GOYO

An AI-Based Active Noise Control System for Smart Home Environments

Taerim Kim
Dept. Information System
Hanyang University
Seoul, Republic of Korea
trnara5375@gmail.com

Wongyu Lee
Dept. Information System
Hanyang University
Seoul, Republic of Korea
onew2370@hanyang.ac.kr

Junill Jang
Dept. Information System
Hanyang University
Seoul, Republic of Korea
jang1161@hanyang.ac.kr

Hoyoung Chung
Dept. Information System
Hanyang University
Seoul, Republic of Korea
sydney010716@gmail.com

***Abstract*—This study proposes an AI-based noise cancellation system designed to effectively manage household noise in situations requiring high concentration, such as studying, remote working, or reading. In everyday environments, users often experience reduced focus due to continuous low-frequency noise generated by home appliances such as air conditioners or refrigerators. To address this issue, the proposed system operates within an IoT environment that continuously detects and analyzes ambient noise in real time while distinguishing it from desired sounds such as speech or media playback. Based on the analyzed information, the system generates white noise or phase-inverted signals to actively cancel unwanted sounds. This approach demonstrates the potential of a smart home–based acoustic control solution that helps maintain users' concentration and reduces auditory fatigue in various home environments.**

***Index Terms*—Smart Speaker, Real-Time Noise Analysis, Active Noise Control, Internet of Things, Smart Home, AI-Driven Sound Control**

Roles	Name	Task description and etc.
User/Customer	Wongyu Lee	The User/Customer seeks to enhance daily life through intuitive smart home technology. They interact via voice or text commands, with the system recognizing emotions and context. They want a service that simplifies device control and creates a responsive environment that improves comfort and efficiency with minimal effort.
AI Developer	Taerim Kim	An AI Developer builds a multimodal service that generates optimal smart home routines. They design and implement the AI architecture, including text generation and speech recognition with emotion and context analysis. They apply transfer learning, ensure library integration, and monitor the system to continuously improve recommendation accuracy and adaptability.
Software developer	Junill Jang	A Software Developer transforms project requirements into functional, high-quality applications. Working with the team, they design scalable architectures, write maintainable code, and handle testing, debugging, and quality assurance. They also document processes and functionalities to support future updates and ensure project continuity.
Development manager	Hoyoung Chung	A Development Manager oversees the software development lifecycle, setting goals and methodologies to ensure timely, high-quality deliverables. They coordinate communication between clients and developers and align cross-functional efforts. Through oversight of design, development, and testing, they ensure products meet user requirements and follow best practices.

I. Introduction

A. Motivation

In modern smart home environments, activities that demand sustained concentration—such as remote work, studying, reading, and immersive entertainment—have become increasingly common. However, continuous background noise generated by

household appliances like air conditioners, refrigerators, and water purifiers can significantly reduce users' comfort and focus, often leading to auditory fatigue over time. Conventional passive noise reduction methods, such as soundproofing materials or ear-covering devices, are often impractical for open or shared living spaces.

To address this challenge, active noise control (ANC) technologies offer a promising alternative. Similar techniques have already been successfully applied in automotive systems, where adaptive filtering algorithms and real-time acoustic analysis are used to suppress engine and road noise within confined spaces. These vehicle-based ANC solutions demonstrate how dynamic and environment-aware control can effectively mitigate unwanted sound.

Inspired by these developments, this study aims to explore how similar ANC principles—augmented with adaptive and AI-based algorithms—can be applied to indoor environments. Unlike vehicle cabins, residential spaces exhibit diverse acoustic reflections, complex sound propagation paths, and structural vibrations, making noise control inherently more challenging. By extending ANC methodologies from the automotive domain to everyday living environments, this research seeks to develop an intelligent, spatially adaptive system for real-time household noise suppression.

B. Problem Statement (Client's Needs)

Despite the growing demand for quiet and focused environments in modern homes, current noise management solutions remain limited. Passive soundproofing materials are often costly, space-inefficient, and ineffective against low-frequency noise generated by household appliances. Moreover, personal noise-cancelling devices such as headphones or earphones isolate the user from their surroundings, making them unsuitable for long-term use or shared spaces.

