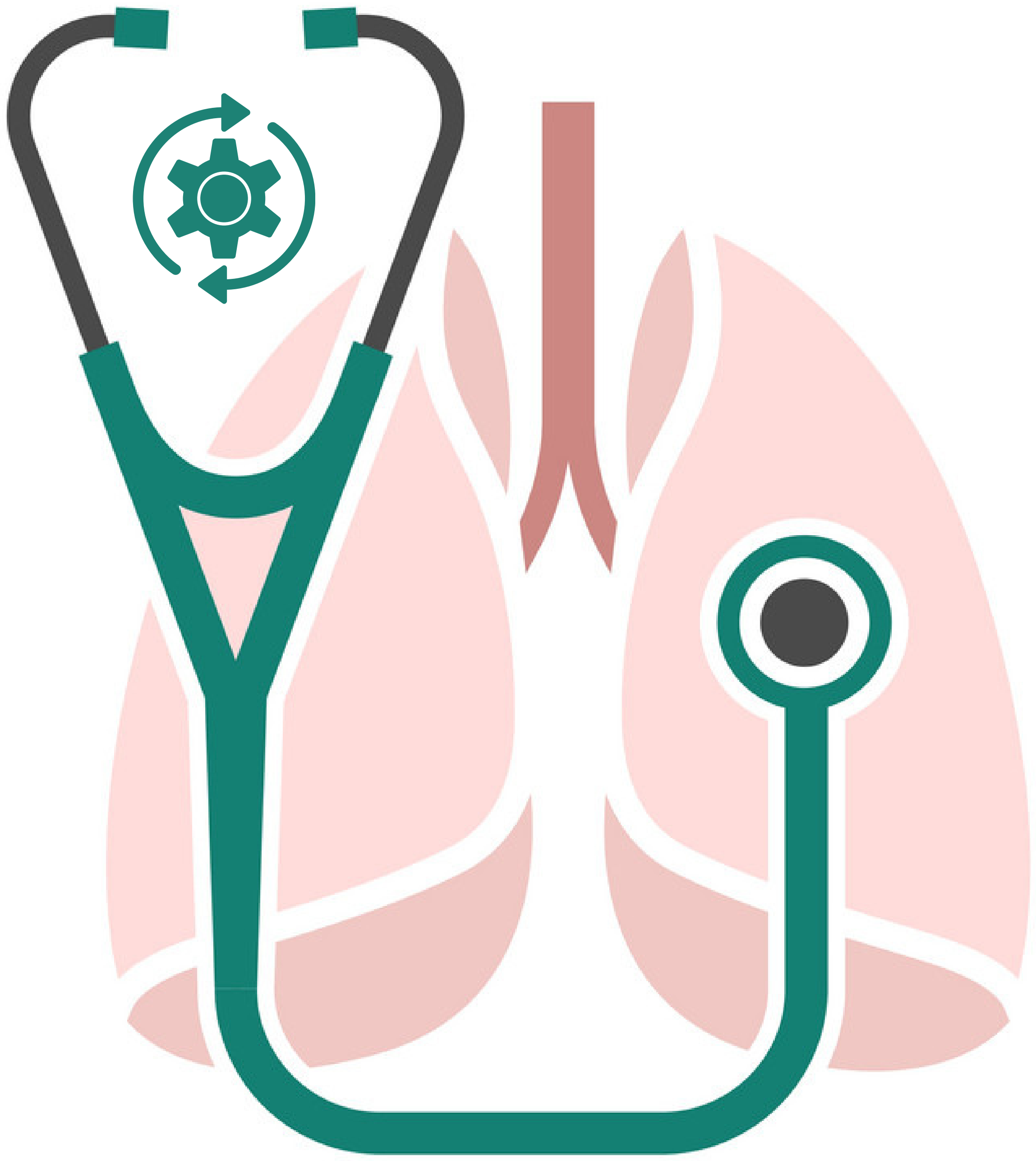


LUNG SOUND RECORDER

The Quad Chips

Team 38

Sai Praneeth - 2022101097

Jahnavi - 2022101118

Vaishnavi Priya - 2022101108

Sai Divya - 2022101090

Associated Professor - Abhishek Srivatsava

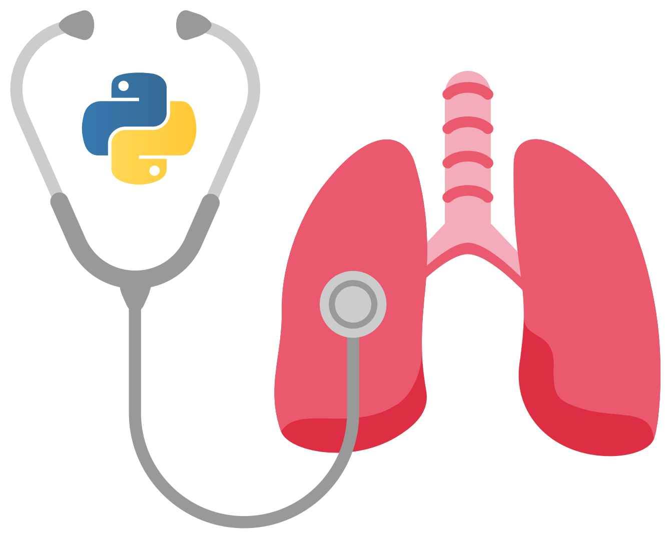
Associated TA - Santhoshini Thota

