

AI Phone Call & TTS Control System

System Overview

This system consists of **two connected parts**:

Component	Location	Purpose
1. NIPA Cloud Server	Hosted on NIPA Cloud	Runs the core AI services — Text-to-Speech (FishSpeech), Chatbot, and Phone Call APIs. Handles AWS Connect integration, DynamoDB, and S3 audio uploads.
2. Local Control Panel (Web UI)	Run on local PC	A simple bilingual (KR/EN) interface used to initiate phone calls, generate greetings, and test TTS with uploaded reference audio.

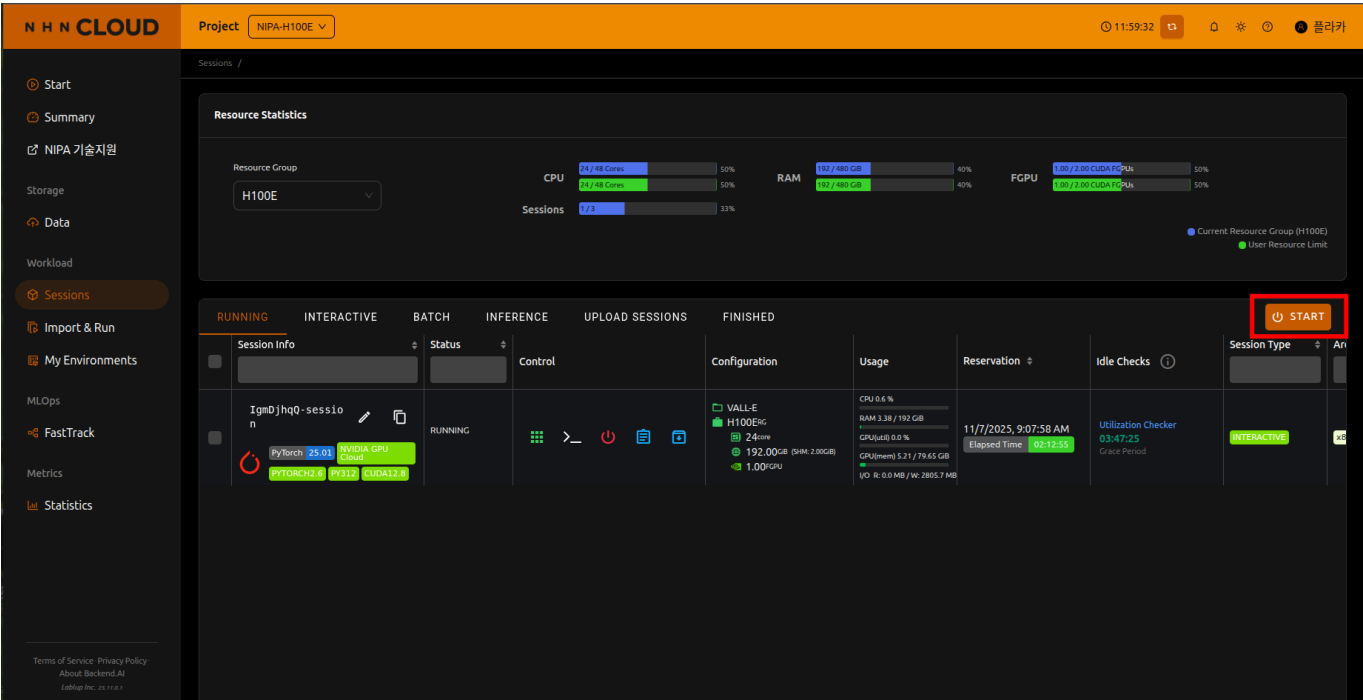
The local interface connects to the NIPA server via public domain.

1. Server Access (NIPA Cloud)

Item	Details
Link	https://nipa.nhncloud.com/
ID	nipa-gpu2025-668@ktcloud.com
Password	_insunetfc6276
Server URL	https://honest-trivially-buffalo.ngrok-free.app