There is therefore a clear need for an intelligent, space-oriented noise control system capable of automatically detecting, analyzing, and reducing unwanted noise without physically isolating the user. Such a system should adapt to varying acoustic conditions, preserve natural sound cues (e.g., speech or media playback), and seamlessly integrate into daily life through IoT connectivity and AI-driven control.

C. Definition of Key Terms

- **Noise Cancellation:** Pass noise through a filter that tends to suppress the noise while leaving the signal relatively unchanged. [1]
- **Active Noise Control (ANC):** A technique that reduces unwanted sound by generating a secondary sound field that destructively interferes with the primary noise. It exploits the long wavelengths of low-frequency noise by electronically controlling secondary sources such as loudspeakers to produce anti-noise signals. [2]
- **Digital Signal Processing (DSP):** The manipulation of digital or digitized signals using digital technologies. DSP systems are implemented through mathematical abstractions, software, or hardware, often under strict real-time and precision requirements, demanding integrated knowledge of signal theory and technology. [3]
- **Second Path:** Entire transfer path between the secondary loudspeaker and the error microphone, including electronic and acoustic components such as the D/A converter, power amplifier, acoustic propagation in air, the microphone, and the A/D converter. Modeling this path accurately is essential, as it determines how the anti-noise signal is physically transformed before reaching the listener. [4]
- **Least Mean Squares (LMS) Algoritm:** An adaptive filtering method that iteratively updates filter coefficients to minimize the mean square error between the desired and actual signals. It uses a gradient-descent approach to adjust weights based on the instantaneous error, making it simple, efficient, and widely applicable in real-time signal processing and noise cancellation systems. [5]
- **FxLMS Algorithm:** A gradient-based algorithm used to identify an unknown system, such as an ANC controller, in the presence of a secondary path. It differs from the standard LMS algorithm in that the reference signal is filtered through an estimated secondary path model before adaptation, allowing compensation for the secondary path effect. [6]

D. Research on Related Software

Several existing software and systems demonstrate the application of AI and IoT integration for noise management and environmental optimization.

1) **Hyundai Motor's In-Vehicle ANC System**
   Hyundai Motor's In-Vehicle ANC system actively reduces unwanted cabin noise by generating phase-inverted sound waves through the car's audio speakers. Unlike passive insulation, it uses DSP and adaptive filtering algorithms—particularly FxLMS method—to analyze and counteract noise in real time. Microphones installed throughout the cabin capture sounds generated by the engine, road surface, and wind, which are then processed by a central controller. The system continuously adjusts its output based on environmental factors such as vehicle speed, road type, and cabin conditions. Hyundai has further advanced this technology through its Road Noise Active Noise Control (RANC) system, which integrates accelerometers to predict and cancel vibration-induced noise before it reaches the cabin. This approach demonstrates how AI-assisted adaptive control can enhance in-vehicle acoustic comfort and serves as a foundation for extending ANC principles to smart home environments. [7]
2) **Apple AirPods ANC System**
   Apple's AirPods employ an ANC system that minimizes unwanted ambient sounds through real-time signal processing. The system uses outward-facing microphones to detect environmental noise and inward-facing microphones to monitor the sound inside the ear canal.

Based on this information, the internal processor generates phase-inverted sound waves that effectively cancel the incoming noise through destructive interference. The algorithm continuously adapts to changes in the acoustic environment—such as movement, wind, or variations in ear fit—ensuring stable noise reduction performance across various conditions. This approach demonstrates how compact wearable devices can achieve efficient, adaptive noise control by integrating real-time sound analysis and feedback-based signal adjustment.

3) **LG ThinQ**
LG ThinQ is LG's AI-powered smart home platform that connects and controls various household appliances through a unified app interface. It enables users to monitor device status, automate routines, and optimize energy usage based on behavior patterns. By integrating AI and IoT technologies, LG ThinQ enhances convenience, efficiency, and personalized home management.

