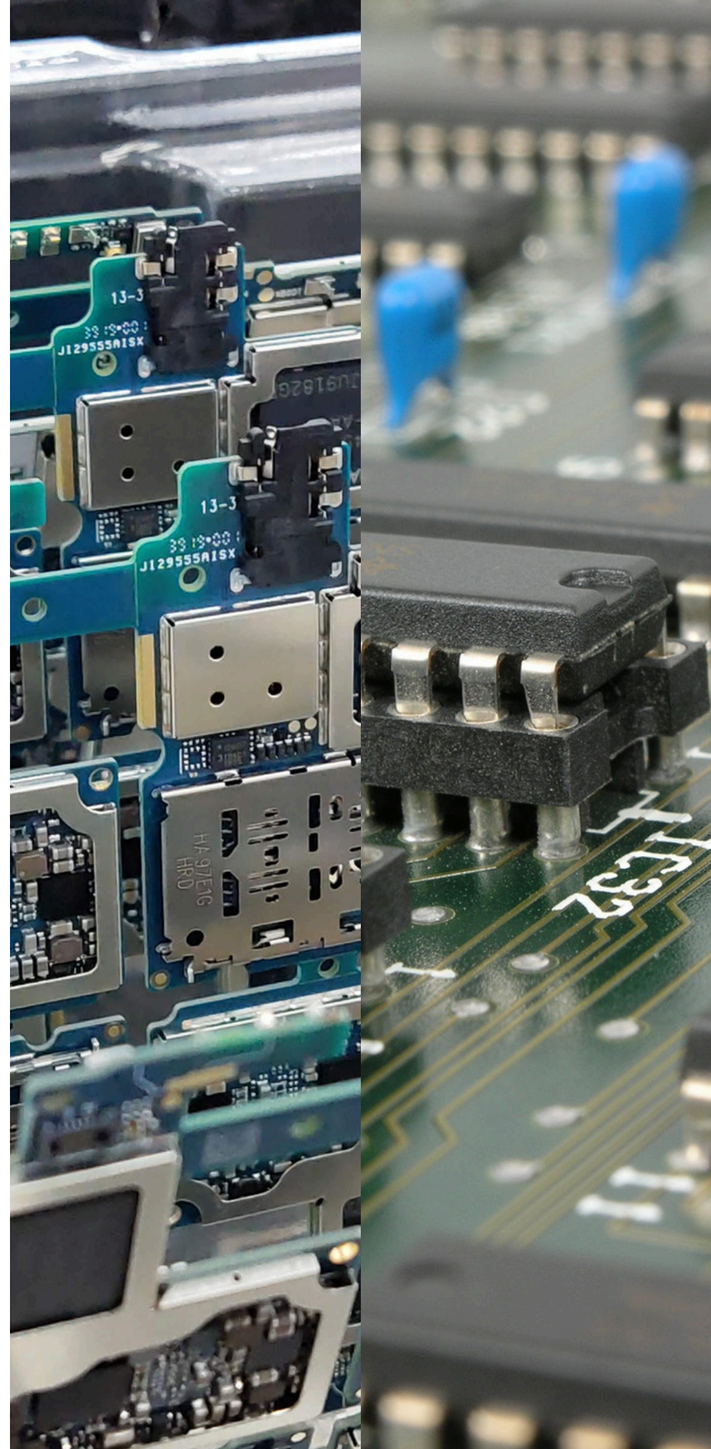
DESIGN OF A SMART MEMORY DEVICE FOR SPEECH AND TEXT PROCESSING

Neha Singh(2022UGEC038)
Somesh Ghosh (2022UGEC013)
Kaushiki Bhattacharyya (2022UGEC032)
Prince Agastya Jaiswal(2022UGEC005)
Shivakant Yadav(2022UGEC002)

INTRODUCTION

Memory-related challenges, such as those faced by people with dementia, require innovative solutions to enhance communication and day-to-day functioning. This project focuses on the design and implementation of a **Smart Memory Device** utilizing **Python and Large Language Models (LLMs)** to perform advanced speech and text processing. The system efficiently converts speech to text and text to speech, storing processed data securely for future use. It leverages pre-trained models like **SpeechBrain's speaker recognition system** to ensure high accuracy and robustness in speech-based applications. The device aims to bridge communication gaps by converting spoken words to text and vice versa, while securely storing information for future reference. Such a system can help individuals with dementia recall important conversations, access stored information, and interact more effectively with their surroundings.