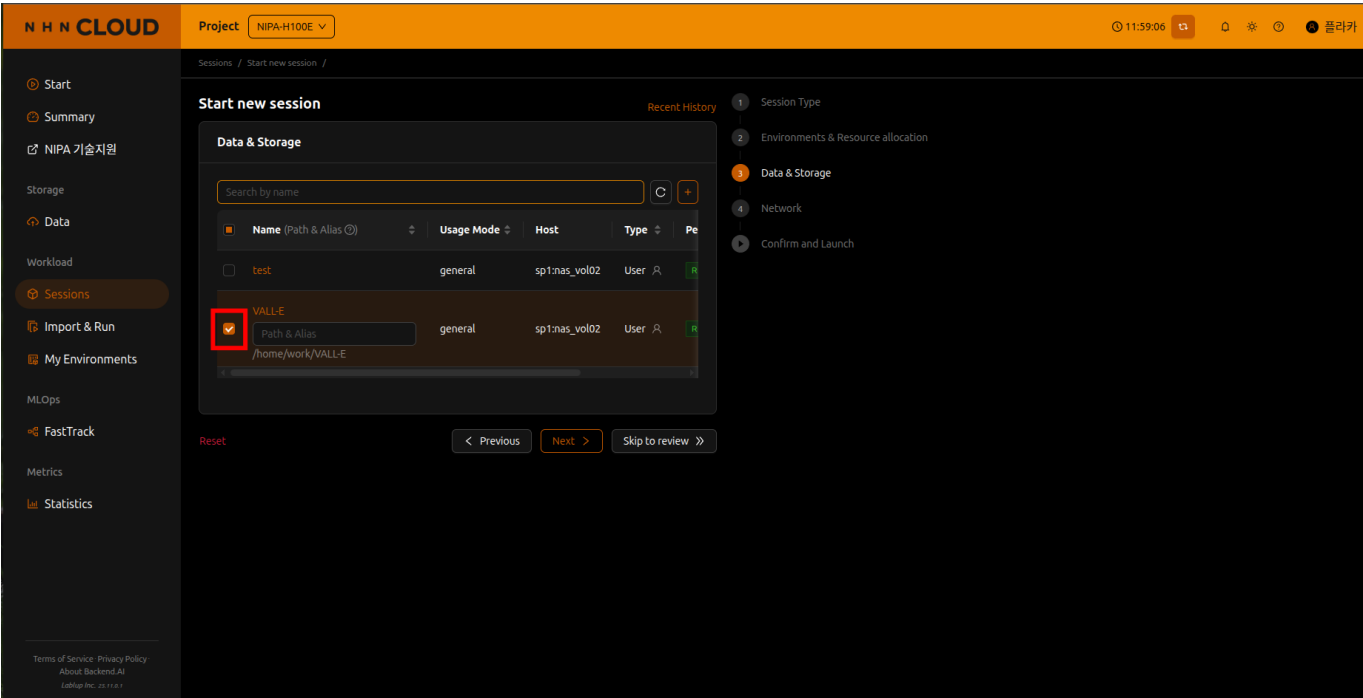
2. Starting the NIPA Server

Step 1 — Starting the server



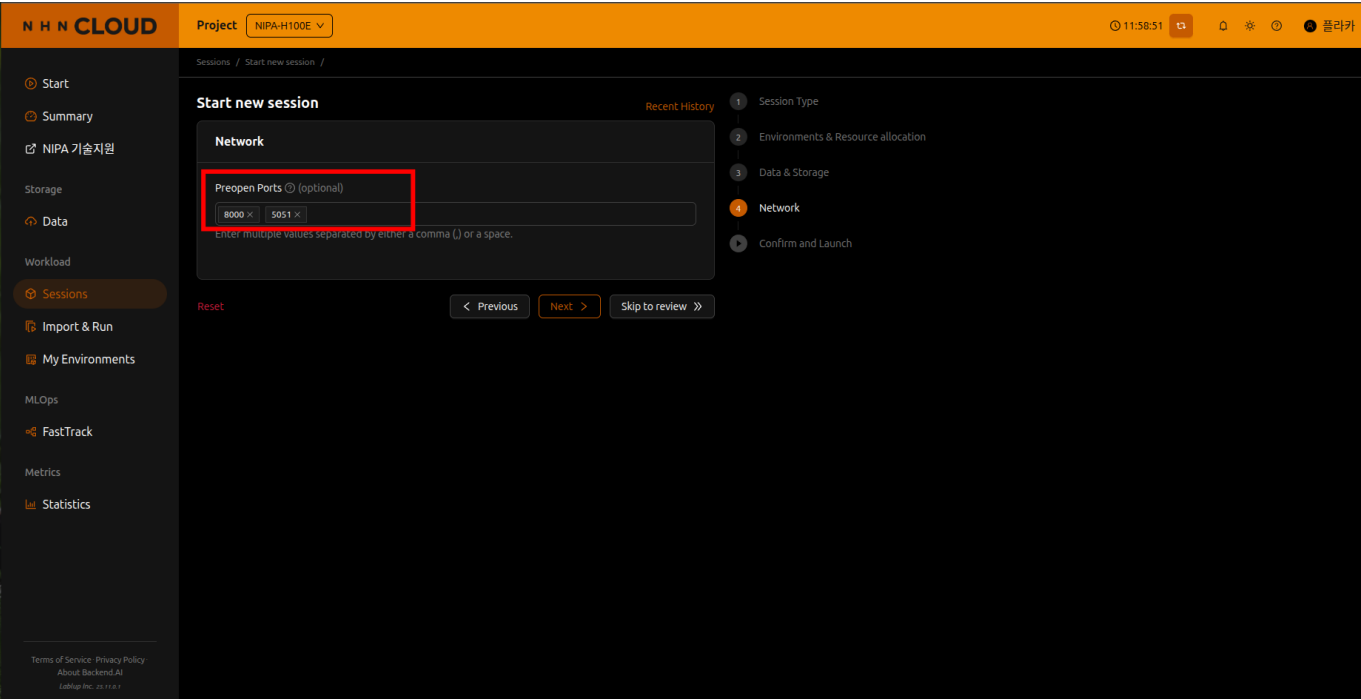
Step 2 — Mount VALL-E directory

