ARIA – AN AUTOMATED VOICE TRAINER FOR SINGERS

FINAL REPORT

TU858
BSC IN COMPUTER SCIENCE

Author

Elitsa Koleva

C21320836

Supervisor

Svetlana Hensman

School of Computer Science

Technological University, Dublin

09/04/25

Abstract

The *Aria* project aims to provide access to vocal training by offering an application designed for amateur singers, hobbyists, and students without access to professional vocal coaching. Singing is a beneficial practice, enhancing physical, emotional, and social well-being, yet access to high-quality training remains limited due to financial and geographic barriers. *Aria* addresses this gap by providing a comprehensive tool to improve singing abilities through personalised feedback, structured exercises, progress tracking, and community engagement.

This report outlines the development of *Aria*, detailing its features, architecture, and methodology. Leveraging technologies like Next.js for the client side, FastAPI for the backend, and Firebase for real-time data handling, *Aria* combines modern web development practices with vocal pedagogy principles.

Key features include a Vocal Analysis Test to assess users' vocal range, pitch accuracy, and lung capacity; daily exercises and singing practice sessions with real-time feedback. The application also includes foundational theory lessons and progress tracking to motivate users and ensure consistent practice.

Through rigorous research, user feedback, and iterative design, *Aria* aims to offer an accessible and engaging solution for vocal training. This report documents the system design, implemented features, and testing plans, providing a roadmap for the continued development of the application. By blending technology with vocal pedagogy, *Aria* aspires to empower users to achieve their singing potential while promoting vocal health and artistic expression.

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

Elitsa Koleva

Elitsa Koleva

09/04/25

Acknowledgements

I would like to extend my heartfelt gratitude to Svetlana Hensman for her invaluable assistance and guidance throughout this project. Her expertise, insights, and unwavering support have been instrumental in shaping the direction and execution of my work.

Her encouragement and constructive feedback have not only helped me overcome challenges but have also inspired me to push the boundaries of my understanding. I am deeply appreciative of the time and effort she has dedicated to helping me achieve my goals.

Thank you, Svetlana, for being a vital part of this journey.

Table of Contents

1. Introduction

1.1. Project Background

The health benefits of singing are well-established, with research highlighting its positive impacts on various aspects of respiratory health, emotional health, and social well-being. Participation in choir singing has been shown to create a sense of community, provide support for individual's mental health, and reduce feelings of isolation, particularly benefiting older adults and vulnerable populations [1]. Singing has also been shown to improve lung function in individuals with respiratory conditions through improving breath control and respiratory muscle strength [2] and another study focusing on the effects of group singing in individuals with early-stage dementia found that well-being was enhanced with the absence of adverse effects [3].

In the UK, an estimated 2.14 million people regularly participate in choirs, and in Ireland, there were 185 singing groups active between October 2020 and April 2021, 56 of which were designed for clinical or specialized populations, such as retired individuals, those with dementia, and individuals recovering from cancer or managing mental health conditions [4][5].

While singing offers notable benefits, a substantial portion of the population has no access to proper vocal training due to financial or geographical limitations. In Ireland, only 36.8% of singing groups offered free participation [5].

The lack of access to professional vocal training can lead to the development of improper vocal techniques. Research shows that individuals without formal vocal training are less likely to adopt protective habits, such as vocal warm-ups, and are more prone to strain by singing outside their natural vocal range or overusing their voices [6]. Furthermore, popular style singers often learn vocal techniques through imitation, which increases their risk of developing damaging vocal habits [7]. These bad practices can lead to symptoms such as hoarseness, coughing, or pain in the larynx [6].

These facts were part of the inspiration to create an automated vocal training application that can be used by amateur singers to learn proper vocal techniques and keep their voices healthy. It can also boost the singer's confidence and improve their mental well-being.

In the following sections, I will describe how this project will tackle the issues associated with the lack of professional vocal training and allow individuals to realise their full singing potential. I will also discuss the difficulties encountered with defining a set of criteria that define good singing that stem from the complicated nature of the vocal pedagogy field.

1.2. Project Description

Aria is a singing voice trainer application targeting hobbyists, amateur singers and students who have no other means of accessing professional vocal training. The app will help users improve their singing skills by building up good vocal health habits (e.g. warmups and cool downs), providing skill-level appropriate exercises with AI-powered guidance, offering theory lessons, providing progress tracking and community features.

When the user opens Aria for the first time, they will be asked to perform a step-by-step vocal analysis test. The analysis test's aim is to gather information about the user's current singing level. It is based on the basic components of singing such as pitch accuracy and pitch range, vocal timbre and tone, lung capacity and note sustain, and lastly vibrato.

The test is conducted as follows: A reminder to correct posture, say a couple of words in natural speaking voice, sing an estimated scale based on speaking voice pitch, and time inhalation and exhalation. The results will then be calculated and displayed to the user with feedback and daily exercises will be recommended.

After the test, the user can proceed to complete the recommended exercises for the day. Aria will keep a streak count to motivate the user to complete the exercises every day.

The exercises are separated into 5 different categories. Pitch Accuracy and Range, Breath Control and Support, Resonance and Tone Quality, Register Blending, Dynamic Control and Expression.

The daily exercises follow a set structure: Posture Check, Warm Up, Weakest Category 1, Weakest Category 2, Weakest Category 3, Cool Down.

For new users the weakest category will be deducted from the initial vocal analysis test, for user's who have used the app, it will be calculated based on their previous exercise performance.

The theory lessons will be assessable to the user in their own section of the application, with a visual representation of the lessons being completed or not completed.

The Singing Practice/Karaoke feature will be like the daily exercises, however instead of pre-selected exercises the user can pick a song from the app's song library and sing along to it with visual live feedback and end score and detailed feedback.

Figure 1. ARIA SYSTEM OVERVIEW – MIND MAP

1.3. Project Aims and Objectives

The primary aim of Aria is to improve the singing abilities of its users. To achieve this aim, a series of key objectives have been set to guide the project development.

The core objectives include:

Vocal Analysis Test: Develop a tool capable of gathering data on essential vocal attributes, including pitch accuracy and range, vocal timbre and tone, lung capacity and note sustain, as well as vibrato.

Vocal Analysis Test Interpretation and Personalised Exercise System: Create an algorithm that interprets the Vocal Analysis Test results and generates customised daily exercises based on a pre-set structure - Warm Up, 3 sets of exercises targeting the user's weakest areas and a Cool Down.