LITERATURE REVIEW

The integration of Large Language Models (LLMs) like **Llama3**, as discussed in the article "**Llama3 RAG on Google Colab**", provides robust retrieval-augmented generation (RAG) capabilities for efficient data retrieval and context-aware responses, essential for intelligent memory systems.

LangChain, as introduced in its documentation, serves as a powerful framework for building applications that utilize LLMs and vector databases, enabling seamless orchestration of the device's memory management and text-processing functionalities.

In addition, the practical implementation of generative AI on edge devices, as described in "**Llama in a Box: Running Generative AI out of a Raspberry Pi**", highlights the feasibility of deploying resource-intensive AI models on compact hardware like the Raspberry Pi. This aligns with the project’s goal of developing a portable and scalable memory device.

The guide "**Setting up Raspberry Pi**" provides foundational steps for configuring the Raspberry Pi, ensuring a smooth hardware integration process.

Furthermore, "**Configuring Microphone and Speaker to Raspberry Pi**" offers valuable instructions for enabling audio input and output, critical for the speech-to-text and text-to-speech functionalities. These references collectively inform the design and development of the Smart Memory Device, combining the latest in AI-driven text and speech processing with practical implementation strategies on hardware platforms like Raspberry Pi to create a robust and user-friendly assistive tool.

OBJECTIVES

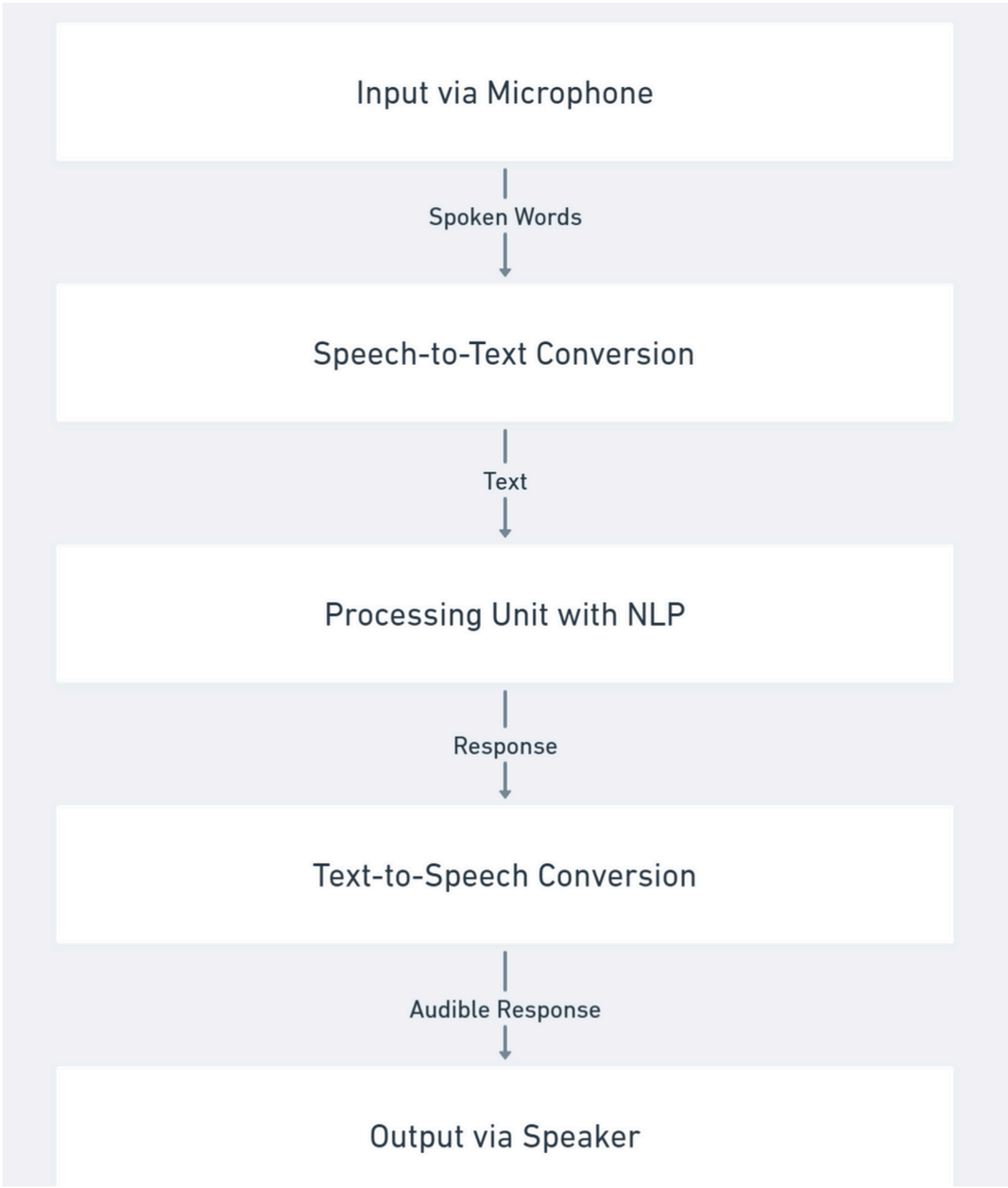
1. **Enhanced Communication:** The device bridges communication gaps by converting speech to text and text to speech, enabling individuals with memory challenges to recall and interact effectively.
2. **Secured data storage:** Secure data storage and retrieval empower users and caregivers to access important information, improving the quality of life for individuals with memory-related conditions like dementia.