Select VALL-E directory for mounting



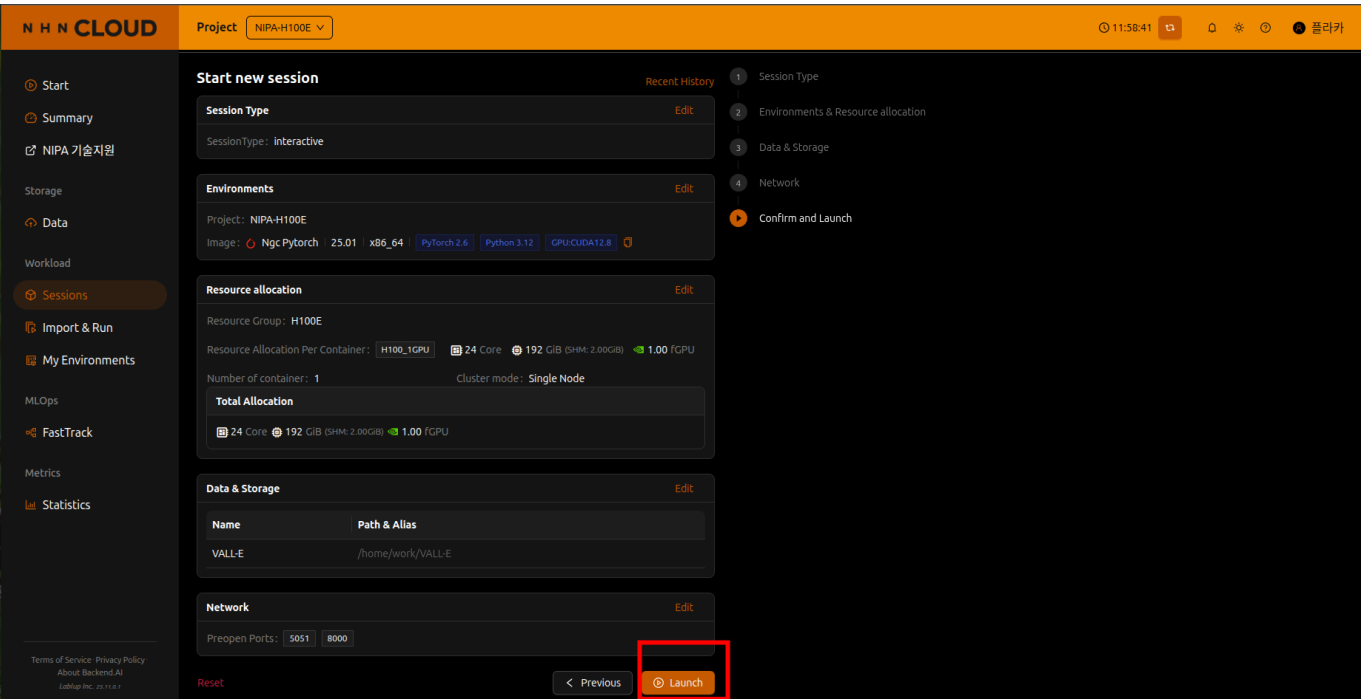
Step 3 — Select pre-open ports

Input ports 8000 and 5051



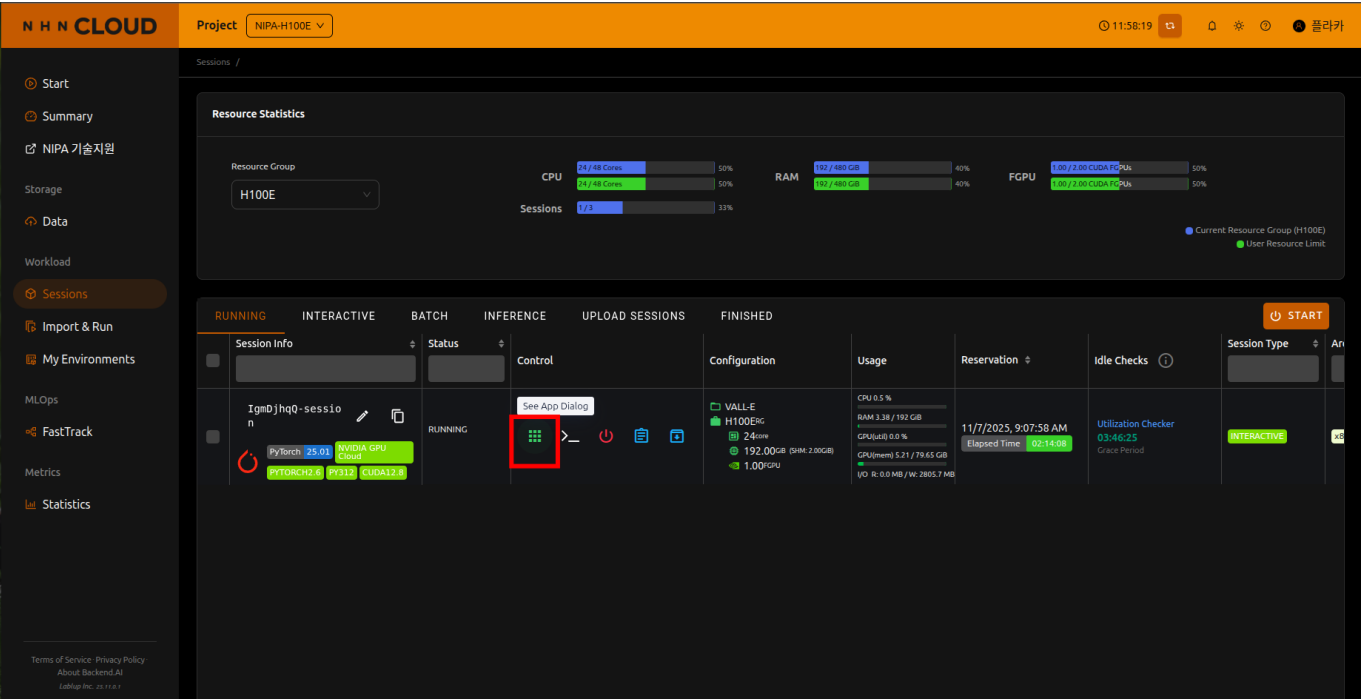