Targeted Exercise Sets: Design an initial set of exercises for each of the five key categories —Pitch Accuracy and Range, Breath Control and Support, Resonance and Tone Quality, Register Blending, and Dynamic Control and Expression—allowing users to practice live with real-time visual feedback.

Foundational Theory Lessons: Provide introductory lessons on vocal health and anatomy to support users' understanding of proper singing techniques.

Progress Tracking System: Implement a system to record and display users' progress across all exercises completed.

To prepare Aria for production, the project includes the following extended objectives:

Expand Targeted Exercise Sets: Develop at least three sets of exercises for each of the five forementioned categories.

AI Feedback: Implement algorithms able to provide useful natural language feedback during exercises and accurately interpret results from the Vocal Analysis Test.

Streaks: Provide a progress incentive to users to encourage consistent daily exercise practice.

Community Features: Implement social elements such as a simple forum and a streak leaderboard.

Singing Practice: Implement at least five songs for follow-along style singing practice. The user should be able to follow the lyrics and required pitch on screen as well as the current pitch they are singing. At the end there will be detailed feedback with the areas needing improvements highlighted, progress statistics updated and a score for that song.

Extensive Testing: Perform unit tests for each of the app's features and a final end-to-end test to ensure the app is working correctly.

1.4. Project Scope

Defining a set of criteria upon which to evaluate a singer's ability to sing well is challenging as Vocal Pedagogy is a wide field, with a range of conflicting opinions especially when it comes to different singing styles and definitions of good singing [8,11]. There are also conflicting opinions on whether to use science-based or imagery-based approaches when it comes to teaching singing with most voice teachers using a blended approach [9, 10].

For the purposes of time, Aria will only focus on popular singing and be built on the assumption that common criteria used to evaluate popular singing such as tone, breathing, pitch accuracy and use of common techniques like vibrato or register blending are sufficient for the purposes of evaluating the user's singing skills and track their progress [11].

It is also important to note that the app will not attempt to replicate the psychological role of vocal coaches. Singing is a form of art and allows people to express their emotions and experiences. A human vocal coach would be able to help their students express their emotions through singing and advise them on the life choices that are impacting their singing abilities such as smoking or drinking alcohol. They would also be able to detect if a student is having difficulties singing due to past life experiences that can be negatively impacting their singing [12]. Replicating this would be exponentially challenging and outside the scope of this project.

Lastly, while Aria will not be looking out for potential bad vocal health signs such as vocal fatigue or hoarseness. Therefore, this project is not meant to be a vocal health diagnostics tool, and its main goal will be to improve average people's singing ability.

2. Literature Review

2.1. Introduction

This chapter presents an overview of the foundational topics essential for understanding the basis of this project, including vocal pedagogy, vocal production, the art of singing and the requirements gathering process. Each of these fields is extensive and intricate; therefore, for clarity and conciseness, the discussion will focus on core concepts that are directly relevant to the project's objectives.

Additionally, this chapter reviews existing technologies and alternative solutions aimed at democratizing access to vocal instruction, setting the stage for assessing their applicability and limitations in addressing the project's goals.

2.2. Alternative Existing Solutions to Your Problem

Singing has been a part of human history for hundreds of years and singing development begins as early as a foetus inside the mother's womb [13, 12]. Some of the first records of vocal pedagogy date back to the beginning of the 13th century. They discussed the concepts of vocal registers, vocal resonance, voice classification, breath support, diction and tone quality and developed the "Bel Canto" (Italian for 'beautiful singing') singing style [14]. As science has developed, the way singing is taught has split into two ideologies, traditional methods that use imagery as examples to explain singing methods and Evidence-Based Voce pedagogy which acknowledges traditional methods but uses recent research as the basis for teaching [15]. This has caused a lot of confusion when it comes to terminology used when it comes to defining a singing voice and evaluating singing and has sparked a lot of research that attempts to gather and group and define all this terminology [11,16,17].

Despite the complexity involved in teaching people how to sing. Singing is still a popular hobby with 100 to 1000 people in Ireland searching for "singing lessons" online every month.[1]

To meet this demand, a trend has emerged that attempts to provide singing lessons using technology. The following section describes some of the solutions discovered.

2.2.1 Vocal Image – AI Voice Coach

Vocal Image [25] is a mobile application that uses a combination of artificial intelligence, theory lessons, and video exercises to help users improve their voices. The app focuses on enhancing various vocal aspects, including public speaking, voice feminisation or masculinisation, speech recovery, and singing training.

The app focuses on improving the voice in a wide range of areas; public speaking, voice feminisation or masculinisation, speech recovery and singing training.

[1] Data collected from Google Keyword Planner on the 6th of November 2024 for the period of October 2023 to September 2024.

