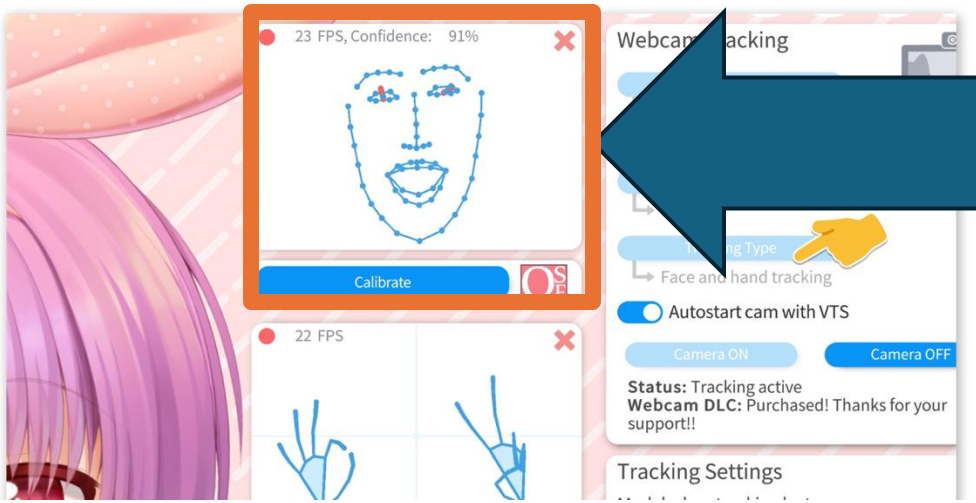
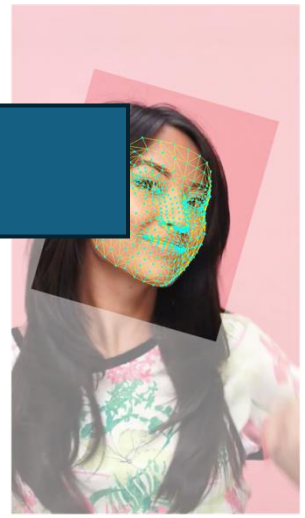
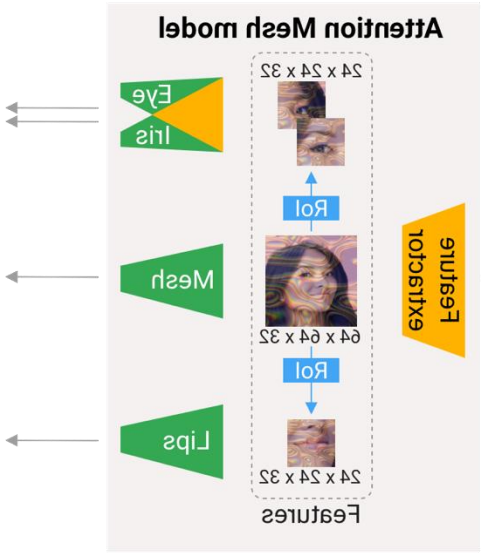
AI 활용 특강
LLM for ~~AI~~
~~Vtuber(2)~~

