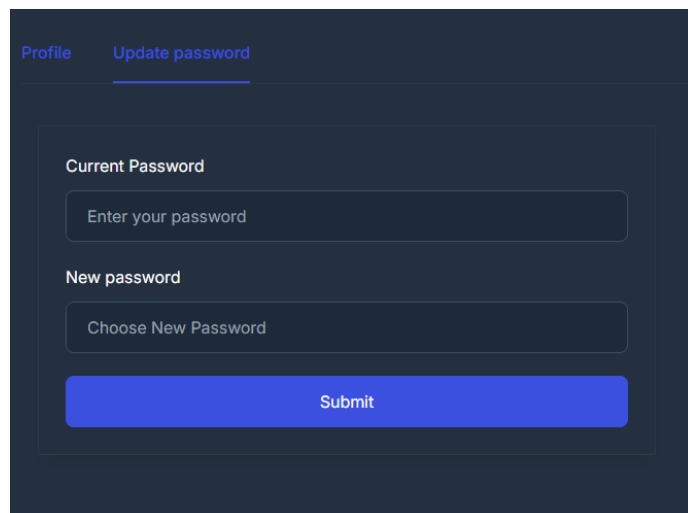
Slika 9 Zvočno sporočilo

Lahko si tudi urediš profil:



Slika 10 Urejanje profila

**Lahko si spremeniš svoje geslo:**



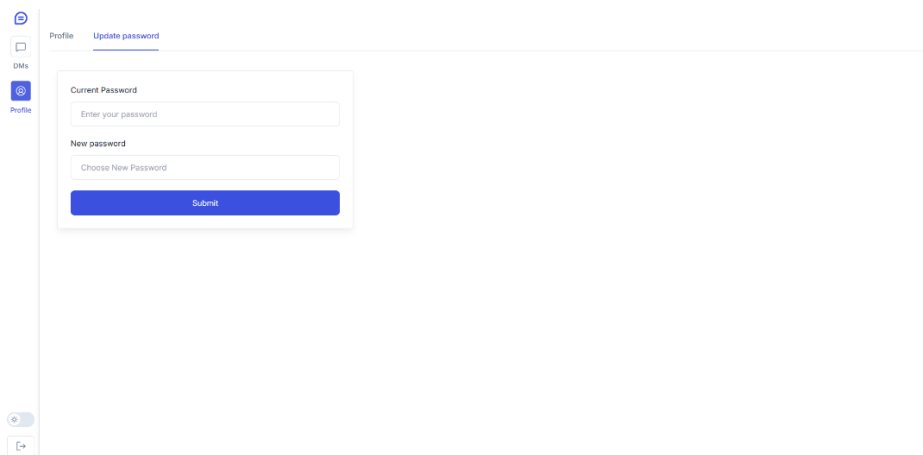
*Slika 11 Spreminjanje gesla*

**Lahko si spremeniš svetlobo, iz temnega načina na svetlega ali obratno, s pritiskom na ta gumb:**



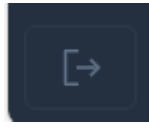
*Slika 12 Spreminjanje teme aplikacije*

**Tukaj je še primer svetlega načina:**



*Slika 13 Primer druge teme*

**S tem gumbom se odjavis:**



*Slika 14 Gumb za odjavo*

## **PREDSTAVITEV REZULTATOV RAZISKOVALNE NALOGE**

Pri razvoju naše aplikacije smo se osredotočili na izboljšanje odzivnosti, stabilnosti in varnosti, še posebej ob večjih obremenitvah uporabnikov. Z izbiro ogrodja Laravel za oba dela aplikacije, tako čelni kot zaledni, smo si prizadevali doseči izboljšave v vseh teh ključnih aspektih.

Za pridobitev točnih in relevantnih rezultatov smo aplikacijo primerjali z lanskoletno različico, ki je temeljila na Vue.js za čelni del in PHP za zaledje. Testiranje smo izvedli v identičnih pogojih: enako omrežje, enak strežnik z 4vCPU jedri, 6GB RAM pomnilnika in 100GB NVMe SSD diskom ter enakimi vnosi v podatkovno bazo (10.000 zapisov). Rezultati so pokazali pomembne izboljšave v vseh ključnih vidikih aplikacije.

Ena izmed večjih prednosti je bila izboljšava odzivnega časa pri vseh CRUD operacijah (ustvari, preberi, posodobi, izbriši). Na primer, čas, potreben za prijavo uporabnika, se je zmanjšal z 240 ms na 130 ms, kar je omogočilo hitrejši dostop do začetne strani aplikacije. Z uporabo mehanizmov za zaščito sej in nadzor zahtevanih podatkov je bila prav tako povečana varnost sistema, saj smo preprečili nepooblaščen dostope in zaščitili uporabniške podatke.

Uporaba Redis podatkovne baze je pomembno pripomogla k izboljšanju zmogljivosti strežnika. Z implementacijo predpomnjenja smo odpravili ponavljajoče se poizvedbe, kar je pripomoglo k večji učinkovitosti. Čas, potreben za izvedbo poizvedb, se je skrajšal

## CONVIBE

za približno 30 ms, saj Redis uporablja RAM za hitro dostopanje do podatkov, ki je skoraj takojšen (pod 2 ms).