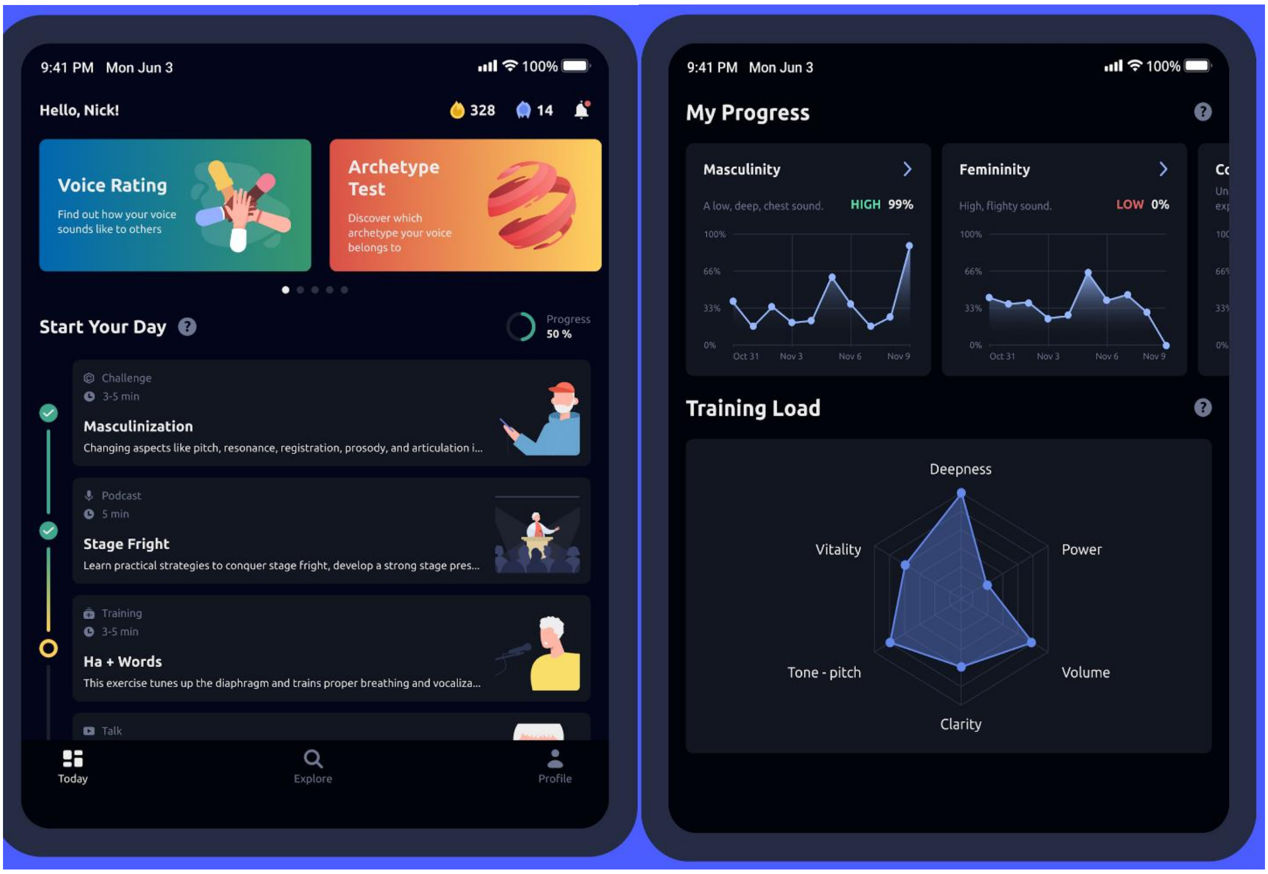


Figure 2 Vocal image user interface

Upon signing up, users are greeted with a clean and intuitive interface. The initial questionnaire collects information on age, gender, vocal goals, and experience level, allowing the app to tailor its content to individual needs. The main dashboard presents a variety of vocal tests and improvement programs, organized for easy navigation.

The vocal tests follow a simple format. The user is required to record an audio sample of their voice after which the app interprets the results and lays them out on a graph. The results are saved to track the user's progress. For example, the *Voice Test* requires users to read out a short paragraph, after which it displays the percentage of perceived masculinity versus femininity, confidence versus weakness, and the level of monotonicity. It additionally offers pitch-related metrics; average pitch, median pitch, and the highest and lowest pitches, alongside volume measurements, such as average, median, and peak volume in decibels.

In the Singing Test, users are prompted to sing for 10-15 seconds. After processing, the results provide information about the user's vocal range in octaves, volume range, voice type (i.e. soprano, mezzo-soprano, baritone, alto, tenor, and bass), and an associated pitch graph.

To help guide the user, an AI chat bot is available that can answer questions. There are also numerous types of resources available including books, videos, podcast, and blog posts.

Vocal Image works well for a few groups of people. The first group; people who want to improve their public speaking skills. Their tests provide enough information for such users to understand how their voice sounds to others and their exercise plans would theoretically help them slowly improve. The second group is transgender people looking to masculinise or feminise their voice. Vocal Image provides exercise plans for voice masculinisation and for voice feminisation which in combination with their vocal tests become a valuable tool.

When it comes to learning singing there are a few areas that do not work well. The singing test does not provide analysis on important aspects of singing such as tone, breath control or technique, instead it only focuses on volume and pitch. Their singing lessons are valuable however without live feedback there is an increased risk of users performing the exercises wrong and damaging their voice.

2.2.2 Youcisian

Figure 3 Yousician Imitation Type Learning

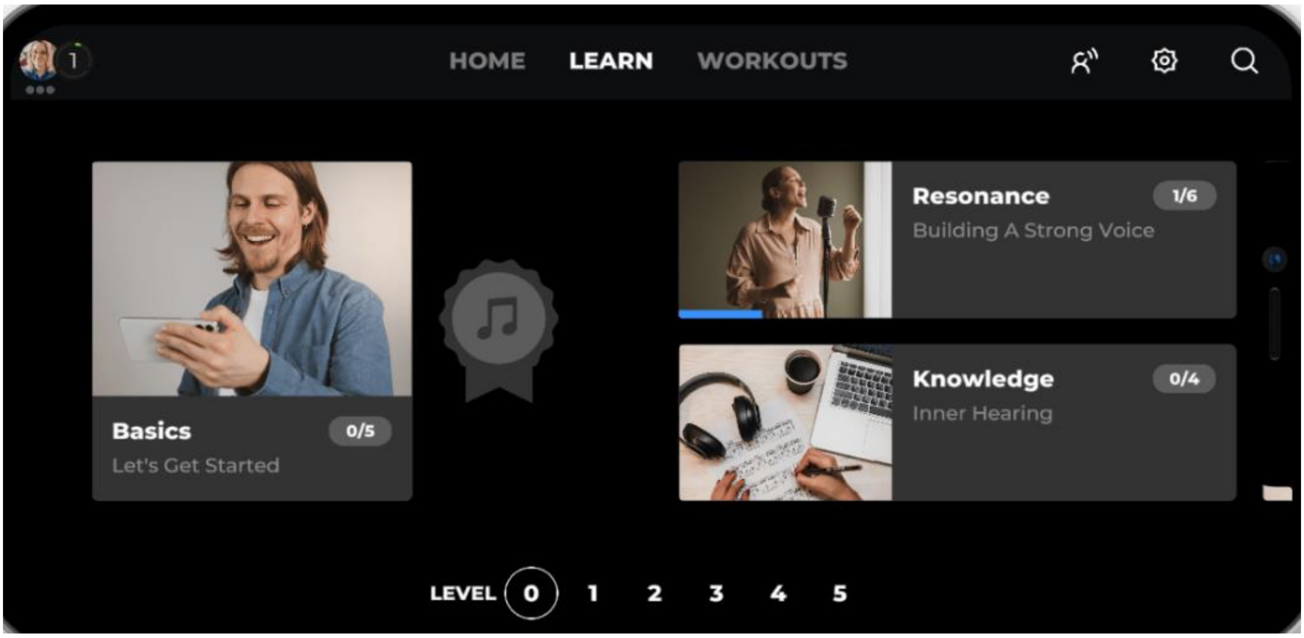


Figure 4 Yousician Home Page

Yousician is an interactive app that offers learning plans for various musical instruments, including guitar, piano, ukulele, bass, and singing. The app's mission is to make musical education accessible and engaging through a gamified learning experience.

