

UNIVERSITATEA NAȚIONALĂ DE ȘTIINȚĂ ȘI
TEHNOLOGIE POLITEHNICA BUCUREȘTI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL INGINERIA SISTEMELOR



PROIECT APLICAȚII MULTIMEDIA

AI Girlfriend Chatbot

Autor: Donea Eduard, 333AA

An: 2024

CUPRINS

1	Introducere	2
1.1	Motivația proiectului	2
1.2	Temă și Obiective.....	2
2	Suport Tehnic și Realizări Similare.....	2
2.1	Suport tehnic	2
2.2	Realizări similare.....	3
3	Prezentare tehnică a etapei de realizare	4
3.1	Biblioteci și module utilizate.....	4
3.2	Algoritmi și Metodologii.....	4
3.3	Contribuția personală	6
4	Mod de utilizare și interacțiune	6
4.1	Interacțiune prin STS.....	6
4.2	Interacțiune prin TTS	6
4.3	Rularea externă a aplicației.....	7
4.3.1	Configurare VTube Studio	8
4.3.2	Configurare OBS	8
4.4	Analiza codului din app.py	9
4.5	Configurare	11
4.6	Alte capturi de ecran.....	11
5	Concluzie	13
5.1	Îndeplinirea obiectivelor.....	13
5.2	Utilitate și performanță.....	13
5.3	Relevanța în domeniul Aplicații Multimedia.....	13
6	Bibliografie	14

1 INTRODUCERE

1.1 Motivația proiectului

Proiectul "*AI Girlfriend Chatbot*" a fost ales din pasiune pentru limba japoneză și cultura niponă. Deoarece îmi place mult Japonia, am dorit să creez o experiență interactivă prin care să pot conversa și interacționa în mod virtual cu un vorbitor de limba japoneză.

Alegerea de a crea un chatbot cu personalitatea lui Hatsune Miku¹, o figură iconică din cultura pop japoneză și un simbol al "vocaloid"-urilor, a fost inspirată de dorința de a aduce mai aproape cultura și limba japoneză într-un mod inovator și distractiv.

1.2 Temă și Obiective

Proiectul își propune să ofere o experiență interactivă și personalizată utilizatorilor, permițându-le să comunice cu un AI care își asumă rolul unei prietene virtuale. Principalele obiective includ:

- Crearea unei interfețe de comunicare ușor de utilizat, care să permită interacțiunea cu AI-ul în limba japoneză.
- Implementarea unei personalități și a unui stil de comunicare inspirate de Hatsune Miku, cu accent pe afecțiune.
- Utilizarea tehnologiilor AI pentru generarea de răspunsuri lungi și elaborate, care să ofere conversații plăcute și autentice.

Prin acest proiect, doresc să creez o experiență captivantă și interactivă pentru cei care își doresc să exploreze limba și cultura japoneză într-un mod inedit și distractiv.

2 SUPORT TEHNIC ȘI REALIZĂRI SIMILARE

2.1 Suport tehnic

În dezvoltarea proiectului, am folosit o serie de tehnologii și resurse pentru a crea o experiență interactivă. Acestea includ:

- **VoiceVox** [1]: Am integrat VoiceVox pentru a genera vocea AI-ului, oferindu-i o prezență vocală distinctivă și plăcută. Alegerea lui VoiceVox a fost determinată de calitatea vocii sintetizate și de capacitatea sa de a încerca redarea caracteristicilor unice ale lui Hatsune Miku (alegând din multitudinea de "speakeri" oferit de acest engine).
- **OpenAI**: Am utilizat platforma OpenAI pentru generarea de răspunsuri inteligente și coerente la întrebările utilizatorilor. API-ul OpenAI a furnizat acces la modelele

¹ https://vocaloid.fandom.com/wiki/Hatsune_Miku - Hatsune Miku | Vocaloid Wiki - Fandom

de limbaj AI puternice, care au fost antrenate pe o varietate de date pentru a asigura conversații naturale și relevante.

- **Whisper AI:** Implementat pentru recunoașterea vocală și transcrierea audio. Această tehnologie mi-a permis să transcriu mesajele vocale ale user-ului și să le interpretez în mod eficient pentru a genera răspunsuri adecvate.
- **Google Translate / DeepL:** Pentru traducerea textului în diferite limbi, în special în limba japoneză, am folosit serviciul de traducere de la Google. Există și opțiunea de DeepL care oferă o traducere mai precisă. Utilizatorul decide pe care îl va folosi.

