



Srednja šola za kemijo, elektrotehniko in računalništvo

Pot na Lavo 22

3000 Celje

GOVOREČI VIRTUALNI AVATAR

Raziskovalna naloga

Avtorji: Jure Ločnikar, Filip Juršinič, Maj Krajnc

Mentor: Matic Holobar

Mestna občina Celje, Mladi za Celje

Celje, 2025

IZJAVA*

Mentor Matic Holobar 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 Govoreči virtualni avatar, katere avtorji so Filip Juršinič, Maj Krajnc, Jure Ločnikar:

- 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, 4. 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.

Kazalo vsebine

1	Uvod	10
2	Raziskava.....	11
2.1	Namen raziskovalne naloge.....	11
2.2	Cilji naloge	11
2.3	Hipoteze.....	12
3	ChatGPT	12
3.1	GPT-3.5-Turbo	12
3.1.1	Glavne značilnosti	13
3.1.2	Uporabnost	13
3.1.3	Razlogi za izbiro modela GPT-3.5-Turbo	13
3.2	TTS-1.....	15
3.2.1	Glavne značilnosti	15
3.2.2	Uporabnost	15
3.2.3	Razlagi za izbiro modela TTS-1	16
4	Yarn	17
4.1	Ključne značilnosti	18
4.2	Razlogi za izbiro Yarn.....	18
5	Node.js.....	19
5.1	Značilnosti	19
5.2	Uporabnost	19
5.3	Razlogi za izbiro Node.js.....	20
6	React Three Fiber	20
6.1	Pomembne lastnosti.....	21
6.2	Uporabnost	21
6.3	Razlogi za izbiro React Three Fiber za 3D upodabljanje	21

7	Rhubarb Lip Sync.....	22
7.1	Delovanje.....	23
7.2	Uporaba	23
7.3	Razlogi za izbiro Rhubarb Lip Sync za sinhronizacijo govora z animacijo.....	23
8	FFmpeg.....	24
8.1	Delovanje.....	24
8.2	Uporaba	25
8.3	Razlogi za izbiro FFmpeg za obdelavo in pretvorbo zvoka	25
9	WebStorm.....	26
9.1	Funkcionalnosti in značilnosti	26
9.2	Prednosti in slabosti.....	27
10	JavaScript	27
10.1	Delovanje.....	28
10.2	Uporaba	28
11	HTML.....	29
11.1	Delovanje.....	29
11.2	Struktura	29
11.3	HTML 5.....	30
12	CSS.....	30
12.1	Delovanje.....	30
13	React.....	31
13.1	Delovanje.....	31
14	Ready Player Me	32
14.1	Delovanje.....	32
14.2	Uporaba	32
14.3	Prednosti in slabosti.....	33

15 Govoreči virtualni avatar	34
15.1 Izgled in funkcionalnosti spletne aplikacije	34
15.2 Evalvacija rezultatov	35
15.2.1 Razmerje med ceno in zanesljivostjo v primerjavi s pravo osebo.....	35
15.2.2 Razlikovanje med umetno inteligenco in človekom.....	36
15.2.3 Zanimivost v primerjavi s pravo osebo	36
16 Zaključek	37
16.1 Povzetek doseženih ciljev.....	37
16.2 Razprava o rezultatih in njihov pomen.....	37
16.3 Možnosti za nadaljnje izboljšave.....	37
17 Ovrednotenje hipotez.....	38
18 Viri in literatura	39

Kazalo slik

Slika 1 Ustvarjanje API ključa	14
Slika 2 Uporaba API ključa in navodila tekstovnemu API modelu	14
Slika 3 Navodila zvočnemu modelu in uporaba API ključa.....	17
Slika 4 Uporabniški vmesnik naše spletne aplikacije.....	34
Slika 5 Cenik tekstovnega API modela	35
Slika 6 Cenik glasovnega API modela	35

Zahvala

Iskreno se zahvaljujemo vsem, ki so kakorkoli pripomogli k nastanku te raziskovalne naloge. Posebna zahvala gre našemu mentorju gospodu profesorju Maticu Holobarju, ki nam je že od samega začetka svetoval in nas spodbujal v vseh fazah dela. Zahvaljujemo se tudi sošolcem in prijateljem za pomoč pri testiranju spletne aplikacije in odgovarjanju na ankete.

Povzetek

V raziskovalni nalogi predstavljamo razvoj in uporabo spletne aplikacije, ki omogoča pogovor z umetno inteligenco preko 3D-avatarja. Avatar se prikaže kot virtualni sogovornik in z uporabo tehnologij za sintezo govora (Text-To-Speech) ter sinhronizacijo ustnic (lipsync) ustvarja vtis, da se z uporabnikom pogovarja dejanska 3D-oseba. Jedro aplikacije temelji na ChatGPT API-ju (za generiranje besedilnih odzivov in TTS) in knjižnici React Three Fiber (za 3D-prikaz avatarja). Za lipsync obdelavo smo uporabili orodje Rhubarb LipSync, ki na podlagi zvočne datoteke generira ključne točke premikov ustnic, pri čemer smo si pomagali tudi z FFmpeg za pretvorbe zvočnih formatov. Avatarja smo zasnovali s pomočjo platforme Ready Player Me, upravljanje strežniške logike pa poteka v okolju Node.js in Yarn.

Ključne besede: 3D-avatar, lipsync, ChatGPT, TTS, umetna inteligenca

Abstract

In this research paper, we present the development and use of a web application that enables conversation with an artificial intelligence through a 3D avatar. The avatar appears as a virtual interlocutor and, by using text-to-speech (TTS) technology and lip synchronization (lipsync), creates the impression that the user is interacting with a real 3D person. The core of the application is based on the ChatGPT API (for generating text responses and TTS) and the React Three Fiber library (for the 3D avatar display). For lipsync processing, we used the Rhubarb LipSync tool, which generates the key points of lip movements from an audio file, and we also used FFmpeg for audio format conversions. We designed the avatar using the Ready Player Me platform, and the server-side logic is managed in a Node.js and Yarn environment.

Keywords: 3D avatar, lipsync, ChatGPT, TTS, artificial intelligence

1 Uvod

V naši raziskovalni nalogi se posvečamo razvoju in uporabi spletnega orodja, ki združuje interaktivnega 3D-avatarja z lipsync animacijo ter pogovornim modelom ChatGPT. Ideja izvira iz opažanja, da klasična – torej le besedilna – komunikacija včasih ne pritegne dovolj pozornosti in je lahko za uporabnika precej dolgočasna. S povezavo 3D-avatarja, ki se “pogovarja” in hkrati premika ustnice v realnem času, želimo zagotoviti bolj privlačen in življenjski način komunikacije, kar bi pomagalo pri zanimanju uporabnikov.

Številne študije s področja interakcije človek–računalnik so pokazale, da vizualni odziv digitalnih sogovornikov močno poveča angažiranost uporabnikov. Premikanje ustnic avatarja v realnem času in njegova podoba kot virtualne osebe povečata občutek prisotnosti in naravnosti pogovora (Cassell et al., 2000). Takšne izkušnje ne zgolj pritegnejo več pozornosti, temveč tudi omogočajo bolj učinkovito posredovanje informacij, saj se uporabniki bolj vključijo v pogovor.

Naša spletna aplikacija sicer deluje podobno kot drugi pogovorni sistemi (chatboti), vendar dodaja ključno novo plast: glasovno predvajanje odgovorov in istočasno animacijo avatarjevih ustnic. S tem želimo preveriti, kako tak pristop vpliva na uporabnike. Ker se pri takšnih rešitvah pogosto pojavi vprašanje stroškov (na primer naročnine za AI-storitve), nas zanima tudi, ali je takšna rešitev cenovno ugodna in kako se primerja z najetjem pravih ljudi, ki bi nudili podobno vrsto storitev.

V nadaljevanju bomo predstavili glavne teoretične podlage in opisali postopek, s katerim smo izdelali prototip. Pri tem smo uporabili tehnologije, kot so React Three Fiber, Ready Player Me, Rhubarb LipSync in ChatGPT API za umetno inteligenco in sintezo govora. V zaključku bomo predstavili rezultate naše raziskave ter odzive testnih uporabnikov.