ABOUT SMART MEMORY DEVICES

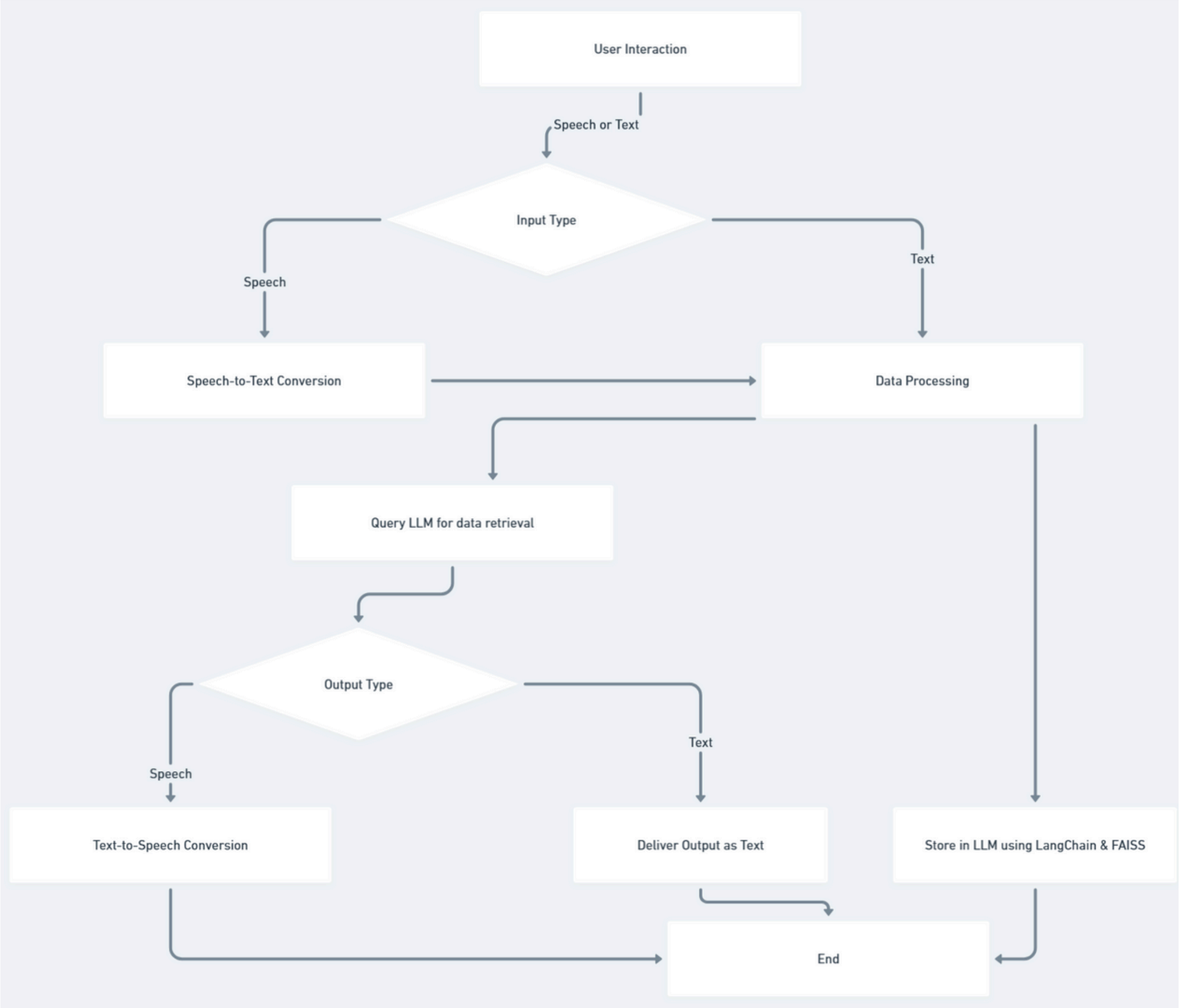
Smart Memory Devices are electronic systems designed to help users remember, store and recall information effectively. These devices focus on storing data from user interactions (like conversations) and providing easy access later. Our Smart Memory Device uses speech processing to capture spoken words, convert them to text and securely save them for later use. This approach enhances user interaction by allowing people to talk to the device and retrieve information verbally, making it a helpful tool for personal memory assistance