Upon launching Yousician, users are presented with a sleek interface that guides them through selecting their instrument of choice. For singing, the app provides a structured roadmap of lessons that cater to different skill levels, from beginners to advanced singers.

There are two types of lessons. Theory lessons comprising of text with some imagery explaining a particular topic and sing-along lessons which are done using imitation type learning [18]. Users listen to vocal examples and replicate them, receiving real-time feedback on their performance. As they sing, the app uses the device's microphone to analyse their pitch accuracy, which is visually displayed on the screen. This immediate feedback allows users to correct their mistakes on the spot. (See Figure 3 Yousician Imitation Type Learning).

Yousician's lessons cover various aspects of singing, including pitch control, timing, and articulation. The app also features a vast library of songs across multiple genres, enabling users to practice with tracks they enjoy. The gamification elements—such as scoring, achievements, and leaderboards—motivate users to practice regularly and track their progress.

Yousician is not free, it uses a subscription model with cheap pricing compared to traditional vocal coaching. However, while the app provides a way to practice singing it might not be detailed enough for users with limited singing experience or for those who are trying to improve other aspects of their singing than pitch.

2.2.3 Sing Sharp

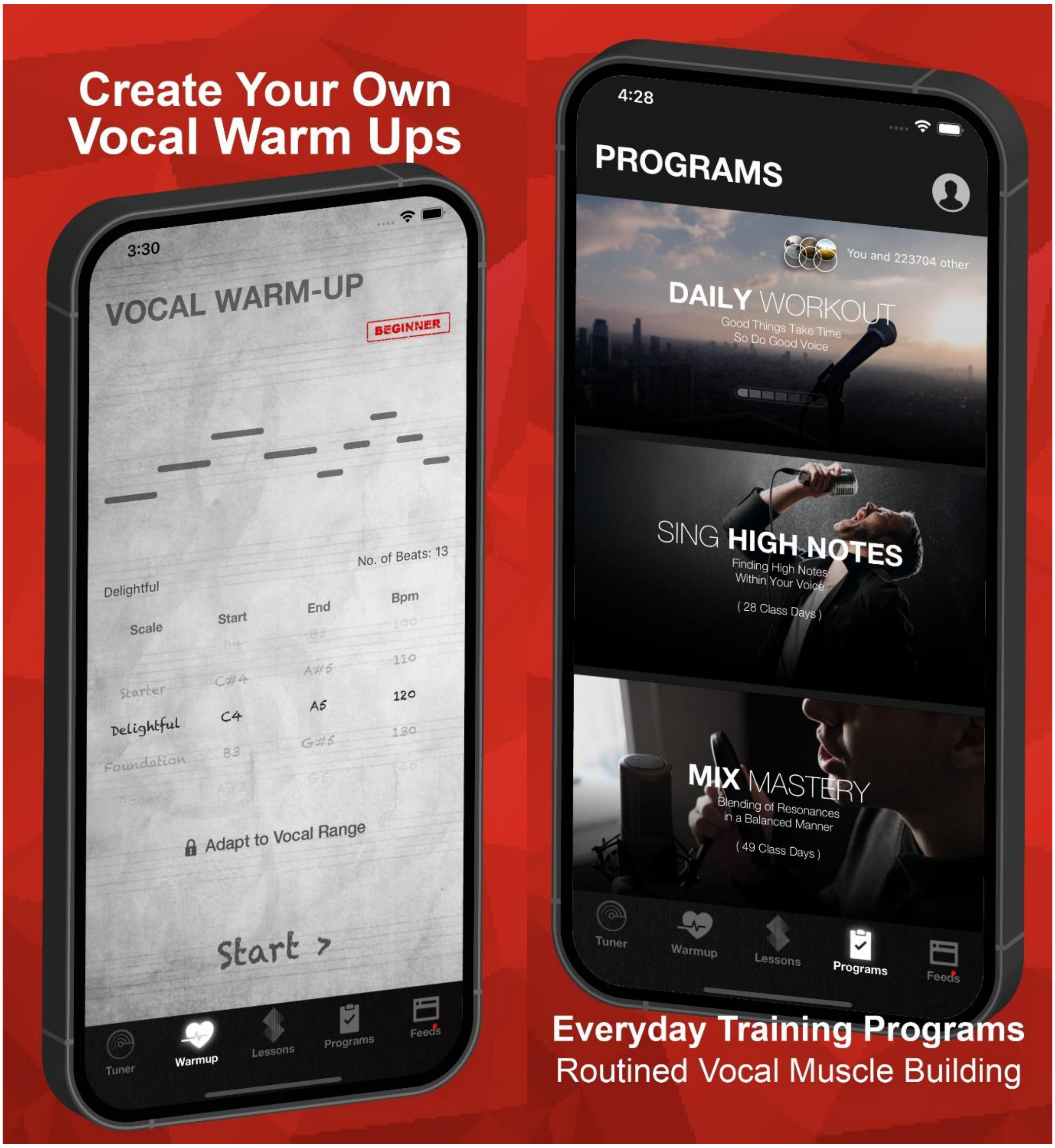


Figure 5 SING SHARP UI

Sing Sharp [27] is a mobile application designed to help users improve their singing through personalised exercises and real-time feedback. The app claims to utilise advanced pitch detection algorithms to evaluate users' singing and provides immediate visual and auditory feedback to correct pitch inaccuracies.

Sing Sharp also offers a variety of features, including vocal range detection, breathing exercises, and vocal warm-ups. The app includes a library of songs for practice, allowing users to sing along and receive feedback on their performance. It also tracks users' progress over time, displaying improvements in pitch accuracy and vocal range.

The user interface of Sing Sharp is unintuitive, users have complained that the “Lessons are too fast,” “There is very little instruction with this, and it's not intuitive” and that “The voice tracker is really glitchy and doesn't work properly” [2]. There were also a few users complaining about the pricing of the app being too expensive.

2.2.4 Evaluation of Existing Solutions

In the landscape of vocal training applications, Vocal Image, Yousician, and Sing Sharp each offer unique features catering to different user needs. However, they also present certain limitations that leave room for improvement, which is where the proposed project, Aria, aims to fill the gaps.

Aria is designed to address these shortcomings by offering a comprehensive, user-friendly vocal training experience. It focuses on all essential components of singing, including breath control, resonance, tone quality, register blending, and dynamic expression. By conducting an initial vocal analysis, Aria personalises daily exercises to target the user's weakest areas, ensuring efficient progress. The app provides AI-powered guidance to offer real-time, intelligent feedback, helping users correct techniques and prevent unhealthy habits. With an emphasis on intuitive design and community features, Aria aims to make vocal training more effective and enjoyable for hobbyists, amateur singers, and students who lack access to professional coaching, filling the gaps left by existing solutions.