구현해야 하는 것들

Devlop AI Vtuber target

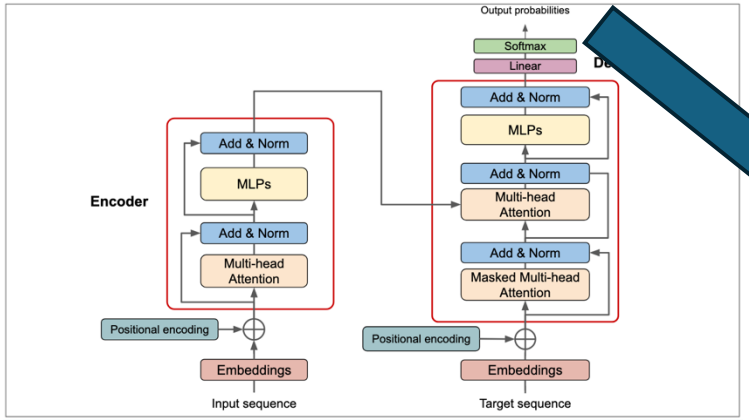




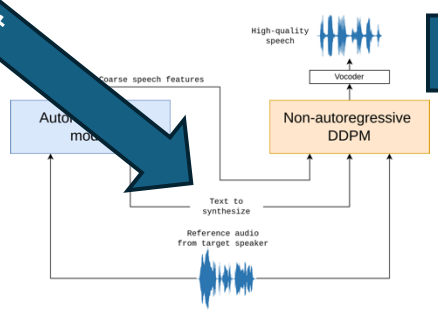


Media pipe

Virtual camera



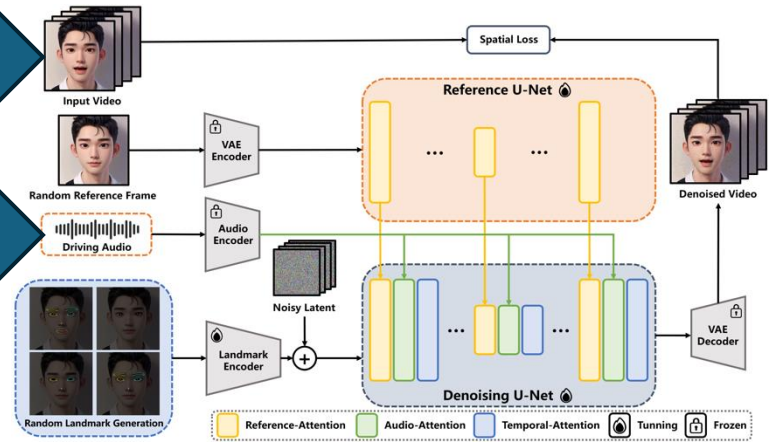
text



Audio-Driven + OuteTTS

Input face img

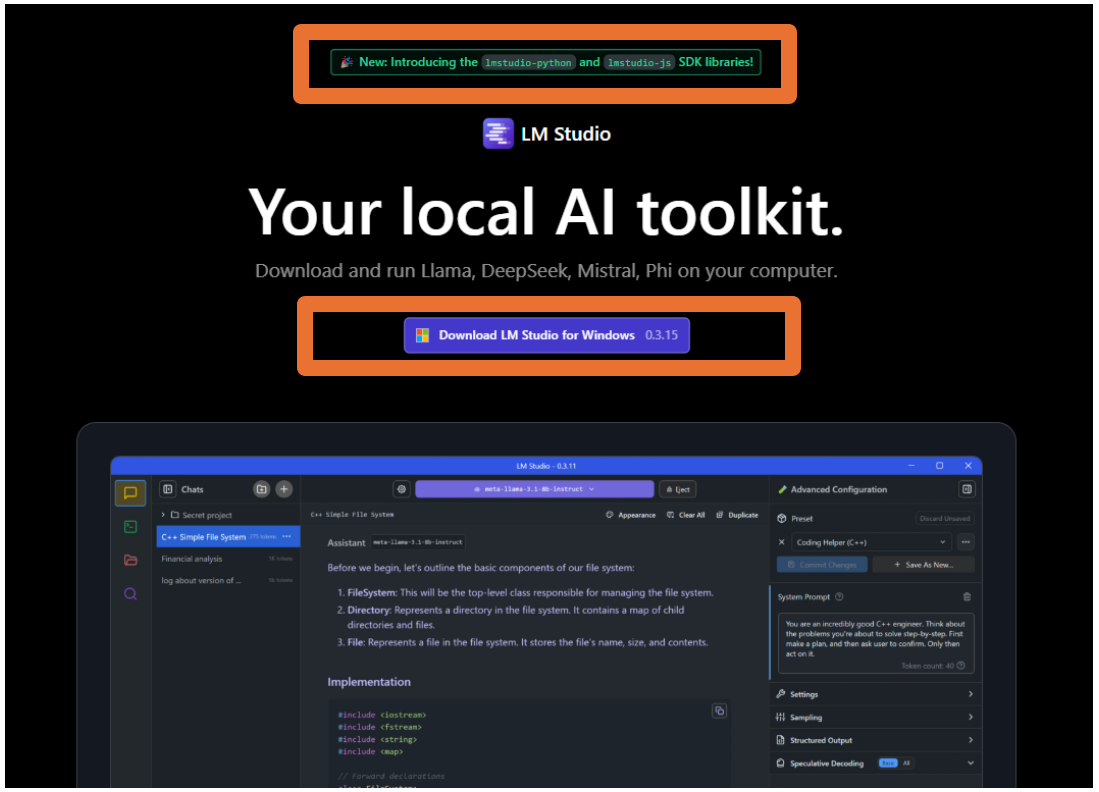
