



*Think Global... Choose Global...*

# Major Project I

**ZENO: Your Private Ai Assistant**


**Guided By : Helly Patel**

**Ansari Aasif**  
2302144022



## **Abstract :**

ZENO is a privacy-first desktop assistant designed to run entirely offline on a local machine. Unlike cloud-based assistants, ZENO ensures complete data security by processing queries without an internet connection. It serves as a reliable desktop partner, capable of answering questions and performing tasks through natural voice commands. With its focus on privacy, efficiency, and accessibility, ZENO redefines the concept of a personal assistant for everyday computing needs.






## Introduction :


ZENO is a privacy-first desktop assistant developed to address the growing concerns of data security in AI-driven tools. It is not like conventional assistants that depend on cloud servers, ZENO runs entirely on a local machine, ensuring that no personal information is shared or exposed online. This makes it a secure and dependable solution for users who prioritize privacy while still seeking the convenience of modern AI.

As a desktop partner, ZENO is designed to stay on the user's desk and provide continuous support. It responds to natural voice commands, answers questions, and performs essential tasks without relying on an internet connection. By functioning fully offline, it remains accessible in all conditions while giving users complete control over their data.





## Objective And Scope

- To design and develop a privacy-focused desktop assistant that runs entirely on a local machine.
  - To enable users to access information and perform tasks through voice commands without requiring an internet connection.
  - To reduce user dependency on opening laptops and typing for simple queries, making the interaction more natural and effortless.
  - Provide quick responses to general questions and system-related tasks through natural speech.
  - Serve as a trusted personal companion on the desk, available at all times for hands-free interaction
- 

# Tools and Technologies Used



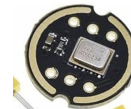
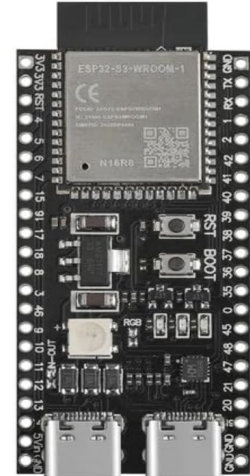
## Hardware:

- ESP32 Microcontroller
- INMP441 Omnidirectional Microphone
- 3W Speaker



## Software:

- Python (Server-side, LLM integration, speech processing)
- C++ (ESP32 programming)
- Piper TTS (Text-to-Speech)
- Speech Recognition Libraries
- WebSockets (real-time communication)