2 Raziskava

2.1 Namen raziskovalne naloge

Glavni namen naloge je oblikovati uporabniku prijazno platformo, kjer bo lahko le-ta interaktivno komuniciral s 3D-avtarjem preko pogovornega okna tako, da bo uporabnik vanj vprašanje napisal, avatar pa bo odgovarjal s pomočjo napredne umetne inteligence, hkrati pa bo sistem poskrbel za ustrezno animacijo ustnic.

2.2 Cilji naloge

Ustvarili smo spletno aplikacijo, ki omogoča uporabnikom to, kar smo želeli. Lahko podajo svoje vprašanje v pogovorno okno, odgovor pa prejmejo preko avatarja, ki istočasno odpira in premika usta.

Cilji v teoretičnem delu:

- Predstaviti uporabljene programske jezike, knjižnice in okolja;
- Predstaviti uporabljen API (tekstovni in glasovni);
- Predstaviti zgradbo in delovanje naše spletne aplikacije;

Cilji v praktičnem delu:

- Narediti delujočo spletno aplikacijo, ki pravilno prikazuje 3D avatarja;
- Uspešno integrirati text-to-speech in lipsync v naš program;
- Ustvariti kar se da prijetno uporabniško izkušnjo;

2.3 Hipoteze

Hipoteza 1: Uporaba takšnega orodja za odgovarjanje na vprašanja je cenovno ugodnejša in vsaj enako zanesljiva kakor oseba, ki bi jo plačevali za to.

Hipoteza 2: Na trgu je zelo malo proizvajalcev oz. ponudnikov, ki bi uporabljali tovrstno storitev ali pa jo ponujali.

Hipoteza 3: Ljudje hitro prepoznajo, ali se z njimi pogovarja človek, ali pa jim odgovarja umetna inteligenca.

Hipoteza 4: Ljudem bo pogovor z interaktivnim avatarjem bolj zanimiv kot dopisovanje s pravo osebo.

3 ChatGPT

Je zmogljiv tekstovni model, ki temelji na arhitekturi GPT (Generative Pre-trained Transformer). Zasnovan je tako, da razume in generira naravni jezik, kar mu omogoča ustvarjanje pogovorov, odgovarjanje na vprašanja, pisanje besedil in razlago različnih tem. Ključna prednost ChatGPT-ja je sposobnost kontekstnega razumevanja, saj se lahko med daljšim pogovorom navezuje na prejšnje dele in ohranja rdečo nit pogovora.

3.1 GPT-3.5-Turbo

GPT-3.5-Turbo je različica GPT-3.5 modela, posebej prilagojena in optimizirana za pogovore in interaktivno izmenjavo sporočil prek API-ja. Namenjena je razvijalcem in podjetjem, ki želijo v svoje aplikacije, storitve ali procese integrirati napredno obdelavo naravnega jezika. GPT-3.5-Turbo uporablja metode učenja, zaradi katerih je hitrejši in učinkovitejši v primerjavi z nekaterimi prejšnjimi različicami GPT.

3.1.1 Glavne značilnosti

Je optimiziran za pogovor, kar pomeni, da je izredno dober pri ohranjanju konteksta pogovora, to pa je ključno za podporo uporabniku v realnem času. Ima visoko stopnjo razumevanja in je sposoben zaznati različne stile, tone in namene v besedilu ter se ustrezno odzove. OpenAI pa tudi redno posodablja svoje modele, kar pripomore pri odpravljanju pomanjkljivosti in izboljšanju natančnosti.

3.1.2 Uporabnost

GPT-3.5-Turbo lahko uporabimo na veliko različnih področij, ker je zelo vsestransko “orodje”. Lahko služi kot podlaga za virtualne pomočnike, ki odgovarjajo na vprašanja, rešujejo težave ali usmerjajo uporabnike. Model je uporaben za ustvarjanje opisov, člankov, družbeno medijskih objav, e-poštnih predlog in čeprav ne gre za namenske prevajalske modele, GPT-3.5-Turbo pogosto omogoča solidno “osnovno” prevajanje ali učinkovito strnitev besedila. Njegova vsestranskost izhaja iz arhitekture, ki omogoča učenje iz velike količine besedil in hitro prilagajanje novim nalogam – kar je bilo potrjeno tudi v raziskavi, kjer so ugotovili, da se takšni primeri učijo le z nekaj primeri (Brown et al., 2020).

3.1.3 Razlogi za izbiro modela GPT-3.5-Turbo

OpenAI je s predstavitvijo GPT-3.5-Turbo naredil pomemben korak naprej v razvoju jezikovnih modelov. Ta nadgradnja prinaša izboljšano natančnost in optimizacijo stroškov, zaradi česar je še privlačnejša za širok spekter aplikacij. Odločitev za razvoj in uporabo GPT-3.5-Turbo temelji na več ključnih prednostih:

Izboljšana natančnost in zmanjšani stroški: GPT-3.5-Turbo ponuja boljšo natančnost pri generiranju besedil, hkrati pa znižuje operativne stroške, kar je ključno za podjetja in razvijalce, ki se zanašajo na visoko zmogljive jezikovne modele za svoje aplikacije.

Podpora za daljše kontekste: Z zmogljivostjo obdelave do 4 tisoč žetonov, GPT-3.5-Turbo omogoča obdelavo daljših kontekstov, kar znatno izboljša razumevanje in generiranje

kompleksnejših besedil. To je še posebej koristno za naloge, ki zahtevajo poglobljeno analizo.

Optimizacija za hitrejše in cenejše klice API: Zmožnost skrajšanja pozivov do 90 % z vključitvijo navodil neposredno v model zmanjšuje čas, potreben za vsak klic API, in optimizira stroške, kar je še posebej pomembno za aplikacije z velikim obsegom zahtev.

Inovativne funkcije za interakcijo z zunanjimi orodji: GPT-3.5-Turbo uvaja možnost klicanja funkcij, ki uporabnikom omogoča, da modelu opišejo funkcije in kako naj model odda objekt JSON z argumenti za te funkcije. To odpira nove možnosti za integracijo GPT-ja z različnimi zunanjimi orodji in API-ji, kar širi uporabnost modela v realnih aplikacijah. Za naš projekt je bil ta model ChatGPT API-ja najboljša izbira.

NAME	SECRET KEY
raziskovalna	sk-...00EA

Slika 1 Ustvarjanje API ključa

Slika prikazuje naš skriti API ključ z naslovom, ki smo ga določili. Ta ključ je uporabljen v kodi za oba, tekstovni in zvočni model.

```
console.log('[TTS] ChatGPT question:', userQuestion);
const chatResponse : AxiosResponse<any> = await axios.post(
  url: 'https://api.openai.com/v1/chat/completions',
  data: {
    model: 'gpt-3.5-turbo', // ali 'gpt-4'
    messages: [
      { role: 'system', content: 'Please respond in Slovenian.' },
      { role: 'user', content: userQuestion }
    ]
  },
  config: {
    headers: { Authorization: `Bearer ${OPENAI_API_KEY}` }
  }
);
```

Slika 2 Uporaba API ključa in navodila tekstovnemu API modelu

Druga slika prikazuje, kako je ključ implementiran v kodo. Zaradi varnosti in zaščite smo ga shranili v spremenljivko v ločeno datoteko, tu pa se na spremenljivko le sklicujemo. Vidimo tudi izbiro tekstovnega modela in navodila, po katerih se ravna pri odgovarjanju ter kako prejme uporabnikovo vprašanje.

3.2 TTS-1

TTS-1 je ime oz. oznaka za storitev, ki omogoča pretvorbo besedila v govor (Text-To-Speech). Čeprav obstajajo različni ponudniki in rešitve, TTS-1 na splošno označuje API ali programsko knjižnico, ki s pomočjo strojnega učenja simulira naravni človeški govor, v našem primeru, pa je s TTS-1 mislimo model, ki ga je izdal OpenAI, ne tako dolgo nazaj.

