



Using the Web Audio API to Create a Background Noise Test in the Context of Respiratory Audio
Capture via a Web Browser

by
Justin Hendy

Supervisor: Nathan Morgan
Second Marker: Tim Bashford

Project submitted as part of the requirements for
the award of MSc Software Engineering (Web Development)

January 2026

Contents

1	Introduction	3
1.1	Aims and Objectives	3
1.1.1	Aim	3
1.1.2	Objectives	3
1.2	Report	3
2	Literature Review	4
2.1	Respiratory Diseases and their Audio Biomarkers	4
2.1.1	Asthma and Chronic Obstructive Pulmonary Disease (COPD)	4
2.1.2	Asthma and COPD Audio Biomarkers	4
2.2	Remote Audio Capture for use in Clinical Settings	6
2.2.1	Shift Towards Digital and Remote Monitoring	6
2.2.2	Currently Available Systems	7
2.2.3	Issues Associated with Capturing User-Controlled Audio Recordings	7
2.2.4	The Web Audio API	8
2.2.5	Noise Reduction Techniques	9
2.3	MERN Stack	11
2.4	Why Web Audio API with DSP?	12
3	Methodology	12
3.1	Introduction	12
3.2	Research Methodology	12
3.3	Analysis and Design Activities	13
3.4	Evaluation Criteria	13
3.5	Materials and Tools	13
3.6	Procedures	14
3.7	Requirements and Design Documentation	14
4	Design	14
5	Implementation	14
6	Results and Discussion	14
7	Conclusions and Recommendations	14

1 Introduction

Respiratory diseases and conditions continue to place a heavy burden on global populations. Data from the Global Burden of Disease 2023 study estimates that as many as 569.2 million people manage some form of Chronic Respiratory Disease (CRD), with approximately 4.2 million CRD-related deaths in the same year. In addition to personal costs, respiratory conditions impose a continuing economic burden on governments and health care providers worldwide. The National Health Service spends an estimated £800 million per year on the management of Chronic Obstructive Pulmonary Disease (COPD). In the US, estimates place COPD costs growth at \$40 billion over the next 20 years and, in 2023, respiratory disease direct costs within the European Union accounted for an estimated 6% of its total healthcare budget. The ability to continue supporting and monitoring conditions remotely is an ever-growing requirement of modern medicine, as acutely demonstrated by the 2020 COVID-19 pandemic. The need for instant personal monitoring and greater connectivity between patients and healthcare professionals has seen an increase in web-based and mobile-based health and monitoring tools, with 2019 estimates placing consumer spending on digital health products at approximately \$49 billion.

Several respiratory-based solutions on the market use proprietary hardware and/or smartphone applications such as StethoMe, iMedic, Felix, Eupnoos, and Hyfe Cough Tracker. There are a few web-based services, such as kopo.com’s breathing analyser and AudibleHealth AI via coughscreening.com; both complete the audio capture process via mobile phone. These options highlight the growing use of audio capture in respiratory and clinical contexts. Given the accessibility advantages of a web-based system, there is a need for a purely web-based service that offers, at the very minimum, web-based audio capture via desktop or mobile devices. The Web Audio API, available across browsers since April 2021, is a tool for real-time audio capture and processing in a web browser and provides a system suitable for developing such a product. This project will complete an engineering investigation into whether noise-responsive respiratory audio recording is attainable within the constraints of browser-based audio systems using the Web Audio API, when integrated into an adaptive, semi-automated digital signal processing pipeline with quality checking and post-processing.

1.1 Aims and Objectives

1.1.1 Aim

Design and implement a web-based audio recording system using the Web Audio API that guides the user through an environmental test to assess background noise interference.

1.1.2 Objectives

- Design an audio recording workflow with a microphone setup guide for use within a web browser
- Implement environment testing with additional noise profile creation where necessary
- Add fixed lowpass and highpass filters to limit bandwidth to the respiratory audio frequency range
- Produce multiple output files representing each stage of the process

1.2 Report

Following this introduction is a literature review outlining the current burden of respiratory diseases and the need for greater access to remote detection and management systems, and examining the Web Audio API and noise suppression techniques. The methodology follows the literature review, which

provides an outline of the procedures used to generate a solution for background noise detection using the Web Audio API. The following chapter outlines the project’s design stage, detailing the decisions made in order to convert requirements into actionable tasks. The system implementation follows the design chapter, providing information on the tools and techniques used to develop the product, as well as details on the issues encountered. The results and discussion chapter describes the finished, working product and evaluates its performance against the given aims and objectives. Finally, a conclusion and recommendations chapter provides a summary of how the actions taken throughout this project contributed to meeting the objectives set out in the introduction, as well as a reflection on the process as a whole.

2 Literature Review

2.1 Respiratory Diseases and their Audio Biomarkers

2.1.1 Asthma and Chronic Obstructive Pulmonary Disease (COPD)

In 2017, global estimations put the number of people suffering from a chronic respiratory disease (CRD) at roughly 550 million, representing a 40% increase when compared with figures from 1990, with CRD-related deaths totalling roughly 4 million, making it the third leading cause of death worldwide [1]. More specifically, Hang et al. reported 41,555,628 new asthma cases in 2021, of which Western Europe accounted for 1.6 million. These figures represent a slight decrease between 1990 and 2021. Although asthma-related deaths during the same time period have increased, the mortality rate, how often a death occurs in a specific time period, has decreased from 9.64% to 5.2% per 100,000 [2]. Hang et al. conclude that asthma remains highly prevalent, especially in countries with higher Social-demographic Index (SDI) and among individuals under the age of 9, where SDI is a measurement of average earnings per person, schooling level and fertility rates [3]. Global figures for COPD cases vary due to differing definitions over the years, with estimates ranging from 212 to 400 million worldwide [4] [5]. Boers et al. estimate that respiratory disease direct costs in the European Union account for 6% of total annual healthcare expenditure, with 56% (€38.6 billion) allocated to COPD. Similarly, estimates relating to the United States’ annual COPD costs indicate an increase to \$40 billion over the next 20 years. Within the UK, estimates place the number of people affected by COPD at around 3 million; however, as previously mentioned, COPD estimates are notoriously variable [6][7]. Rayner et al. also highlight the estimated cost to the NHS of around £800 million per year, with annual deaths of 30,000. Scully et al. sought to standardise datasets on the number of new cases of asthma, COPD and ILD (Interstitial Lung Disease) across England, Scotland and Wales, per 1,000 people per year. For 2019, England, Scotland, and Wales, respectively, saw an estimated 50,000, 3,700, and 2,000 new asthma cases; 46,000, 5,800, and 2,100 new COPD cases; and 183,000, 9,000, and 4,300 new ILD cases [8]. Both asthma and COPD continue to carry a significant health and cost burden in both the UK and global populations.

2.1.2 Asthma and COPD Audio Biomarkers

A 2024 study by Lozano-Garcia et al. sought to demonstrate the feasibility of measuring respiratory sound intensity (i.e., signal amplitude) for diagnosing and monitoring COPD, citing work by Fiz et al., who demonstrated a link between the two, particularly during inhalation [9]. Lozano-Garcia et al. used four piezoelectric contact microphones that detect sound via the movement of solid objects rather than the displacement of air particles. They placed them at various points on the subject’s skin. They concluded that such measurements could detect a reduction in airflow performance among COPD patients compared with healthy patients, and that the results were consistent with peak flow

measurements obtained using the traditional spirometry method [10]. Ashraf and Moussani compared the performance of a purpose-built piezoelectric microphone and a standard commercial omnidirectional condenser microphone when used to capture tracheal breathing sounds. Although they found that the piezo microphone outperformed the condenser microphone in noisy environments, the measured power spectral density (PSD), or distribution of power across frequencies, did not differ much under normal conditions, as evidenced by the two charts below [11].