4) **Google Nest Audio**
Google Nest Audio is a smart speaker designed to optimize sound output through adaptive room acoustics. It uses on-device machine learning to adjust equalization in real time based on ambient noise and playback type, indirectly contributing to noise comfort and sound clarity in smart homes.

II. Requirements

A. Common Features

1) **Sign Up**
The registration process requires users to provide an email address (serving as the user ID), phone number, password, and desired nickname. Passwords must contain at least 8 characters, incorporating a combination of letters, numbers, and special characters. The chosen nickname becomes the user's default display name within the application. Upon registration, a verification email is sent to activate the account before login access is granted.

2) **Log in**
The system supports standard authentication using email and password credentials. Invalid login attempts trigger clear error messages to guide users. Successful authentication redirects users to the Main page.

3) **Account Recovery**
The system provides an account recovery feature that allows users to retrieve their ID and reset their password. To find their ID, users verify their phone number, after which the registered email address is displayed. For password recovery, if the entered email exists in the system, a verification code is sent to that email, and upon successful verification, users can proceed to reset their password.

B. In-Application Features

1) **Splash Screen**
Upon application launch, a splash screen displaying the system logo against a minimalist background appears while the system initializes device connections and loads user configurations.

2) **Main Page**
The application opens with the Home page as the default view. This page includes a Start/Stop button for controlling the main functionality and provides an overview of the current sound environment detected by connected external microphones. The interface displays a live frequency spectrum and noise suppression graph, allowing users to visualize ambient sound levels and system performance. Users can enable or disable noise reduction for each detected sound and adjust the suppression intensity. Each noise type can also be deleted, and all changes are reflected in real time. A menu bar provides quick access to three main sections of the application.
- **Home:** This button allows users to navigate from other pages to the main page.
- **Device Management:** Lists connected microphones and speakers with connection controls.
- **Profile:** Manages user settings.

3) **Device Management**
This page enables users to register, monitor, and manually control connected audio devices. Microphones and speakers are detected automatically via wired or audio interface connections. Users can test the audio input and output and disconnect devices as needed. Each connected microphone can be designated as a Reference or Error microphone, allowing the system to optimize ANC performance according to its role. Status updates of the device (connected, active, or disconnected) are reflected in real time.

4) **Profile Page**
Users can view and update personal information, including their name. The application stores user-specific ANC configurations and maintains data logs of noise patterns, suppression statistics, and device usage history.

C. AI Features

1) **Noise Classification and Analysis**
The AI module analyzes incoming audio data to classify environmental noise types such as low-frequency hums, fan noise, and transient impact sounds. Identifies dominant frequency bands and temporal patterns to support accurate noise profiling and trend visualization. The classification results are used by other modules for adaptive control and user feedback.

2) **Adaptive ANC Tuning**
AI-based adaptive control dynamically adjusts the ANC parameters based on classified noise patterns and real-time environment feedback. The system learns optimal suppression profiles for different acoustic conditions to maintain effective performance under changing noise environments.

3) **White Noise Masking for Irregular Noises**
For irregular or unpredictable noises that cannot be

effectively canceled through phase inversion, the system generates a low-level white noise signal to mask disruptive sounds and improve perceived comfort. The AI module determines when masking is preferable over active cancellation based on real-time noise variability and user preferences.

4) **Smart Speaker Interaction**
AI enables natural interaction with connected smart speakers. Users can issue voice commands to control ANC features or request environment analysis, and the system responds through text-to-speech feedback summarizing detected noise conditions or ANC status.

D. ANC Features

The system shall include a real-time DSP module responsible for noise analysis, anti-noise signal generation, and adaptive control. Incoming audio from connected microphones is processed with minimal latency to identify dominant noise frequencies and generate corresponding anti-phase signals for ANC. The DSP module continuously monitors residual noise through feedback and dynamically adjusts signal parameters to maintain optimal suppression performance.

It also performs essential filtering, gain control, and noise smoothing to ensure stable operation under varying acoustic conditions. Environmental calibration measures room and echo characteristics to further enhance ANC effectiveness.