3.2.1 Glavne značilnosti

Model TTS-1 je precej prilagodljiv, saj omogoča izbiro različnih glasov, hitrosti govora in intonacij, da se čim bolj ujema z željami uporabnika. Integriranje je precej preprosto, tako kot pri modelu GPT-3.5-Turbo, podpira pa tudi številne različne jezike, med drugim tudi slovenščino. Prednost tega modela je njegova naravna intonacija in izgovorjava, ki se približa človeški. Napredni sistem TTS, kot je TTS-1, temeljijo na sodobnih pristopih strojnega učenja in obdelave zvoka, kar omogoča skoraj naravno govorno komunikacijo. Po raziskavi podjetja Google (Shen et al., 2018) takšni modeli že dosegajo kakovost govora, ki je zelo blizu človeški izgovorjavi.

3.2.2 Uporabnost

Samodejno lahko pretvarja katerokoli besedilo v zvočne datoteke. Uporabniki lahko tako v realnem času odgovore slišijo, namesto da bi jih morali prebrati. Uporabljen je lahko tudi v klicnih telefonskih centrih, ampak seveda v primernem jeziku (Angleščino govori lepše, bolj naravno in tudi bolj tekoče, kot slovenščino).

3.2.3 Razlagi za izbiro modela TTS-1

Pri razvoju našega sistema smo se odločili za uporabo TTS-1, naprednega modela sinteze govora, ki ga je razvil OpenAI. Njegova uporaba prinaša naraven zvočni izhod, boljšo intonacijo in optimizirano obdelavo, kar omogoča učinkovitejšo integracijo v aplikacije, ki zahtevajo visokokakovostno pretvorbo besedila v govor. Naša izbira temelji na več ključnih lastnostih tega modela:

Naravna kakovost govora in učinkovita obdelava: Model TTS-1 izstopa po realistični in tekoči izgovorjavi, ki se približuje naravnemu govoru. Posebno pozornost posveča poudarkom, premorom in dinamiki izgovorjenih besed, kar omogoča boljšo razumljivost in prijetnejšo izkušnjo poslušanja. Poleg tega je model optimiziran za učinkovito procesiranje zvoka, kar omogoča hiter odziv sistema brez nepotrebnih zamikov.

Prilagodljivost glasov in tonalitete: Ena izmed pomembnih prednosti TTS-1 je možnost izbire različnih glasov, kar omogoča večjo prilagodljivost pri oblikovanju govorne izkušnje. Uporabniki lahko izberejo glas, ki najbolj ustreza specifičnim potrebam aplikacije.

Hitrost generiranja zvoka: Model zagotavlja hitro sintezo govora, saj omogoča skoraj takojšnje ustvarjanje zvočnih datotek.

Podpora za različne zvočne formate in enostavna implementacija: TTS-1 omogoča generiranje zvoka v več formatih, kot so MP3, OGG in drugi, kar zagotavlja prilagodljivost pri shranjevanju in predvajanju v različnih okoljih. Integracija s sistemom je preprosta, saj model uporablja standardiziran API, ki omogoča enostavno povezovanje s programsko opremo.

```

console.log('[TTS] Calling OpenAI TTS...');
const ttsPayload = {
  model: 'tts-1',
  voice: 'alloy',
  input: textResponse
};
const ttsRes : AxiosResponse<any> = await axios.post(
  url: 'https://api.openai.com/v1/audio/speech',
  ttsPayload,
  config: {
    responseType: 'arraybuffer',
    headers: {
      Authorization: `Bearer ${OPENAI_API_KEY}`,
      'Content-Type': 'application/json'
    }
  }
);

```

Slika 3 Navodila zvočnemu modelu in uporaba API ključa

Na zgornji sliki ponovno podamo spremenljivko za naš ključ in izberemo zvočni model ter njegov glas besedilo za govor.

4 Yarn

Yarn je upravitelj paketov, zasnovan za okolje Node.js. Omogoča, da razvijalci preprosto nameščajo, posodablajo in upravljajo zunanje knjižnice oziroma pakete, ki jih potrebujejo v svojih JavaScript projektih. Yarn je bil razvit kot alternativa NPM-ju (Node Package Manager) z namenom izboljšati hitrost nameščanja, delovanje v ekipnih okoljih in doslednost nameščenih paketov.

Sam po sebi ne vključuje vgrajenega strežnika za poganjanje spletnih aplikacij na naslovu localhost. Običajno pa se v kombinaciji z razvojnim ogrođjem ali skripto sproži t. i. razvojni strežnik na lokalnem naslovu. To je odvisno predvsem od konfiguracije v datoteki »package.json« in uporabljenih orodij.

4.1 Ključne značilnosti

Yarn je bil zasnovan tako, da v primerjavi s tradicionalnim NPM dobičke v hitrosti dosega z vzporednimi prenosi paketov. Za razliko od starejših izvedb NPM, ki so pakete prenašale zaporedno, Yarn podpira prenašanje in nameščanje paketov v vzporednih nitih, kar občutno skrajša čas nameščanja v projektih z večjim številom odvisnosti. Yarn uporablja uradni NPM register, zato so vsi paketi, ki so dostopni prek NPM, dostopni tudi prek Yarn. Uporabnikom se zato ni treba prilagajati drugim registrskim repozitorijem, razen če to želijo.

4.2 Razlogi za izbiro Yarn

Izbira tega orodja temelji na več ključnih prednostih, ki pripomorejo k boljši učinkovitosti in stabilnosti projekta.

Izboljšana hitrost in optimizirana namestitev paketov: Ena izmed glavnih prednosti Yarn je njegova hitrost, saj uporablja paralelno obdelavo, kar pomeni, da lahko hkrati prenaša in namešča več paketov, kar znatno skrajša čas postavitve okolja.

Natančno in zanesljivo upravljanje odvisnosti: Yarn zagotavlja deterministično upravljanje odvisnosti, kar pomeni, da bo vsaka namestitev zagotovila enake verzije paketov na vseh sistemih. To omogoča stabilno in predvidljivo razvojno okolje, kar je ključnega pomena za zagotavljanje doslednega delovanja aplikacije.

Podpora za delo brez povezave: Zaradi lokalnega keširanja lahko Yarn deluje tudi v načinu brez povezave, kar pomeni, da lahko razvijalci ponovno uporabijo že prenesene pakete, tudi če trenutno nimajo dostopa do interneta. To pripomore k večji prilagodljivosti pri razvoju in zmanjšuje nepotrebno porabo pasovne širine.

Integracija z različnimi ekosistemi: Yarn je združljiv z NPM (Node Package Manager) in drugimi orodji v ekosistemu JavaScript, kar omogoča nemoteno integracijo v obstoječe projekte brez večjih prilagoditev. Poleg tega ponuja napredne funkcije, kot so Workspaces, ki omogočajo učinkovitejše upravljanje več projektov znotraj istega repozitorija.

5 Node.js

Node.js je odprtokodno, večplatformno izvajalno okolje za JavaScript, ki omogoča poganjanje JavaScript kode zunaj brskalnika. Uporablja se predvsem za razvoj strežniških aplikacij in omogoča izgradnjo hitrih in učinkovitih aplikacij.

5.1 Značilnosti

Node.js uporablja enojni nitni model, ki obdeluje več hkratnih zahtev. To pomeni, da ne čaka na zaključek ene operacije, preden začne izvajati naslednjo (npr. branje iz baze podatkov se zgodi v ozadju, medtem pa strežnik še naprej obdeluje druge zahteve). Node.js je zato izjemno učinkovit za aplikacije v realnem času, kot so spletni klepeti, igre in API-ji.

Namesto da bi čakal na dokončanje operacij (kot je branje podatkov iz datoteke ali baze), Node.js uporablja dogodkovno zanko, ki omogoča hkratno obdelavo več zahtev brez upočasnitev. To je idealno za strežnike, kjer se pogosto obdeluje veliko število hkratnih zahtev. Node.js uporablja V8 pogon, ki hitro pretvarja JavaScript v strojno kodo. V8 je optimiziran za hitrost in uporablja JIT (Just-In-Time) prevajanje, kar omogoča hitro izvajanje kode.

5.2 Uporabnost

Node.js pogosto nadomešča klasične strežnike, kot sta Apache ali Nginx, predvsem pri aplikacijah v realnem času. Node.js je odlična izbira za realno časovne aplikacije, kot so klepetalnice. Omogoča tudi obdelavo velikih količin podatkov, kot so npr. branje večjih datotek.