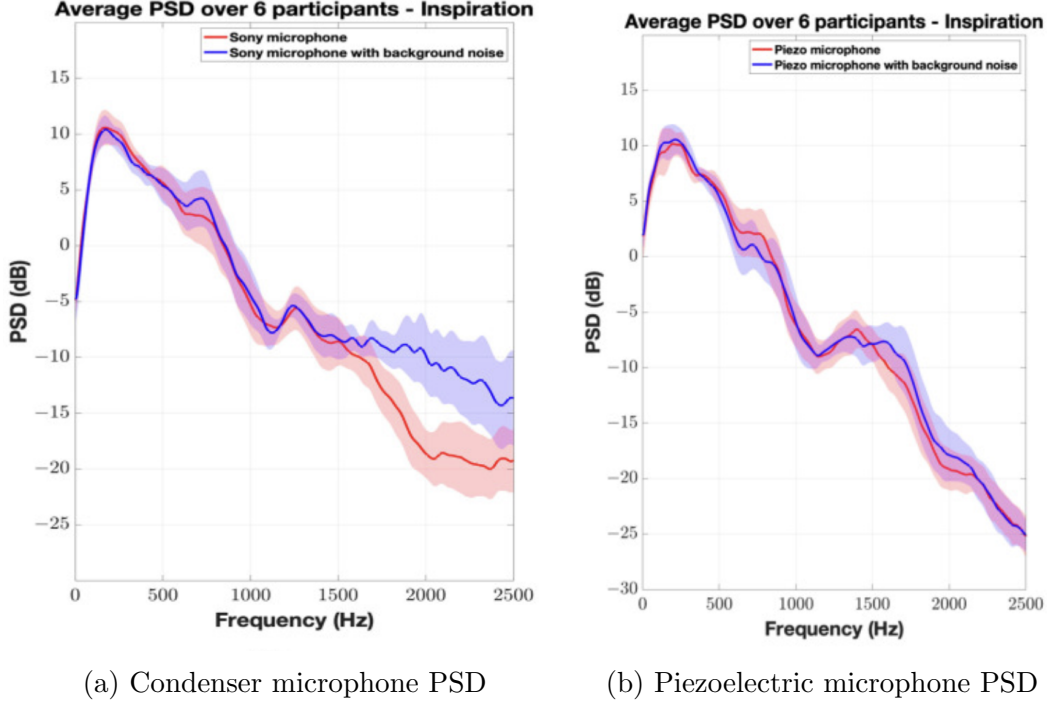


Figure 1: Comparison of PSD [11]

With this in mind, it is safe to assume that standard microphones can also measure respiratory sound intensity, albeit with reduced performance in noisy environments.

B.T. et al highlight that typical asthmatic symptoms include coughing, wheezing and dyspnea (difficulty breathing) [12]. To develop a system for classifying asthmatic and healthy children, B.T. et al. used recordings of cough and /:a/ (ah) sounds to train a machine learning model. Results showed that the model successfully aligned with clinical decisions, achieving 95.3% accuracy for cough sounds and 72.2% for /:a/ sounds; however, the cough model utilised 2029 samples, whereas the /:a/ model used only 333. All samples were captured using a smartphone and recorded in a noisy hospital environment. Vowel sound classification uses the identification of three formants or spaced peaks in a frequency spectrum plot and their relative intensities [13]. The chart below demonstrates average resonances of each vowel sound for men, women and children, where /:a/, as used in the word 'had', has a range of 660 Hz for the first formant for men, up to 3320 Hz for the third formant for children [14].

	Formant	heed	head	had	hod	haw'd	who'd
Men	F1	270	530	660	730	570	300
	F2	2290	1840	1720	1090	840	870
	F3	3010	2480	2410	2440	2410	2240
Women	F1	310	610	860	850	590	370
	F2	2790	2330	2050	1220	920	950
	F3	3310	2990	2850	2810	2710	2670
Children	F1	370	690	1010	1030	680	430
	F2	3200	2610	2320	1370	1060	1170
	F3	3730	3570	3320	3170	3180	3260

Figure 2: Formant Chart

Acoustic features of cough sounds include the amplitude envelope, i.e., changes in amplitude over time, and the relative intensity of grouped frequencies [15]. Ghrabli et al. studied the "unique spectral fingerprints in cough sounds" and deemed it unnecessary to capture frequencies beyond 4000 Hz, evidenced by their use of a lowpass filter set to said frequency, and with their resulting analysis focusing on bands of 50 Hz to 900 Hz, 450 Hz to 1000 Hz and 1400 Hz to 1600 Hz of particular interest in distinguishing between pneumonia and asthma [16]. Both studies demonstrate that the spectral significance of respiratory audio lies within a specific range, from 50 Hz at the lower limit to 4000 Hz at the upper limit.

2.2 Remote Audio Capture for use in Clinical Settings

2.2.1 Shift Towards Digital and Remote Monitoring

Due to the importance of early detection and monitoring in improving quality of life and reducing hospitalisation, remote assessment via electronic devices is increasingly prevalent across various healthcare domains [17]. Tools, such as speech-based remote monitoring of asthma and COPD exacerbations, hold great promise for diagnosing and monitoring these conditions and could reduce the burden on patients and healthcare providers [17]. The ability to capture audio samples via a mobile phone has also led to the potential development of at-home spirometry tests that, when used in conjunction with AI deep learning models, can accurately correlate measured forced expiration audio to lung functionality, albeit with an error rate that is currently too high for clinical settings [18]. Alsagri et al., through their systematic literature review of emerging technologies and trends in lung volume estimation, highlight the need for more rigorous clinical testing and standardisation when assessing the impact of remote tracking devices on a patient's health and care in order to fully realise the full potential of digital and remote monitoring [19]. Mahboubian et al. assessed the perspectives of seven key groups involved in COPD care regarding remote patient monitoring (RPM) [20]. They found that RPM offers significant opportunities to improve care, particularly in ongoing assessment between visits, resource allocation, and symptom tracking. However, health care providers noted concerns about the usability of technology and the need for clear guidelines. The key concern for patients is any change in traditional practice, such as "reduced in-person care and overreliance on data" [20]. Yan et al. discuss the increase in the acceptance and use of smart devices in clinical settings and how the development and understanding of digital biomarkers provide opportunities to develop new diagnostic and monitoring methods. COVID-19 and the subsequent pandemic ushered in a new era of development in home-based mobile health monitoring, and the existence of telehealth and RPM will provide opportunities to improve care practices more equitably [21]. Digital and remote monitoring solutions will play a key role in future healthcare developments,

and the ability to collect and correlate accurate data will be vital to safely and correctly introduce any such system into clinical use.