III. Development environment

A. Choice of Software Development Platform

Development Platform:

1) **macOS**
macOS is Apple's operating system providing a robust development environment centered around Xcode for Apple ecosystem development. Its Unix foundation offers powerful command-line capabilities, while package managers like Homebrew facilitate tool management. The platform excels in native development through Swift and Objective-C support, and includes comprehensive debugging and performance optimization tools. macOS integrates smoothly with Apple services like iCloud and TestFlight for deployment, while maintaining security through features like Gatekeeper. Its UNIX certification and virtual machine support enable versatility across different development scenarios.

Language and Framework:

1) **Python**
Python is a versatile and widely used high-level programming language, praised for its simplicity and readability. This makes it particularly attractive for both beginners and experienced developers. Python's extensive standard library and rich ecosystem of third-party libraries provide powerful tools for various tasks, including web development, data analysis, and artificial intelligence. The language's strong support for object-oriented, imperative, and functional programming paradigms allows developers to choose the style that best fits their needs. Furthermore, Python is heavily utilized in the AI community due to its robust frameworks and libraries that facilitate tasks such as data preprocessing, model building, and evaluation.

2) **Flutter**
Flutter serves as a robust and versatile framework for mobile application development, enabling the creation of high-performance applications on both iOS and Android platforms from a single codebase. Its rich set of predesigned widgets allows developers to implement highly customizable and visually appealing user interfaces efficiently. Flutter's hot-reload feature accelerates the development cycle by instantly reflecting code changes, thereby enhancing productivity. Moreover, Flutter integrates seamlessly with native APIs and third-party packages, providing flexibility to incorporate diverse functionalities. Extensive community support and comprehensive documentation ensure that development challenges can be quickly addressed, making Flutter an ideal choice for cross-platform mobile development.

3) **FastAPI**
FastAPI is a modern, high-performance web framework for building APIs with Python, designed to enable fast development and high efficiency. It leverages Python's type hints to provide automatic data validation, serialization, and comprehensive API documentation through OpenAPI and Swagger. FastAPI supports asynchronous programming using Python's async and await syntax, allowing for scalable and non-blocking request handling. Its simplicity, combined with robust performance comparable to Node.js and Go, accelerates backend development while maintaining reliability. Furthermore, FastAPI integrates seamlessly with popular Python libraries, database ORMs, and authentication systems, offering flexibility and extensive community support to quickly resolve development challenges.

Cost estimation:

Item	**Price**
Laptop	$1,199
Speaker	$9

Development Environment:

1) On Local Machine

Name	**Computer Resource**
Taerim Kim	Apple M2 8GB RAM
Wongyu Lee	Apple M4 16GB RAM
Junill Jang	Apple M2 8GB RAM
Hoyoung Chung	Apple M3 8GM RAM Ubuntu 24.04 / AMD 5600x, rtx3070 8gb, 32GB RAM

2) Cloud Platform

Purpose	**Computer Resource**
Back-end server deployment	AWS EC2 (t2.micro) 1 vCPU 1GB RAM Ubuntu 20.04
AI server deployment	TBD