Audio



실습해보기

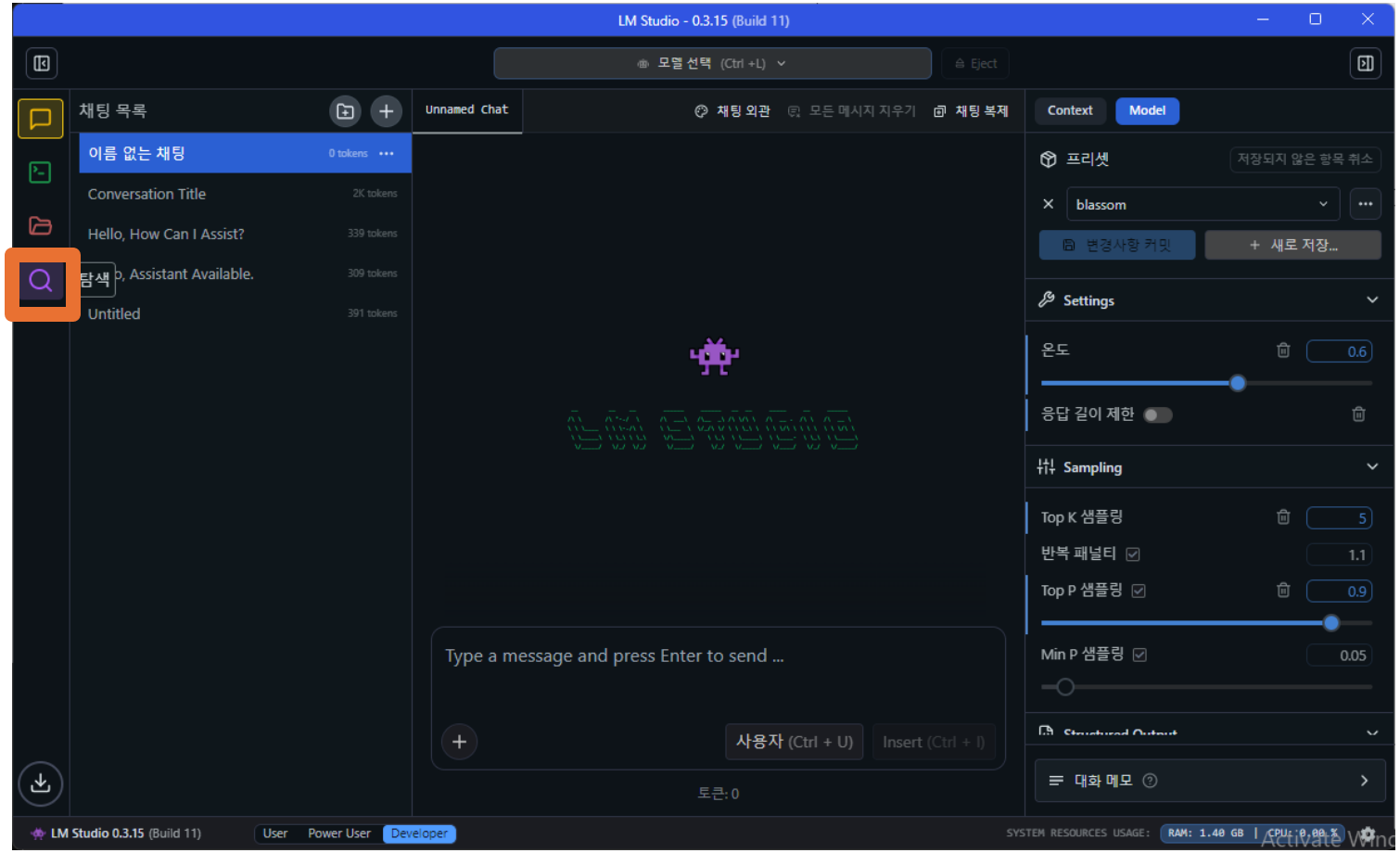
버튜버 만들기 위해 필요한 기능 구동해보기

Running LLM for GUI Envorment

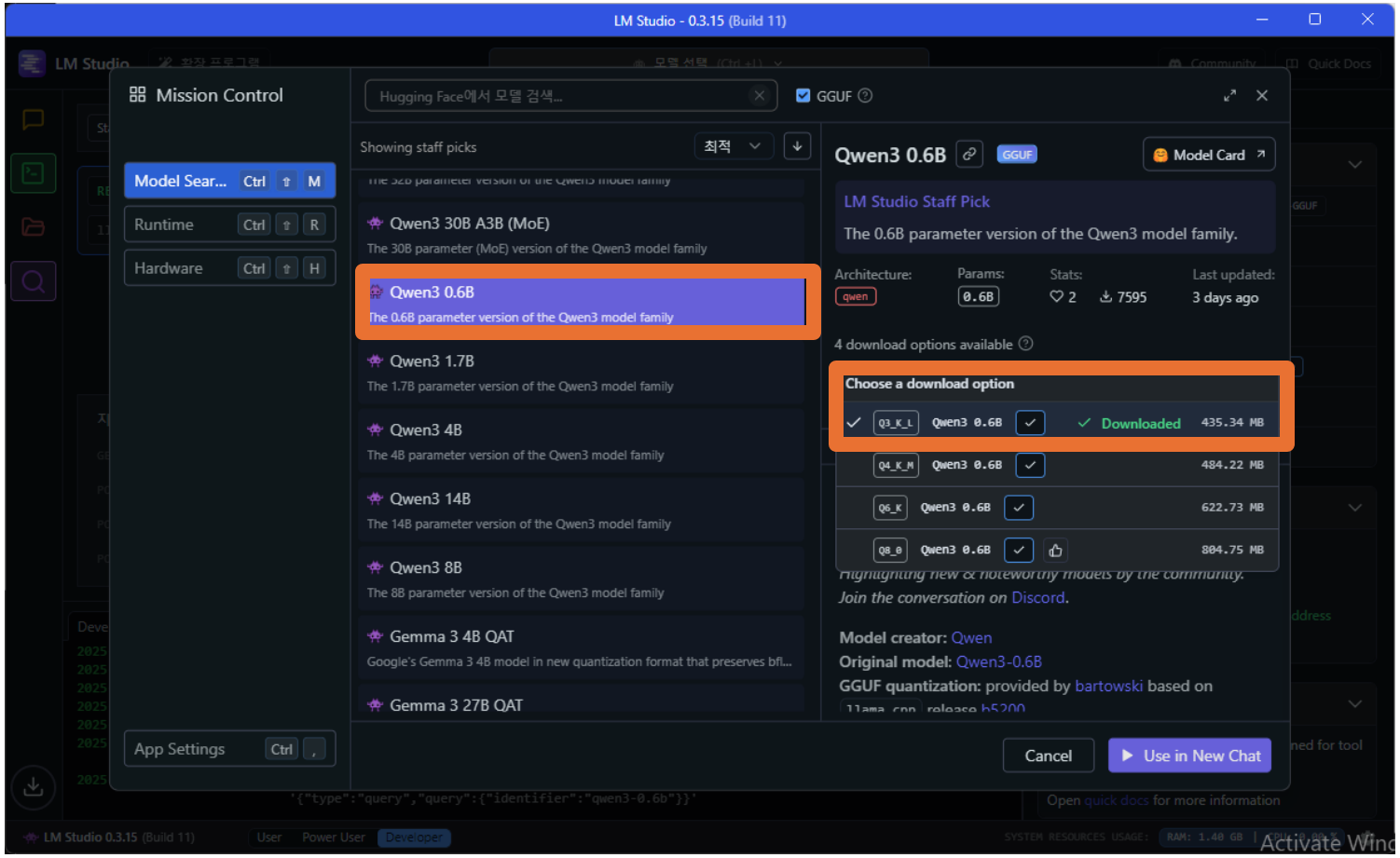


- GUI 환경에서 LLM On Device 구동 가능.
- (진짜 따끈따끈한)최근 python API 쉽게 불러 올 수 있게 만들어져 있음.
  - API 관련 문서는 여기로