2.2.2 Currently Available Systems

There are existing mHealth apps for respiratory disease detection and management that focus entirely on audio capture and analysis, such as the Hyfe Cough Tracker, a smartphone-based AI cough monitoring app that can detect, record, and count coughs from cough-like sounds when used in acoustically controlled environments [22], the iMedic application, a smartphone-based self-auscultation tool for AI-Powered pediatric respiratory assessment [23] or Eupnoos, a breath analysis tool described as "Shazam for diagnosing respiratory diseases" [24]. Several applications utilise connected devices such as StethoMe and Felix. Both StethoMe and Felix are smartphone applications that connect to a wireless stethoscope and utilise AI to provide in-home examinations. Welch et al. concluded that the Eupnoos application has "promising" accuracy for diagnosing asthma or COPD, but cited limitations due to the small number of disease cases used [25]. Similarly, for StethoMe, scheduled clinical trials have not yet commenced at the time of writing [26]. To further highlight this point, Safavi et al. estimated in 2019 that by 2020, consumers would spend approximately \$49 billion on digital health products; however, they found that the majority of top-funded US-based digital health companies did not measure the "clinical effectiveness" of their products or assess their use within populations considered "high-cost, high-burden" such as those suffering with chronic illnesses, older in age or diagnosed with some form of disability. Of the 156 studies, the largest group evaluated healthy patients (34); however, only 5 sought to assess clinical effectiveness. The four respiratory-related areas of asthma, COPD, ILD (Interstitial Lung Disease), and influenza garnered a total of eight studies. Of these eight, three were asthma-related, and two were COPD-related; all five studies focused on patients deemed to be high-cost/high-burden, but only two of the three asthma studies evaluated clinical effectiveness [27].

There are a few web-based services, such as kopo.com's breathing analyser and AudibleHealth AI via coughscreening.com. Kopo.com offers both smartphone and web browser capture processes; however, it only detects conditions in children aged from 6 months to 7 years. AudibleHealth AI, whilst accessible through a web browser, only offers audio capture via mobile phone.

2.2.3 Issues Associated with Capturing User-Controlled Audio Recordings

There are several issues associated with end-user or patient-captured audio recordings. In their study examining the use of cough AI systems to detect COVID-19, Khanzada et al. found that varying audio quality was the most significant barrier to developing an algorithm for this task, with particular mention of poor recording environment associated with high levels of background noise, distorted audio as a result of signal clipping (audio amplitude levels going beyond what the system is capable of handling), audio compression and wide ranging microphone quality [28]. Other issues include the need for a sufficient signal-to-noise ratio (SNR), a comparison of the wanted signal level to the unwanted signal level. When using audio recordings to determine proper inhaler use amongst patients with asthma or COPD by measuring peak inspiratory flow rate, inspiratory capacity and inhalation ramp time, Taylor et al. found that 13 dB (decibels) represented an acceptable SNR, providing 80% accuracy when estimating inhalation parameters; however, lower SNR levels, such as 10 dB, were still able to provide accuracy of around 70%. Busquet et al. discuss the importance of microphone placement, in terms of distance and angle to the user, having a significant impact on the consistency and quality of the captured audio recording [29]. Distance can affect spectral quality through the proximity effect, an increase in low-frequency spectral content caused by the lack of distance between the subject and the audio device [30], and the angle of the speaker to the device will also cause unwanted spectral changes to the

recorded signal, as will the microphone’s stated frequency response [29]. Below is an example polar pattern and a frequency response for a consumer condenser microphone, with the polar pattern showing changes in performance as a function of angle to the microphone transducer, and the frequency response demonstrating changes in the signal at different frequencies [31].

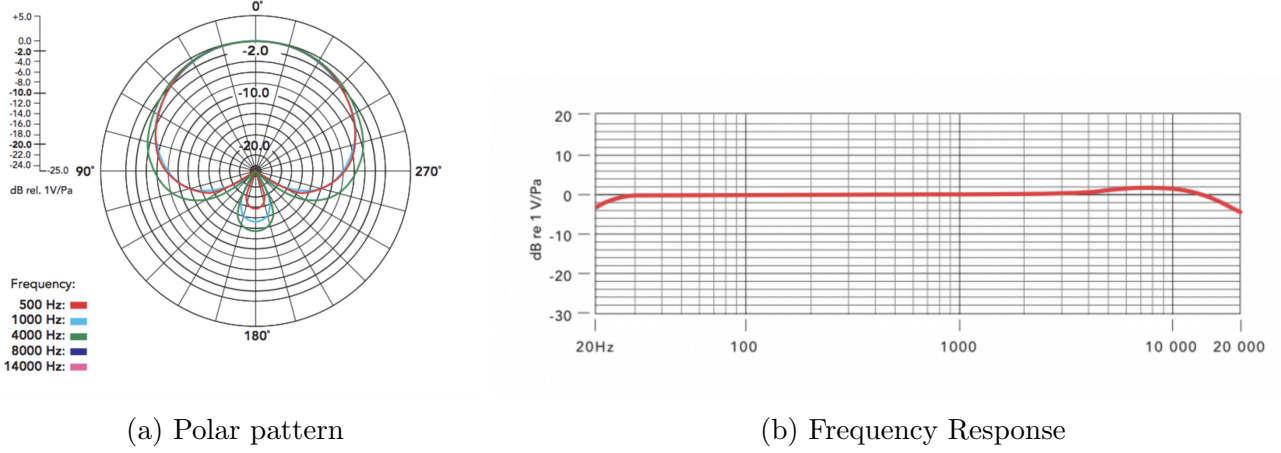


Figure 3: Rode NT1-A Polar Pattern and Frequency Response [31]

Given the current trends in remote diagnosis and management of respiratory conditions, it is clear that to fully realise the potential of user-controlled audio recordings in this setting, whether for training AI models or for human analysis, minimising the impact of these issues is vital to any such endeavour.

2.2.4 The Web Audio API

The Web Audio API, introduced in June 2021, is a W3C official web standard for audio creation and manipulation. It provides users with several "modular building blocks" that require no additional plugins, enabling the creation of a wide variety of audio-based applications for both websites and web applications on mobile and desktop devices [32]. Being a W3C standard has positive implications for long-term support and stability when creating a new platform. Further benefits include support across all major desktop and mobile browsers, without requiring users to install any additional extensions or natively installed applications, dramatically increasing accessibility and adoption, especially for users who are inexperienced or not comfortable installing software [33]. In addition to browser compatibility, Web Audio API applications can be deployed as part of Progressive Web Applications (PWAs), furthering their accessibility and use cases [34]. In 2017, Jillings et al. demonstrated the potential of the Web Audio API by developing a purely web-based digital audio workstation (DAW), using AngularJS for the user interface (UI), Web Audio for the audio engine, and a relational database for storage. A DAW is a tool for recording and manipulating multiple audio channels for playback and balancing, used across a variety of music technology disciplines such as music production and sound design. They further highlight the potential for research and analysis by collecting data on user interactions and decisions within the field of music technology [35].

The key component of the Web Audio API is the audio graph and the ability to build a modular system of nodes. The API provides access to tools for manipulating and analysing audio streams, including filtering, gain control, and real-time analysis. Examples of such nodes include BiQuadFilterNode for creating highpass and lowpass filters, the GainNode, and the AnalyserNode. End-to-end connectivity between inputs and outputs is available via the MediaStreamSourceNode and AudioDestinationNode,

which allows for connection to speakers and/or analysis streams all within the confines of the web browser [36] [37].