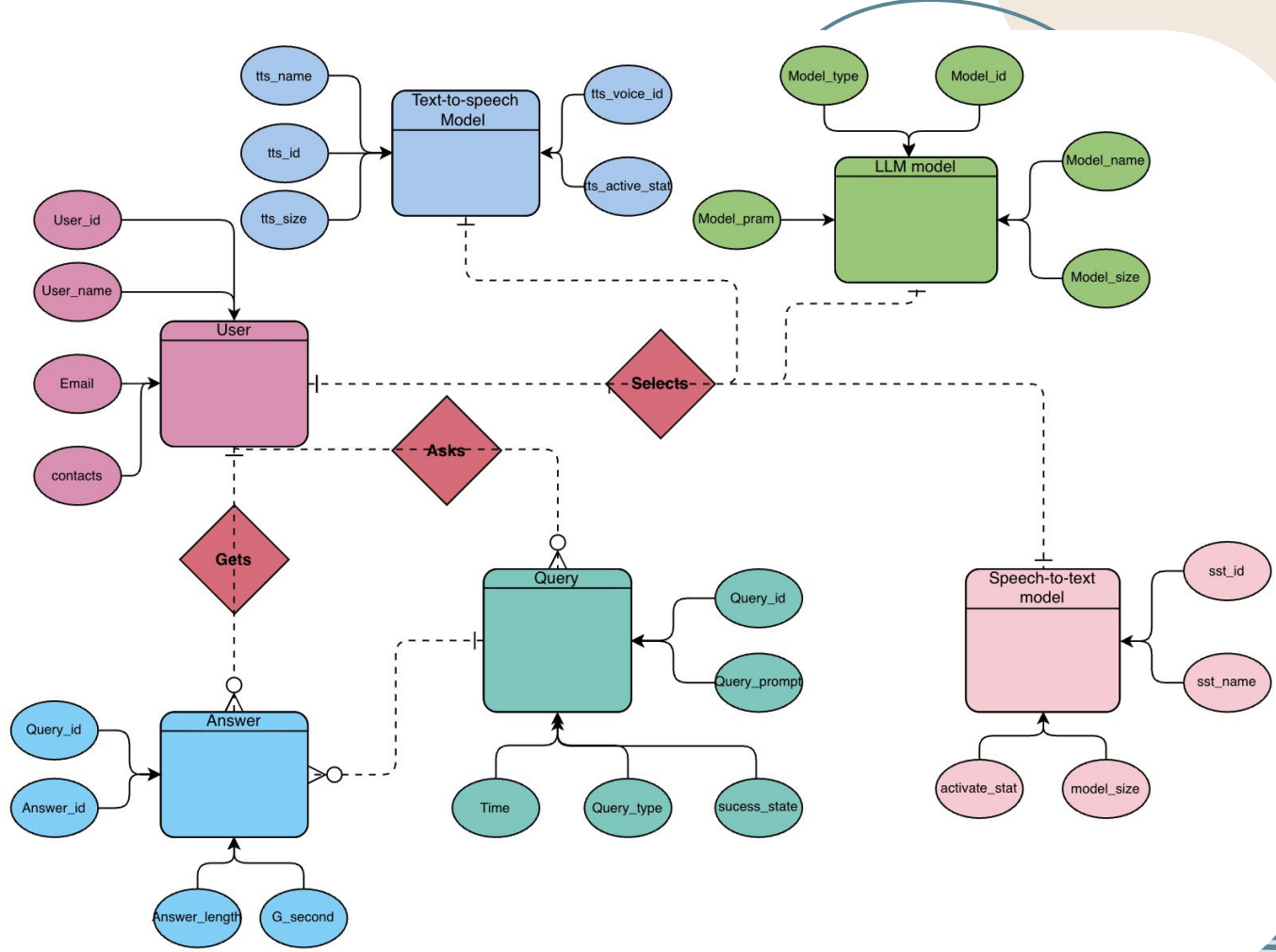


**Piper TTS**

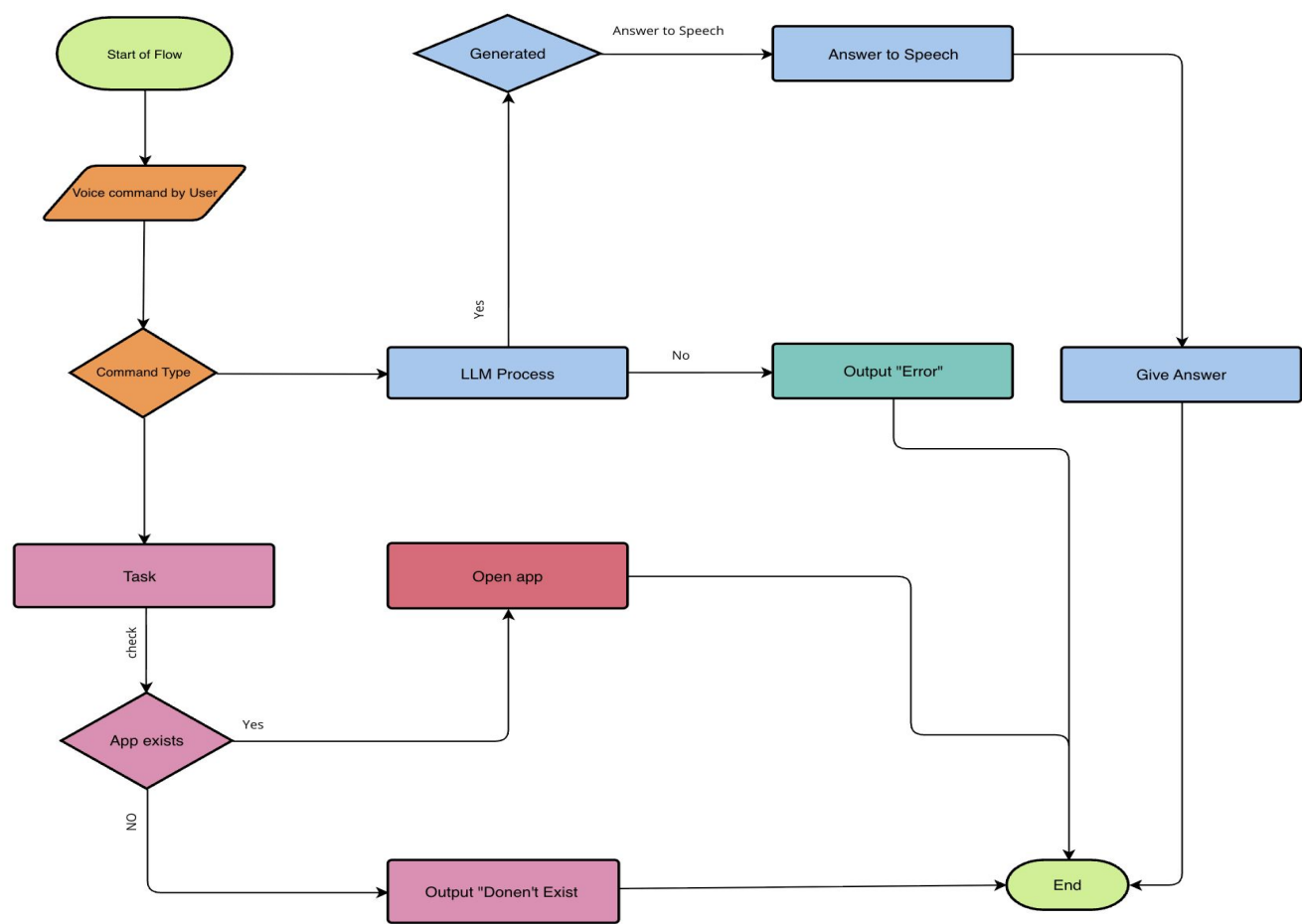


[huggingface.co](https://huggingface.co)

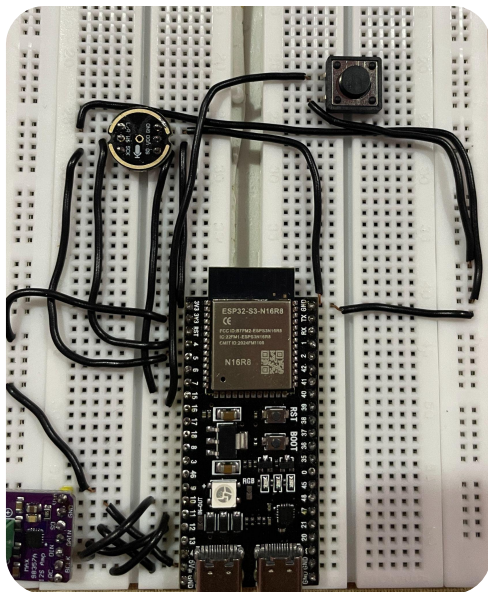
ER Diagram:



Flow chart:



## Screenshot and Modules:





```
1 import sounddevice as sd
2 from faster_whisper import WhisperModel
3 import wave
4 import ollama
5 import piper
6 import numpy as np
7 import subprocess as sb
8 #setting up the tts piper model for text to speech.
9 tts_model = '/Users/ansariaasif/Desktop/esp32/flask_server/voice_models/en_US-hfc_female-medium.onnx'
10 tts = piper.PiperVoice.load(tts_model)
11
12 #speech to text converting using whisper
13 whiser_model = WhisperModel('tiny',compute_type='int8',device='cpu')
14
15 samplerate = 16000
16 duration = 5
17 filename = "speech.wav"
18
19 while True:
20     print("# Listening...")
21     recording = sd.rec(int(duration * samplerate), samplerate=samplerate, channels=1, dtype='int16')
22     sd.wait()
23
24     with wave.open(filename, "wb") as wf:
25         wf.setnchannels(1) # mono
26         wf.setsampwidth(2) # 16-bit
27         wf.setframerate(samplerate)
28         wf.writeframes(recording.tobytes())
29
30     segments, info = whiser_model.transcribe('speech.wav', beam_size=1, vad_filter=True, language='en')
31     prompt = ''
32     for segment in segments:
33         prompt+=segment.text
34
35     print(f'Prompt: {prompt}')
36     answer = ''
37
38     lower_prompt = prompt.strip().lower()
39
40     if lower_prompt.startswith("open"):
41         app_list = {
42             "chrome": "/Applications/Google Chrome.app",