Download Model

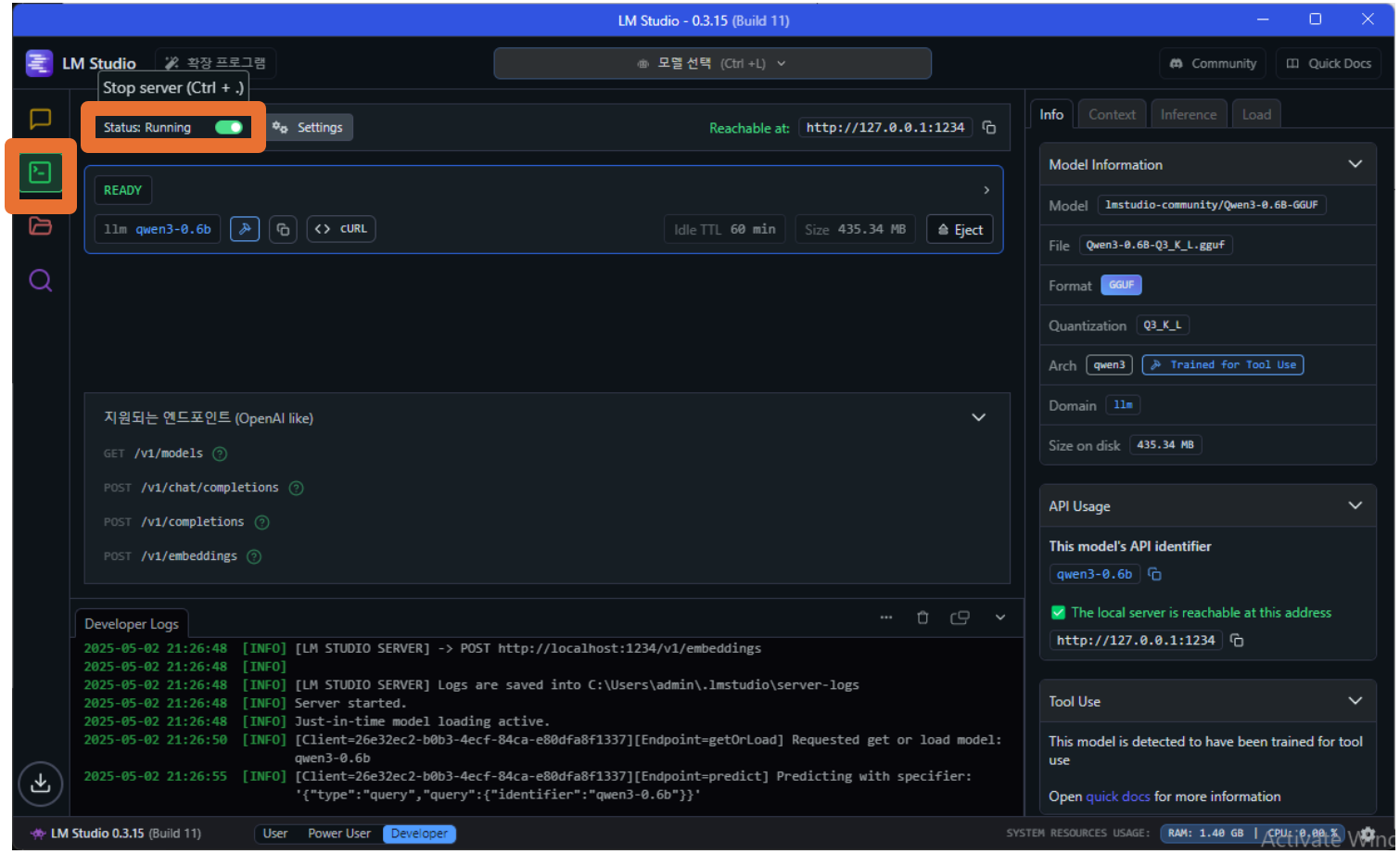


경량형 LLM QWEN 3.0 0.6B



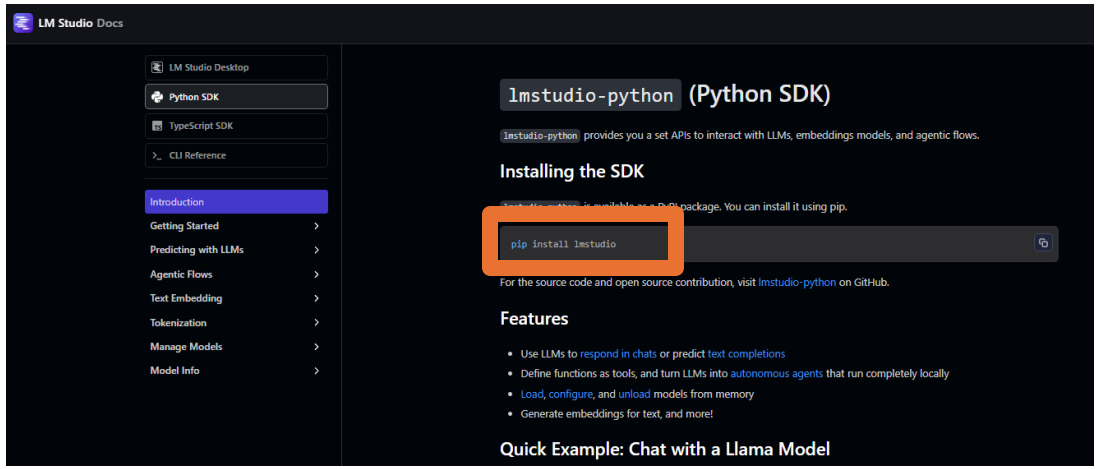
* 모델파일 자체적으로 500MB도 안한다!....ㅎㄷㄷ

LM Studio API 실행하기



LM Studio API 파이썬에서 호출

- Pip install lmstudio 설치 ㄱ

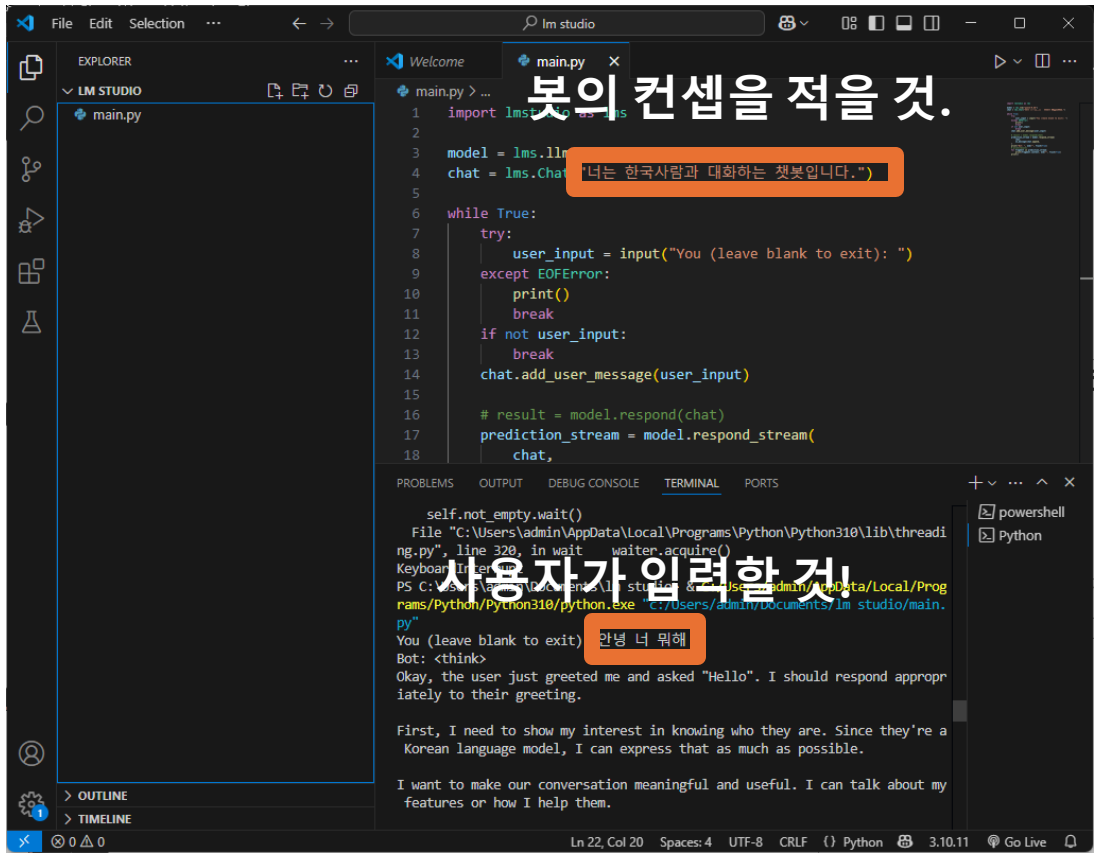


LM Studio example Chat for Python

Example: Multi-turn Chat

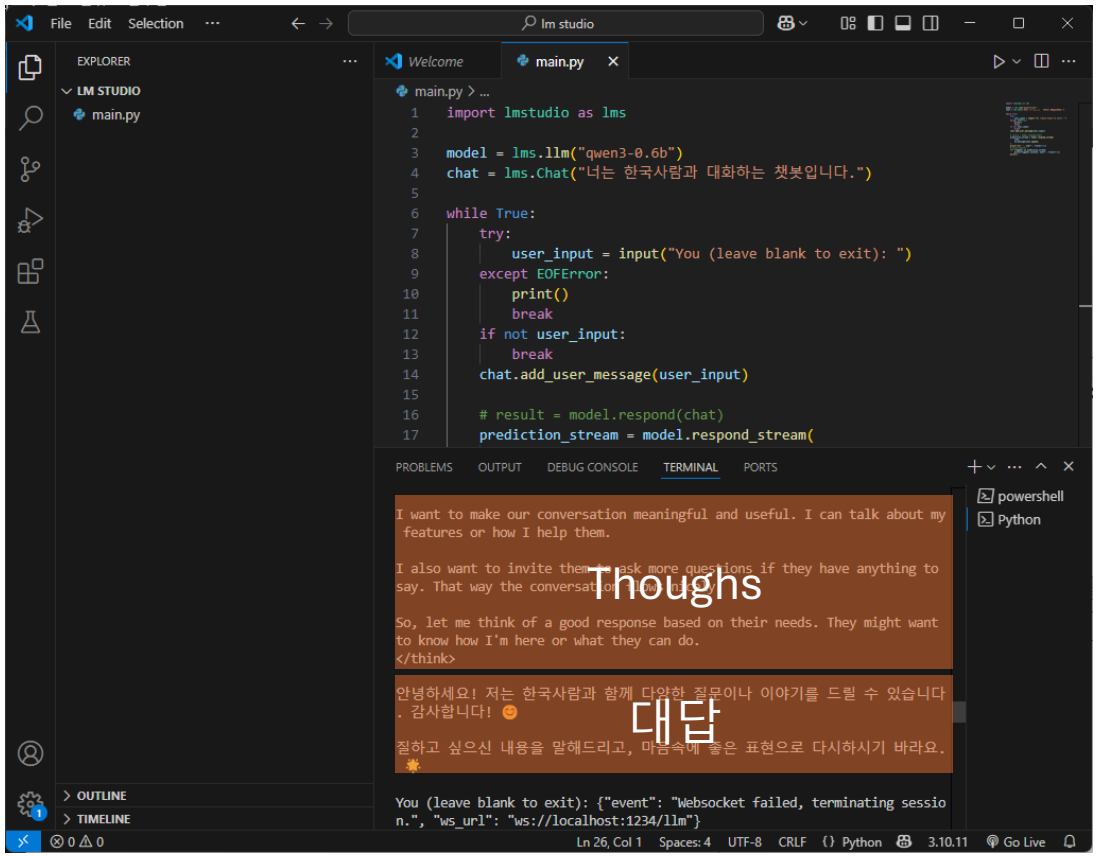
chatbot.py

```python
import lmstudio as lms

model = lms.llm()
chat = lms.Chat("You are a task focused AI assistant")

while True:
    try:
        user_input = input("You (leave blank to exit): ")
    except EOFError:
        print()
        break
    if not user_input:
        break
    chat.add_user_message(user_input)
    prediction_stream = model.respond_stream(
        chat,
        on_message=chat.append,
    )
    print("Bot: ", end="", flush=True)
    for fragment in prediction_stream:
        print(fragment.content, end="", flush=True)
    print()
```



https://lmstudio.ai/docs/python/llm-prediction/chat-completion

결과 확인할 것



- </think> 이후로 답변을 추출할 필요가 있음.
- 실제 String으로 되어 있는 값을 추출할 필요 있음.