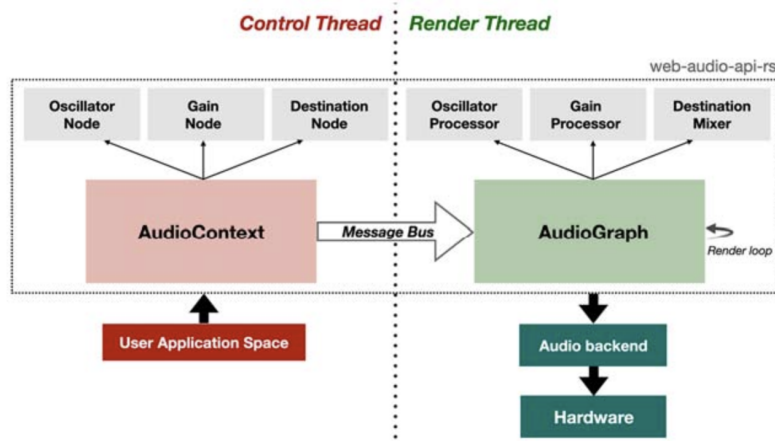


Figure 4: Web Audio API Architecture [36]

The ability to programmatically access a user’s microphone and manipulate and analyse any incoming audio stream provides developers with the opportunity to create in-browser real-time digital signal processing pipelines that can provide user feedback and support automated or user control. AudioWorklets further extend the adaptability and potential of this API by providing the ability to develop DSP processors that are not part of the standard, which run on a separate thread to the user interface (UI), providing additional low-latency (a delay between input/action and output/response) control over the audio stream [37]. The analysis node’s ability to provide live data enables the development of detailed user feedback UIs, such as waveform generation, level meters, spectrograms, etc. [33]. A W3C paper from 2013 outlines several use cases and requirements for the API standard, highlighting its potential for real-time audio manipulation, a clear indicator that the API provides opportunities to create tools for guiding and correcting user microphone usage before recording a respiratory sample for submission [38].

2.2.5 Noise Reduction Techniques

The ability to improve audio intelligibility and accuracy without adversely affecting the primary signal has many uses in speech and audio enhancement, and over the years, a large number of speech enhancement algorithms (SEAs) have emerged [39]. There are several ways to categorise SEAs, with a key definition being whether they require single-channel or multi-channel audio capture [39] [40] [41].

Adaptive noise cancelling is one such multi-channel method that, as its name suggests, can adapt to a changing noise signal and attempt to subtract the noise captured using a secondary input from a primary input containing both noise and wanted signal [42]. Given the limitations of home audio setups regarding multi-channel audio capture, this solution is not suitable within this context [43].

Basic filtering, the removal of unwanted frequencies using static filters set to specific frequencies, can provide an element of noise reduction and have several benefits [44]. As previously mentioned, they are easy to implement using the Web Audio API via the BiQuadFilterNode. In addition, they have a low computational cost due to their relatively simple arithmetic algorithm (see figure below), providing stable, predictable outputs for given inputs, which, in turn, makes them suitable for real-time audio pipelines [45] [46].

$$y_t = b_0x_t + b_1x_{t-1} + b_2x_{t-2} - a_1y_{t-1} - a_2y_{t-2}$$

Figure 5: Infinite Impulse Response (IIR) Difference Equation

While these benefits certainly have a place in any audio processing pipeline dealing with limited-bandwidth audio, such as respiratory sounds and speech, there are several limitations when specifically attenuating fixed-frequency bands in a recording. The main issue with basic filtering is the lack of knowledge about which part of the signal is noise and which is the desired signal, leading to potentially unwanted changes if applied aggressively in those key frequency bands [45] [44].

Spectral subtraction is a technique that removes noise from a signal containing the desired audio output using noise-spectrum estimates derived from non-speech audio segments. Spectral subtraction assumes that noise follows an additive model, in which the desired signal and the noise combine to produce the output signal [47]. The alternative is the multiplicative noise model, in which the noise’s strength directly alters the desired signal’s strength, or vice versa. Another assumption or requirement is that the background noise is relatively stationary in terms of spectral power and level; noise from HVAC systems, fans, steady traffic, etc. [48]. To perform the subtraction, for each input signal frequency or collection of frequencies, the noise power for that same frequency/frequencies is subtracted from the noisy input signal. Unlike standard basic filtering, spectral subtraction responds directly to the noise in a given environment. However, like basic filtering, it is computationally efficient and suitable for use in low-resource settings [47] [43]. Limitations include the introduction of “musical noise” into the resulting signal, characterised as sporadic “whistles or hums” that have a detrimental effect on the listening experience [47] [49]. Whilst its use of a static noise estimate is a benefit over basic filtering, the fact that it can only work with static noise is also a limitation, as any error in the applied noise estimate will result in an inaccurate, potentially over-suppressed desired audio signal, also known as the mean square error (MSE) [50].

Closely related to spectral subtraction is the Wiener filter [43]. Wiener filtering minimises the MSE by replacing the direct subtraction used in spectral subtraction with a weighted gain applied to the input signal at each frequency or frequency band. The weighted gain is calculated from the ratio of signal power to the sum of the signal and noise powers at that frequency (see below).

$$\text{Weight} \approx \frac{\text{Signal Power}}{\text{Signal Power} + \text{Noise Power}}$$

Figure 6: Wiener Filter Weight Calculation

With this calculation, the weighting ranges from 0 (complete attenuation) to 1 (no attenuation). When the signal power equals the noise power, the gain is set to 0.5, providing gentle attenuation of the signal. The process described here is the “optimal estimate” or “optimal statistical estimate” and reduces or minimises the mean square error (MMSE) [43]. The strengths of this approach are similar to those of spectral subtraction, but with the added benefit of less “musical noise” from weighted gain application. The limitations are also similar: it is best utilised with static or non-variable background noise and requires an accurate noise estimation/profile [51].

Machine learning, using neural networks such as convolutional neural networks (CNNs), can map degraded, noisy signals to a clean signal. Anvekar et al. compared the performance of Wiener and band-pass filtering with an autoencoder (an artificial neural network trained on clean audio with added white noise) for denoising a given audio signal contaminated with white noise. The autoencoder outperformed

both Wiener and bandpass filters in terms of MSE and signal-to-noise ratio [44]. Zhu et al. demonstrate that artificial neural networks also perform well when reducing the impact of non-stationary background noise or "babble noise" [52]. This improved performance comes at a high computational cost, with Zhu et al. noting that traditional noise reduction methods, such as Wiener filtering and spectral subtraction, are still widely used due to their low computational cost and effectiveness in pre-processing environments [53] [52]. ML models often require sensitive data for training, and concerns exist about the safety and security of that data, as machine learning is vulnerable to model inversion and extraction attacks, as well as membership-inversion attacks [54].

Given the relative strengths and limitations of each technique, Wiener filtering would suit a real-time, lightweight, browser-based audio enhancement pipeline. To increase the effectiveness of such a system, adding basic lowpass and highpass filtering with conservative cutoff frequencies to the incoming signal will remove much of the unnecessary bandwidth from the frequency spectrum, thereby reducing the attenuation required by the Wiener filter. In addition to such measures, a noise power and variability check can ensure that users recording in uncontrollable environments beyond the Wiener filter's operating window are not permitted to continue, preventing the production of an audio file unsuitable for further processing and assessment. To determine signal strength and variability, time-based signal analysis will be required.

A standard approach to audio signal analysis involves creating short, fixed-length frames containing collections of audio samples. Any data derived from that frame then represents the entire frame, as any change is assumed insignificant over such a short period. Measurements taken from each frame are then compared across larger collections of frames to assess variability within a signal. In audio signal processing, frame lengths of 20-40ms balance time resolution with local stationarity assumptions, with 25ms being a commonly adopted choice [55] [56].