Pridobivanje profilnih podatkov iz baze je postalo še hitrejše, saj je bil čas za pridobitev podatkov skrajšan za kar 60 ms – zdaj traja le 30 ms, medtem ko je prejšnji sistem potreboval 90 ms. Implementacija **paginacije** je dodatno zmanjšala obremenitev strežnika pri večjih količinah podatkov. Namesto nalaganja celotnih seznamov smo zdaj podatke razdelili v manjše skupine, kar je zmanjšalo povprečen čas pridobivanja podatkov na 33 ms za 5 zapisov, medtem ko je bil čas v prejšnjem sistemu včasih med 300 ms in 500 ms.

Na podlagi teh rezultatov lahko zaključimo, da je aplikacija po optimizaciji znatno hitrejša, odzivni časi pa so bili izboljšani za približno 60 % do 150 %, kar pomeni, da je aplikacija zdaj zelo odzivna, z le nekaj primeri, kjer odzivni čas preseže 100 ms, vendar so ti primeri zelo redki.

## HIPOTEZE

### 1.hipoteza:

Aplikacija bo v enakih pogojih za vsaj 40 % hitrejša v primerjavi s podobnimi platformami.

To hipotezo zavračamo, saj brez konkretnih testov in primerjav ni mogoče zagotoviti točne odstotne razlike v hitrosti. Hitrost je odvisna od več dejavnikov, vključno s strežniško infrastrukturo, optimizacijo kode in omrežnimi pogoji.

### 2.hipoteza:

Uporaba ogrodja Node.js v kombinaciji z React bo zagotavljala boljšo odzivnost kot tradicionalne monolitne arhitekture.

Ta hipoteza je upravičena, saj Node.js omogoča asinhro obdelavo zahtevkov, kar izboljšuje odzivnost, React pa z virtualnim DOM-om prispeva k hitrejšemu posodabljanju uporabniškega vmesnika.

### 3.hipoteza:

Naša aplikacija bo bolj zavarovana z »verification code« kot so aplikacije WhatsApp, Viber...

To hipotezo zavračamo, saj WhatsApp in Viber že uporabljata preverjanje z verifikacijsko kodo, poleg tega pa imata dodatne varnostne ukrepe, kot je end-to-end šifriranje. Sama uporaba verifikacijske kode ne zagotavlja boljše varnosti v primerjavi z že uveljavljenimi rešitvami.

## 7. ZAKLJUČEK

V zaključku naše raziskovalne naloge želimo poudariti ključna spoznanja, ki smo jih pridobili skozi razvoj naše aplikacije.

Med razvojem projekta smo poglobili svoje znanje pri uporabi različnih tehnologij, kot so **JavaScript**, **CSS**, **MySQL** in **C#**, ki so bile ključne za uspešen razvoj naše spletne aplikacije. **JavaScript** je bil osnova za dinamične in interaktivne funkcionalnosti aplikacije, medtem ko smo za oblikovanje privlačnega in funkcionalnega uporabniškega vmesnika uporabili **CSS**. S pomočjo **MySQL** smo zagotovili učinkovito shranjevanje in obdelavo podatkov v podatkovni bazi, medtem ko smo za napredno obdelavo in upravljanje z logiko aplikacije uporabili **C#**.

S tem projektom smo si pridobili dragocene izkušnje pri delu s podatkovnimi bazami, optimizaciji poizvedb ter izboljšavi varnosti podatkov. Pri razvoju aplikacije smo se osredotočili na optimizacijo hitrosti delovanja in odzivnosti sistema, pri čemer so imeli pomembno vlogo učinkoviti algoritmi za obdelavo in shranjevanje podatkov.

Pri testiranju aplikacije smo izboljšali hitrost nalaganja strani in odzivni čas poizvedb z optimizacijo vmesnikov in zmanjšanjem nepotrebnih poizvedb v podatkovni bazi. Dodatno smo se naučili, kako pomembno je, da so aplikacije prilagodljive in enostavne za širitev, kar pomeni, da so naše rešitve fleksibilne za prihodnje izboljšave in nadgradnje.

Tako smo skozi ta projekt pridobili dragoceno znanje in veščine, ki nam bodo v veliko pomoč pri nadaljnjem razvoju. Sodelovanje v ekipi je prav tako izjemno izboljšalo naše sposobnosti komuniciranja, sodelovanja in reševanja težav, kar je ključno za uspeh v večjih projektih in podjetjih na trgu dela.

V prihodnosti bi želeli nadgraditi našo aplikacijo z dodatnimi funkcionalnostmi, kot je izboljšanje varnosti in optimizacija uporabniške izkušnje z novimi tehnologijami. Prav tako bomo veseli priložnosti za preizkušanje aplikacije v realnem okolju in morebitno sodelovanje s partnerji, kar bi nam omogočilo še boljšo analizo njene uspešnosti v praksi.

## VIRI IN LITERATURA

W3Schools json (brez datuma) Pridobljeno 12.12.2024:

[https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp)

React.js (brez datuma) Pridobljeno 23.11.2024:

<https://react.dev>

MongoDB (brez datuma) Pridobljeno 11.1.2025:

<https://www.mongodb.com>

SupaBase (brez datuma) Pridobljeno 12.1.2025:

<https://supabase.com>

CONVIBE

Redux.js (brez datuma) Pridobljeno 21.2.2025:

<https://redux.js.org>

In veliko pomoči z github forms.