2.3. Technologies Research

In this section multiple technology options will be compared against each other, evaluating the benefits they can bring to the project.

2.3.1 Client-Side Technologies

When it comes to client-side or frontend technology for a singing trainer app, there are multiple options. Considering that this project aims to reach as many people as possible, only front-end technologies made for websites will be considered.

The first technology that was researched was React.js.

React is a library that supports client-side-rendering and Next.js is a React framework that offers features such as, hot code reloading automatic routing and server-side rendering on top of React. They are commonly used it web development, React being used by 39.5% of developers worldwide[3]

Being a framework, React supports the use of class components and functional components allowing for the creation of easily maintainable and reusable code.

Another benefit is the virtual DOM which makes it possible to re-render every state change and allows for the creation of live feedback exercises and good user experience.

---

2 “Review comments from: Learn to Sing - Sing Sharp – Apps on Google Play.” Accessed: Nov. 26, 2024. [Online]. Available: https://play.google.com/store/apps/details?id=com.harmonynetwork.singsharp&hl=en_GB

3 “Most used web frameworks among developers 2024,” Statista. Accessed: Nov. 07, 2024. [Online]. Available: https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/

There are multiple other similar frameworks like React for example Vue.js, Angular, Svelte etc. however the differences between them when it comes to functionality are minor, therefore picking one over the other comes down to preference and experience.

A different option that the traditional JavaScript frameworks is ASP .NET Core.

This is a cross-platform, open-source framework developed by Microsoft for building modern applications.

There are multiple options it offers on the front-end for building web applications.

ASP .NET Core Blazor is a web UI framework that uses C# instead of the traditional use of JavaScript. It offers features like Blazor WebAssembly which allows C# code to run directly in the browser with near-native performance and compatibility with all modern browsers, without needing plugins.

It can interop with JavaScript making it possible to use existing JavaScript libraries and integrate components with existing MVC, Razor Pages or JavaScript based apps. And lastly as a part of the .NET ecosystem, existing .NET libraries such as ML.NET can be used which will make integrating the projects machine learning aspects easier.

After evaluating both options, Next.js was chosen as the client-side technology for Aria. The decision is based on Next.js's server-side rendering which is beneficial for the dynamic and interactive components required for real-time feedback, the extensive React.js community which provides ample support and readily available libraries for audio processing and UI components and React's component-based architecture which allows for modular development.

2.3.2 Server-Side Technologies

When choosing the server-side technologies, only the ones that are compatible with web development and had a big community with a large catalogue of libraries were chosen for comparison.

For example, Node.js is a JavaScript runtime environment that allows developers to execute JavaScript on the server side, and it usually goes together with Express which is a minimal web application framework for Node.js. Using these two together allows for the use of JavaScript on both the server and client side and provides efficient handling of concurrent requests. However, it is not optimal for heavy computational tasks like audio processing and machine learning and can lead to complex code structures if not managed properly.

Another option is Python which is a versatile programming language with extensive support for scientific computing. It has rich libraries such as Librosa [43] for audio processing and TensorFlow for machine learning. It is simple with little overhead which will allow for rapid development; however, it is not as performant as low-level languages like C++.

FastAPI is a web framework for building APIs with Python. It is known for its high performance due to the support for asynchronous functions. It is built on Starlette and Pydantic and can handle many requests concurrently.

Another popular Python-based web framework is Flask. However, it is synchronous and can handle only one request at a time.

Since Aria could potentially use machine learning algorithms for parts of the vocal analysis, and since it will need to be able to scale and handle multiple users at the same time, Fast API would be the best option. Python is extremely popular for machine learning due to libraries like TensorFlow and the asynchronous functions of Fast API will aid in a successful backend for the project.

2.3.3 Database Technologies

The database is an important part of the application; it will store all the user data and potentially other data such as songs and their metadata.

There are two types of databases that can be used, SQL and NoSQL.

For a traditional SQL option, we have PostgreSQL. PostgreSQL is a powerful open-source relational database system. Its strengths lie in its reliability and data integrity, supported by ACID-compliant transactions, and its advanced features, such as support for complex queries and indexing. Despite these benefits, PostgreSQL faces challenges in scalability, as vertical scaling can be expensive and horizontal scaling more complex, and it requires more effort for setup and ongoing maintenance.

For NoSQL, one option is Firebase.

Firebase is a Backend-as-a-Service platform offering real-time database services, authentication, and hosting. The real-time updates can provide useful when it comes to live feedback and progress tracking features. It also provides simplified authentication and automatic security as well as automatic scalability as the application grows.

Another NoSQL option is MongoDB.

MongoDB is a document database renowned for its flexibility and scalability. It offers several advantages, including a flexible schema that supports dynamic data structures and high performance optimized for read and write operations. However, it has its drawbacks, such as a lack of ACID transactions across multiple documents, which can impact consistency, and challenges in performing complex joins or aggregations compared to SQL databases.

In the case of Aria's database requirements, Firebase will be the best option due to its real-time capabilities, which are crucial for features like live feedback and leaderboards. Additionally, its ease of use simplifies backend development, allowing the team to focus on core functionalities. Firebase also provides built-in services such as authentication and hosting, reducing the need for additional infrastructure.

2.4. Requirements Research

The target audience for a traditional voice coach would primarily be amateur and professional singers, voice actors, people looking to improve their vocal health (e.g. speech therapists, public speakers) and individuals aiming for voice feminisation/masculinisation.

Aria's is focusing on singing training therefore the target audience would primarily include amateur and professional singers. To help understand the users, people in the industry, including students attending singing lessons, hobbyist singers and the musical director of an Irish choir were asked to fill out a questionnaire aimed at exploring the needs of people who sing.

The questionnaire contained 10 questions, 8 multiple-choice questions and 2 long-answer questions and was conducted through the Google Forms platform.