B. Software in use

1) **Visual Studio Code**
   Visual Studio Code (VS Code) is a powerful, open-source code editor developed by Microsoft. It supports a wide range of programming languages and is highly extensible through its rich marketplace of plugins and extensions. VS Code provides an integrated terminal, debugging tools, and Git version control, making it an ideal environment for collaborative development. Features like IntelliSense offer smart code completion and context-aware suggestions, enhancing developer productivity. Its user-friendly interface and customizable settings allow developers to tailor their workspace, facilitating efficient and streamlined coding practices.
2) **Figma**
   Figma serves as the core tool for our project's UI/UX design. The latest web-based version provides real-time collaboration, enabling multiple team members to work simultaneously. Figma allows us to manage the entire design process on a single platform, from wireframe creation to detailed prototype production. Its design system management features help maintain consistent UI elements and facilitate easy extraction of CSS values and assets during the transition from design to development. Moreover, Figma's component-based approach aligns seamlessly with React Native's component structure, ensuring an efficient workflow from design to implementation.
3) **MySQL**
   MySQL is a widely-used open-source relational database management system, implemented with version 8.0. It provides powerful data management capabilities, including complex queries, transaction processing, and stored procedures, all with strong ACID compliance. MySQL supports built-in replication, high availability, and robust security features, such as encrypted connections and role-based access control. Its efficient indexing, caching, and query optimization enable high-performance data operations at scale, making MySQL a standard choice for applications requiring reliable and persistent data storage.
4) **Redis**
   Redis is an open-source, in-memory data structure store, commonly used as a database, cache, and message broker. It supports various data types such as strings, hashes, lists, sets, and sorted sets, providing high-performance read and write operations. Redis offers features like persistence, replication, pub/sub messaging, and Lua scripting, making it highly versatile for real-time applications. Its low-latency and scalable architecture make it an ideal choice for caching frequently accessed data, session management, and handling high-throughput workloads in modern web and mobile applications.
5) **TablePlus**
   TablePlus is a modern, native database management tool that provides a streamlined interface for interacting with multiple relational and non-relational databases. It supports databases such as MySQL, PostgreSQL, SQLite, Redis, and more. TablePlus offers features like syntax highlighting, query editor, table browsing, and secure connections, enabling developers to efficiently manage database structures and perform data operations. Its intuitive interface and fast performance make it a convenient choice for both development and database administration tasks.
6) **GitHub**
   GitHub is a widely-used web-based platform for version control and collaborative software development. It provides Git repository hosting, enabling developers to track changes, manage code versions, and coordinate work across teams efficiently. Features such as pull requests, code reviews, and issue tracking facilitate seamless collaboration, while GitHub Actions allows automated workflows for testing, building, and deployment. The platform's extensive community support and integration with numerous development tools make it an essential resource for modern software projects.
7) **Notion**
   Notion is an all-in-one workspace platform that combines note-taking, task management, and team collaboration tools. It enables teams to organize projects, documentation, and workflows within a single, flexible interface. Users can create databases, kanban boards, calendars, and wikis, facilitating both personal and team productivity. Notion's real-time collaboration features allow multiple team members to edit and comment simultaneously, ensuring smooth communication and coordination. Its customizable templates and integration with other tools make it a versatile solution for project management and knowledge sharing.
8) **Overleaf**
   Overleaf is a cloud-based LaTeX editor that facilitates collaborative document creation and editing. It provides real-time collaboration, version control, and seamless compilation of LaTeX documents without the need for local installation. Overleaf offers a rich set of templates for academic papers, reports, and presentations, streamlining the document preparation process. Its integrated PDF preview and error highlighting features help users quickly identify and correct issues, while collaboration tools allow multiple contributors to work simultaneously, making it ideal for team-based research and technical writing projects.

C. Task distribution

Roles	Name	Task description and etc.
Front-end Developer	Wongyu Lee	Responsible for implementing the user interface and user experience of applications. They work with technologies like HTML, CSS, JavaScript, and frameworks such as React or Flutter to create responsive, interactive, and visually appealing designs. Frontend developers ensure seamless integration with backend services and optimize performance for various devices.
Back-end Developer	Junill Jang	Handles server-side logic, databases, and application infrastructure. They design and implement APIs, manage data storage, ensure security and scalability, and maintain server performance. Backend developers work with frameworks like FastAPI, Spring Boot, or Node.js to provide reliable and efficient services for frontend applications.
AI Developer	Taerim Kim	Develops and deploys artificial intelligence and machine learning models to solve specific problems or enhance application functionality. They preprocess data, train models, optimize algorithms, and integrate AI solutions into applications. AI engineers often work with frameworks like TensorFlow, PyTorch, or scikit-learn.
Data Engineer	Hoyoung Chung	Designs, builds, and maintains data pipelines and infrastructure for collecting, processing, and storing large volumes of data. They ensure data quality, accessibility, and scalability, supporting analytics and AI workflows. Data engineers work with databases, ETL tools, and cloud platforms to enable reliable data-driven decision-making.