5.3 Razlogi za izbiro Node.js

Za razvoj našega sistema smo izbrali Node.js, visoko zmogljivo in učinkovito JavaScript izvajalno okolje, ki omogoča hitro izvajanje strežniških aplikacij. Ta izbira temelji na več ključnih prednostih, ki prispevajo k boljši učinkovitosti in prilagodljivosti našega projekta.

Asinhrona in dogodkovno usmerjena arhitektura: Ena izmed največjih prednosti Node.js je njegova dogodkovno vodena arhitektura, ki omogoča sočasno obdelavo več zahtevkov brez nepotrebnega obremenjevanja sistema. To omogoča hitrejše izvajanje, še posebej pri aplikacijah, ki temeljijo na realno časovni komunikaciji, kot so spletni klepeti, API-ji in pretočne storitve.

Visoka zmogljivost in učinkovitost: Node.js temelji na V8 JavaScript pogonu, ki skrbi za hitro interpretacijo in izvajanje kode. Poleg tega optimizirano upravljanje pomnilnika omogoča, da aplikacije delujejo stabilno in učinkovito tudi pri večji obremenitvi.

Širok ekosistem odprtokodnih knjižnic: Z dostopom do NPM (Node Package Manager) ima Node.js enega največjih ekosistemov odprtokodnih modulov, kar omogoča hitrejši razvoj in ponovno uporabo preverjenih rešitev. To znatno skrajša razvojni čas in omogoča enostavno integracijo različnih funkcionalnosti.

6 React Three Fiber

React Three Fiber (R3F) je knjižnica za React, ki omogoča enostavno uporabo Three.js (popularnega JavaScript 3D grafičnega pogona) v React aplikacijah. Nudi deklarativni pristop k ustvarjanju 3D scen, modelov in interaktivnih animacij znotraj React komponent. Knjižnica je zgrajena na Three.js in uporablja Reactov razmeroma enostaven sintaktični sistem (JSX), zaradi česar je razvoj 3D vsebin bolj intuitiven in organiziran.

6.1 Pomembne lastnosti

Zelo koristen je deklarativni pristop, saj uporablja Reactov JSX sintaktični slog, kar omogoča opisovanje 3D scene na podoben način kot HTML ali React komponente. Namesto proceduralne kode (kot v Three.js) omogoča čistejšo in bolj modularno strukturo. Omogoča boljšo integracijo z obstoječimi React aplikacijami, delo s komponentami in stanji pa je poenostavljeno.

React Three Fiber omogoča učinkovito upravljanje WebGL scene brez nepotrebnih ponovnih renderiranj. Lahko dinamično dodajamo in odstranjujemo objekte brez preobremenitve GPU-ja, z uporabo »instancing« in LOD (Level of Detail) lahko optimiziramo prikazovanje več tisoč objektov. Podpira 3D modele v formatih .glb, .gltf...

6.2 Uporabnost

Knjižnico je priporočljivo uporabiti za razvijanje interaktivnih 3D aplikacij v React-u, če potrebujemo zmogljivo optimizacijo 3D grafike in če želimo združiti React logiko s 3D svetom (stanja, animacije, uporabniški vnosi).

6.3 Razlogi za izbiro React Three Fiber za 3D upodabljanje

Pri implementaciji 3D vizualizacij smo se odločili za uporabo R3F, ki predstavlja zmogljiv in prilagodljiv ovitek za Three.js v okviru React ekosistema. Ta izbira nam omogoča enostavno integracijo 3D-grafike v spletne aplikacije ter bolj intuitiven in modularen razvoj.

Povezava z React ekosistemom: Ena izmed glavnih prednosti React Three Fiber je njegova naravna integracija z Reactom, kar omogoča uporabo JSX sintakse in komponentnega pristopa pri izdelavi 3D prizorov. To razvijalcem omogoča boljšo organizacijo kode in uporabo znanih React konceptov, kot so stanje, reaktivnost in učinki.

Optimizirana zmogljivost in učinkovito upravljanje virov: Ker R3F temelji na Three.js, izkorišča zmogljivosti WebGL za učinkovito upodabljanje 3D grafike. Poleg tega omogoča avtomatizirano upravljanje virov, kar pomeni, da sistem samodejno odstrani nepotrebne objekte, s čimer optimizira pomnilnik in izboljša delovanje aplikacije.

Modularnost in ponovna uporaba komponent: Z R3F je možno razdeliti kompleksne 3D prizore na manjše, večkrat uporabne komponente, kar poenostavi razvoj in vzdrževanje kode. To omogoča lažje sestavljanje interaktivnih 3D vsebin brez potrebe po ročnem upravljanju scene, kamer in luči.

Dostop do Three.js funkcionalnosti: Čeprav React Three Fiber poenostavi delo s Three.js, ohranja polni dostop do njegovega API-ja, kar omogoča uporabo naprednih tehnik kot so senčenja, post-processing učinki in fizikalne simulacije. To zagotavlja popolno prilagodljivost pri ustvarjanju vizualno dovršenih 3D prizorov.

Enostavna integracija z drugimi knjižnicami in orodji: R3F omogoča združevanje z animacijskimi in fizikalnimi knjižnicami, kot so react-spring, drei in cannon-es, kar omogoča realistično interakcijo s 3D elementi in bolj dinamično uporabniško izkušnjo.

7 Rhubarb Lip Sync

Rhubarb Lip Sync je odprtokodno orodje za sinhronizacijo ustnic, ki analizira zvočne posnetke govora in jih pretvori v natančne sekvence gibov ustnic. To omogoča, da se animirani liki premikajo skladno s posnetim govorom, kar je ključno pri razvoju iger, animacij in interaktivnih projektov. Njegova največja prednost je, da uporabniku ni treba ročno animirati vsakega gibanja ustnic, saj program samodejno analizira zvok in ustvari ustrezne fonemske oblike, ki jih lahko neposredno uporabimo v animacijskih orodjih.

7.1 Delovanje

Rhubarb Lip Sync deluje tako, da sprejme zvočno datoteko govora in jo analizira s pomočjo napredne prepoznave zvoka. Namesto uporabe strojnega učenja, ki je lahko počasno in zahteva velike količine podatkov, se zanaša na hitre akustične analize, ki učinkovito prepoznajo foneme v govoru. Po analizi ustvari izhodne podatke v obliki časovnih oznak, ki označujejo, katera oblika ustnic mora biti prikazana ob določenem času. Ti podatki se izvozijo v formate, kot so JSON, XML ali specializirane datoteke za uporabo v animacijskih programih.

7.2 Uporaba

Orodje je še posebej priljubljeno pri razvijalcih iger in animatorjih, saj omogoča enostavno in hitro integracijo govora v animirane sekvence. Podpira širok nabor programskih okolij, vključno z Unity, Godot, Blenderjem in drugimi animacijskimi orodji. To pomeni, da je mogoče rezultate analize neposredno uporabiti pri animiranju likov, ne da bi bilo treba vsak gib ustnic nastavljanje ročno. Tako se znatno skrajša čas produkcije, kar je še posebej pomembno pri projektih z velikimi količinami dialoga.

7.3 Razlogi za izbiro Rhubarb Lip Sync za sinhronizacijo govora z animacijo

Za avtomatizirano sinhronizacijo govora z ustnimi premiki smo izbrali Rhubarb Lip Sync, zmogljivo orodje za analizo zvočnih posnetkov in generiranje lipsync podatkov. Orodje omogoča učinkovito prepoznavanje ustreznih fonemov in ustvarjanje animacijskih podatkov, ki jih lahko uporabimo za animiranje likov v različnih okoljih.

Natančna analiza govora in avtomatsko generiranje lipsync podatkov: Ena ključnih prednosti Rhubarb Lip Sync je natančno prepoznavanje govora, ki omogoča ustvarjanje časovno usklajenih ustnih premikov glede na zvočni posnetek. Sistem analizira zvoke in foneme ter jih pretvori v niz ustnih oblik, ki jih lahko animiramo.