Step 4 — Launch the server

Click on the launch button



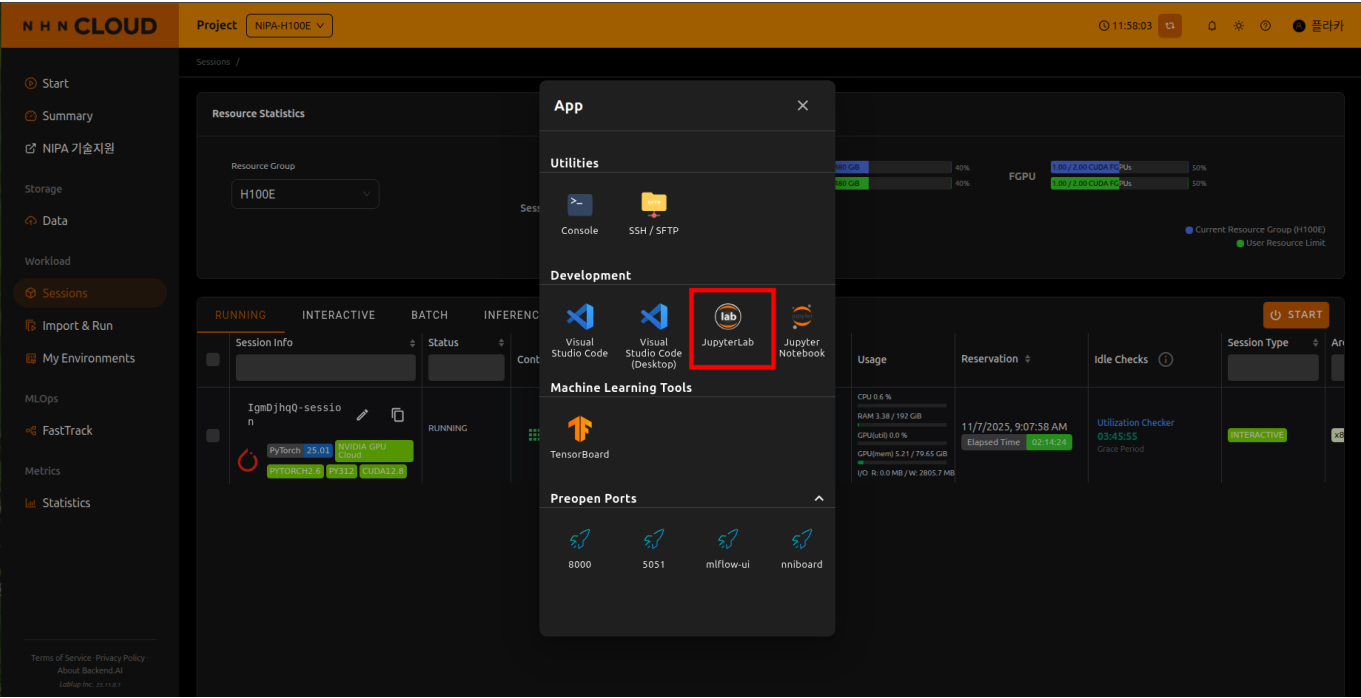
Step 5 — Selecting app dialog for server configuration