43             "code": "/Applications/Visual Studio Code.app",
44             "safari": "Safari",
45             "whatsapp": "/Applications/WhatsApp.app"
```





## Conclusion

**ZENO demonstrates how a desktop assistant can combine privacy, convenience, and intelligence while operating fully offline. By eliminating the need for internet connectivity, it ensures complete data security and protects users from the risks associated with cloud-based systems. With its ability to respond to natural voice commands, ZENO reduces the need to open a laptop and type for simple information, making everyday interactions faster and more effortless. Acting as a reliable desktop partner, it provides a seamless blend of functionality and security, proving that personal assistants can be both user-friendly and privacy-focused.**





## **Future Scope:**

**RAG Integration:** Fetch and process information from local documents/knowledge bases.

**Camera Addition:** Enable facial recognition, gesture commands, and visual input processing.

**Task Automation:** Add scheduling, reminders, and calendar management.

**Application & IoT Control:** Manage desktop apps and smart devices directly.

**Personalization:** Adapt responses to user preferences with context awareness.

**Multilingual Support:** Extend accessibility to a wider range of users.



## Reference :

1. **ESP32 Datasheet:** <https://www.espressif.com/en/products/socs/esp32/resources>
2. **Python WebSockets Library:** <https://websockets.readthedocs.io/>
3. **PlatformIO:** <https://platformio.org/>
4. **YouTube – Electronic Components Tutorials:** <https://www.youtube.com/@GreatScottLab>