Podpora različnim izhodnim formatom: Orodje omogoča izvoz lipsync podatkov v različne formate, kot so JSON, XML, TSV in DAT, kar zagotavlja prilagodljivost pri uporabi v različnih animacijskih in razvojnih okoljih. To nam omogoča enostavno integracijo s 3D modeli, 2D animacijami in igralnimi pogoni.

Podpora za več jezikov in prilagodljivost prepoznavanja: Čeprav Rhubarb Lip Sync uporablja PocketSphinx za prepoznavanje angleškega govora, omogoča tudi fonetični način prepoznavanja, ki je primeren za druge jezike. S tem lahko dosežemo boljše prilagoditev za specifične projekte in večjo jezikovno podporo.

Optimizirana zmogljivost in avtomatska obdelava: Orodje je zasnovano za hitro analizo zvočnih posnetkov, kar omogoča avtomatsko generiranje lipsync podatkov brez ročnega označevanja. To močno skrajša čas produkcije animacij in izboljša učinkovitost dela.

8 FFmpeg

FFmpeg je zmogljivo odprtokodno orodje za obdelavo avdio in video datotek, ki omogoča pretvorbo, urejanje, zajemanje in pretočno predvajanje večpredstavnostnih vsebin. Gre za niz ukaznih vrstic in knjižnic, ki razvijalcem in uporabnikom omogočajo izvajanje kompleksnih operacij na zvočnih in video datotekah brez potrebe po grafičnem vmesniku. Zaradi svoje prilagodljivosti in podpore širokemu naboru formatov je FFmpeg eno najpogosteje uporabljenih orodij v večpredstavnostni industriji.

8.1 Delovanje

FFmpeg deluje prek ukazne vrstice in uporablja modularno arhitekturo, ki vključuje različne komponente za obdelavo medijskih datotek. Glavna komponenta je ffmpeg, ki omogoča manipulacijo video in zvočnih datotek, medtem ko ffmpeg-ffplay služi kot preprost predvajalnik, ki temelji na knjižnicah FFmpeg. Tretja pomembna komponenta je ffmpeg-ffprobe, ki omogoča analizo in prikaz informacij o datotekah, kot so kodeki, ločljivost in trajanje.

FFmpeg uporablja različne knjižnice, kot so libavcodec (za kodiranje in dekodiranje), libavformat (za manipulacijo z večpredstavnostnimi formati) in libswscale (za spreminjanje velikosti slik). Te knjižnice delujejo skupaj in omogočajo učinkovito obdelavo zvočnih in video datotek z minimalno izgubo kakovosti.

8.2 Uporaba

Ena glavnih funkcionalnosti FFmpeg je pretvorba med različnimi zvočnimi in video formati. S pomočjo enostavne ukazne vrstice lahko uporabnik hitro in učinkovito spremeni format datoteke brez potrebe po dodatni programski opremi.

8.3 Razlogi za izbiro FFmpeg za obdelavo in pretvorbo zvoka

Za potrebe pretvorbe, obdelave in optimizacije zvočnih datotek smo izbrali FFmpeg, zmogljivo in široko uporabljano orodje za manipulacijo avdio in video vsebin. Njegova hitrost, prilagodljivost in podpora različnim formatom omogočajo učinkovito integracijo v naš sistem.

Široka podpora za zvočne formate: FFmpeg podpira praktično vse priljubljene zvočne formate, kot so MP3, OGG, WAV, AAC in številni drugi. Ta vsestranskost nam omogoča prilagajanje in pretvorbo zvočnih datotek glede na specifične zahteve aplikacije, ne glede na format vhodnih podatkov.

Visoka zmogljivost in optimizacija zvoka: Orodje omogoča hitro in učinkovito obdelavo zvoka, vključno s kompresijo, normalizacijo glasnosti, odstranjevanjem šumov ter prilagajanjem zvočnih parametrov. Zaradi svoje optimizirane arhitekture lahko FFmpeg izvaja visokokakovostne pretvorbe z minimalno izgubo kakovosti.

Samodejna pretvorba iz MP3 v OGG: V našem sistemu uporabljamo FFmpeg za avtomatsko konverzijo MP3 datotek v OGG format, kar zagotavlja boljšo združljivost z različnimi predvajalniki in aplikacijami. Ta proces se izvaja v ozadju brez uporabniškega posredovanja, kar omogoča nemoteno delovanje sistema.

Kompaktna in hitra izvedba: FFmpeg je zasnovan tako, da omogoča hitro izvajanje obdelav z nizko porabo sistemskih virov. To je še posebej pomembno pri obdelavi večjih količin zvočnih podatkov, saj lahko hkrati izvaja več nalog brez opaznega vpliva na delovanje strežnika.

Integracija v avtomatizirane delovne procese: Ena izmed ključnih prednosti FFmpeg je njegova možnost uporabe v kriptiranih in avtomatiziranih procesih. Omogoča izvajanje naprednih zvočnih operacij preko ukazne vrstice, kar je idealno za brezhibno integracijo z ostalimi komponentami v našem sistemu.

9 WebStorm

WebStorm je napreden in zmogljiv IDE (Integrated Development Environment), ki je posebej zasnovan za razvoj v JavaScript-u, TypeScript-u in drugih sodobnih spletnih tehnologijah. Razvil ga je JetBrains, podjetje, znano po svojih kakovostnih razvojnih orodjih, kot so IntelliJ IDEA, PyCharm in PhpStorm. WebStorm je priljubljen med spletnimi razvijalci, saj ponuja bogato zbirko funkcionalnosti, ki izboljšajo produktivnost, zmanjšajo napake in omogočajo učinkovitejše programiranje.

Njegova glavna prednost je pametna podpora za JavaScript ekosistem, vključno z React, Angular, Vue.js, Node.js in drugimi tehnologijami. Ponuja napredne funkcije, kot so avtomatsko dopolnjevanje kode, močno orodje za refaktoriranje, vgrajeno odpravljanje napak ter integracijo s priljubljenimi orodji za različico kode, kot je Git.

9.1 Funkcionalnosti in značilnosti

WebStorm je zasnovan tako, da razvijalcem omogoča hitro in učinkovito delo z JavaScript, TypeScript in drugimi povezani jeziki. Ena izmed ključnih funkcij je avtomatsko dopolnjevanje kode, ki analizira kodo v realnem času in ponuja inteligentne predloge za funkcije, metode in spremenljivke. To zmanjšuje število napak in pospešuje proces razvoja.

9.2 Prednosti in slabosti

WebStorm ponuja številne prednosti, zaradi katerih je priljubljena izbira med razvijalci spletnih aplikacij. Njegova inteligentna pomoč pri kodiranju, vgrajena orodja za razhroščevanje in testiranje ter odlična podpora za JavaScript ekosistem omogočajo hitrejši in učinkovitejši razvoj. Poleg tega njegova močna integracija z Git-om in razširljivost omogočata enostavno sodelovanje med razvijalci v ekipah.

Slabost WebStorm-a je njegova plačljiva licenca, saj gre za komercialni izdelek, ki zahteva naročnino. Vendar pa JetBrains ponuja brezplačno različico za študente in odprtokodne projekte, kar olajša dostop do orodja za tiste, ki si ne morejo privoščiti plačljive različice. Poleg tega je WebStorm nekoliko težji za računalniške vire kot nekateri lažji urejevalniki kode, kot je VS Code, kar pomeni, da lahko pri večjih projektih deluje počasneje na starejših računalnikih.

10 JavaScript

Je vsestranski programski jezik, ki se uporablja predvsem za razvoj interaktivnih spletnih strani in spletnih aplikacij. Prvotno je bil zasnovan kot skriptni jezik za brskalnike, ki omogoča dinamično spreminjanje vsebine spletnih strani, vendar se je sčasoma razširil in postal ključna tehnologija v modernem spletnem razvoju. Danes se JavaScript uporablja ne le v brskalnikih, temveč tudi na strežnikih (npr. z Node.js), v mobilnih aplikacijah in celo pri razvoju namiznih aplikacij.

Temelji na dogodkih in omogoča izvajanje kode v realnem času brez potrebe po osveževanju strani. Zaradi svoje asinhrono narave je še posebej primeren za razvoj odzivnih in hitrih aplikacij, kot so spletni klepeti, interaktivne igre in dinamične spletne strani. Poleg tega podpira številne paradigme programiranja, vključno z objektno usmerjenim, funkcijskim in imperativnim programiranjem, kar ga naredi izjemno prilagodljivega za različne vrste projektov.