Click on app dialog button




Step 6 — Select and open JupyterLab

Select JupyterLab



Step 7 — Start the NIPA cloud server



The screenshot displays the Visual Studio Code (VS Code) interface. On the left, the File Explorer sidebar shows the project directory structure. The 'Name' column lists files and folders, and the 'Modified' column shows their last update times. The file 'app.py' is highlighted. The main editor area on the right shows a terminal window with the command './VALL-E/run_server.sh' being executed. The terminal output is currently empty.

Name	Modified
asr_bridge	14d ago
audio_samples	23h ago
chatbot	yesterday
data	last mo.
fishspeech	2d ago
installations	2d ago
miniconda3	2mo ago
miniconda3.bak	2mo ago
nlk_data	2mo ago
phone_call	43m ago
app.py	50m ago
KsponSpeech_da...	5mo ago
Miniconda3-lates...	3mo ago
ngrok.log	22m ago
requirements.txt	2mo ago
run_server.sh	2h ago
stable-ngrok-v3-li...	2mo ago

```
work@main[igb0]hq-session:~$ ./VALL-E/run_server.sh
```

This is the preview if the server launch is completed

```
File Edit View Run Kernel Tabs Settings Help
└─ /VALL-E/
  Name Modified
  asr_dir 14d ago
  audio_samples 23h ago
  chatbot yesterday
  data last mo.
  fishspeech 2d ago
  installations 2d ago
  miniconda3 2mo ago
  miniconda3.bak 2mo ago
  nltk_data 2mo ago
  phone_call 43m ago
  app.py 50m ago
  xpsongspeech_data 5mo ago
  Miniconda3-lates... 3mo ago
  ngrok.log 8m ago
  requirements.txt 2mo ago
  run_server.sh 2h ago
  stable-ngrok-v3-l... 2mo ago

Terminal 1 x app.py
workon1[1]@ip[redacted]:~$ -s -VALL-E/run_server.sh
[*] direnv already installed.
[*] Allowing envrc in /home/work/VALL-E
direnv: loading ~/.VALL-E/.envrc
[*] Registering ngrok auth token...
[*] Environment loaded: conda:fishspeech | AWS + ngrok ready
direnv: export +AWS_ACCESS_KEY_ID +AWS_REGION +AWS_SECRET_ACCESS_KEY +CONDA_DEFAULT_ENV +CONDA_EXE +CONDA_PREFIX +CONDA_PROMPT_MODIFIER +CONDA_PYTHON_EXE +CONDA_SHLVL +GSETTINGS_SCHEMA_DIR +GSETTINGS_SCHEMA_DIR_CC
NO BACKUP +NGROK_AUTH_TOKEN +NGROK_FIXED_DOMAIN +NGROK_HOST +NGROK_PORT +NGROK_TZ +PROJECT_ROOT +UIVCORR_HOST +UIVCORR_PORT +CE_CONDA +CE_M -PATH -PYTHONPATH
[*] Starting ngrok.
[*] ngrok log (last 20 lines):
[*] Starting uicorn...
/home/work/VALL-E/miniconda3/envs/fishspeech/lib/python3.10/site-packages/torch/_init_.py:1617: UserWarning: Please see the new API settings to control TF32 behavior, such as torch.backends.cudnn.compatibility_split = 'tf32' or torch.backends.cudnn.compatibility_split = 'ieee'. Old settings, e.g., torch.backends.cudnn.matmul.allow_tf32 = True, torch.backends.cudnn.allow_tf32 = True, allowTF32OnNan() and allowTF32OnBFloat16() will be deprecated after Pytorch 2.0. Please see https://pytorch.org/docs/main/note/cuda.html#tensorfloat-32-tf32-on-ampere-and-later-devices (Triggered internally at /pytorch/aten/src/ATen/Context.cpp:86).
C_set_float32_matmul_precision(precision)
2025-11-07 02:23:33.142 INFO fishspeech.fish_speech.app: <module>:254 - [PATCH] TTSLanguageEngine.inference to use REF_AUDIO
2025-11-07 02:23:33.519 INFO fishspeech.fish_speech.app: warmup fishspeech:1455 - [FishSpeech] Starting warmup...
2025-11-07 02:23:33.521 INFO fishspeech.fish_speech.app: init :1244 - Launching LLM queue.
2025-11-07 02:23:33.547 INFO fishspeech.fish_speech.app: Loading model from /home/work/VALL-E/fishspeech/fish_speech/checkpoints/opensudo1.1-mini. config: DualMMModelArgs{
model_type:'dual_ar', vocal_size=15776, n_layer=20, n_head=16, dim=1024, intermediate_size=3072, n_local_heads=16, head_dim=128, rope_base=1000000, norm_eps=1e-06, max_seq_len=4192, dropout=0.0, tie_word_embeddings=False, attention_qkv_bias=False, attention_o_bias=False, attention_k_norm=True, codebook_size=4096, num_codebooks=10, use_gradient_checkpointing=True, initializer_range=0.03125, is_reward_model=False, scale_codebook_embeddings=True, n_fast_layer=4, fast_dim=1024, fast_n_local_heads=16, fast_head_dim=64, fast_intermediate_size=3072, fast_attention_qkv_bias=False, fast_attention_k_norm=False, fast_attention_v_norm=False, n_codebooks=10}