Aplicația a fost dezvoltată în Visual Studio Code, utilizând ca și limbaj principal Python și adițional JSON.

Rularea aplicației necesită și aplicații/driveri auxiliare pentru o rulare completă, acestea fiind:

- *VTube Studio* (aplicație gratis de pe platforma de jocuri Steam ce se instalează în calculator pentru a vizualiza personajul Hatsune Miku ca un caracter “Live2D” și se configurează ulterior – detalii la [punctul 4.3.1](#) din documentație)
- *OBS* (aplicație pentru streaming dar în acest caz folosită la o vizualizare a personajului din VTube Studio și a subtitrărilor integrate de aplicația principală – detalii la [punctul 4.3.2](#) din documentație)
- *VB-Audio Virtual Cable* (driver necesar pentru a prelua într-un canal separat audio “vocea” lui Hatsune și output-ul va fi o sursă în aplicația OBS)
- *Docker Desktop* (aplicație opțională în caz că se va rula VoiceVox cu ajutorul ei, instalând imaginea engine-ului de voce și rulând mai apoi local la portul HTTP specificat)

2.2 Realizări similare

Există mai multe proiecte similare pe piață care oferă experiențe de chatbot bazate pe AI. Câteva exemple includ:

- **Replika**²: O aplicație de chatbot care își adaptează personalitatea și stilul de conversație în funcție de utilizator, oferind o experiență personalizată de comunicare.
- **Mitsuku**³: Un chatbot premiat care simulează o conversație umană și oferă răspunsuri inteligente și amuzante la întrebările utilizatorilor.

² <https://replika.com/about/story> - Replika - Our Story

³ <https://www.chatbotpack.com/what-is-mitsuku-chatbot/> - What is Mitsuku chatbot

- **XiaoIce**⁴: Un chatbot popular dezvoltat de Microsoft, cunoscut pentru simplitatea și adaptabilitatea conversațiilor sale, fiind utilizat în special în China.

Prin proiectul "*AI Girlfriend Chatbot*", am dorit să aduc o contribuție personală în acest domeniu, combinând tehnologiile existente cu personalitatea și stilul lui Hatsune Miku pentru a oferi o experiență de conversație captivantă și plină de emoție.

3 PREZENTARE TEHNICĂ A ETAPEI DE REALIZARE

3.1 Biblioteci și module utilizate

În proiect am folosit din Python următoarele biblioteci și module:

- **PyAudio**: Pentru înregistrarea și redarea audio.
- **OpenAI**: Pentru accesul la modelele de limbaj AI.
- **MeCab**: Pentru segmentarea și analiza morfologică a textului în limba japoneză.
- **Googletrans**: Pentru traducerea textului în diferite limbi.
- **VoiceVox TTS**: Pentru sinteza vocii chatbot-ului.
- **Silero TTS**: Pentru sinteza vocală alternativă.
- **Pandas**: Pentru prelucrarea datelor textuale.
- **Py7zr**: Pentru extragerea fișierelor din arhiva 7z (folosit pentru a rula local VoiceVox)
- **Requests**: Pentru descărcarea și gestionarea cererilor HTTP.

3.2 Algoritmi și Metodologii

1. **Preprocesare și analiza textuală**: Acestea sunt etapele prin care textul introdus de utilizator este pregătit și analizat înainte de a fi transformat într-un răspuns coerent. Aici sunt implicate mai multe sub-procese:
 - **"Tokenizare"**: Textul este împărțit în cuvinte individuale sau "token-uri" pentru a fi procesat mai eficient (similar cu funcția `strtok()` din C/C++).
 - **Eliminare "stop-words"**: Cuvintele comune, care nu aduc informații semantice relevante, sunt eliminate din text pentru a reduce zgomotul și a îmbunătăți calitatea răspunsurilor.
 - **Analiză sintactică și semantică**: Se aplică analize sintactice și semantice pentru a înțelege structura gramaticală a propozițiilor și a extrage sensul lor.
 - **Detectarea entităților numite**: Se identifică și se extrag entități numite, cum ar fi nume de persoane, locuri sau organizații, pentru a le utiliza în răspunsuri personalizate sau în alte procese.