10.1 Delovanje

JavaScript deluje kot interpretirani jezik, kar pomeni, da ga brskalnik ali drugo izvajalno okolje izvaja neposredno brez potrebe po predhodni kompilaciji. To omogoča hitro in dinamično izvajanje kode, saj spremembe lahko vidimo takoj. Brskalniki uporabljajo posebne JavaScript pogone, kot so V8 (Chrome, Node.js), SpiderMonkey (Firefox) in JavaScriptCore (Safari), ki prevajajo in optimizirajo JavaScript kodo v strojno kodo, kar izboljšuje njeno zmogljivost.

Ko brskalnik naleti na JavaScript kodo, jo izvede v kontekstu DOM-a (Document Object Model), kar omogoča spreminjanje vsebine spletne strani v realnem času. Poleg tega lahko JavaScript komunicira s strežnikom prek AJAX-a ali Fetch API-ja, kar omogoča dinamično nalaganje podatkov brez ponovnega nalaganja strani.

JavaScript uporablja enojni nitni model z dogodkovno zanko, kar pomeni, da izvaja kodo zaporedno, vendar lahko upravlja asinhrono operacije, kot so zahteve na strežnik ali časovni zamiki, brez blokiranja glavne niti. To omogoča gladko delovanje aplikacij in izboljša uporabniško izkušnjo.

10.2 Uporaba

JavaScript je temelj sodobnega spletnega razvoja in se uporablja za ustvarjanje dinamičnih in interaktivnih spletnih aplikacij. Omogoča manipulacijo s HTML in CSS, kar pomeni, da lahko s kodo spreminjamo vsebino, slog in postavitev spletnih strani.

Poleg tega omogoča komunikacijo s strežnikom prek AJAX-a in Fetch API-ja, kar pomeni, da lahko spletne aplikacije pridobivajo in pošiljajo podatke brez osveževanja strani. To je ključno pri razvoju aplikacij, kot so spletne trgovine, socialna omrežja in nadzorne plošče.

11 HTML

HTML (HyperText Markup Language) je osnovni jezik za oblikovanje spletnih strani, ki določa njihovo strukturo in vsebino. Uporablja se za organizacijo besedila, slik, povezav, gumbov in drugih elementov, ki sestavljajo spletno stran. Gre za označevalni jezik, kar pomeni, da se ne izvaja kot programski jezik, temveč se uporablja za opisovanje in strukturiranje vsebine, ki jo nato brskalnik interpretira in prikaže uporabniku.

11.1 Delovanje

HTML dokumenti so sestavljeni iz oznak, ki opisujejo posamezne elemente na strani. Oznake so zapisane v oglatih oklepajih (< >) in so lahko parne (npr. <p> ... </p>) ali samozapiralne (npr.). Parne oznake imajo začetno (<tag>) in končno oznako (</tag>), ki označujeta začetek in konec določenega elementa.

Vsaka spletna stran se začne z osnovno HTML strukturo, ki vključuje glavo (<head>) in telo (<body>). Glava vsebuje metapodatke o strani, kot so naslov in povezave na zunanje vire (npr. CSS datoteke), medtem ko telo vsebuje glavno vsebino, ki jo vidi uporabnik.

11.2 Struktura

HTML dokument je sestavljen iz več pomembnih delov, ki določajo njegovo strukturo.

- <!DOCTYPE html> – Določa, da gre za dokument HTML5;
- <html> – Glavni vsebnik celotne strani;
- <head> – Vsebuje metapodatke o strani, povezave na CSS in druge zunanje vire;
- <title> – Nastavi naslov strani, ki se prikaže v zavihku brskalnika;
- <meta> – Podatki o kodiranju znakov, odzivnosti in drugih lastnostih;
- <body> – Vsebuje dejansko vsebino, ki jo vidi uporabnik (besedilo, slike, povezave...);

11.3 HTML 5

HTML5 je najnovejša različica HTML-ja, ki prinaša številne izboljšave in nove elemente.

Med njimi so:

- Semantični elementi (<header>, <nav>, <section>, <article>, <footer>) za boljšo organizacijo vsebine;
- Večpredstavnostni elementi (<audio> in <video>), ki omogočajo vdelavo zvoka in videoposnetkov brez dodatnih vtičnikov;
- Napredni obrazci z novimi vrstami vhodnih polj (<input type="email">, <input type="date">, <input type="number"> itd.);
- Canvas API (<canvas>) za risanje grafike neposredno na spletni strani;

12 CSS

CSS je slogovni jezik, ki se uporablja za oblikovanje in urejanje vizualnega videza spletnih strani. Omogoča spreminjanje barv, pisav, postavitve elementov in drugih oblikovnih lastnosti, ne da bi bilo treba spreminjati osnovno strukturo HTML dokumenta. CSS je ključen za ločevanje vsebine (HTML) in oblikovanja, kar omogoča večjo prilagodljivost in enostavnejše vzdrževanje spletnih strani.

Deluje po principu kaskadnega oblikovanja, kar pomeni, da lahko več pravil vpliva na isti element, pri čemer imajo nekatera pravila prednost pred drugimi glede na specifičnost in vrstni red določanja slogov.

12.1 Delovanje

CSS deluje tako, da cilja na določene HTML elemente in jim dodeli določene slogovne lastnosti. To se izvaja prek selektorjev, ki določajo, na katere elemente se slogovna pravila nanašajo, ter prek lastnosti, ki določajo obliko teh elementov. CSS pravila so običajno zapisana v ločeni datoteki s pripono .css ali pa so vključena neposredno v HTML dokument.

13 React

React je priljubljena odprtokodna JavaScript knjižnica, ki omogoča gradnjo uporabniških vmesnikov (UI). Razvil ga je Facebook (Meta) in je zasnovan za izdelavo hitrih, odzivnih in modularnih spletnih aplikacij. Omogoča komponentni pristop k razvoju, kar pomeni, da so uporabniški vmesniki sestavljeni iz več samostojnih, ponovno uporabnih komponent, ki obdelujejo lasten UI in stanje.

Temelji na konceptu virtualnega DOM-a, kar omogoča optimizirano upodabljanje in izboljšano zmogljivost. To pomeni, da namesto neposrednega posodabljanja celotnega DOM-a brskalnika, React najprej ustvari virtualno kopijo DOM-a, primerja spremembe in posodobi le tiste dele, ki so se spremenili.

13.1 Delovanje

Uporablja komponente, ki predstavljajo posamezne dele UI-ja. Komponente so lahko funkcijske ali razredne in se lahko sestavljajo v večje celote.

Najosnovnejša React aplikacija je sestavljena iz glavne komponente, ki upodobi HTML vsebino. Uporablja JSX (JavaScript XML), kar omogoča pisanje HTML podobne sintakse znotraj JavaScript kode.

14 Ready Player Me

Ready Player Me je platforma za ustvarjanje personaliziranih 3D avatarjev, ki jih lahko uporabniki uporabljajo v različnih virtualnih okoljih, igrah in aplikacijah. Platforma omogoča ustvarjanje avatarjev na podlagi fotografije ali ročnega prilagajanja ter podpira integracijo z več kot 900 aplikacijami in igrami, vključno z VR/AR izkušnjami, spletnimi aplikacijami in družbenimi platformami. Omogoča enostavno integracijo avatarjev v igre, aplikacije in virtualne svetove, kar razvijalcem omogoča, da uporabnikom ponudijo prilagodljivo in personalizirano izkušnjo.

14.1 Delovanje

Ready Player Me uporabnikom omogoča, da v nekaj preprostih korakih ustvarijo svoj edinstveni 3D avatar. Postopek je hiter in enostaven:

1. Ustvarjanje avatarja: Uporabnik lahko naloži svojo fotografijo ali pa ročno prilagodi videz avatarja (spol, obrazne poteze, pričesko, oblačila itd.).
2. Prilagajanje podrobnosti: Avatar lahko uporabnik nadalje prilagodi z različnimi možnostmi, kot so oblačila, dodatki in barve kože.
3. Izvoz avatarja: Ko je pripravljen, ga lahko uporabnik izvozi v različnih formatih in uporabi v podpornih aplikacijah, kot so VR igre, aplikacije ali spletni klepeti.
4. Integracija z aplikacijami: Razvijalci lahko Ready Player Me integrirajo v svoje projekte prek API-ja in SDK-ja, kar omogoča enostavno dodajanje avatarjev v igre in aplikacije.