IV. Specifications

A. Common Features

1) Sign Up

ID	Name	Description
01	Signup-Page	GOYO requires four user information to sigh up for membership: E-mail, phone number, password, and name.
02	Signup-Email	The Email field serves as the unique identifier (primary key) for user login. The entered value must follow a valid email format (e.g., contain '@' and domain). Upon submission, the system checks the PostgreSQL database for duplicate entries. If found, an error message such as "This email is already in use" is displayed. Invalid formats prompt "Please enter a valid email address."
03	Signup-Password	The Password field requires at least 8 characters, combining two or more of the following: letters, numbers, and special symbols. During input, the password should be displayed on the screen in the format of asterisks '****'.
04	Signup-Name	The Name field is a mandatory input representing the user's real name. Upon successful registration, this value is stored in the users table under the name column and used as the user's default nickname within the system. Input validation ensures that the field cannot be left empty.
05	Signup-PhoneNum	The mobile phone number is mandatory for user verification. In case user forget their login information, they can restore account access through mobile identity authentication.

2) Login

ID	Name	Description
06	Login-Page	The Login page provides access control for registered users of the GOYO system. It includes two primary input fields (Email and Password), one action button (Sign In), and a navigation link to the Sign-Up page. Two account recovery buttons are also provided: Find ID and Reset Password.
07	Login-Success	When both ID and password are correctly entered and match an existing entry in the user DB, the login is successful and the backend returns an HTTP 200 response. The user is then redirected to the main homepage.
08	Login-Failure	If the entered login information does not match any existing record in the user DB, the system will deny access and display a "Enter your ID/PW again" message.

3) Account Recovery

ID	Name	Description
09	RecoverID-Page	The ID Recovery page allows users to retrieve their registered account ID by verifying their personal information. It includes two primary input fields (Name and Phone Number) and an action button (Send Verification Code). Once a valid user is found, an additional input field (Verification Code) appears for SMS verification. After successful verification, the user's registered ID (email address) is displayed on the screen. If the information does not match any record, an error message "User information not found" is shown.
10	ResetPswd-Page	The Password Reset page allows users to verify their identity before resetting their account password. It includes two primary input fields (Email and Phone Number) and an action button (Send Verification Code). Once a valid user is identified, an additional input field (Verification Code) appears for SMS verification. After successful verification, the user is redirected to the password reset page to create a new password. If the provided information does not match any record, an error message "User information not found" is displayed.
11	ResetPswd-Reset	The Password Reset (New Password) page allows verified users to create a new account password. It includes two input fields (New Password and Confirm Password) to ensure accuracy. The password creation rules are identical to those on the Sign-Up page but additionally, users cannot reuse their previous password for security reasons. If the confirmation does not match or the new password violates the rules, an appropriate error message is displayed.

B. In-Application Features

1) Main Page

- Active Noise Control
  The Active Noise Control witch allows users to glob-

ally enable or disable AI-based real-time noise suppression. When toggled ON, the system activates the embedded model that classifies ambient sound from connected microphones and generates anti-phase signals to reduce low-frequency noise. When OFF, all suppression processes are paused to conserve resources. The switch's label dynamically displays the current ANC status ("ON" / "OFF").

- Noise Rules
  The Noise Rules section lists predefined noise types (e.g., Low-frequency hum, Fan noise) that are analyzed by the AI engine. Each rule contains a threshold value (in dB) determining when suppression should be triggered. These baseline categories are included by default for common household noise sources.
- Add Noise Rule
  The Add (+) button allows users to include additional noise categories from a predefined list provided by the system. When selected, a dialog appears showing available noise types(e.g., keyboard noise, appliance hum). Users can choose one or more items, adjust the detection threshold if necessary, and add them to their personal noise rules.
- Noise Rule Toggle
  Each noise rule includes an on/off toggle switch that determines whether the AI should recognize and suppress that particular noise type. When the switch is ON, the system monitors incoming audio frames for the rule's frequency band and activates adaptive filtering accordingly. When OFF, the AI model ignores that category in classification.
- Noise Rule Edit
  The Edit (pencil icon) function allows modification of the rule's name or threshold value. The updated configuration is immediately applied to the current ANC session. Threshold changes dynamically adjust the sensitivity level of noise detection and suppression intensity.
- Noise Rule Delete
  The Delete (trash icon) function permanently removes a noise rule from the user's configuration. Upon deletion, the rule is excluded from future classification cycles and the local settings are updated to reflect the change. The system requests user confirmation before executing deletion.