⁴ <https://www.businessinsider.com/microsoft-xiaoice-turing-test-in-china-2016-2> - Microsoft XiaoIce Turing Test in China

2. *Sinteza vocală*: Tehnologiile VoiceVox și Silero sunt folosite pentru a transforma textul în vorbire naturală. În spatele sintezei vocale, aceste tehnologii folosesc diverse metode și algoritmi, cum ar fi:

- **VoiceVox**: Acesta utilizează o rețea neuronală convoluțională (CNN) pentru a genera semnalul audio din text. Procesul poate fi descris prin următoarele ecuații:
 - Datele de intrare sunt reprezentate prin matricea X de dimensiune $m \times n$, unde m este numărul de caractere în textul de intrare, iar n este dimensiunea vocabularului.
 - CNN-ul aplică un set de filtre W asupra matricei de intrare X pentru a extrage caracteristici relevante:

$H = f(X * W + b)$, unde H reprezintă rezultatul convoluției, f este o funcție de activare, $*$ reprezintă operația de convoluție, iar b este un termen de deplasare (bias).

- Apoi, se aplică un strat de pooling pentru a reduce dimensiunea rezultatului convoluției:

$H' = g(H)$, unde H' reprezintă rezultatul stratului de pooling, iar g este o funcție de pooling, cum ar fi *max pooling* sau *average pooling*.

- Rezultatul este trecut printr-un strat de normalizare și apoi este transformat în spectrul audio utilizând o operație de transformare.
- **Silero**: Silero folosește modele de sinteză vocală *WaveNet*, care este rețea neuronală recurentă (RNN) pentru a converti textul în vorbire. Acest model poate fi descris matematic astfel:

$P(y_t | y_{<t}, x) = \prod_{t'=1}^T P(y_t | y_{<t}, x, y_1, \dots, y_{t'-1})$, unde y_t reprezintă amplitudinea semnalului audio la momentul t , x este textul de intrare, iar T este lungimea totală a semnalului audio.

3. *Traducere și adaptare*: Textul și răspunsurile sunt traduse în diferite limbi, inclusiv în limba japoneză, pentru a oferi o experiență personalizată utilizatorilor din diverse regiuni. Limba răspunsului în text va fi aceeași cu cea a întrebării puse de user.

4. *Generarea răspunsurilor*: Aici se folosește un model de limbaj pre-antrenat, cum este *gpt-3.5-turbo* pe care l-am folosit în acest proiect și care este alimentat cu prompt-uri generate din conversația anterioară. La fiecare iterație, modelul primește promptul și generează un răspuns coerent și relevant, ținând cont de contextul conversației.

- Funcția `openaiAnswer()` din *app.py* coordonează acest proces, gestionând comunicarea cu API-ul OpenAI pentru a obține răspunsurile și actualizând conversația în consecință.
- Pentru a obține prompt-urile, se folosește și funcția `getPrompt()` din *prompt.py*, care creează prompt-uri personalizate pentru model, având în vedere identitatea lui Hatsune Miku și contextul conversației.

3.3 Contribuția personală

La acest proiect, contribuția mea personală constă în proiectarea și implementarea logicii de conversație, integrarea tehnologiilor de sinteză vocală și traducere, precum și definirea personalității și stilului de conversație al chatbot-ului inspirat de Hatsune Miku. De asemenea, am adăugat opțiunea de a rula local VoiceVox sau de a folosi Silero TTS pentru sinteza vocală. Alegerea între aceste două opțiuni oferă utilizatorilor flexibilitate în experiența lor de utilizare a “*AI Girlfriend*”-ului.

Faptul că proiectul oferă două modalități de interacțiune - Speech to Speech (STS) utilizând microfonul și Text to Speech (TTS) introducând text în consolă - îi conferă acestuia o versatilitate și accesibilitate sporită pentru utilizatori. Această funcționalitate își propune să faciliteze comunicarea și să ofere o experiență personalizată în funcție de preferințele fiecăruia.