Motivation

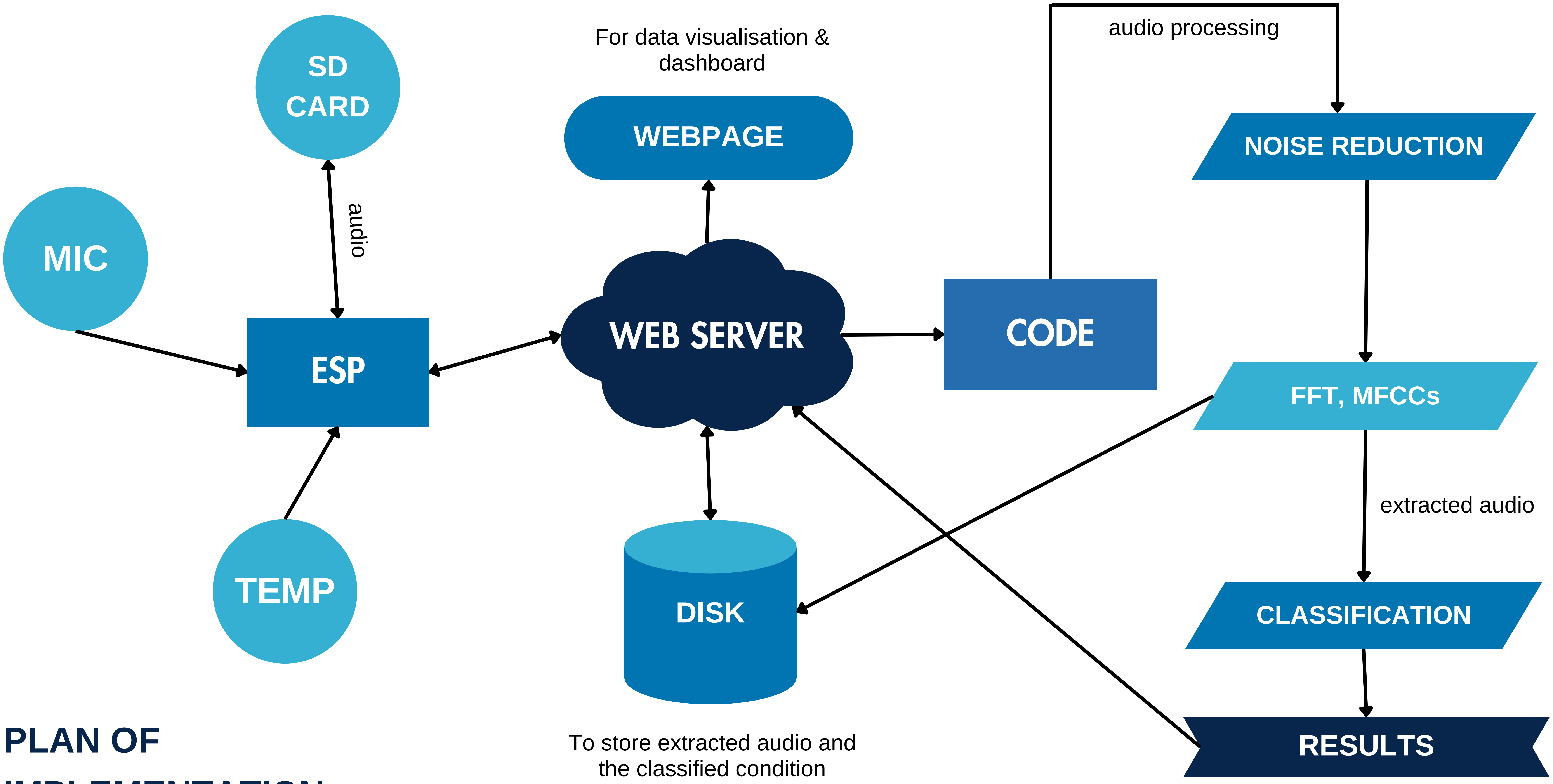
The early diagnosis of respiratory related diseases in children below the age of 5 is crucial

- for effective treatment
- lung sounds provide valuable
- essential to develop a device that can extract and record these sounds.