2025-11-07 02:23:40.833 INFO fishspeech.models.text2semantic.llama: from pretrained:498 - Loaded weights with error: <All keys matched successfully>
2025-11-07 02:23:40.847 INFO fishspeech.models.text2semantic.inference: init model:1357 - Restored model from checkpoint
2025-11-07 02:23:40.847 INFO fishspeech.models.text2semantic.inference: init model:1360 - Using DualMMTransformer
/home/work/VALL-E/miniconda3/envs/fishspeech/lib/python3.10/site-packages/torch/nn/utils/weight_norm.py:144: FutureWarning: 'torch.nn.utils.weight_norm' is deprecated in favor of 'torch.nn.utils.parametrize.v2.weight_norm.apply(module, name, dim)'
2025-11-07 02:23:43.761 INFO fishspeech.models.dac.inference: load model:46 - Loaded model: <All keys matched successfully>
2025-11-07 02:23:43.761 INFO fishspeech.fish_speech.app: init :1251 - Constructing TTSLanguageEngine.
2025-11-07 02:23:44.839 INFO fishspeech.fish_speech.app: torch.compile not available; continuing
2025-11-07 02:23:44.839 INFO fishspeech.fish_speech.app: load target_se once:1364 - [FishSpeech] Loaded target SE from /home/work/VALL-E/audio_samples/cached_ref.pt
2025-11-07 02:23:44.839 INFO fishspeech.fish_speech.app: warmup fishspeech:1462 - [FishSpeech] Running dummy inference for warmup...
2025-11-07 02:23:44.840 INFO fishspeech.app: patched inference:149 - @ [PATCH] Loading default embedding from /home/work/VALL-E/audio_samples/cached_ref.pt
2025-11-07 02:23:44.841 INFO fishspeech.fish_speech.app: patched inference:158 - [PATCHED] Done loading embeddings. Using 1 prompt token(s).
2025-11-07 02:23:44.842 INFO fishspeech.fish_speech.app: patched inference:165 - set seed: 42
2025-11-07 02:23:44.853 INFO fishspeech.models.text2semantic.inference: generate_log:457 - Encoded text: 언연하씨요. 테스트입니다.
2025-11-07 02:23:47.862 INFO fishspeech.models.text2semantic.inference: generate_log:491 - Generated 37 tokens in 3.01 seconds, 12.30 tokens/sec
2025-11-07 02:23:47.862 INFO fishspeech.models.text2semantic.inference: generate_log:494 - Bandwidth achieved: 10.58 GB/s
2025-11-07 02:23:47.862 INFO fishspeech.models.text2semantic.inference: generate_log:497 - GPU Memory used: 5.02 GB
2025-11-07 02:23:47.866 INFO fish_speech.inference.engine.vq_manager: decode_vq_tokens:20 - VQ features: torch.Size([10, 36])
2025-11-07 02:23:49.640 INFO fishspeech.fish_speech.app: synthesize:1335 - [Requested 36] model:36 - Training model done
2025-11-07 02:23:49.640 INFO fishspeech.fish_speech.app: warmup fishspeech:1476 - [FishSpeech] Warmup complete. Dummy audio length: 73728 samples at 44100 Hz
[2025-11-07 02:23:49.159] INFO [real_accelerator.py:203:get_accelerator] Setting ds_accelerator to cuda (auto detect)
INFO: Started server process [5050]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
```

Step 1 — Open terminal in project folder

Step 2 — Install dependencies

```
pip install fastapi uvicorn httpx
```

Step 3 — Start the web app

```
uvicorn local_app:app --host 0.0.0.0 --port 5051
```


Then open your browser and go to:
🖱️ <http://localhost:5051>

4. Using the Local Web Interface (Updated)

Overview

The local web interface is now a **single unified page** with three main functional areas:

Area	Name	Function
1	Phone Call Panel	Start calls and generate personalized greetings
2	Chat Interface	Test and interact with the 차집사 chatbot (Chajipsa / GPT engines)
3	TTS Utility (Optional)	Generate voice audio via the local TTS server

 Placeholder images shown below. Replace them with real screenshots later.