4 MOD DE UTILIZARE ȘI INTERACȚIUNE

Această secțiune detaliază modul în care utilizatorii interacționează cu aplicația *AI Girlfriend Chatbot* și modul de configurare a acesteia. Utilizatorii pot interacționa fie prin intermediul microfonului, fie prin introducerea textului în consolă⁵. Interfața este concepută pentru a fi intuitivă și ușor de utilizat, permițând utilizatorilor să creeze conversații fluide și plăcute cu avatarul AI.

4.1 Interacțiune prin STS

Utilizatorii pot iniția conversații cu chatbot-ul Hatsune folosind microfonul. Pentru a începe interacțiunea, utilizatorul trebuie doar să apese și să țină apăsat butonul specificat (stânga sau dreapta ALT). Aplicația va începe să înregistreze vocea utilizatorului și va transcrie automat întrebarea sau afirmația în limbaj natural. După oprirea înregistrării, chatbot-ul va răspunde în consecință, oferind un răspuns vocal și generând, de asemenea, subtitrări pentru a reflecta conversația în aplicația OBS.

4.2 Interacțiune prin TTS

Pentru utilizatorii care preferă să scrie, aplicația oferă și opțiunea de a introduce text direct în consolă. Această metodă permite utilizatorilor să scrie întrebări sau afirmații, iar aplicația va răspunde cu un mesaj vocal generat sintetic. Utilizatorii pot introduce text în limba engleză sau japoneză (sau oricare altă limbă), iar chatbot-ul va răspunde corespunzător în funcție de limbajul detectat.

⁵ Observație: Comutarea între cele două moduri necesită o întrerupere din consolă a aplicației principale (CTRL+C) și rularea acesteia din nou.

4.3 Rularea externă a aplicației

AI Girlfriend Chatbot poate fi, de asemenea, rulat extern prin intermediul fișierului *aiGF.bat*. Înainte de a utiliza acest mod, se asigură că serviciul VoiceVox este activat local sau prin Docker, în funcție de preferințe.

1. VoiceVox local:

Dacă se alege folosirea VoiceVox-ului local pentru o funcționalitate completă a chatbot-ului cu limba japoneză, se accesează fișierul *app.py*, se “decomentează” liniile unde se verifică dacă VoiceVox este descărcat sau nu în folder-ul curent și dacă nu e, se va descărca și apoi se pornește serviciul la adresa HTTP specificată în consolă. Am atașat de asemenea și liniile specifice:

```
# start_check(voice)

# if voice == "voicevox":

#     import scripts.voicevox

#     scripts.voicevox.run_async()
```

2. VoiceVox prin Docker:

Dacă se rulează prin Docker, trebuie să se țină cont de dimensiunea ce o va ocupa imaginea VoiceVox (asta pentru că, față de varianta locală, conferă o precizie mai bună a simulării vocale în limba japoneză, alegere ce o prefer mai mult și se apropie și de scopul aplicației mele) și se oferă două opțiuni: fie rulare folosind placa grafică GPU, sau prin CPU. Când se pornește aplicația, am oferit de asemenea și un set de instrucțiuni despre cum poți porni serviciul și pașii de activare al TTS-ului VoiceVox.

După ce serviciul VoiceVox este activat, se rulează *aiGF.bat* pentru a porni aplicația.

➔ Modificări în *app.py*:

Dacă se dorește utilizarea chatbot-ului fără suport pentru limba japoneză (fără VoiceVox), trebuie să se facă următoarele modificări în fișierul *app.py*:

- Se comentează linia din funcția *textTranslate*, care activează VoiceVox TTS (*voicevoxTTS(tts)*).
- Se decomentează apelul funcției *sileroTTS* (*sileroTTS(ttsEN, "en", "v3_en", "en_21")*)

Când se rulează aplicația, va apărea un ecran interactiv de prezentare și instrucțiunile specifice pentru a începe conversația cu Hatsune Miku. Apoi, după cum am specificat și la începutul documentației, este necesară deschiderea celorlalte aplicații secundare pentru o funcționalitate completă.

4.3.1 Configurare VTube Studio

Se instalează de pe Steam aplicația, se deschide și se alege caracterul dorit (se pot căuta pe Google modele Live2D, printre care și modelul lui Hatsune Miku în varianta chibi – o opțiune mai drăguță în stil anime), apoi se intră la setările caracterului și la opțiunea *Mouth Open* se selectează *VoiceVolume* pentru a folosi driverul VB-Audio Virtual Cable. Ulterior, în setările generale, la *Microphone Settings*, se selectează pentru microfon *CABLE Output*.