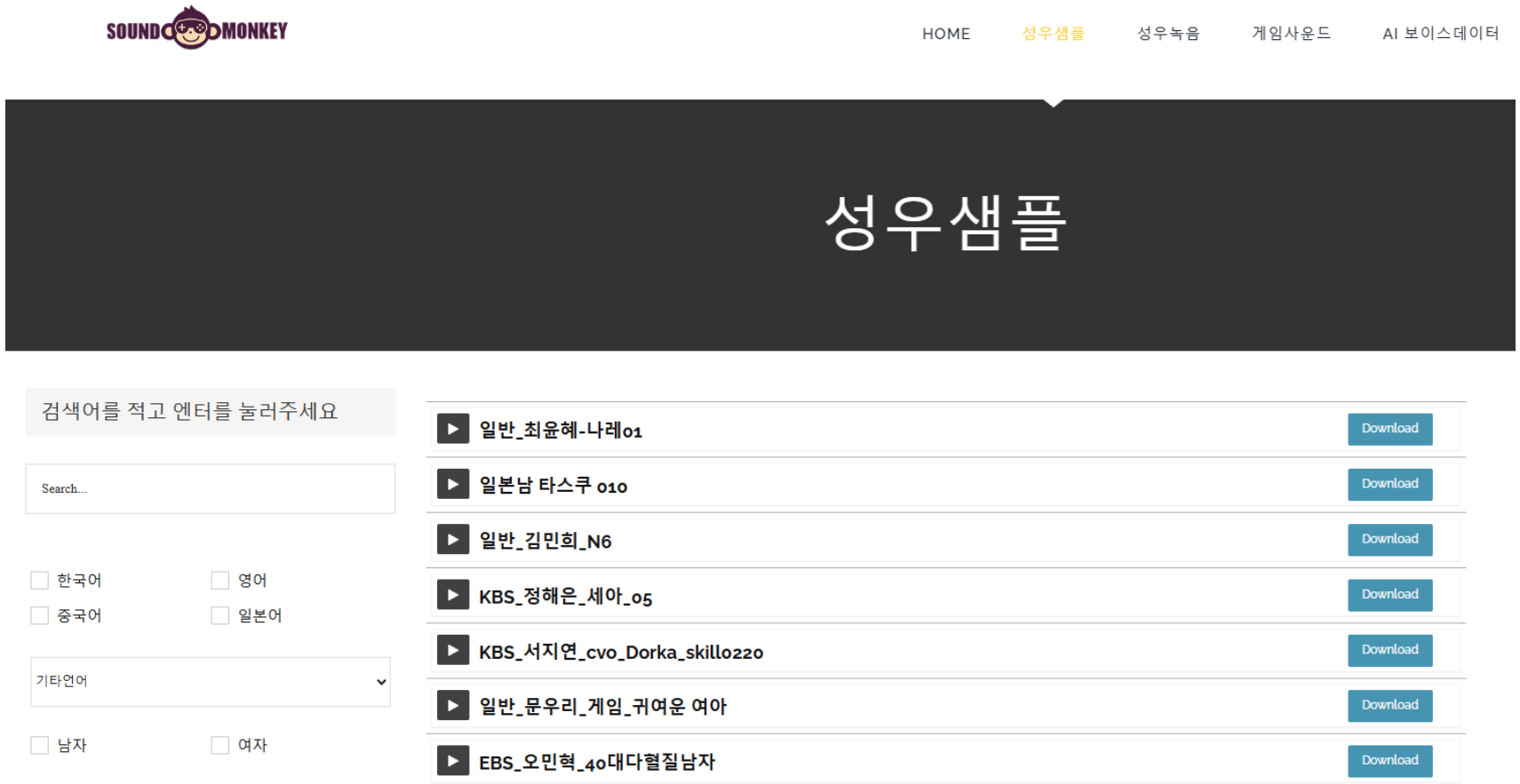
실시간 값 확인하면서 텍스트 추출

```python
import lmstudio as lms
from emoji import core

model = lms.llm("qwen3-0.6b")
chat = lms.Chat("너는 한국사람과 대화하는 챗봇입니다.")

while True:
    try:
        user_input = input("You (leave blank to exit): ")
    except EOFError:
        print()
        break
    if not user_input:
        break
    chat.add_user_message(user_input)

    prediction_stream = model.respond_stream(
        chat,
        on_message=chat.append,
    )
    print("Bot: ", end="", flush=True)

    for fragment in prediction_stream:
        print(fragment.content, end="", flush=True)
    print()

    bot_result = core.replace_emoji(prediction_stream.result().content.split('</think>')[1].strip())
    print(bot_result)
```

- 이모티콘이 있으니 제거 해봅시다!
- pip install emoji 설치 ㄱㄱ

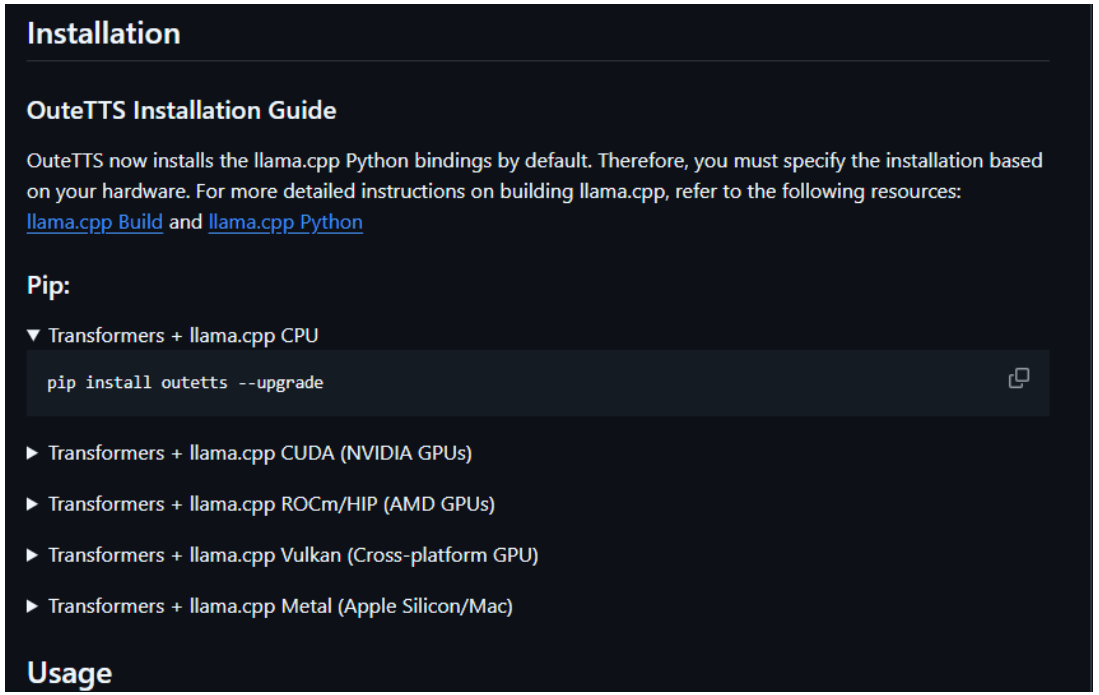
```
You (leave blank to exit): 안녕
Bot: <think>
Okay, the user just said "안녕" which translates to "Hello" in English. Let me think about how to respond appropriately.

Also, maybe I can offer some help or assistance to make the conversation more friendly. That way both parties feel comfortable and connected.

Let me check if there's anything else they might need. Maybe ask them about their current situation or interests in case we need further information.

I should keep my tone warm and welcoming, making sure the conversation flows naturally. Let me make sure I'm responding appropriately without adding any unnecessary information.
</think>

안녕하세요! 저는 항상 신뢰하고 건강하게 대화하며 도움을 드릴 수 있습니다. 어떤 주제나 관심 분야에 대해 이야기해주세요. 😊
안녕하세요! 저는 항상 신뢰하고 건강하게 대화하며 도움을 드릴 수 있습니다. 어떤 주제나 관심 분야에 대해 이야기해주세요.
You (leave blank to exit):
```

Text to Speech sample record mp3 download



https://studiomonkey.co.kr/voicesample/ | 예제 음성 파일은 일반_한호진_강현_03 임. 아무거나 해도 ㄱㅊ음.
위 해당 링크로는 mp3로 제공함. Wav 파일로 변환해서 구동해야할 것!