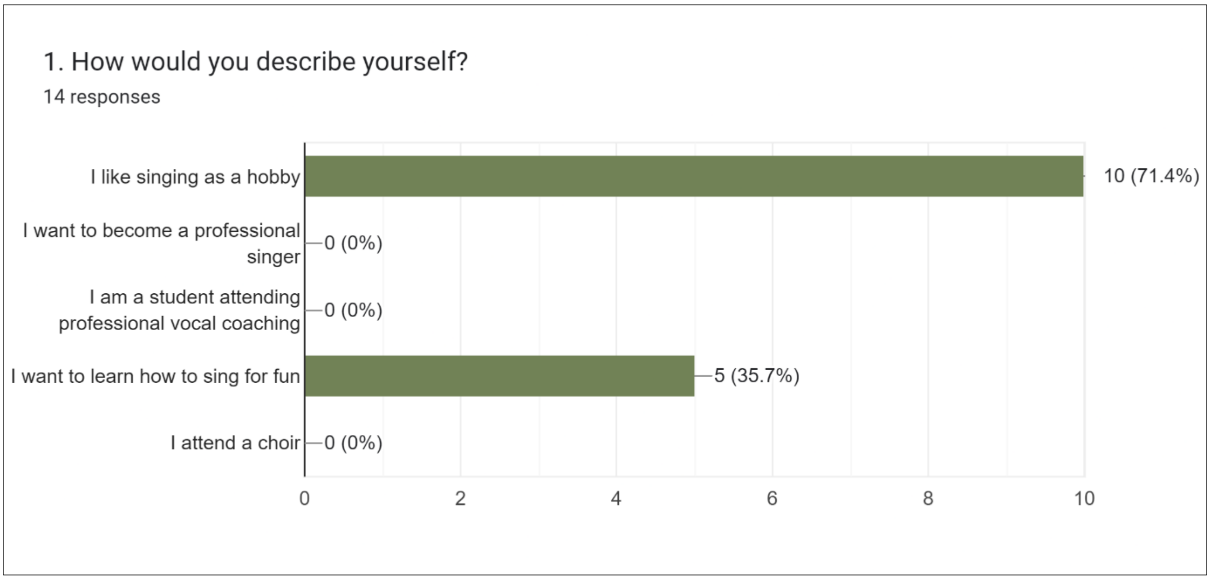


Figure 6 USER REQUIREMENTS QUESTIONNAIRE – QUESTION 1

Based on the responses for question one, most respondents enjoy singing as a hobby and the rest of the respondents specified that they want to learn how to sing for fun. This shows that most of the respondents who sing do not want to become professional singers or attend a choir.

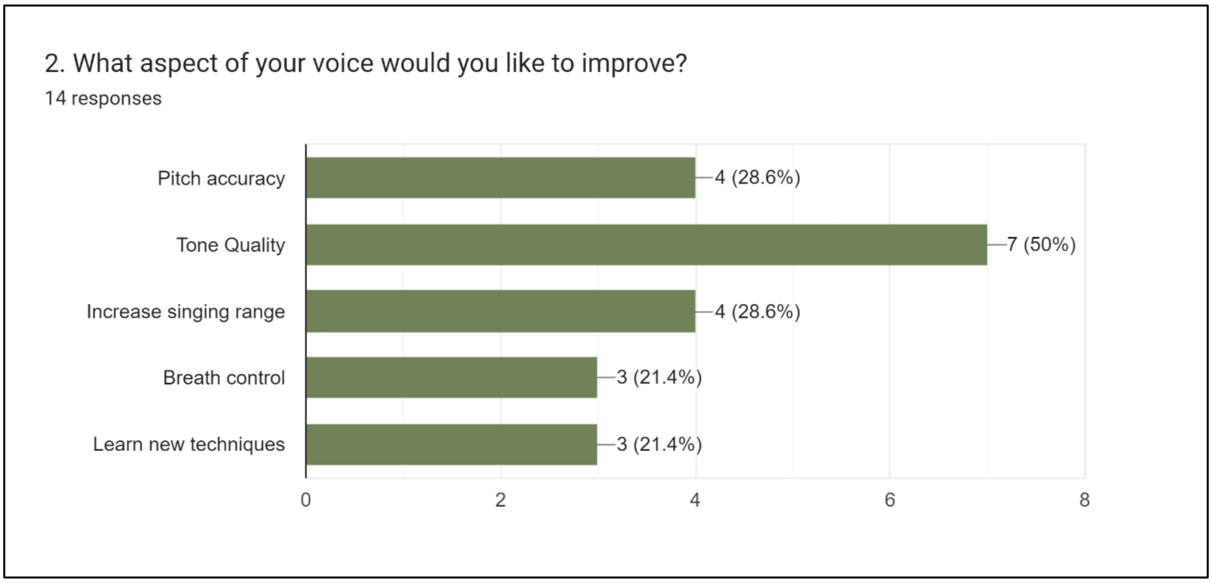


Figure 7 USER REQUIREMENTS QUESTIONNAIRE – QUESTION 2

For question two, most respondents answered that they want to improve the tone of their voice, with the rest of the respondents being evenly divided between the rest of the categories and none responding with other answers.

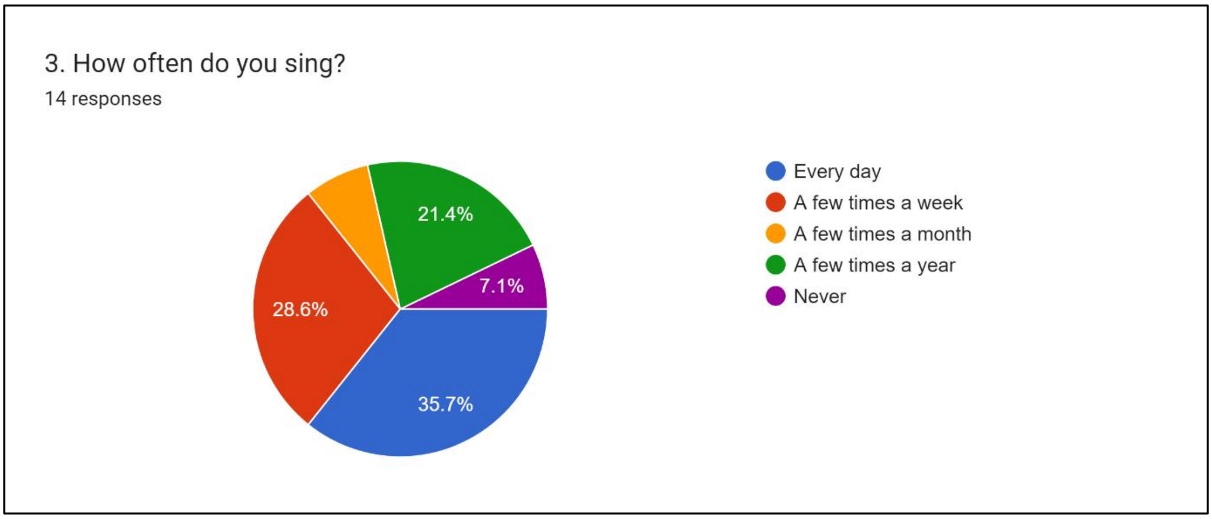


Figure 8 USER REQUIREMENTS QUESTIONNAIRE – QUESTION 3

Based on question three's results, most respondents sing from a few times a week to every day. This shows that for this project it would be ideal to offer the user to select their time-commitment for singing practice. For example, there can be options for daily exercises, three times a week plan and once a week plan.