14.2 Uporaba

Ready Player Me je zasnovan tako, da je združljiv z različnimi tehnologijami, kot so Unity, Unreal Engine in WebXR, kar pomeni, da ga lahko razvijalci preprosto vključijo v svoje igre in virtualne svetove. Avatarji so optimizirani za VR, AR in spletne aplikacije, kar pomeni, da jih je mogoče uporabljati v različnih okoljih, ne glede na napravo ali platformo.

Platforma je še posebej uporabna v:

- VR/AR igrah (npr. VRChat, Spatial, Mozilla Hubs);
- Spletnih aplikacijah (npr. video klici, virtualna srečanja);
- Metaverse okoljih (npr. Somnium Space, Mona);
- NFT in blockchain projektih, kjer se avatarji lahko uporabljajo kot digitalne identitete ali unikatni NFT-ji;

14.3 Prednosti in slabosti

Ponuja številne prednosti, ki ga postavljajo v ospredje na področju ustvarjanja 3D avatarjev:

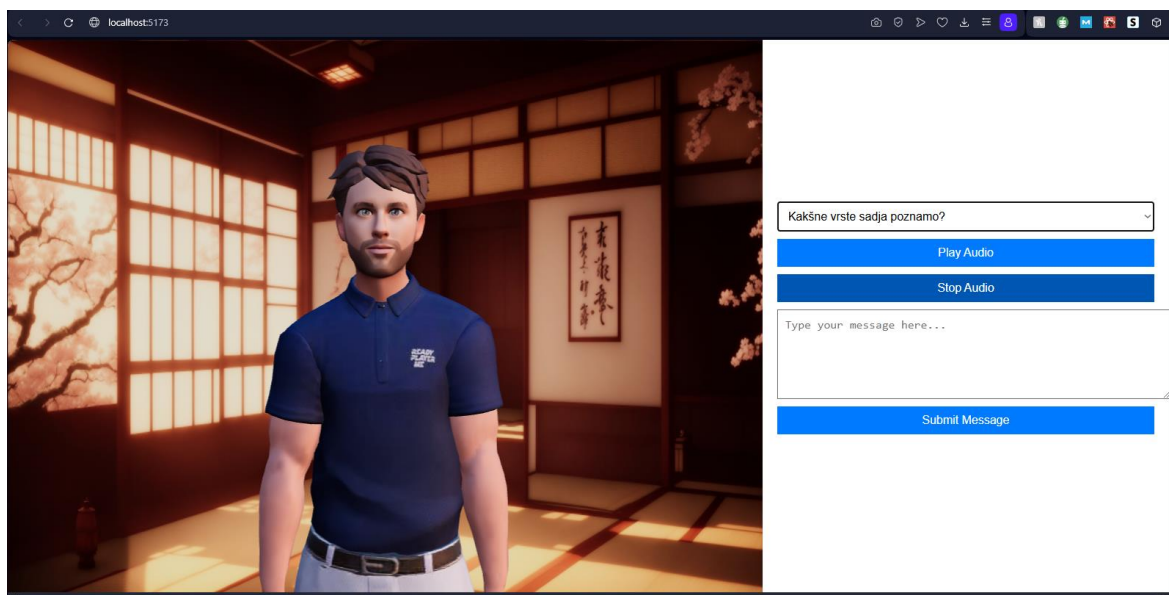
- Enostavna uporaba – Uporabniki lahko ustvarijo avatar v nekaj sekundah;
- Združljivost – Podpira več kot 900 aplikacij in platform;
- API in SDK podpora – Enostavna integracija za razvijalce;
- Optimizacija za VR in AR – Avatarji so lahki in delujejo dobro v realnem času;
- Brezplačna uporaba – Osnovna različica Ready Player Me je brezplačna;

Kljub številnim prednostim pa ima platforma tudi nekaj omejitev:

- Omejene možnosti prilagajanja – Čeprav ponuja številne prilagoditve, ne omogoča popolne svobode pri ustvarjanju unikatnih modelov;
- Odvisnost od interneta – Platforma temelji na oblaku, kar pomeni, da je potrebna povezava za dostop do avatarjev;
- Manj primerno za fotorealistične igre – Avatarji so stilizirani in niso primerni za hiperrealistična okolja;

15 Govoreči virtualni avatar

15.1 Izgled in funkcionalnosti spletne aplikacije



Slika 4 Uporabniški vmesnik naše spletne aplikacije

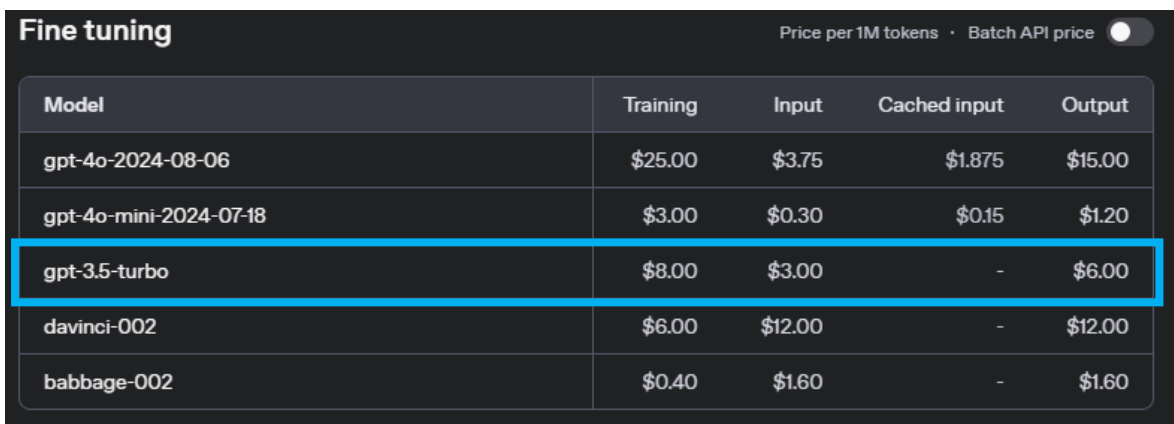
Slika prikazuje naš uporabniški vmesnik. Viden je 3D avatar z ozadjem, prostor za vpisovanje vprašanja ter zgodovina že odgovorjenih zahtev. Uporabnik lahko komunicira z avatarjem, kot smo planirali. V okvirček za vprašanje vnese svojo zahtevo, nato počaka, da avatar pridobi odgovor in začne govoriti. Ko uporabnik prejme odgovor, se ta shrani in je pripravljen za ponovno predvajanje, če želi pa lahko predvajanje tudi, kadarkoli prekine s klikom na gumb »Stop audio«.

15.2 Evalvacija rezultatov

15.2.1 Razmerje med ceno in zanesljivostjo v primerjavi s pravo osebo

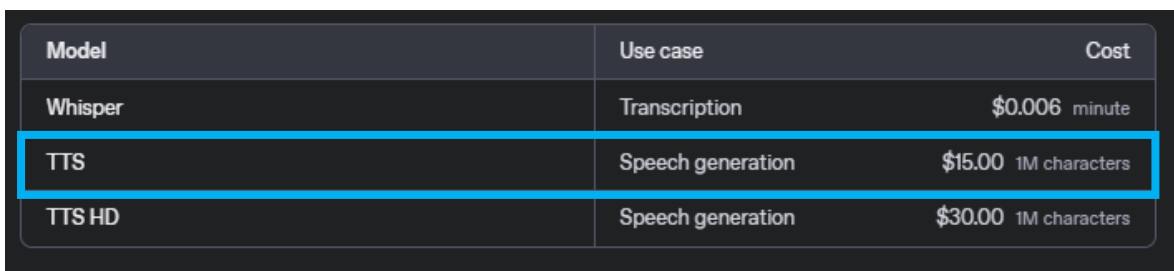
Na podlagi naših izračunov in odgovorov, ki smo jih prejeli od podjetij, smo prišli do ugotovitev, da so v podjetjih zaposleni, ki zgolj odgovarjajo na vprašanja uporabnikov, plačani okoli 1200 € mesečno.