2) **Device Manager**

- Device manager page
  The Device Manager page provides an interface for registering, monitoring, and controlling all audio devices connected to the GOYO system. Both microphones and speakers are detected automatically via local Wi-Fi or Bluetooth Low Energy (BLE). Users can also pair devices manually by entering their unique device IDs. Real-time device status (Connected, Active, or Disconnected) is displayed for each entry.
- Scan device
  Scan Devices button initiates a discovery process for nearby microphones and speakers. Detected devices are listed with their current connection type (Wi-Fi/BLE) and measured latency. Users can select a device to establish a secure pairing connection. Once connected, latency and signal strength are monitored continuously.
- Device info
  Tapping a device item opens a detailed information panel showing device name, type (microphone/speaker), connection protocol, and latency in milliseconds. From this view, users can perform actions such as testing audio input/output or checking signal stability.
- Rename device
  The Edit (pencil icon) function allows users to rename a device for easier identification (e.g., "Room Mic", "Desk Speaker"). Updated names are stored locally and synchronized with the PostgreSQL devices table, ensuring consistent labeling across sessions.
- Delete device
  The Delete (trash icon) function permanently removes a device from the paired list. Upon confirmation, the device entry is deleted from both the local cache and backend database. A confirmation dialog prevents accidental deletions.

3) **Profile**

- Profile page
  The Profile Page allows users to view and modify personal information and configure their preferred sound environments. Each user's ANC (Active Noise Control) settings and usage history are managed independently, providing a personalized listening experience optimized for different activity modes such as Study, Sleep, and Focus.
- User info
  The User Information section displays the user's registered name and offers the ability to edit it if needed. When a new name is entered and saved, the updated value is stored in the PostgreSQL users table and immediately reflected throughout the application interface.
- Sound mode
  The Preferred Sound Mode section lets users choose between predefined sound environments: Study, Sleep, and Focus. Each mode has its own ANC preset, consisting of automatic adjustment options and suppression intensity levels. Selecting a mode updates the active ANC configuration in real time. The "Automatic Mode" toggle enables adaptive suppression based on ambient noise levels, while the Suppression Intensity Slider allows manual control (0-100%).
- User stats
  Usage & Noise Stats section summarizes past noise analysis data, including detected noise patterns, sup-

pression performance, and device usage duration. This information is stored in the usage_logs table and can be displayed as visual graphs to help users understand how ANC settings performed over time.

- Save changes
  The Save Changes button confirms and applies all modifications made within the Profile Page. Successful updates trigger a confirmation toast message ("Profile updated successfully").