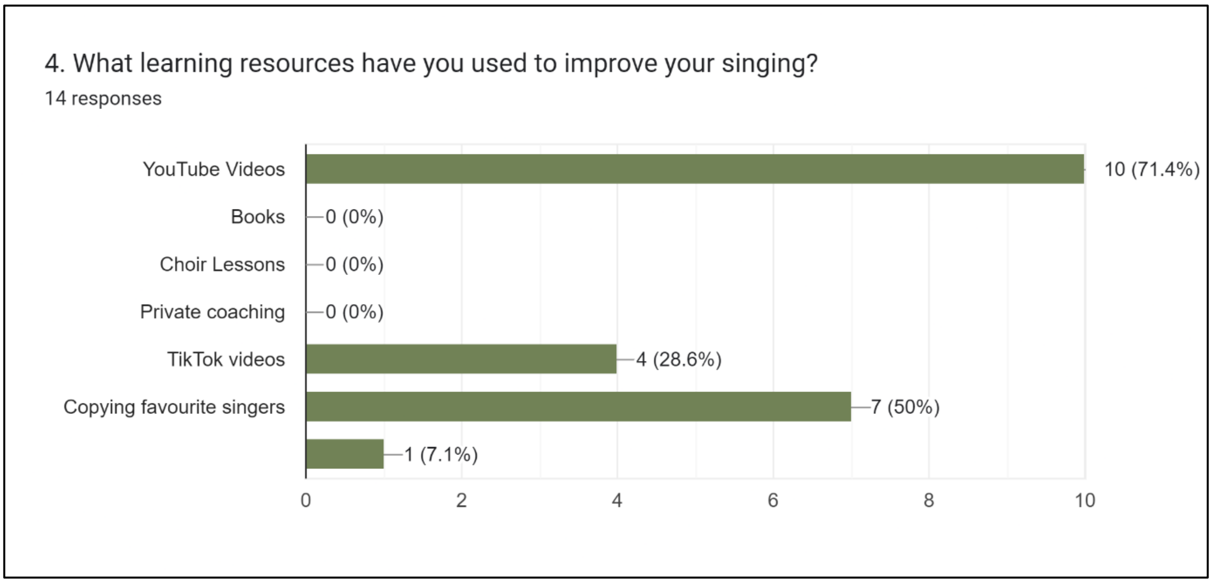


Figure 9 USER REQUIREMENTS QUESTIONNAIRE – QUESTION 4

After reviewing question four's responses, the most popular resource for learning is YouTube videos, with the second most popular options being copying favourite singers or imitation learning. This is in line with previous research where it is shown that amateur singers and popular singers are likely to learn through imitation [7]. Therefore, for this project, introducing elements of follow-along and imitation learning during the exercises would be crucial.

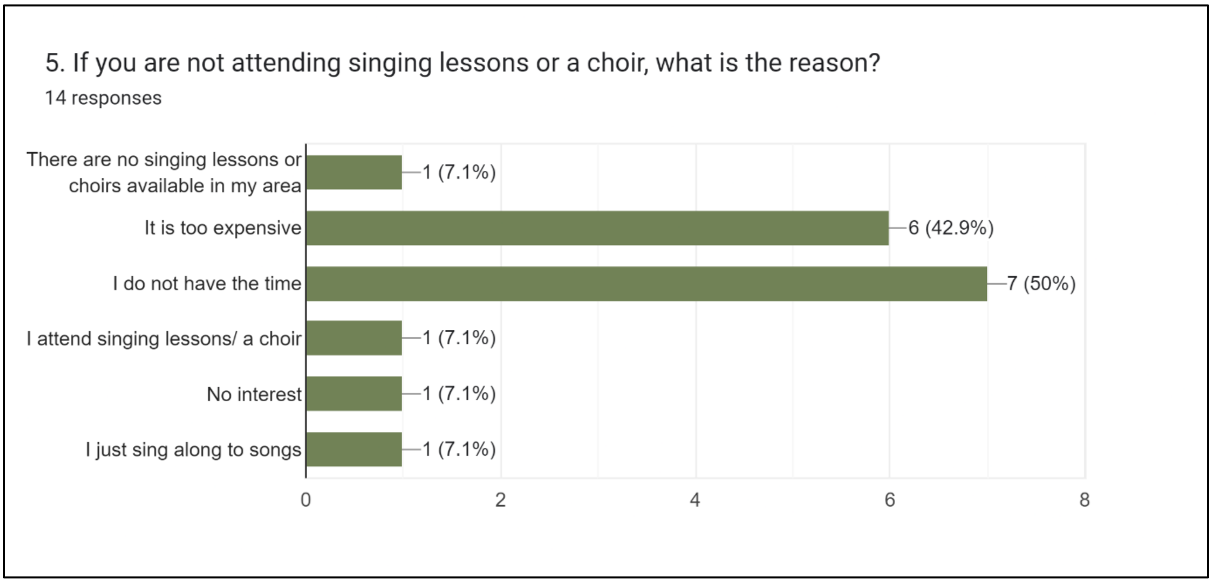


Figure 10 USER REQUIREMENTS QUESTIONNAIRE – QUESTION 5

Question five's responses suggest that financial and time constraint reasons are preventing people from joining choirs. This suggests that Aria should be free to access and should not require prolonged periods of use.