CORE COMPONENTS AND TECHNOLOGIES

- **Speech-to-Text Conversion**: This component allows the device to "listen" to spoken words and convert them to written text. Using the SpeechRecognition library in Python, the device captures audio through a microphone and processes it into text that can be stored or used for further functions.
- **Text-to-Speech Conversion:** The Text-to-Speech (TTS) function, powered by the pyttsx3 library in Python, allows the device to "speak" the text.
- **Speaker Recognition:** Speaker recognition, enabled by SpeechBrain’s pre-trained models, allows the device to identify individual users based on their voice.



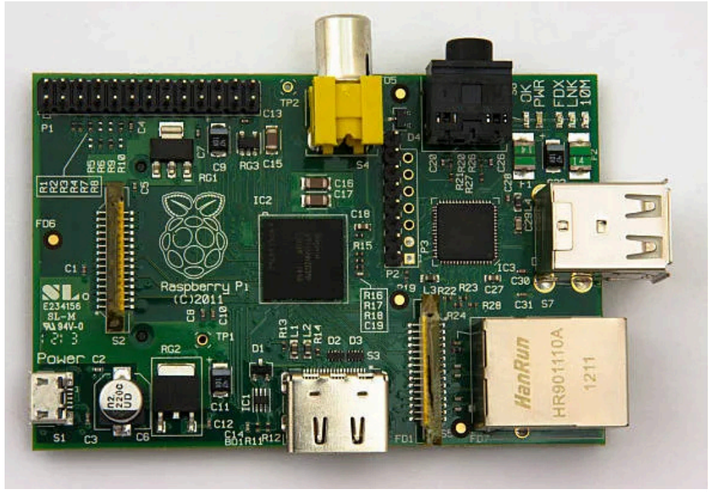
METHODOLOGY



WORKING AND METHODOLOGY

Our project leverages the Tiny Dolphin model to build an intelligent system enabling seamless speech and text interactions. Users provide input as speech or text, which is processed accordingly. Speech inputs are converted to text using advanced speech-to-text modules, while text inputs are directly processed. The data is then queried through the Tiny Dolphin language model, integrated via LangChain, ensuring accurate and context-relevant responses.