2.3 MERN Stack

MERN is a technology stack for developing web applications and includes MongoDB (database), ExpressJS (backend framework), React (frontend library) and NodeJS (server environment). MERN applications are deployable on mobile devices and web browsers and known for their responsiveness, thanks in part to React's Virtual DOM (Document Object Model - programming interface for representing a web page) and Node's non-blocking I/O (prevents input and output tasks like reading a file from blocking further code execution) [57]. Other stacks are available, such as the MEAN stack (replace React with Angular), and whilst there are differences (e.g., the use of the virtual or real DOM), the key reason for choosing one over the other is author familiarity. When working with audio files of unknown size, MongoDB's GridFS system supports storing files larger than 16 MB. GridFS also optimises retrieval by splitting files into chunks, distributing them, and reconstructing them as needed [58]. NodeJS is a JavaScript runtime environment that allows JavaScript code to execute outside the browser, thereby enabling app development via a unified language [59]. ExpressJS is a server-side framework for NodeJS and provides methods for handling different "HTTP requests, and parsing URL parameters", and provides no more than the core functionality required to implement HTTP-based APIs [60]. ExpressJS is an extremely popular framework, as evidenced by its current weekly downloads exceeding 58.5 million (at the time of writing). Other frameworks, such as NextJS, NestJS, and KoaJS, each have benefits of their own, such as NextJS's folder-based page routing structure; however, as with React vs Angular, author familiarity is key.

2.4 Why Web Audio API with DSP?

Given the previously outlined issues with user-controlled respiratory audio capture, conventional DSP implemented via the Web Audio API offers a good opportunity to deliver a multi-platform, multi-device solution for real-time audio quality enhancement.

Respiratory audio biomarkers are subtle and sensitive to poor recording environments, such as background noise, low signal-to-noise ratios, clipping/distortion, and inconsistent positioning and technique. However, the biomarkers also exist within known frequency ranges and, given the right conditions, can be captured successfully using consumer-based technology. Whilst post-processing solutions, and in particular ML models, offer opportunities for enhancement that surpass the capabilities of conventional DSP, they require substantial computational power, are privacy-sensitive, and cannot provide the user with real-time feedback. The goal at the start of a respiratory audio tool is to ensure the captured audio is of the highest quality. From here, ML models and other analysis tools can extract features and markers from the audio where required; however, downstream performance or output will diminish if the audio is not of a particular standard.

The Web Audio API provides real-time audio graph processing with low-latency audio nodes for standard audio DSP control. Beyond this, access to both frequency and time-domain data enables developers to build real-time user feedback UIs and data-controlled pipelines. The `AudioWorkletProcessor` class enables developers to build bespoke DSP units, such as a Wiener filter. The modular nature of the Web Audio API also allows for a clear understanding of each stage of the pipeline and prevents the "black boxing" of specific processes. It allows for deterministic behaviour: the system will produce the same output for a given input, and this will apply across every process/step of the pipeline. All JavaScript/AudioWorklet code executes locally in the browser runtime; all audio samples from the user's device microphone are passed through the audio graph and processed in real time. Hence, avoiding machine learning at this stage ensures all processing is on-device, improving privacy and reducing data transmission until the file is ready for storage. Any submission of files will require the user to take direct action. Beyond the technical capability, the API is a cross-platform standard, requires no installation and offers immediate access via consumer devices.

3 Methodology

3.1 Introduction

The purpose of this study is to investigate the use of digital signal processing techniques in a web browser audio pipeline that utilises the Web Audio API, in the context of building an end-user respiratory audio capture system. The developed artefact will consist of an environment noise level and variability testing system that will either pass or fail, or generate a noise profile for potential use in a noise suppression system.

3.2 Research Methodology

The chosen methodologies for this study are the Design Science Research Methodology (DSRM) alongside experimental testing. This study aligns with DSRM, where the structured process of problem identification, objective definition, design and development, demonstration, evaluation, and communication align with the stated aims and outcomes of this study [61] [62] [63]. Producing a research contribution is a significant aspect of the DSRM, with the overall aim being an artefact and solution to an identified and "understood research problem", and where the artefact can consist of any number of outputs, such as a new method or technical product [63].

Experimental testing allows independent variables to be tested in controlled environments to ascertain their effect on dependent variables [64]. In the context of this study, the experimental tests need to demonstrate that, under different noise conditions, the developed application responds appropriately to a range of different environments.

3.3 Analysis and Design Activities

The literature review in Chapter 2 identified problems caused by poor-quality remote respiratory audio capture, whether for clinical assessment or for passing to an ML model. Issues included poor microphone placement, suboptimal recording environments due to background noise, signal clipping, and poor recording levels. The literature review also identified the lack of web-based respiratory audio services, with many current respiratory-based remote monitoring and diagnosis applications requiring downloading and installing on a mobile device, as well as some form of proprietary hardware. These issues defined the audio pipeline’s functional requirements, which in turn became user stories. Due to the number of requirements for the complete pipeline, the initial environment test and subsequent noise profile generation became the primary focus.

The client-server system architecture supported browser-based audio capture and storage and facilitated the use of the Web Audio API. Prior to implementation, an end-to-end capture and processing workflow outlined each stage of the pipeline, from microphone to storage and retrieval. Where appropriate, literature-derived criteria and values defined the decision logic for the environment test, but further refinement occurred through iterative testing throughout the implementation. Ensuring visual feedback on the live audio signal level was key to the user interface design, as was providing clear instructions for each stage of the environment test.

3.4 Evaluation Criteria

The evaluation criteria consisted of measuring environmental noise quality using metrics for level, variability, and frequency distribution. Depending on whether the environment meets usability thresholds, the system will either pass, pass but with a noise profile captured, or fail the environment under test. The testing method involved simulating various background noise environments, with each simulation chosen for specific characteristics that allow testing against specific criteria. Tests will also assess behaviour close to threshold levels to ensure correct categorisation. Each test generated several system logs and, where necessary, visual spectrum plots showing relationships among noise levels, noise variability, noise frequency distributions, and evaluation outcomes, supporting verification of system robustness.

3.5 Materials and Tools

This study used a MERN (MongoDB, Express, React, and Node) stack with TypeScript, providing a type-safe framework for rapid web-based prototyping to develop a web-based environment noise test. React, Vite, Tailwind CSS, and React-Router implemented the user interface and managed page navigation. The Web Audio API provided the necessary tools to develop the DSP pipeline for environmental noise testing and analysis through its modular routing and audio node system. Use of the Winston logging library enabled the development of JSON logs that provided structured test results to support the verification and analysis of outcomes. Microsoft Excel enabled the conversion of JSON noise profile data points to visual frequency plots.

3.6 Procedures

As the requirements were primarily defined upfront based on the literature review, the development process followed a Waterfall approach [65]. Minor iterations refined the decision-logic variables following initial testing to ensure the system correctly categorised environmental noise. However, limited flexibility in the outcome meant this was not a truly iterative development process. Structured approaches such as Yourdon or Ward-Mellor emphasise the need for data flow diagrams and formal modelling, but are unnecessary for developing a single pipeline [66]. The Agile/Scrum approach requires stakeholder input, sprints and reviews. Given that this is a single-developer project with no stakeholders, this again does not fit the purpose of the study [65]. Rapid Application Development, with iterative refinement, evaluation and artefact development, provided a possible alternative [67]. However, given the fixed requirements and the limited scope of this component, Waterfall still fits more closely.

3.7 Requirements and Design Documentation

Requirements derived from the literature review provided the basis for a series of user stories acting as lightweight specification and design documentation. Further design documentation includes an environment test logic and decision flowchart, as well as a high-level user interaction and noise capture lifecycle state machine model.