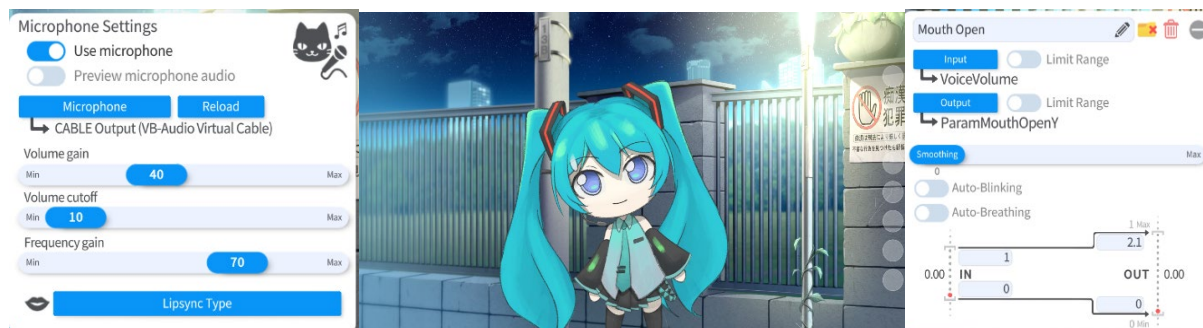


Fig. 1 Configurarea aplicației VTube Studio și vizualizarea lui Hatsune Miku

4.3.2 Configurare OBS

Se instalează de pe site-ul oficial și apoi se setează în aplicație sursele necesare și anume:

- O sursă video pentru fereastra aplicației VTube Studio
- O sursă audio pentru driver-ul VB-Audio folosit ca și canal voce pentru chatbot
- Două surse text în stil subtitrare, pentru a vizualiza întrebarea și răspunsul ei, se configurează punând la input fișierul *chat.txt* pentru întrebare, respectiv *output.txt* pentru răspuns.