Aim



To design a portable healthcare device that will capture lung sounds and temperature readings using a microphone and stethoscope, and transmits them to dashboard for further analysis.



PLAN OF IMPLEMENTATION

Progress until now

NOISE REDUCTION

Wavelet denoising capacity to effectively suppress noise while retaining crucial respiratory characteristics

NORMALIZATION

Peak Normalization to maintain consistent loudness levels across different audio files

MEL SPECTOGRAM

Forming Mel spectrogram of the given audio. As a part of similarity matching, we segment the audio and form mel spectrograms for each.

CLASSIFICATION

Classifying based on similarity with the reference audios (wheeze, crackle, etc.)

Working

```python
import os
import pywt
import pywt.data
import librosa
import librosa.display
import numpy as np
import scipy.io.wavfile as wavfile
import matplotlib.pyplot as plt
from sklearn.metrics.pairwise import cosine_similarity

sample_rate, noisy_signal = wavfile.read('pnenoria.wav')

wavelet = 'haar'
level = 3
threshold_mode = 'hard'
coeffs = pywt.wavedec(noisy_signal, wavelet, level=level)

threshold_value = np.std(coeffs[-1]) * np.sqrt(2 * np.log(len(noisy_signal)))

coeffs = [pywt.threshold(c, threshold_value, mode=threshold_mode) for c in coeffs]

denoised_signal = pywt.waverec(coeffs, wavelet)

audio_file = "denoised_audio.wav"
wavfile.write(audio_file, sample_rate, denoised_signal.astype(np.int16))

# for similarity checking in segments
overlap = 0.2

def save_spectrogram(mel_spectrogram, path):
    mel_spectrogram_db = librosa.power_to_db(mel_spectrogram, ref=np.max)
```

Further Plans

Working on the new mic sensor:

Developing a front-end interface: Data visualization, admin controls - for setting the thresholds.

Real-time audio integration: Currently, audio is processed on some trigger. We will try to automate that by recording audios with some delay and processing it in between (alternating). And build an alerting system based on the real-time processed data.

Setting thresholds for each reference audio: We further try to set suitable thresholds for each reference (based on testing various sounds). We judge the existence of the sound based on that similarity (exist if > threshold).

Classifying diseases: Each disease has certain set of lung sounds. Based on similarities with the reference audios and thresholds, we classify.

Localized processing on a remote device: If time and resources permit

Challenges faced.

Finding reliable data set: Hard time finding the perfect dataset. Datasets usually have lung sounds related with diseases. But the reference audios we needed (wheeze, stridor, crackle, etc.) are from different dataset (or recorded from a different environment). Due to this, there are problems in classifying.

Coincidence of sound in multiple diseases: Challenging to differentiate between different diseases based solely on the sounds heard.
For example: wheezing or crackling sounds in the lungs can be indicative of various conditions, including asthma, bronchitis, pneumonia etc.

Problems setting up the hardware: We didn’t get the mic sensor we needed. Lately, after trying a lot, we realized that given sensor isn’t suitable for recording. And we didn’t receive the required sensor in time. So, had to concentrate on the software implementation.

Project Deliverables

Integration with ESP32 for Real-time Data Collection

Development of User-Friendly Dashboard for Data Visualization

Classifying the lung sounds - outputting the related disease

Implementing a localized diagnostic device

Individual Works

Sai Praneeth:

Studying Lung sounds, Working on implementation of Hardware,
Testing of hardware components,
Collection of datasets,
Improvised the similarity matching code - by a segmenting technique
Sending data to ThingSpeak, server handling

Jahnavi:

Testing of temperature sensor,
Researching about FFT,
Studied on computing STFT and mapping it onto MEL scale,
Spectrogram images, MFCC derived from MEL spectrogram (code),
Testing of the final integrated code, made videos for the same.

Sai Divya:
Applying noise reduction on sample audios,
Coming up with efficient ways of noise reduction,
Normalization (code), Studying FFT,
calculating similarities between the reference audios and generated MEL (code)
Testing Similarity matching with lung sounds in the dataset.

Vaishnavi:
Preprocessing of Reference audio files,
Wavelet denoising code (code),
Generating reference MELs, MFCCs for Data set for classification,
Studying abnormal Lung Sounds, Peak and Loudness Normalization.
Testing noise reduction code with recorded audio.

Thank you!

Team 38, Group 4