1 — Phone Call Panel

The **top-left** section of the interface handles phone call operations.

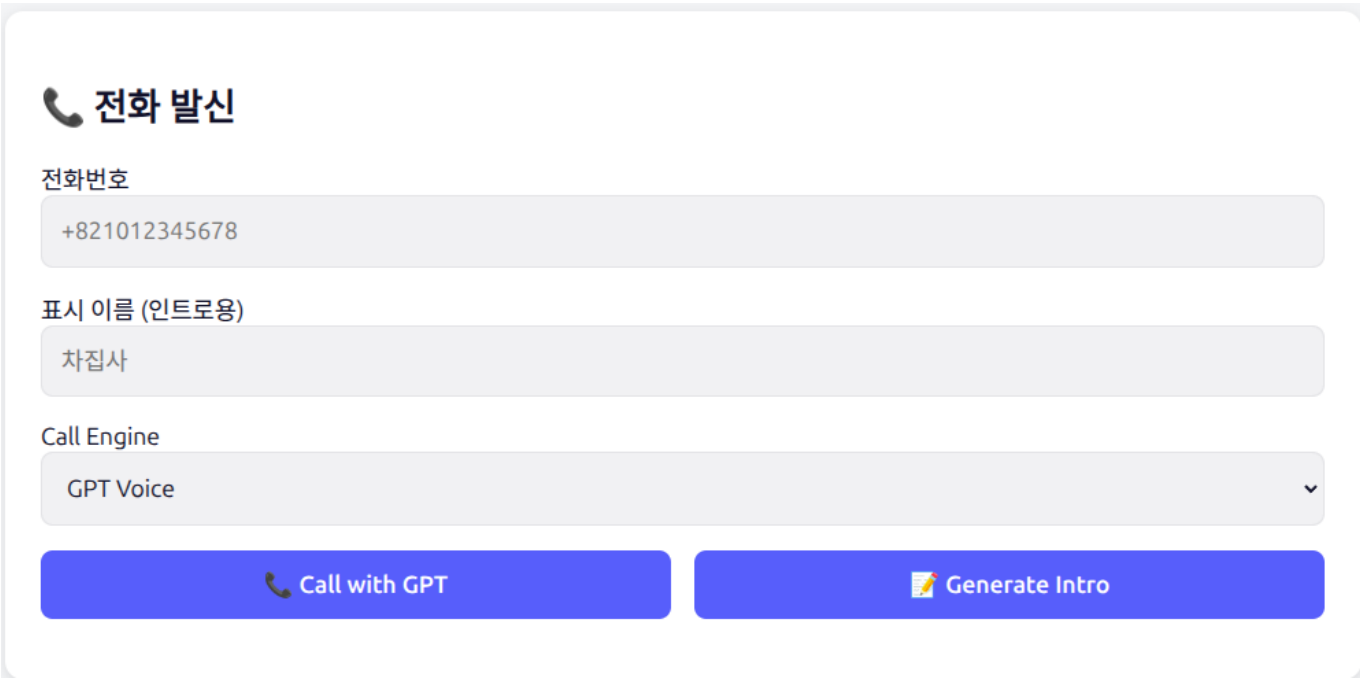
Functions

- Start outbound phone calls
- Enter customer name & phone number
- Generate personalized greetings:
 - Intro greeting (e.g., “안녕하세요, 홍길동 고객님...”)
 - Personalized affirmative clip (“네, 홍길동님”)

How to Use

1. Enter the phone number
2. (Optional) Enter the customer’s name
3. Click **Generate Greeting**
4. Click **Start Call** to begin the outbound call

Screenshot Placeholder



🎤 2 — TTS Utility

Located at the **bottom** of the page.


Functions

- Generate voice audio
- (Optional) Upload reference audio (.wav, .m4a)
- Download or play generated audio

How to Use

1. Enter text into the TTS input
2. (Optional) upload a reference voice file
3. Click **Generate**
4. The result will appear as an audio player with a download option

Screenshot Placeholder

 **Text-to-Speech**

엔진 선택

FishSpeech

텍스트

여기에 텍스트를 입력하세요...


톤 선택 (Tone Selection)

[None] (happy) (sad) (angry) (excited) (friendly) (fearful)

Temperature 0.70

참조 음성 (선택)

Choose File No file chosen

 **Generate Audio**

3 — Chat Interface

Located on the **right side** of the page.