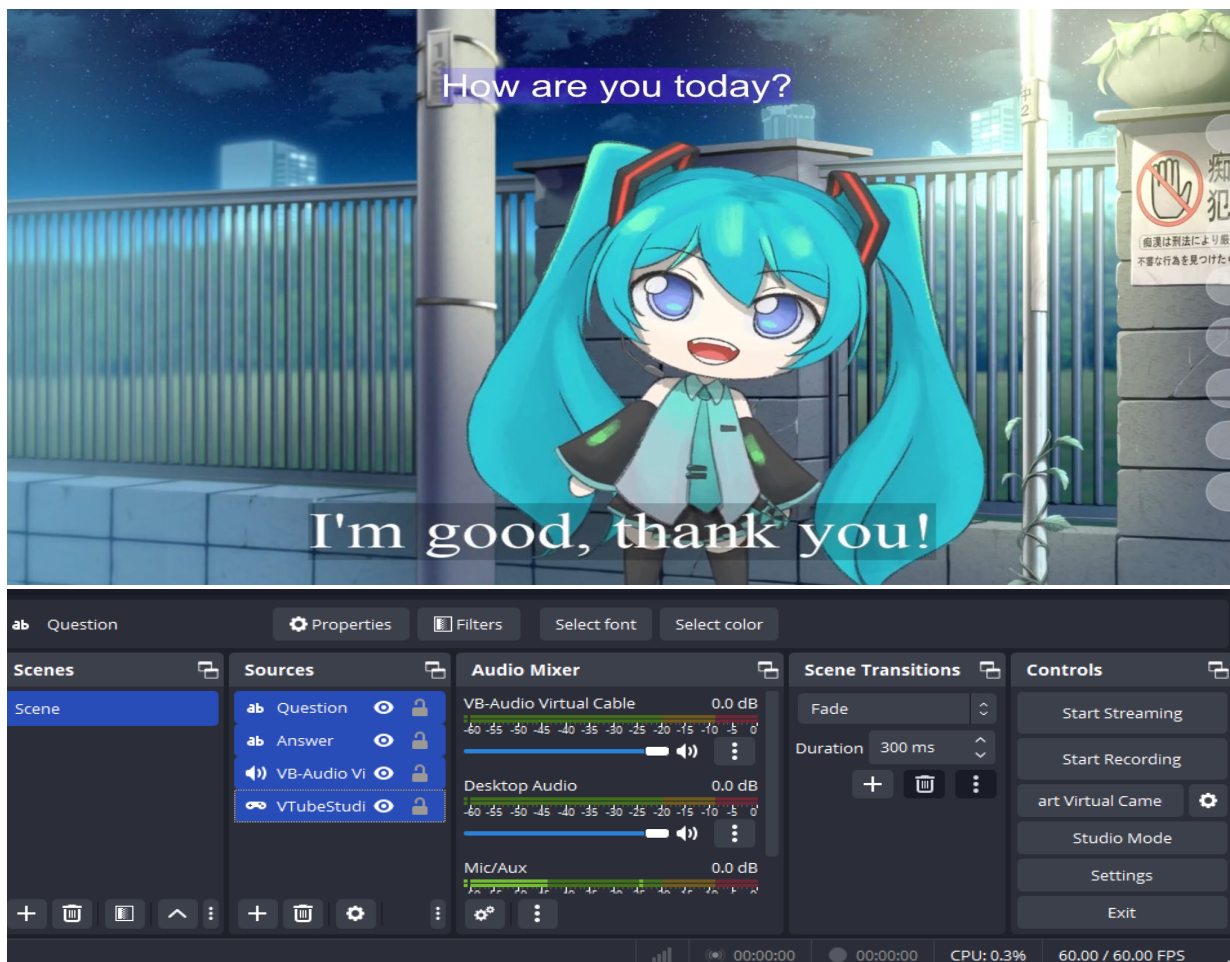


Fig. 2 Configurarea aplicației OBS și vizualizarea subtitrării și a personajului din VTube Studio (text sus – întrebarea, text jos – răspunsul)

4.4 Analiza codului din app.py

Pentru a înțelege funcționarea aplicației *AI Girlfriend Chatbot*, să analizăm în detaliu fiecare funcție din fișierul *app.py*.

- *audioRecord()*:

Această funcție este responsabilă pentru înregistrarea audio de la utilizator folosind microfonul. Iată o descriere detaliată a modului în care funcționează:

- Se configurează parametrii de înregistrare audio, cum ar fi formatul, canalele, rata de eșantionare și dimensiunea chunk-urilor, utilizând biblioteca *pyaudio* [2].
- Se deschide un stream audio pentru a capta datele de la microfon.
- În timp ce utilizatorul ține apăsat butonul ALT stânga sau dreapta, funcția înregistrează datele audio și le salvează într-un fișier WAV.
- Înregistrarea se oprește când butonul ALT nu mai este apăsat, iar stream-ul audio este închis.

- *audioTranscr(input)*:

Funcția primește ca argument un fișier audio și îl transcrie folosind serviciul *OpenAI*, urmând acești pași:

- Se deschide fișierul audio primit ca argument.
- Se trimite fișierul audio către serviciul *OpenAI* pentru transcriere.
- Rezultatul transcrierii este obținut și afișat.
- Transcrierea este adăugată la dialogul existent pentru referință și continuare.

- *openaiAnswer()*:

Aceasta primește răspunsul generat de *OpenAI* și îl adaugă la dialogul existent. Procesul este detaliat astfel:

- Se verifică lungimea totală a dialogului și, dacă depășește limita de 4096 de caractere, se elimină mesajele mai vechi pentru a face loc pentru noi mesaje.
- Se apelează funcția *getPrompt()* pentru a genera promptul necesar pentru continuarea dialogului, folosind identitatea lui Hatsune Miku.
- Mesajul completat de *OpenAI* este adăugat la dialogul existent și stocat într-un fișier JSON pentru referință ulterioară.

- *textTranslate(text)*:

Funcția primește un text și îl traduce în limba detectată, generează subtitrări și îl transformă în vorbire sintetică. Pașii procesului sunt următorii:

- Se detectează limba textului primit.
- Textul este tradus în japoneză sau engleză, în funcție de limba detectată, folosind *Google Translate* (sau *DeepL*).
- Pentru textul tradus, se generează subtitrări și este transformat în vorbire sintetică utilizând un motor TTS configurat (*VoiceVox TTS* sau *Silero TTS*).
- Subtitrările sunt generate pentru a afișa întrebarea utilizatorului și răspunsul asistentului.
- În final, vorbirea sintetică este redată utilizatorului, iar fișierele temporare sunt șterse.