C. AI Features

1) **Real-time Audio Acquisition**
   The system captures a real-time audio stream from the microphone using the sounddevice library. This stream is processed within an audio callback function to ensure continuous data handling with minimal latency. The acquired audio is automatically converted to the YAMNet model's requirements: 16kHz sampling rate and Mono channel. The wav_data (whether from Librosa or sounddevice) is cast to a float32 tensor (waveform) via tf.cast to be ready for model input.
2) **Core Model Construction (via Transfer Learning)**
   The existing YAMNet model showed performance limitations as it is primarily trained on US-based sounds. To overcome this "Domain Mismatch," we constructed a fine-tuned model based on Transfer Learning.
   - **Feature Extractor:** The build_finetuned_model function defines a custom YAMNetLayer. This layer loads YAMNet's original TF function via hub.load() and sets trainable=False, freezing YAMNet's existing knowledge (millions of parameters) to reduce unnecessary training time.
   - **Batch Processing:** The YAMNetLayer's call function uses tf.map_fn to map the batch data (Batch Data) passed by Keras into 'single' data instances that the YAMNet function can process.
   - **Embedding Extraction:** Of the three outputs returned by YAMNet ([scores, embeddings, spectrogram]), only the second item, embeddings (1024 vectors), is extracted.
   - **Classifier Attachment:** The build_finetuned_model function uses tf.keras.Model to take the embeddings output from the frozen YAMNetLayer, flattens it with a Flatten layer, and connects it to a new Dense() layer that we defined. This layer is set to trainable=True and is the only part trained on our custom dataset.
3) **Dataset Construction and Model Training**
   The existing YAMNet model showed performance limitations as it is primarily trained on US-based sounds. To overcome this "Domain Mismatch," we constructed a fine-tuned model based on Transfer Learning.
4) **Inference and Signal Hand-off**
   The trained model is used to run inference on audio files (or real-time streams). This process involves preprocessing the data to the AUDIO_LENGTH_SAMPLES size, adding a batch dimension, and calling model(waveform). The model returns probabilities for our defined NOISE_CLASSES. The argmax() function derives the infered_class_index, which is mapped to our custom class_names list. If this class name corresponds to one of the 20 'NOISE' classes, a 'True signal' is generated and transmitted to activate the 'Noise Canceler' module. This signal acts as the interface between the two AI modules, completing the requirements for the 'Recognition and Classification' module.

D. ANC Features

1) **Measuring Secondary Path**
   The secondary path means how the sound changes when it travels from the speaker (that plays the anti-noise) to the error microphone (placed near the listener's ear). It includes how much the sound is delayed, weakened, or altered in the air. This measured result is stored as a list of numbers (a vector). This vector is later used as a reference for calculations.
2) **Generating Anti-noise Signal**
   When noise is detected, the system starts with an empty filter (no adjustment yet). The system applies a filter to the noise signal and then passes it through the secondary path model to predict how the anti-noise will actually sound in the air. The predicted anti-noise is played through the speaker to cancel the noise.
3) **Checking result and update the filter**
   The error microphone, placed near the listener's ear, listens to the actual sound and checks if the noise level has decreased. If there is still noise, the system adjusts the filter based on the difference between the target and the actual sound. This process repeats continuously and very quickly, allowing the system to gradually improve the noise cancellation performance.

Additional explanation:

- Secondary path vector: A set of numbers that describes how sound changes from the speaker to the microphone.
- Filter: A simple set of rules (multiplying and adding numbers) that transforms the incoming noise into an opposite sound wave
- Core idea: First, measure how sound travels (secondary path), then create an opposite sound, and finally keep adjusting it based on the microphone feedback.

REFERENCES

[1] B. Widrow *et al.*, "Adaptive noise cancelling: Principles and applications," *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692–1716, Dec. 1975, doi: 10.1109/PROC.1975.10036.

[2] S. J. Elliott and P. A. Nelson, "Active noise control," *IEEE Signal Processing Magazine*, vol. 10, no. 4, pp. 12–35, Oct. 1993, doi: 10.1109/79.248551.

[3] A. R. Thompson, J. M. Moran, and G. W. Swenson, Jr., *Digital Signal Processing*, in *Interferometry and Synthesis in Radio Astronomy*, 3rd ed., Taylor & Francis, 2017.

[4] M. Zhang, H. Lan, and W. Ser, "Cross-updated active noise control system with online secondary path modeling," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, pp. 598–602, July 2001, doi: 10.1109/89.928924.

[5] B. Widrow and M. E. Hoff Jr., ”Adaptive switching circuits,” in *1960 IRE WESCON Convention Record*, Part 4, pp. 96–104, August 1960.
[6] I. T. Ardekani and W. H. Abdulla, ”FxLMS-based Active Noise Control: A Quick Review,” in *APSIPA Annual Summit and Conference*, Xi’an, China, 2011.
[7] 김성현, M. E. Altinsoy, and 김중관, ”차량 능동 소음 제어 시스템에서 제어 위치 선택에 따른 소음 저감 성능에 관한 예비 평가,” *한국소음진동공학회논문집*, vol. 32, no. 6, pp. 544–551, 2022, doi: 10.5050/KSNVE.2022.32.6.544.