Text to Speech – OuteTTS install



Installation

OuteTTS Installation Guide

OuteTTS now installs the llama.cpp Python bindings by default. Therefore, you must specify the installation based on your hardware. For more detailed instructions on building llama.cpp, refer to the following resources: llama.cpp Build and llama.cpp Python

Pip:

▼ Transformers + llama.cpp CPU

```
pip install outetts --upgrade
```

▶ Transformers + llama.cpp CUDA (NVIDIA GPUs)

▶ Transformers + llama.cpp ROCm/HIP (AMD GPUs)

▶ Transformers + llama.cpp Vulkan (Cross-platform GPU)

▶ Transformers + llama.cpp Metal (Apple Silicon/Mac)

Usage

- llama.cpp Build, llama.cpp Python 설치해야한다!

https://github.com/edwko/OuteTTS?tab=readme-ov-file#installation | 각자 환경 맞추어서 설치할 것!

llama.cpp install

Build llama.cpp locally

To get the Code:

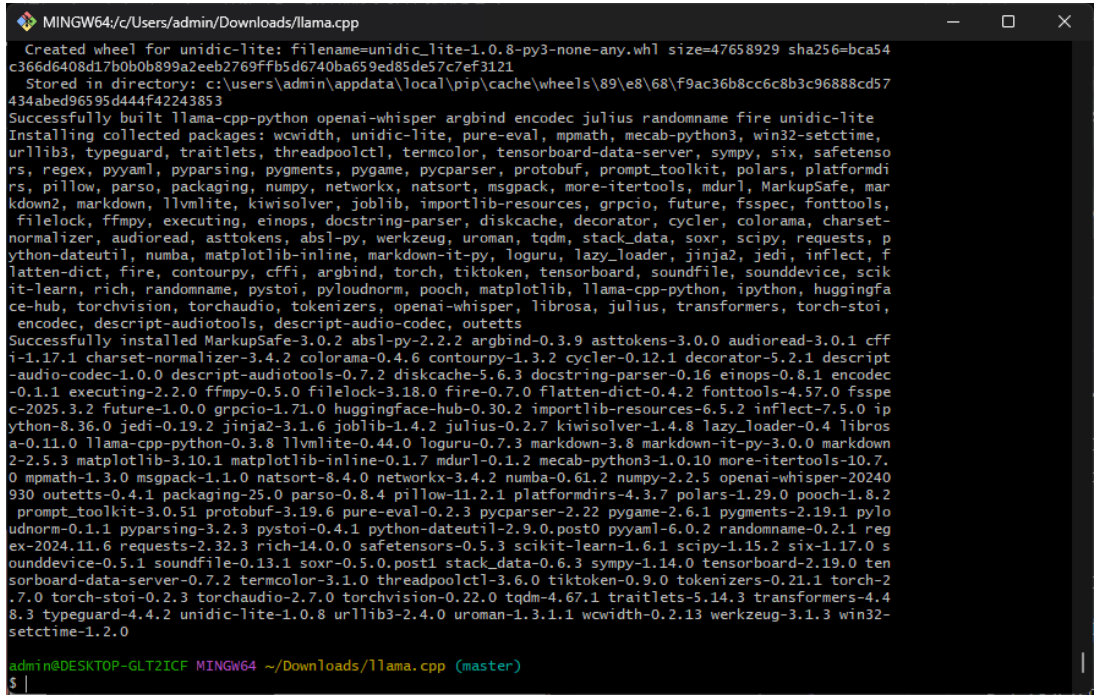
```
git clone https://github.com/ggml-org/llama.cpp
cd llama.cpp
```

The following sections describe how to build with different backends and options.

CPU Build

Build llama.cpp using `CMake`:

```
cmake -B build
cmake --build build --config Release
```



중요! cmake 설치 필요한 학생(윈도우 기준) : https://ndb796.tistory.com/365 들어가서 설치
이와 같은 것이 안된다면, Visual Studio 2022 데스크탑 C++ 설치할 것! ~~여기에서 대부분 많이 막힐거다 ㅋㅋ..~~

Llama cpp python install

Installation Configuration

`llama.cpp` supports a number of hardware acceleration backends to speed up inference as well as backend specific options. See the llama.cpp README for a full list.

All `llama.cpp` cmake build options can be set via the `CMAKE_ARGS` environment variable or via the `--config-settings / -C` cli flag during installation.

▼ Environment Variables

```
# Linux and Mac
CMAKE_ARGS="-DGGML_BLAS=ON -DGGML_BLAS_VENDOR=OpenBLAS" \
  pip install llama-cpp-python
```

```
# Windows
$env:CMAKE_ARGS = "-DGGML_BLAS=ON -DGGML_BLAS_VENDOR=OpenBLAS"
pip install llama-cpp-python
```

OuteTTS install

```
PS C:\Users\admin\Documents\lm studio> pip install outetts --upgrade
Requirement already satisfied: outetts in c:\users\admin\appdata\local\programs\python\python310\lib\site-packages (0.4.1)
Requirement already satisfied: llama-cpp-python==0.3.8 in c:\users\admin\appdata\local\programs\python\python310\lib\site-packages (from outetts) (0.3.8)
Requirement already satisfied: torch in c:\users\admin\appdata\local\programs\python\python310\lib\site-packages (from outetts) (2.7.0)
Requirement already satisfied: torchvision in c:\users\admin\appdata\local\programs\python\python310\lib\site-packages (from outetts) (0.22.0)
Requirement already satisfied: torchaudio in c:\users\admin\appdata\local\programs\python\python310\lib\site-packages (from outetts) (2.7.0)
Requirement already satisfied: scipy in c:\users\admin\appdata\local\programs\python\python310\lib\site-packages (from outetts) (1.15.2)
Requirement already satisfied: einops in c:\users\admin\appdata\local\programs\python\python310\lib\site-packages (from outetts) (0.8.1)
Requirement already satisfied: pyyaml in c:\users\admin\appdata\local\programs\python\python310\lib\site-packages (from outetts) (6.0.2)
Requirement already satisfied: huggingface-hub in c:\users\admin\appdata\local\programs\python\python310\lib\site-packages (from outetts) (0.30.2)
Requirement already satisfied: encodec in c:\users\admin\appdata\local\programs\python\python310\lib\site-packages (from outetts) (0.1.1)
Requirement already satisfied: matplotlib in c:\users\admin\appdata\local\programs\python\python310\lib\site-packages (from outetts) (3.10.1)
Requirement already satisfied: transformers==4.48.3 in c:\users\admin\appdata\local\programs\python\python310\lib\site-packages (from outetts) (4.48.3)
Requirement already satisfied: soundfile in c:\users\admin\appdata\local\programs\python\python310\lib\site-packages (from outetts) (0.13.1)
```