4 Design

5 Implementation

6 Results and Discussion

7 Conclusions and Recommendations

References

- [1] J. B. Soriano, P. J. Kendrick, K. R. Paulson, V. Gupta, E. M. Abrams, R. A. Adedoyin, T. B. Adhikari, S. M. Advani, A. Agrawal, E. Ahmadian, F. Alahdab, S. M. Aljunid, K. A. Altirkawi, N. Alvis-Guzman, N. H. Anber, C. L. Andrei, M. Anjomshoa, F. Ansari, J. M. Antó, J. Arabloo, S. M. Athari, S. S. Athari, N. Awoke, A. Badawi, J. A. M. Banoub, D. A. Bennett, I. M. Bensenor, K. S. S. Berfield, R. S. Bernstein, K. Bhattacharyya, A. Bijani, M. Brauer, G. Bukhman, Z. A. Butt, L. A. Cámara, J. Car, J. J. Carrero, F. Carvalho, C. A. Castañeda-Orjuela, J.-Y. J. Choi, D. J. Christopher, A. J. Cohen, L. Dandona, R. Dandona, A. K. Dang, A. Daryani, B. de Courten, F. M. Demeke, G. T. Demoz, J.-W. De Neve, R. Desai, S. D. Dharmaratne, D. Diaz, A. Douiri, T. R. Driscoll, E. E. Duken, A. Eftekhari, H. Elkout, A. Y. Endries, I. Fadhil, A. Faro, F. Farzadfar, E. Fernandes, I. Filip, F. Fischer, M. Foroutan, M. Garcia-Gordillo, A. K. Gebre, K. B. Gebremedhin, G. G. Gebremeskel, K. E. Gezae, A. G. Ghoshal, P. S. Gill, R. F. Gillum, H. Goudarzi, Y. Guo, R. Gupta, G. B. Hailu, A. Hasanzadeh, H. Y. Hassen, S. I. Hay, C. L. Hoang, M. K. Hole, N. Horita, H. D. Hosgood, M. Hostiuc, M. Househ, O. S. Ilesanmi, M. D. Ilic, S. S. N. Irvani, S. M. S. Islam, M. Jakovljevic, A. A. Jamal, R. P. Jha, J. B. Jonas, Z. Kabir, A. Kasaeian, G. G. Kasahun, G. M. Kassa, A. T. Kefale, A. P. Kengne, Y. S. Khader, M. A. Khafaie, E. A. Khan, J. Khan, J. Khubchandani, Y.-E. Kim, Y. J. Kim, S. Kisa, A. Kisa, L. D. Knibbs, H. Komaki, P. A. Koul, A. Koyanagi, G. A. Kumar, Q. Lan, S. Lasrado, P. Lauriola, C. La Vecchia, T. T. Le, J. Leigh, M. Levi, S. Li, A. D. Lopez, P. A. Lotufo, F. Madotto, N. B. Mahotra, M. Majdan, A. Majeed, R. Malekzadeh, A. A. Mamun, N. Manafi, F. Manafi, L. G. Mantovani, B. G. Meharie, H. G. Meles, G. G. Meles, R. G. Menezes, T. Mestrovic, T. R. Miller, G. Mini, E. M. Mirrakhimov, B. Moazen, K. A. Mohammad, S. Mohammed, F. Mohebi, A. H. Mokdad, M. Molokhia, L. Monasta, M. Moradi, G. Moradi, L. Morawska, S. M. Mousavi, K. I. Musa, G. Mustafa, M. Naderi, M. Naghavi, G. Naik, S. Nair, V. Nangia, J. R. Nansseu, J. Nazari, D. E. Ndwandwe, R. I. Negoï, T. H. Nguyen, C. T. Nguyen, H. L. T. Nguyen, M. R. Nixon, R. Ofori-Asenso, F. A. Ogbo, A. T. Olagunju, T. O. Olagunju, E. Oren, J. R. Ortiz, M. O. Owolabi, M. P. A, S. Pakhale, A. Pana, S. Panda-Jonas, E.-K. Park, H. Q. Pham, M. J. Postma, H. Pourjafar, H. Poustchi, A. Radfar, A. Rafiei, F. Rahim, M. H. U. Rahman, M. A. Rahman, S. Rawaf, D. L. Rawaf, L. Rawal, R. C. Reiner Jr., M. B. Reitsma, L. Roever, L. Ronfani, E. M. Roro, G. Roshandel, K. E. Rudd, Y. D. Sabde, S. Sabour, B. Saddik, S. Safari, K. Saleem, A. M. Samy, M. M. Santric-Milicevic, B. P. Sao Jose, B. Sartorius, M. Satpathy, M. Savic, M. Sawhney, S. G. Sepanlou, M. A. Shaikh, A. Sheikh, M. Shigematsu, R. Shirkoohi, S. Si, S. Siabani, V. Singh, J. A. Singh, M. Soljak, R. Somayaji, M. Soofi, I. N. Soyiri, Y. M. Tefera, M.-H. Temsah, B. E. Tesfay, J. S. Thakur, A. T. Toma, M. Tortajada-Girbés, K. B. Tran, B. X. Tran, L. Tudor Car, I. Ullah, M. Vacante, P. R. Valdez, J. F. M. van Boven, T. J. Vasankari, Y. Veisani, F. S. Violante, G. R. Wagner, R. Westerman, C. D. A. Wolfe, D. Z. Wondafrash, A. B. Wondmienen, N. Yonemoto, S.-J. Yoon, Z. Zaidi, M. Zamani, H. J. Zar, Y. Zhang, and T. Vos, “Prevalence and attributable health burden of chronic respiratory diseases, 1990–2017: a systematic analysis for the Global Burden of Disease Study 2017,” *The Lancet Respiratory Medicine*, vol. 8, no. 6, pp. 585–596, Jun. 2020.
- [2] Y.-q. Hang, X. Piao, J. Wu, Q.-w. Jiang, and Y. Han, “The epidemiology of asthma and its attributable risk factors from 1990 to 2021: A systematic analysis based on the Global Burden of Disease Study 2021 and Mendelian Randomization Studies,” *Public Health*, vol. 243, p. 105731, Jun. 2025.
- [3] A. J. e. a. Ferrari, “Global incidence, prevalence, years lived with disability (YLDs), disability-adjusted life-years (DALYs), and healthy life expectancy (HALE) for 371 diseases and injuries in 204 countries and territories and 811 subnational locations, 1990–2021: a systematic analysis for