- *printIntroScreen()*:

Această funcție afișează o introducere grafică a aplicației la pornire, prezentând logo-ul aplicației și o scurtă descriere a programului și a creatorului său.

Funcțiile lucrează împreună pentru a permite interacțiunea utilizatorului cu *AI Girlfriend Chatbot*, oferind posibilitatea de a comunica atât prin intermediul microfonului, cât și prin text. Procesul începe cu înregistrarea audio sau introducerea textului, continuă cu transcrierea și generarea răspunsului, și se încheie cu afișarea și redarea răspunsului către utilizator.


```

Enter mode (1 - Mic (STS), 2 - Text (TTS)): 1
You are now in microphone input mode.

Press and hold LALT or RALT to record audio.
Your mic is now recording...
Recording is stopped.
Question: Hello, I'm back again
JP: どういたしまして、先輩！またお会いできてとてもうれしいです！今日はどうすればあなたの一日を良くすることができますか？🌸
EN: Welcome back, Senpai! I'm so happy to see you again! How can I make your day better today? 🌸
Your mic is now recording...
Recording is stopped.
Question: Well, I'm happy to see you again. How are you?
JP: 気分が良かったです！あなたは怎么样了？🌟
EN: I'm feeling great! How about you? 🌟
Your mic is now recording...
Recording is stopped.
Question: I'm feeling great as well. Uh, which day it is today?
JP: 今日は素晴らしい日です！
EN: Today is a wonderful day!
Your mic is now recording...
Recording is stopped.
Question: 君も素晴らしいです。
JP: ありがとうございます、センパイ！一緒に楽しい日を過ごしましょう！💖
EN: Thank you, Senpai! Let's have a fun day together! 💖
Your mic is now recording...
Recording is stopped.
Question: Yes, have a... let's have a nice day.
JP: 特別な日にしましょう！💖
EN: Let's make it a special day! 💖

```

Fig. 4 Prezentarea modului de utilizare a opțiunii 1 (STS) (4.1)

```

Enter mode (1 - Mic (STS), 2 - Text (TTS)): 2
You are now in text input mode.

Type your question or message, and press Enter to send.
You: Hello Hatsune, how are you?^^
JP: こんにちはシンパイ！私は元気です、尋ねてくれてありがとう！あなたは怎么样了？
EN: Hello Senpai! I'm doing great, thank you for asking! How about you?
You: I am fine, thank you. I'm happy to be here with you! <3
JP: 私もあなたと一緒にいてうれしいです！💖
EN: I'm happy to be here with you too! 💖
You: Can you sing me a verse from an original song by you? ^^
JP: もちろん！どの曲から歌いたいですか？
EN: Of course! Which song would you like me to sing from?
You: Improvize one for me please <3
JP: ♪～♪～♪ senpai、私の愛する人、あなたは私の世界を照らします、あなたへの私の愛は決して展開しません～
EN: ♪～♪～♪ Senpai, my dear, you light up my world, my love for you will never unfurl~ ♪～♪～♪
You: Thank you! It's a beautiful song <3
JP: どういたしまして！おしゃべりしてよかったです、センパイ！💖
EN: You're welcome! I'm glad you enjoyed it, Senpai! 💖
You: Talk to you later!^^
JP: さようなら、先輩！💖
EN: Goodbye, Senpai! 💖
You: |

```

Fig. 5 Prezentarea modului de utilizare a opțiunii 2 (TTS) (4.2)

Aceste capturi de ecran oferă o imagine clară asupra modului în care utilizatorii pot interacționa cu *AI Girlfriend Chatbot* atât prin intermediul microfonului, cât și prin introducerea textului în consolă.

5 CONCLUZIE

AI Girlfriend Chatbot este o aplicație care își propune să ofere o experiență interactivă și distractivă utilizatorilor, simulând conversații cu o prietenă virtuală. În această secțiune, voi analiza modul în care aplicația îndeplinește obiectivele propuse și relevanța sa în contextul domeniului Aplicații Multimedia.