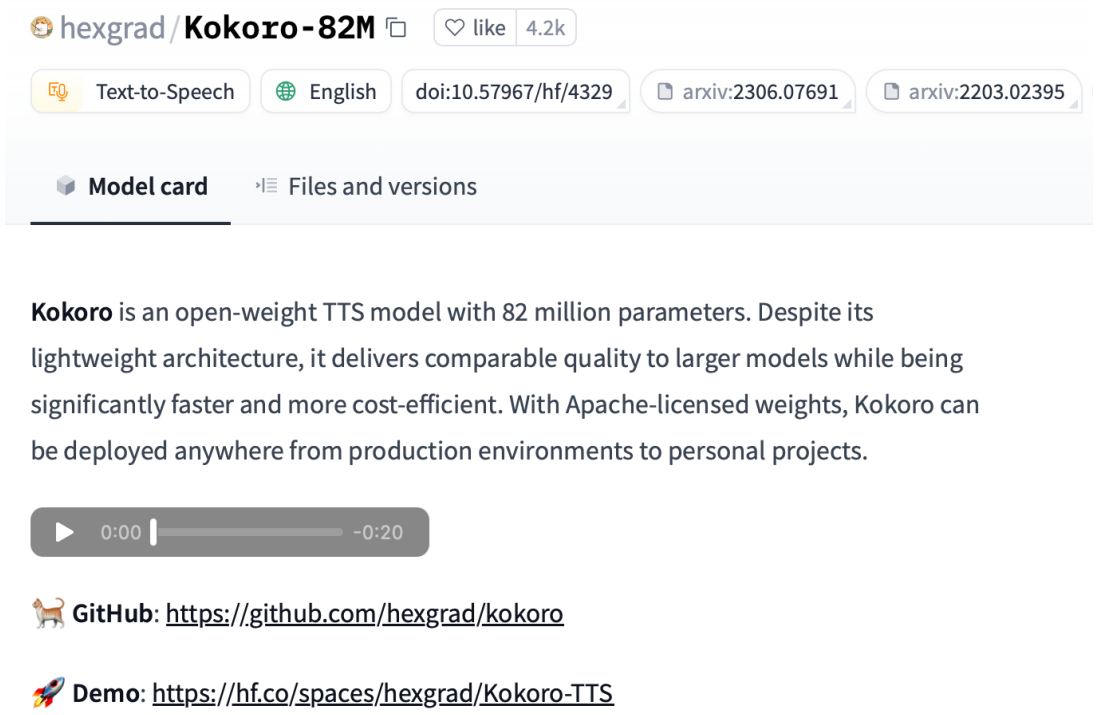
OuteTTS 목소리 추출 & 생성

```python
import outetts

# Initialize the interface
interface = outetts.Interface(
    config=outetts.ModelConfig.auto_config(
        model=outetts.Models.VERSION_1_0_SIZE_1B,
        # For llama.cpp backend
        backend=outetts.Backend.LLAMACPP,
        quantization=outetts.LlamaCppQuantization.Q2_K # 양자화 모델, 작은걸로 함
    )
)

speaker = interface.create_speaker("audio.wav") # 생성할 목소리 원본파일 넣기
interface.save_speaker(speaker, "speaker.json")
```

```python
import outetts

# Initialize the interface
interface = outetts.Interface(
    config=outetts.ModelConfig.auto_config(
        model=outetts.Models.VERSION_1_0_SIZE_1B,
        # For llama.cpp backend
        backend=outetts.Backend.LLAMACPP,
        quantization=outetts.LlamaCppQuantization.Q2_K
        # For transformers backend
        # backend=outetts.Backend.HF,
    )
)

speaker = interface.load_speaker("speaker.json")

# Generate speech
output = interface.generate(
    config=outetts.GenerationConfig(
        text="텍스트 음성 생성할 내용 작성",
        generation_type=outetts.GenerationType.CHUNKED,
        speaker=speaker,
        sampler_config=outetts.SamplerConfig(
            temperature=0.6
        ),
    )
)

# Save to file
output.save("output.wav")
```

https://huggingface.co/OuteAI/Llama-OuteTTS-1.0-1B-GGUF#quick-start-guide

설정하는 것이 오래걸리고 실행 속도 느리다면?



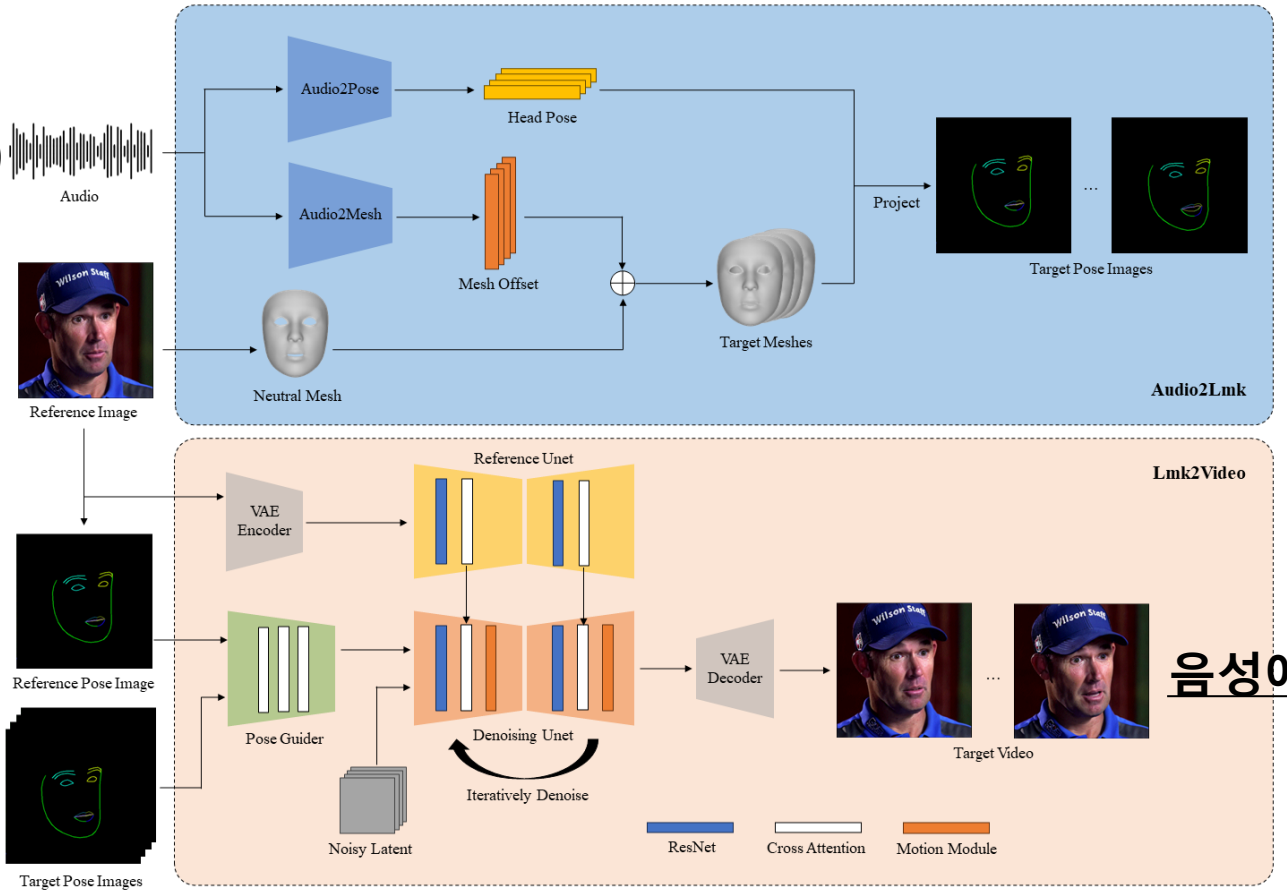
- 일반적인 TTS 사용하는 것으로 해봅시다.