Z našim avatarjem, se mesečni stroški občutno zmanjšajo. Za vhodne podatke preko tekstovnega API modela, OpenAI računa 0,0003 centa na črko, za izhodne pa 0,0006 centa. Izgovorjava pa je malce dražja, in sicer je cena preko glasovnega API modela 0,0015 centa na izgovorjeno črko. Če za primer vzamemo 50.000 znakov, kar je nekaj tisoč več kot jih vsebuje naša raziskovalna naloga, je cena za tekstovne izhodne črke 30 centov, za na glas prebrane pa le 75 centov.



Model	Training	Input	Cached input	Output
gpt-4o-2024-08-06	\$25.00	\$3.75	\$1.875	\$15.00
gpt-4o-mini-2024-07-18	\$3.00	\$0.30	\$0.15	\$1.20
gpt-3.5-turbo	\$8.00	\$3.00	-	\$6.00
davinci-002	\$6.00	\$12.00	-	\$12.00
babbage-002	\$0.40	\$1.60	-	\$1.60

Slika 5 Cenik tekstovnega API modela



Model	Use case	Cost
Whisper	Transcription	\$0.006 minute
TTS	Speech generation	\$15.00 1M characters
TTS HD	Speech generation	\$30.00 1M characters

Slika 6 Cenik glasovnega API modela

15.2.2 Razlikovanje med umetno inteligenco in človekom

Testirali smo 50 dijakov tako, da je vsak moral opraviti nekaj interakcij s pravo osebo na drugi strani in nekaj, kjer so se pogovarjali z UI. V 98 % so dijaki pravilno prepoznali, ali pogovor poteka s pravo osebo, ali ne, kar je bilo z naše strani predvidevano.

15.2.3 Zanimivost v primerjavi s pravo osebo

Isti dijaki, ki so se pogovarjali z ljudmi in umetno inteligenco, so na koncu preizkusili še našo spletno aplikacijo. Ko je vsak postavil kakšno vprašanje in poslušal nekaj odgovorov, smo jih prosili, da podajo svoja mnenja. In sicer na vprašanje »Ali vam je bil pogovor z našim interaktivnim avatarjem bolj zabaven in zanimiv kot dopisovanje s pravo osebo?« so v 90 % odgovorili z »da«, 10 % pa je bilo mnenja, da se jim ni zdel nič kaj bolj zanimiv.

16 Zaključek

16.1 Povzetek doseženih ciljev

Naša raziskovalna naloga je dosegla zastavljene cilje, torej sprejemanje vprašanja uporabnika in vrnitev odgovora nanj v zvočni obliki z usklajenim premikanjem avatarjevih ust.

16.2 Razprava o rezultatih in njihov pomen

Rezultati potrjujejo uspešno integracijo TTS in tekstovnega API v našo spletno aplikacijo. S tem smo ustvarili »delavca«, ki deluje 24/7 in rešuje več nalog hkrati, kar bi lahko zmanjšalo število delovnih mest v podpori strankam, a tudi omogočilo višje plače preostalih zaposlenih ali pa vlaganje v razvoj podjetja.

16.3 Možnosti za nadaljnje izboljšave

Za prihodnost naše spletne aplikacije imamo v načrtu izboljšan uporabniški vmesnik ter nekaj dodatnih funkcij, ki bi bile zelo koristne za širšo in množično uporabo. Ena izmed glavnih je podatkovna baza, iz katere bi umetna inteligenca črpala podatke. Poimenovali bi jo »baza znanja« in delovala bi tako, da bi uporabnik, ki bi našega avatarja želel integrirati v svojo spletno stran, lahko vanjo vnesel podatke in besedila, iz katerih bi želel, da UI črpa odgovore.

To bi bilo koristno predvsem za podjetja, ki bi tako lahko nadomestila njihove agente za podporo uporabnikom. Dodali bi tudi zgodovino pogovorov in možnost, da uporabnik odgovor prejme tako v zvočni kot pisni obliki.

17 Ovrednotenje hipotez

Hipotezo 1, ki trdi, da je uporaba takšnega orodja za odgovarjanje na vprašanja, je cenovno ugodnejša in vsaj enako zanesljiva kakor oseba, ki bi jo plačevali za to, lahko potrdimo.

Z OpenAI cenikom za njihove API storitve smo lahko dokazali, da je tovrstna pot bolj ugodna.

Hipotezo 2, ki trdi, da je na trgu zelo malo proizvajalcev oz. ponudnikov, ki bi uporabljali tovrstno storitev ali pa jo ponujali, lahko potrdimo.

Na internetu nismo zasledili nobene spletne strani, ki bi ponujala to, kar imamo mi. Ponujajo storitve kot so 3D avatar, ki govori besedilo, ki ga poda uporabnik. To je zelo koristno za snemanje videov ali oglasov, vendar ni nikogar, ki bi ponujal avatarja, ki v realnem času odgovarja na uporabnikova vprašanja oziroma zahteve.

Hipotezo 3, ki trdi, da ljudje hitro prepoznajo, ali se z njimi pogovarja človek, ali pa jim odgovarja umetna inteligenca, lahko potrdimo.

Rezultati prve ankete so bili, kot že prej omenjeno, 98 % v prid naši hipotezi.

Hipotezo 4, ki trdi, da bo ljudem pogovor z interaktivnim avatarjem bolj zanimiv kot dopisovanje s pravo osebo, lahko potrdimo.

Odgovori na našo drugo anketo dokazujejo, da je v 90 % dijakom pogovor z avatarjem ostal bolj v spominu kot pa pogovor s pravo osebo.

18 Viri in literatura

OpenAI dokumentacija o ChatGPT API. Pridobljeno 24. november 2024 iz

<https://platform.openai.com/docs/>

OpenAI TTS API dokumentacija za sintezo govora. Pridobljeno 9. februarja 2025 iz

<https://openai.com/research/text-to-speech>

OpenAI API cenik. Pridobljeno 10. februarja 2025 iz

<https://platform.openai.com/docs/pricing>

React Three Fiber dokumentacija. Pridobljeno 24. novembra 2024 iz

<https://docs.pmnd.rs/react-three-fiber>

Rhubarb Lip Sync. Pridobljeno 24. novembra 2024 iz

<https://github.com/DanielSWolf/rhubarb-lip-sync>

FFmpeg dokumentacija. Pridobljeno 25. novembra 2024 iz

<https://ffmpeg.org/documentation.html>

Ready Player Me dokumentacija. Pridobljeno 25. novembra 2024 iz

<https://docs.readyplayer.me/ready-player-me>

Node.js dokumentacija. Pridobljeno 25. novembra 2024 iz

<https://nodejs.org/en/docs>

Yarn dokumentacija. Pridobljeno 25. novembra 2024 iz

<https://yarnpkg.com/getting-started>

WebStorm dokumentacija. Pridobljeno 25. novembra 2024 iz

<https://www.jetbrains.com/webstorm/documentation/>

HTML dokumentacija. Pridobljeno 28. novembra 2024 iz

<https://en.wikipedia.org/wiki/HTML>

CSS dokumentacija. Pridobljeno 28. novembra 2024 iz

<https://en.wikipedia.org/wiki/CSS>

JavaScript dokumentacija. Pridobljeno 28. novembra 2024 iz

<https://en.wikipedia.org/wiki/JavaScript>

Cassell, J., Sullivan J., Prevost, S., Churchill, E. (2000) – Embodied Conversational Agents, Cambridge, MA: MIT Press. Pridobljeno 28. novembra 2024 iz

<https://www.media.mit.edu/gnl/pubs/AI.magazine.cassell.PDF>

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Amodei, D. (2020) - Language Models are Few-Shot Learners. Pridobljeno 28. novembra 2024 iz

<https://arxiv.org/abs/2005.14165>

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Wu, Y. (2018) - Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. Pridobljeno 28. novembra 2024 iz

<https://dl.acm.org/doi/10.1109/ICASSP.2018.8461368>