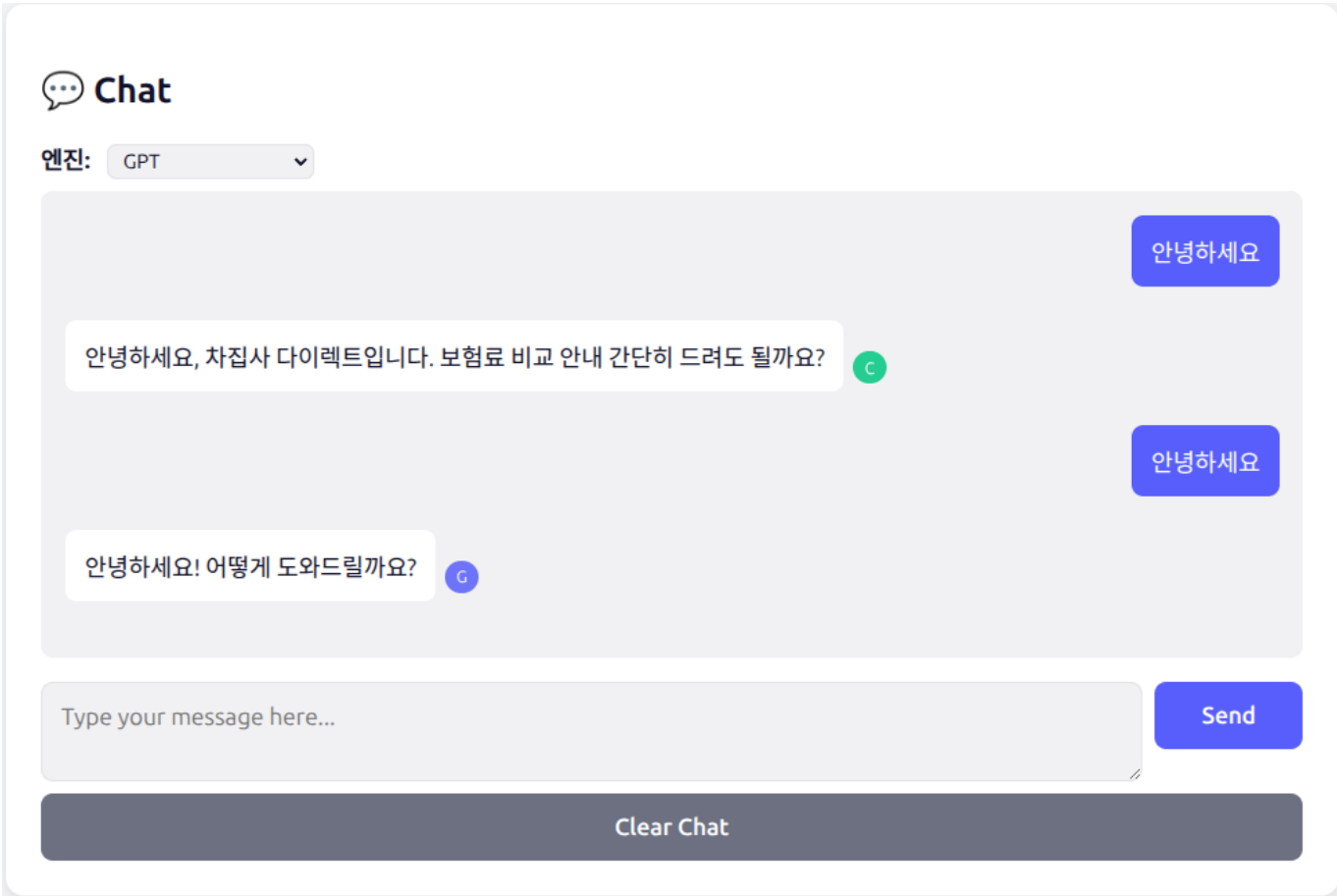
Functions

- Interactive chat with the 차집사 assistant
- Switch between:
 - **Chajipsa engine**
 - **GPT engine**
- Engine icons appear next to each assistant message
- System log output shown underneath messages

How to Use

1. Type your message in the chat box
2. Choose the engine (Chajipsa or GPT)
3. Press **Enter** or click **Send**
4. Chat messages will appear continuously like a messenger UI

Screenshot Placeholder



⚠ 5. Troubleshooting

Issue	Cause	Solution
“Server is not running” message	NIPA server is offline or Ngrok disconnected	Reconnect Ngrok and restart <code>uvicorn app:app</code>
No sound / empty audio	Missing reference file or wrong prefix	Verify <code>/tts/synthesize2</code> works
Call not triggered	AWS Connect permission issue	Check Lambda + Connect integration
Browser stuck at “Generating...”	Timeout or memory issue	Check NIPA logs

🧠 6. Next Steps

- ☒ Web API and Control Panel completed
- ☐ Train new multitask chatbot model (intent + response)
- ☐ Integrate chatbot into phone call flow (optional Tab 3)