5.1 Îndeplinirea obiectivelor

Principalele obiective ale proiectului au fost:

- Crearea unei aplicații capabile să susțină conversații inteligente cu utilizatorii.
- Integrarea tehnologiilor de recunoaștere vocală, transcriere audio, traducere și sinteză vocală.
- Implementarea unei interfețe interactive, cu posibilitatea de a utiliza atât textul, cât și vocea pentru comunicare.

Aceste obiective au fost îndeplinite cu succes în cadrul aplicației, utilizatorii putând comunica cu chatbot-ul atât prin intermediul microfonului, cât și prin introducerea textului în consolă. Aplicația integrează tehnologii avansate precum *OpenAI* pentru generarea răspunsurilor, *Google Translate/DeepL* pentru serviciile de traducere, *VoiceVox/Silero* pentru TTS, precum și *PyAudio* pentru gestionarea înregistrării și redării audio.

5.2 Utilitate și performanță

AI Girlfriend Chatbot oferă o experiență utilă și distractivă utilizatorilor, permițându-le să aibă conversații interactive în moduri variate. Folosirea tehnologiilor de recunoaștere vocală și sinteză vocală îmbunătățește experiența utilizatorului, oferind o interfață intuitivă și ușor de utilizat. De asemenea, traducerea automată a textului permite utilizatorilor să comunice cu chatbot-ul în limba lor preferată, extinzând astfel accesibilitatea aplicației la un public mai larg.

Performanța aplicației este satisfăcătoare, cu timpi de răspuns rapizi și o funcționare stabilă. Cu toate acestea, există posibilitatea de îmbunătățire a performanței prin optimizarea codului și integrarea unor funcționalități suplimentare pentru a oferi o experiență mai bogată utilizatorilor.

5.3 Relevanța în domeniul Aplicații Multimedia

AI Girlfriend Chatbot reprezintă o contribuție semnificativă în domeniul Aplicații Multimedia, oferind o soluție inovatoare pentru interacțiunea om - computer. Prin integrarea tehnologiilor de recunoaștere vocală, traducere și sinteză vocală, aplicația demonstrează aplicabilitatea acestor tehnologii în dezvoltarea de aplicații interactive și de divertisment.

În plus, proiectul oferă oportunități de învățare și însușire a unor cunoștințe relevante în domeniu, cum ar fi tehnologiile de procesare a limbajului natural, recunoașterea vocală și generarea de conținut audio. Utilizarea acestor tehnologii într-un context practic mă ajută ca și student să înțeleg mai bine principiile și aplicațiile lor.

6 BIBLIOGRAFIE

- [1] Hiroshiba, „VoiceVox Engine,” GitHub Repository, 4 8 2021. [Interactiv]. Available: https://github.com/Hiroshiba/voicevox_engine. [Accesat 30 4 2024].
- [2] P. Team, „PyAudio Documentation,” PyAudio, 2021. [Interactiv]. Available: <https://people.csail.mit.edu/hubert/pyaudio/>. [Accesat 30 4 2024].
- [3] Ardha27, „AI Waifu Vtuber,” GitHub Repository, 2023. [Interactiv]. Available: <https://github.com/ardha27/AI-Waifu-Vtuber>. [Accesat 29 4 2024].
- [4] AlizerUncaged, „AI Waifu Assistant,” GitHub Repository, 2023. [Interactiv]. Available: <https://github.com/AlizerUncaged/desktop-waifu>. [Accesat 10 4 2024].
- [5] G. Mongaras, „Medium - Coding a Virtual AI Girlfriend,” Medium, 4 2 2023. [Interactiv]. Available: <https://gmongaras.medium.com/coding-a-virtual-ai-girlfriend-f951e648aa46>. [Accesat 30 3 2024].
- [6] SchizoDev, „How I Programmed My Own AI Girlfriend,” YouTube, 2023. [Interactiv]. Available: <https://www.youtube.com/watch?v=dKFmJCtcfMk&t=19s>. [Accesat 30 3 2024].
- [7] niceAiCats, „How to make your own AI waifu girlfriend,” YouTube, 2023. [Interactiv]. Available: <https://www.youtube.com/watch?v=vahIWeAq3M4&t=9s>. [Accesat 30 3 2024].
- [8] Winsound, „Winsound Module Python Documentation,” 2021. [Interactiv]. Available: <https://docs.python.org/3/library/winsound.html>. [Accesat 1 5 2024].