The system supports dual output modes—text or speech—based on user preferences. Outputs are either displayed as text or converted to speech using text-to-speech modules. Additionally, processed data and outputs are stored in FAISS for efficient retrieval. Deployed on a Raspberry Pi, this accessible system is designed for applications like assisting individuals with dementia, offering real-time responses, and improving user accessibility.



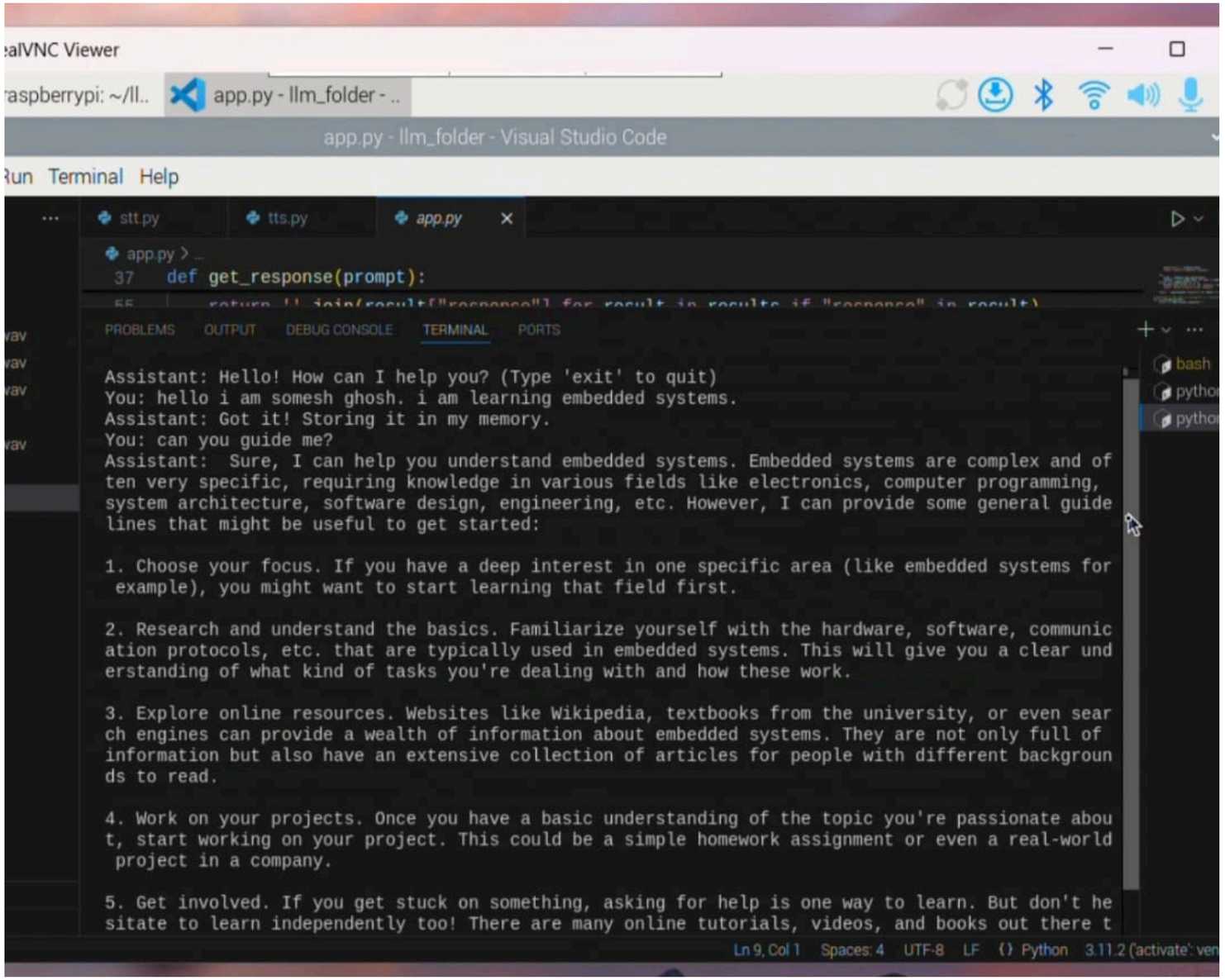


FIG : LLM MODEL OUTPUT

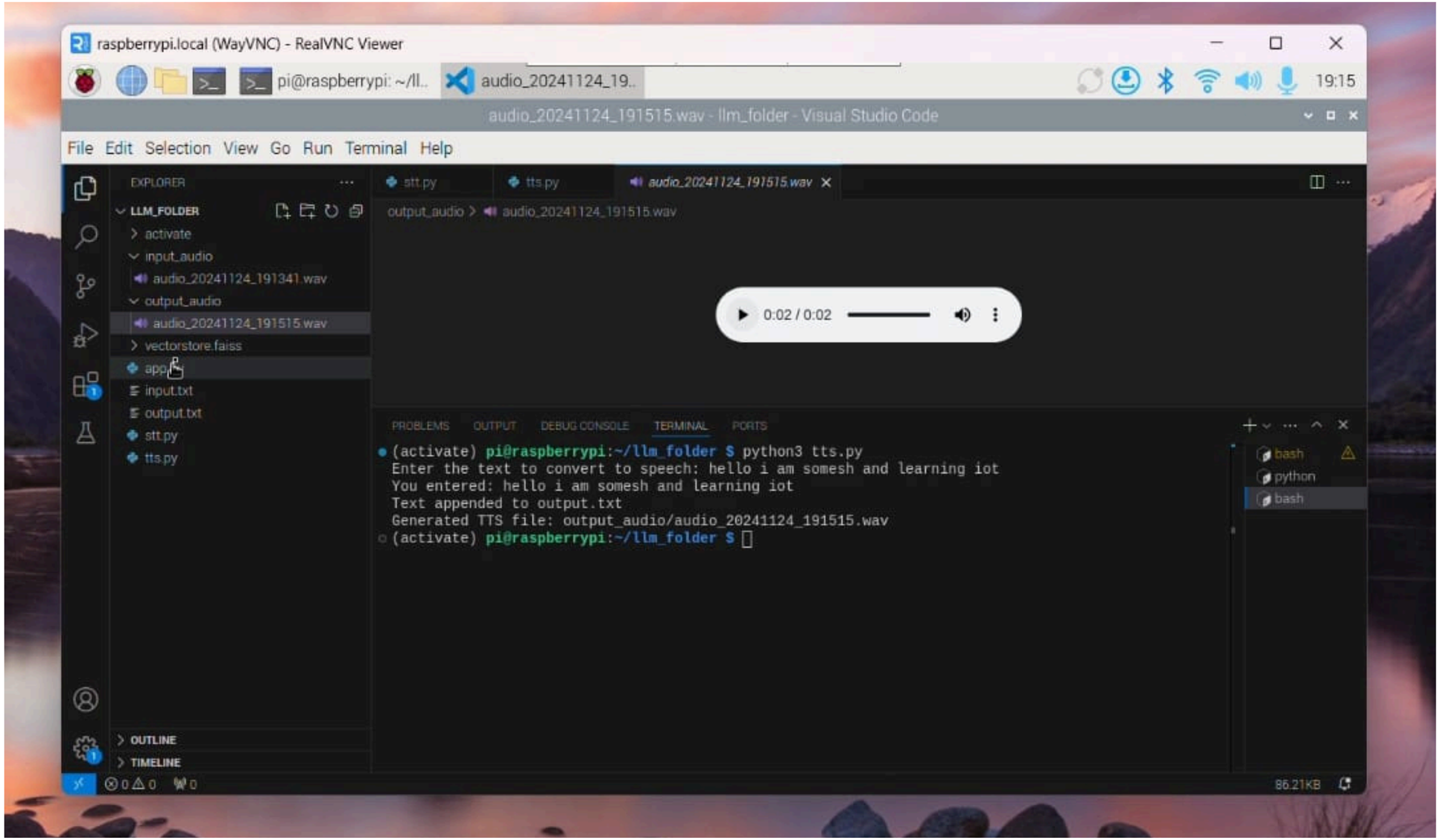


FIG : TEXT TO SPEECH OUTPUT

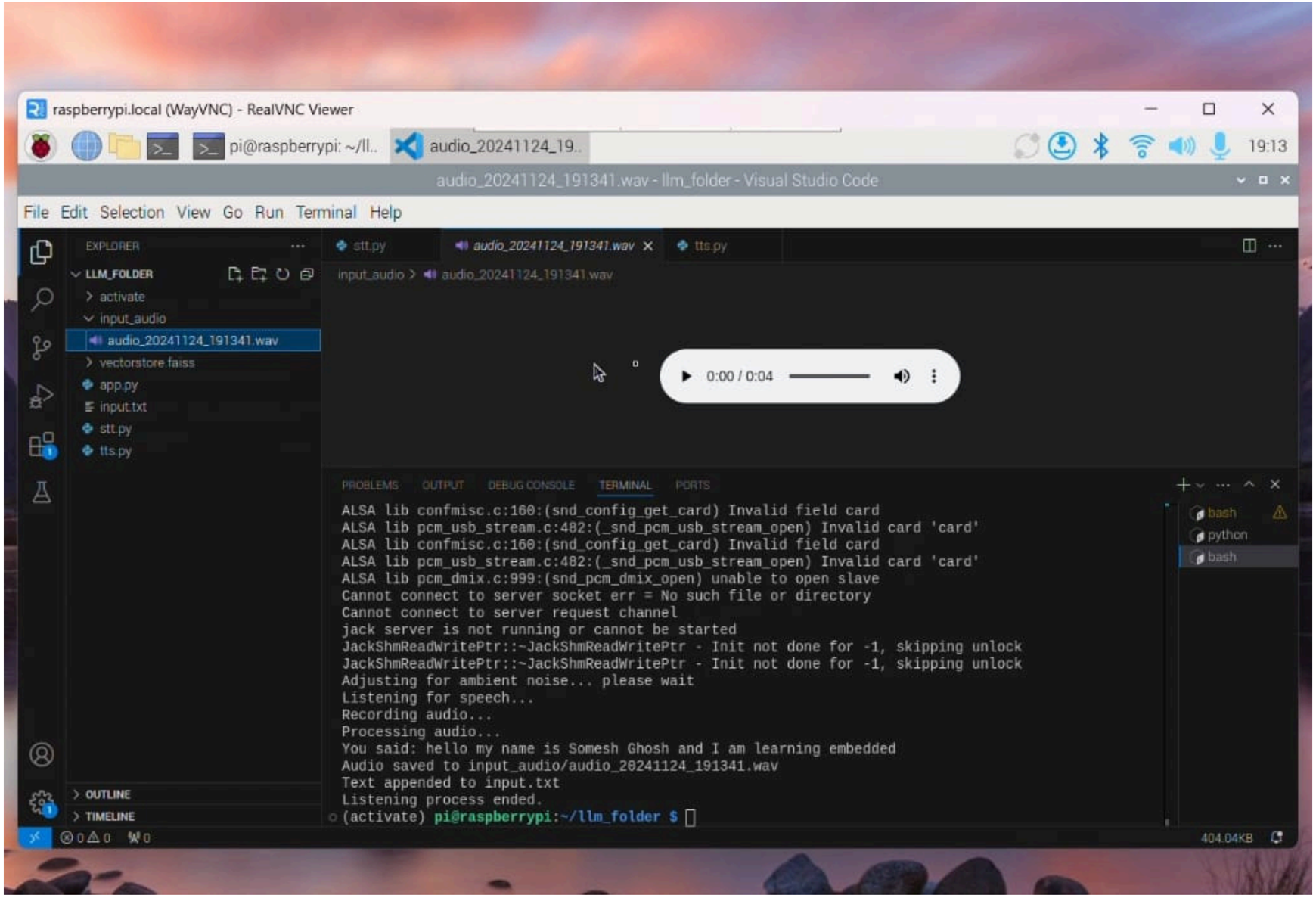


FIG: SPEECH TO TEXT OUTPUT

THANK YOU