- the Global Burden of Disease Study 2021,” *The Lancet*, vol. 403, no. 10440, pp. 2133–2161, May 2024.
- [4] D. Adeloye, P. Song, Y. Zhu, H. Campbell, A. Sheikh, and I. Rudan, “Global, regional, and national prevalence of, and risk factors for, chronic obstructive pulmonary disease (COPD) in 2019: a systematic review and modelling analysis,” *The Lancet Respiratory Medicine*, vol. 10, no. 5, pp. 447–458, May 2022.
 - [5] E. Boers, M. Barrett, J. G. Su, A. V. Benjafield, S. Sinha, L. Kaye, H. J. Zar, V. Vuong, D. Tellez, R. Gondalia, M. B. Rice, C. M. Nunez, J. A. Wedzicha, and A. Malhotra, “Global Burden of Chronic Obstructive Pulmonary Disease Through 2050,” *JAMA Network Open*, vol. 6, no. 12, p. e2346598, Dec. 2023.
 - [6] L. Rayner, J. Sherlock, B. Creagh-Brown, J. Williams, and S. deLusignan, “The prevalence of copd in england: An ontological approach to case detection in primary care,” *Respiratory Medicine*, vol. 132, pp. 217–225, Nov. 2017.
 - [7] R. Halbert, S. Isonaka, D. George, and A. Iqbal, “Interpreting COPD Prevalence Estimates,” *Chest*, vol. 123, no. 5, pp. 1684–1692, May 2003.
 - [8] S. Scully, H. Evans, S. Hatam, S. Cook, H. Whittaker, C. Kallis, I. Farr, C. Orton, A. Sheikh, and J. Quint, “Incidence and prevalence of asthma, copd, and ild across england, wales, and south-east scotland – a harmonised approach to data analysis,” *International Journal of Population Data Science*, vol. 9, no. 5, Sep. 2024.
 - [9] J. A. Fiz, R. Jané, M. Lozano, R. Gómez, and J. Ruiz, “Detecting unilateral phrenic paralysis by acoustic respiratory analysis,” *PLoS ONE*, vol. 9, no. 4, p. e93595, Apr. 2014.
 - [10] M. Lozano-García, F. A. Paredes, C. J. Jolley, and R. Jané, “Respiratory sound intensity as a noninvasive acoustic biomarker in copd,” in *2024 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, Jul. 2024, pp. 1–4.
 - [11] W. Ashraf and Z. Moussavi, “Design and analysis of a contact piezo microphone for recording tracheal breathing sounds,” *Sensors*, vol. 24, no. 17, p. 5511, Aug. 2024.
 - [12] B. B. T., H. I. Hee, O. H. Teoh, K. P. Lee, S. Kapoor, D. Herremans, and J.-M. Chen, “Asthmatic versus healthy child classification based on cough and vocalised ah sounds,” *The Journal of the Acoustical Society of America*, vol. 148, no. 3, pp. EL253–EL259, Sep. 2020.
 - [13] D. Aalto, J. Malinen, and M. Vainio, “Formants,” *Oxford Research Encyclopedia of Linguistics*, Jun. 2018.
 - [14] J. H. Appleton, R. C. Perera, and O. Luening, *The development and practice of electronic music*, J. H. Appleton, R. C. Perera, and O. Luening, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1975, literaturverz. S. 336 - 343.
 - [15] S. Mootassim-Billah, J. Schoentgen, M. De Bodt, N. Roper, A. Digonnet, M. Le Tensorer, G. Van Nuffelen, and D. Van Gestel, “Acoustic analysis of voluntary coughs, throat clearings, and induced reflexive coughs in a healthy population,” *Dysphagia*, vol. 38, no. 6, pp. 1467–1486, May 2023.

- [16] S. Ghrabli, M. Elgendi, and C. Menon, “Identifying unique spectral fingerprints in cough sounds for diagnosing respiratory ailments,” *Scientific Reports*, vol. 14, no. 1, Jan. 2024.
- [17] Y. Yan, L. van Bommel, F. M. Franssen, S. O. Simons, and V. Urovi, “Developing a multi-feature fusion model for exacerbation classification in asthma and copd,” *Computer Methods and Programs in Biomedicine*, vol. 268, p. 108796, Aug. 2025.
- [18] G. de Souza dos Reis, J. Baddini-Martinez², and B. Sanches Masiero¹, “Pulmonary function estimation using smartphone audio and deep learning,” *Jornal Brasileiro de Pneumologia*, p. e20250003, Jul. 2025.
- [19] R. Alsagri, A. Wilde, K. Farrahi, A. Alhomoud, and N. White, “Emerging technologies and trends in lung volume estimation: A review,” *IEEE Sensors Journal*, vol. 25, no. 13, pp. 23 552–23 563, Jul. 2025.
- [20] A. Mahboubian, M. J. Kasteleyn, E. Bašić, N. H. Chavannes, and J. J. Aardoom, “Patients’ and health care professionals’ perspectives on remote patient monitoring in chronic obstructive pulmonary disease exacerbation management: Initiating cocreation,” *Journal of Medical Internet Research*, vol. 27, p. e67666, May 2025.
- [21] S. N. Miller, S. McCabe, E. Morgan, R. Al Hourany, M. Nichols, T. Fowler, W. Smith, C. Durham, M. Pittman, M. Madisetti, R. Teufel, A. Swartz, and C. Strange, “Remote spirometry, education, and action for copd diagnosis and management in rural and medically underserved settings in south carolina: Protocol for an observational feasibility study,” *Journal of Evaluation in Clinical Practice*, vol. 31, no. 1, Feb. 2025.
- [22] M. Galvosas, J. C. Gabaldón-Figueira, E. M. Keen, V. Orrillo, I. Blavia, J. Chaccour, P. M. Small, G. Giménez, M. Rudd, S. Grandjean Lapierre, and C. Chaccour, “Performance evaluation of the smartphone-based ai cough monitoring app - hyfe cough tracker against solicited respiratory sounds,” *F1000Research*, vol. 11, p. 730, Jun. 2023.
- [23] S. G. Jeong and S. E. Kim, “Patient domain supervised contrastive learning for lung sound classification using mobile phone,” 2025.
- [24] A. Systems, “Eupnoos respiratory app for lung function tests,” Website, Oct. 2025. [Online]. Available: <https://www.atta.systems/case-studies/eupnoos#:~:text=Eupnoosisaninnovativerespiratory,tri,als,anduserexperiencedesign>.
- [25] L. Welch, M. Shaban, C. Sheppard, A. Gratiot, D. Kodak, and S. Siddiqui, “S134 eupnoos: advancing early diagnosis of respiratory diseases with smartphone-based audio phenotyping,” *Thorax*, vol. 78, no. Suppl 4, pp. A97–A98, 2023. [Online]. Available: https://thorax.bmj.com/content/78/Suppl_4/A97.2
- [26] Y. Kim, Y. Hyon, S. Lee, S.-D. Woo, T. Ha, and C. Chung, “The coming era of a new auscultation system for analyzing respiratory sounds,” *BMC Pulmonary Medicine*, vol. 22, no. 1, Mar. 2022.
- [27] K. Safavi, S. C. Mathews, D. W. Bates, E. R. Dorsey, and A. B. Cohen, “Top-funded digital health companies and their impact on high-burden, high-cost conditions,” *Health Affairs*, vol. 38, no. 1, pp. 115–123, Jan. 2019.

- [28] A. Khanzada, S. Hegde, S. Sreeram, G. Bower, W. Wang, R. P. Mediratta, K. D. Meister, and A. Rameau, "Challenges and opportunities in deploying covid-19 cough ai systems," *Journal of Voice*, vol. 35, no. 6, pp. 811–812, Nov. 2021.
- [29] F. Busquet, F. Efthymiou, and C. Hildebrand, "Voice analytics in the wild: Validity and predictive accuracy of common audio-recording devices," *Behavior Research Methods*, vol. 56, no. 3, pp. 2114–2134, May 2023.
- [30] J. G. Svec and S. Granqvist, "Guidelines for Selecting Microphones for Human Voice Production Research," *American Journal of Speech-Language Pathology*, vol. 19, no. 4, pp. 356–368, Nov. 2010.
- [31] D. Pines, "RODE NT1 Review: The Condenser Mic for You?" <https://mastering.com/rode-nt1-review/>, May 2018, accessed on 21-01-2026. [Online]. Available: <https://mastering.com/rode-nt1-review/>
- [32] W3C, "Web Audio API enables the creation and manipulation of music and sounds, directly in the browser," <https://www.w3.org/press-releases/2021/webaudio/>, Jun. 2021, accessed on 21-01-2026. [Online]. Available: <https://www.w3.org/press-releases/2021/webaudio/>
- [33] MDN, "Web Audio API," Jan. 2026, accessed on 21-01-2026. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API
- [34] D. Moerkerke, "PWA Audio recording," <https://whatpwacando.today/audio-recording>, Jan. 2026, accessed on 21-01-2026. [Online]. Available: <https://whatpwacando.today/audio-recording>
- [35] N. Jillings, Y. Wang, R. Stables, and J. Reiss, "Intelligent audio plugin framework for the Web Audio API," in *Intelligent audio plugin framework for the Web Audio API*, 08 2017.
- [36] B. Matuszewski and O. Rottier, "The Web Audio API as a Standardized Interface Beyond Web Browsers," *Journal of the Audio Engineering Society*, vol. 71, no. 11, pp. 790–801, Nov. 2023.
- [37] P. Adenot and H. Choi, "Web Audio API 1.1," <https://www.w3.org/TR/webaudio-1.1/>, Nov. 2024, accessed on 21-01-2026. [Online]. Available: <https://www.w3.org/TR/webaudio-1.1/>
- [38] J. Berkovitz and O. Theroux, "Web Audio Processing: Use Cases and Requirements," <https://www.w3.org/TR/webaudio-usecases/>, Jan. 2013, accessed on 22-01-2026. [Online]. Available: <https://www.w3.org/TR/webaudio-usecases/>
- [39] S. T. Yousif and B. M. Mahmmoud, "Speech Enhancement Algorithms: A Systematic Literature Review," *Algorithms*, vol. 18, no. 5, p. 272, May 2025.
- [40] W. Zhang, J. Benesty, and J. Chen, "Single-channel noise reduction via semi-orthogonal transformations and reduced-rank filtering," *Speech Communication*, vol. 78, pp. 73–83, Apr. 2016.
- [41] W. A. Jassim, R. Paramesran, and M. S. Zilany, "Enhancing noisy speech signals using orthogonal moments," *IET Signal Processing*, vol. 8, no. 8, pp. 891–905, Oct. 2014.
- [42] B. Widrow, J. Glover, J. McCool, J. Kaunitz, C. Williams, R. Hearn, J. Zeidler, J. Eugene Dong, and R. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692–1716, 1975.
- [43] N. Upadhyay and R. K. Jaiswal, "Single Channel Speech Enhancement: Using Wiener Filtering with Recursive Noise Estimation," *Procedia Computer Science*, vol. 84, pp. 22–30, 2016.