실습과제
Audio Driven 기능 실행해보기

AniPortrait: Audio-Driven Synthesis of Photorealistic Portrait Animations

내가 말해야하는 음성(TTS 결과)

얼굴 움직여야하는 대상 이미지

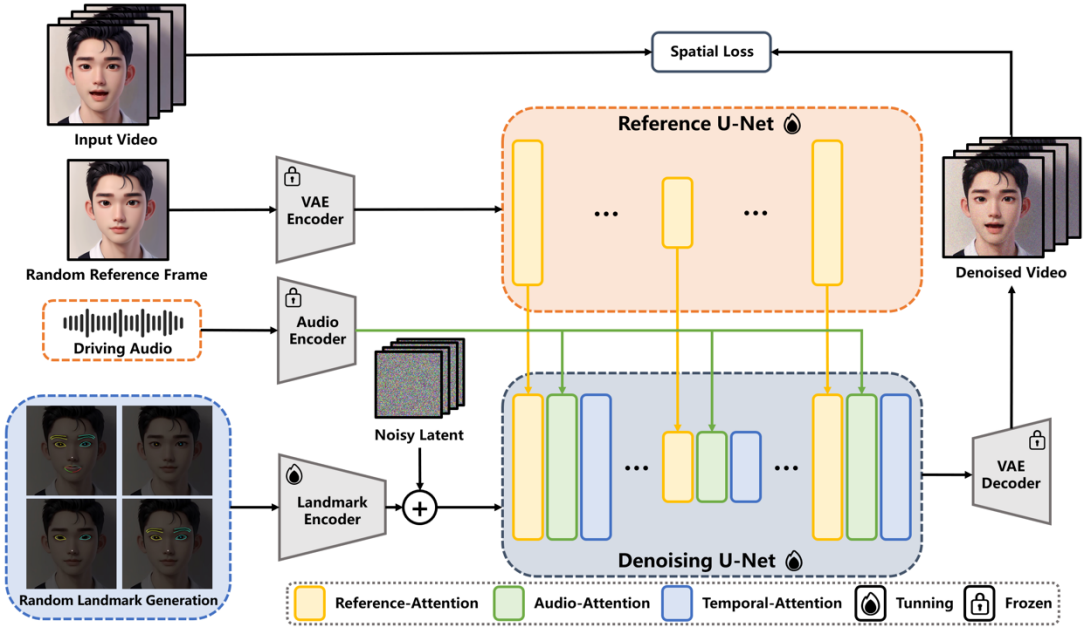


음성에 맞는 얼굴 생성

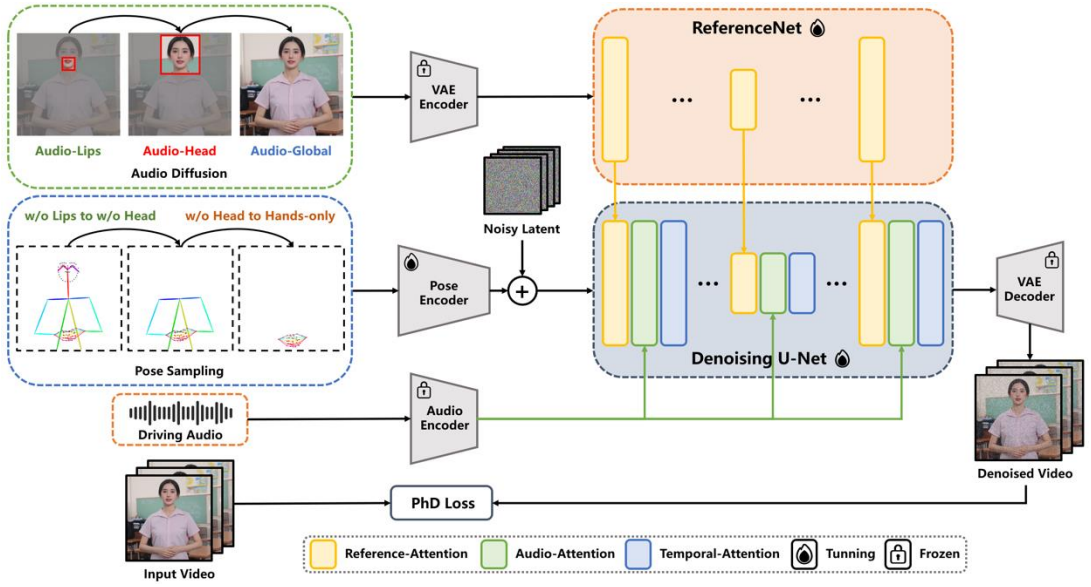
이미지 생성 기술은 Stable Diffusion 이라는 것을 확인할 수 있음..! (SD1.5 version)

EchoMimic: Lifelike Audio-Driven Portrait Animations through Editable Landmark Conditioning

v1



v2



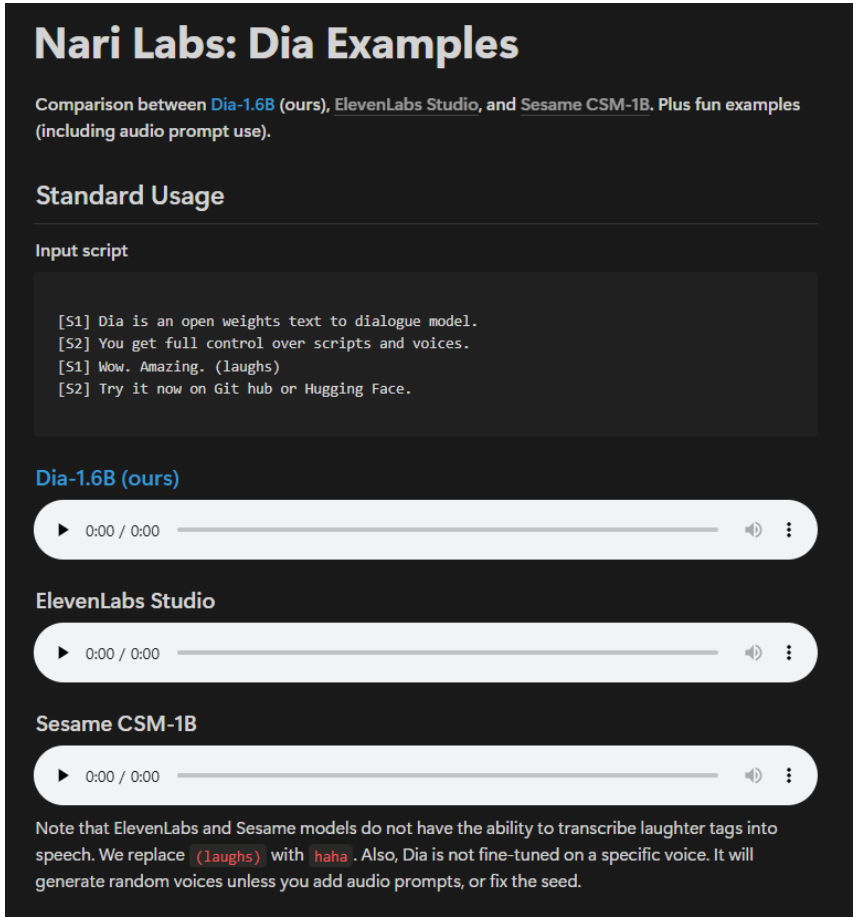
이러한 것들을 구동하기 위해서 필요한 것?

- ~~좋은 컴퓨터 부품과 API 호출할 돈만 준비되면 가능하다!~~
- 요구 사양을 낮추고 싶다면 양자화 모델이나 크기가 작은 모델로 사용하면 된다!

Real-time video audio api

- https://x.com/i/status/1915808257958154284

TTS 음질 좋게 하고 싶다면?



Nari Labs: Dia Examples

Comparison between Dia-1.6B (ours), ElevenLabs Studio, and Sesame CSM-1B. Plus fun examples (including audio prompt use).

Standard Usage

Input script

```
[S1] Dia is an open weights text to dialogue model.
[S2] You get full control over scripts and voices.
[S1] Wow. Amazing. (laughs)
[S2] Try it now on Git hub or Hugging Face.
```

Dia-1.6B (ours)

0:00 / 0:00

ElevenLabs Studio

0:00 / 0:00

Sesame CSM-1B

0:00 / 0:00

Note that ElevenLabs and Sesame models do not have the ability to transcribe laughter tags into speech. We replace `(laughs)` with `haha`. Also, Dia is not fine-tuned on a specific voice. It will generate random voices unless you add audio prompts, or fix the seed.

- https://yummy-fir-7a4.notion.site/dia
- ~~물론 모델 크기가 10GB인걸로...~~