- [44] Anvekar, Rohan V. and Hosalli, Rohit M. and Shindolkar, Samruddhi V. and Tadasalur, Ujwala V. and Manage, Prabhakar S, “White noise Denoising of audio signal using Auto encoder, Band pass Filter and Wiener Filter,” in *2025 6th International Conference for Emerging Technology (INCET)*. IEEE, May 2025, pp. 1–4.
- [45] A. Hashmi, “Perceptual evaluation of speech quality for inexpensive recording equipment,” *Acoustics*, vol. 3, no. 1, pp. 200–211, Mar. 2021.
- [46] A. N. SLOSS, D. SYMES, and C. WRIGHT, *DIGITAL SIGNAL PROCESSING*. Elsevier, 2004, pp. 258–314.
- [47] Z. Ling, Y. Zeng, and P. He, “Research on the Principle, Application and Optimization of Spectral Subtraction Audio Noise Reduction Technology,” in *2025 5th International Conference on Electronic Information Engineering and Computer Science (EIECS)*. IEEE, Sep. 2025, pp. 1084–1088.
- [48] S. Boll, “Suppression of acoustic noise in speech using spectral subtraction,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, Apr. 1979.
- [49] M. T. Sadiq, N. Shabbir, and W. J. Kulesza, “Spectral Subtraction for Speech Enhancement in Modulation Domain,” *International Journal of Computer Science Issues (IJCSI)*, vol. 10, no. 4, pp. 282–285, 07 2013, copyright - Copyright International Journal of Computer Science Issues (IJCSI) Jul 2013; Last updated - 2023-11-20. [Online]. Available: <https://www.proquest.com/scholarly-journals/spectral-subtraction-speech-enhancement/docview/1470806268/se-2>
- [50] L.-y. Sui, X.-w. Zhang, J.-j. Huang, and B. Zhou, “An improved spectral subtraction speech enhancement algorithm under non-stationary noise,” in *2011 International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, Nov. 2011, pp. 1–5.
- [51] J. Chen, J. Benesty, Y. Huang, and S. Doclo, “New insights into the noise reduction Wiener filter,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1218–1234, Jul. 2006.
- [52] J. Zhu, W. Cai, M. Zhang, and Y. Yang, “Self-supervised denoising model based on deep audio prior using single noisy marine mammal sound sample,” *Applied Intelligence*, vol. 53, no. 21, pp. 25 697–25 714, Aug. 2023.
- [53] Z. Chen, “Implementation of Music Noise Removal Technology based on Adaptive Filtering Algorithm,” in *2025 4th International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*. IEEE, Apr. 2025, pp. 1–6.
- [54] Y. Bai, M. Fan, Y. Li, and C. Xie, “Privacy Risk Assessment of Training Data in Machine Learning,” in *ICC 2022 - IEEE International Conference on Communications*. IEEE, May 2022, pp. 1015–1015.
- [55] S. Shahnawazuddin, C. Singh, H. K. Kathania, W. Ahmad, and G. Pradhan, “An experimental study on the significance of variable frame-length and overlap in the context of children’s speech recognition,” *Circuits, Systems, and Signal Processing*, vol. 37, no. 12, pp. 5540–5553, May 2018.
- [56] K. Khaldi, A. Boudraa, and M. Turki, “Voiced/unvoiced speech classification-based adaptive filtering of decomposed empirical modes for speech enhancement,” *IET Signal Processing*, vol. 10, no. 1, pp. 69–80, Feb. 2016.

- [57] R. K. Parida, G. Singh, and A. N. Pal, “Optimizing React Components For Improving The Performance Of MERN Stack Application,” in *2025 Seventh International Conference on Computational Intelligence and Communication Technologies (CCICT)*. IEEE, Apr. 2025, pp. 55–61.
- [58] S. Wang, G. Li, X. Yao, Y. Zeng, L. Pang, and L. Zhang, “A Distributed Storage and Access Approach for Massive Remote Sensing Data in MongoDB,” *ISPRS International Journal of Geo-Information*, vol. 8, no. 12, p. 533, Nov. 2019.
- [59] L. Liang, L. Zhu, W. Shang, D. Feng, and Z. Xiao, “Express supervision system based on NodeJS and MongoDB,” in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*. IEEE, May 2017, pp. 607–612.
- [60] O. Chaplia, H. Klym, M. Konuhova, and A. I. Popov, “Enhancing REST API Handlers Organization for Node.js Microservices,” *SN Computer Science*, vol. 6, no. 7, Aug. 2025.
- [61] M. Obczovsky, A. Bernsteiner, C. Haagen-Schützenhöfer, and T. Schubatzky, “Systematizing decisions in design-based research: From theory to design,” *Science Education*, vol. 109, no. 2, pp. 523–536, Oct. 2024.
- [62] J. R. Venable, J. Pries-Heje, and R. L. Baskerville, “Choosing a design science research methodology,” in *ACIS*, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:115194465>
- [63] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A design science research methodology for information systems research,” *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, Dec. 2007.
- [64] R. Bevans, “A Guide to Experimental Design - 5 Steps and Examples,” Dec. 2022, accessed on 25-01-2026. [Online]. Available: <https://www.scribbr.co.uk/research-methods/guide-to-experimental-design/>
- [65] S. Komai, H. Nakanishi, and H. Saidi, “Guidelines for selecting agile development method in system requirements definition,” in *2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*. IEEE, Nov. 2017, pp. 49–53.
- [66] A. Butterfield, *A Dictionary of Computer Science*, 7th ed., ser. Oxford Quick Reference Ser., G. E. Ngondi and A. Kerr, Eds. Oxford: Oxford University Press, Incorporated, 2016, description based on publisher supplied metadata and other sources.
- [67] P. Beynon-Davies, C. Carne, H. Mackay, and D. Tudhope, “Rapid application development (rad): an empirical review,” *European Journal of Information Systems*, vol. 8, no. 3, pp. 211–223, Sep. 1999.