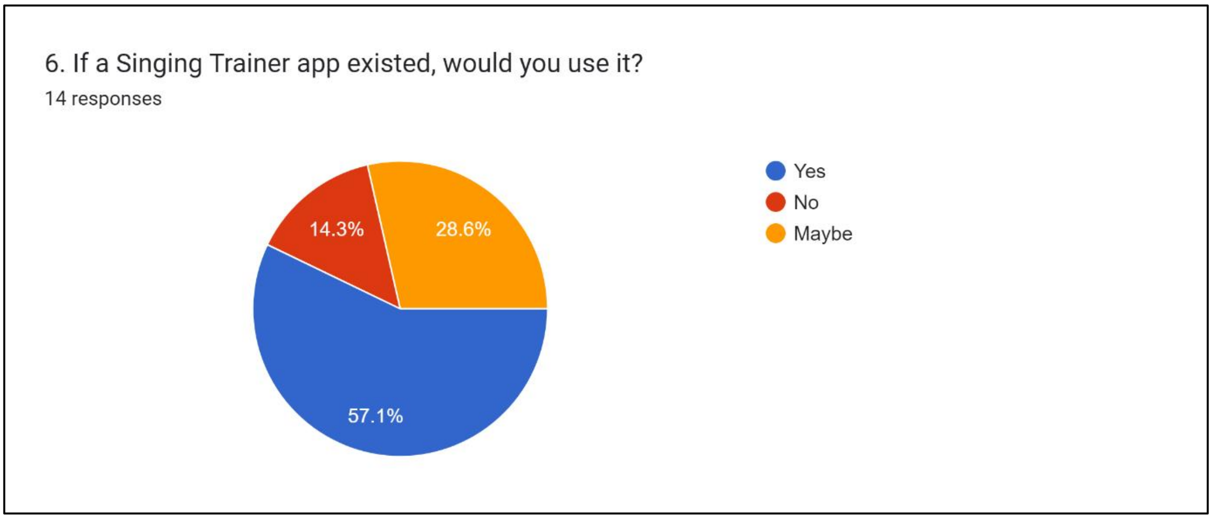
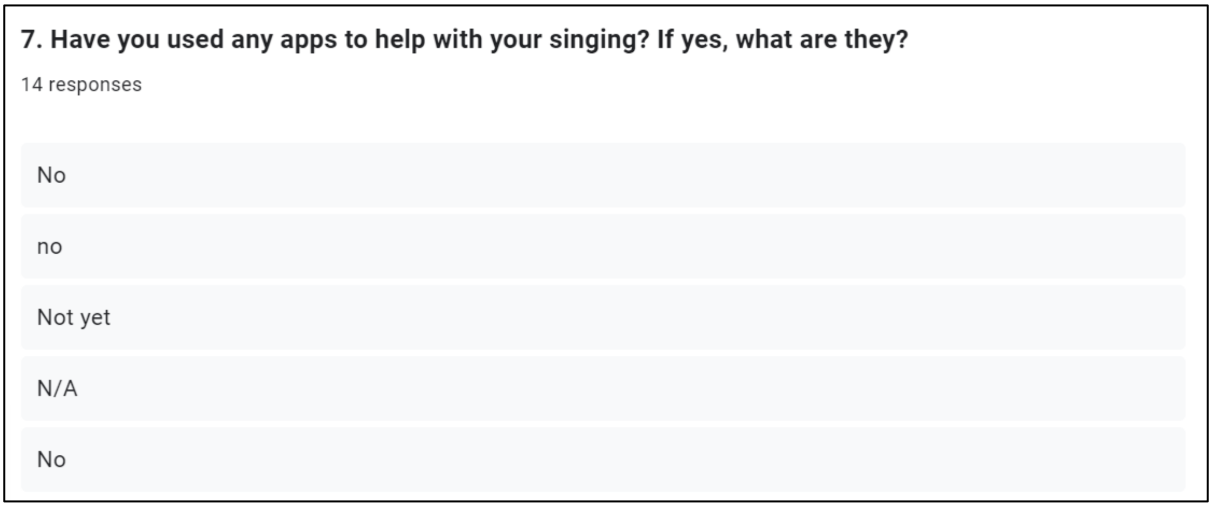


Figure 11 USER REQUIREMENTS QUESTIONNAIRE – QUESTION 6

Based on question six; 79.7% of the respondents are willing to try out a project like Aria to help them improve their singing.


7. Have you used any apps to help with your singing? If yes, what are they?

14 responses

No

no

Not yet

N/A

No

Figure 12 USER REQUIREMENTS QUESTIONNAIRE – QUESTION 7

Question seven shows that currently none of the respondents use applications to help with their singing, even though based on the previous question there is a demand for such applications.

This could mean that the current existing solutions are not sufficient and the development of a project like Aria can fill that gap in the market. It could also mean that the users are unaware of the current solutions due to a lack of efficient targeted advertising.

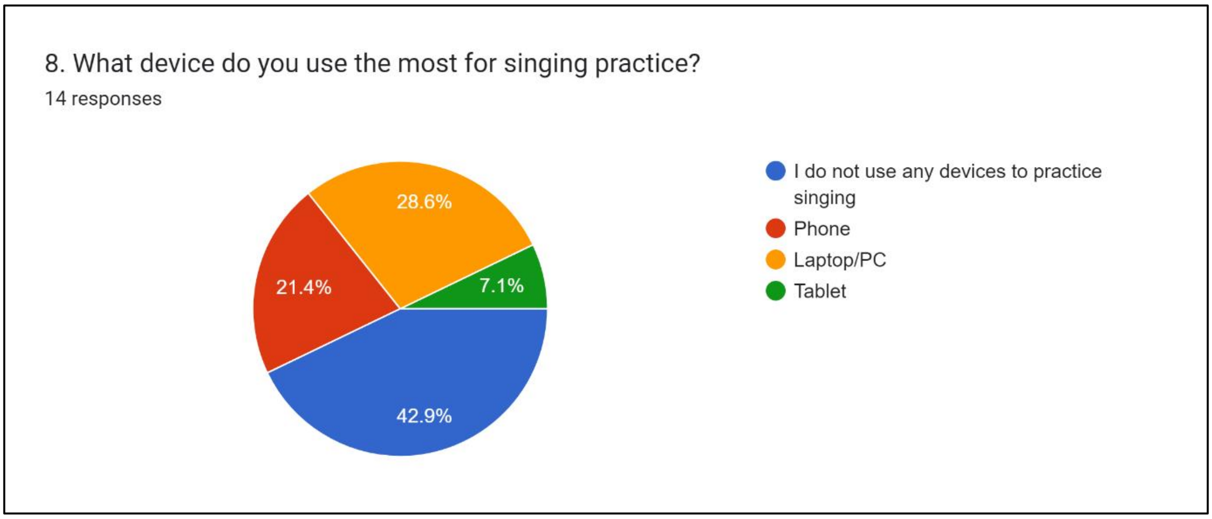


Figure 13 USER REQUIREMENTS QUESTIONNAIRE – QUESTION 8

The results of question eight suggest that this project should be developed in a way that allows it to be used for desktops, tablets, and mobile devices. Solutions can include building the project as a cross-platform web application or developing a native app for each platform. As this is an individual project with limited time, the development of a cross-platform web application is the most viable solution.

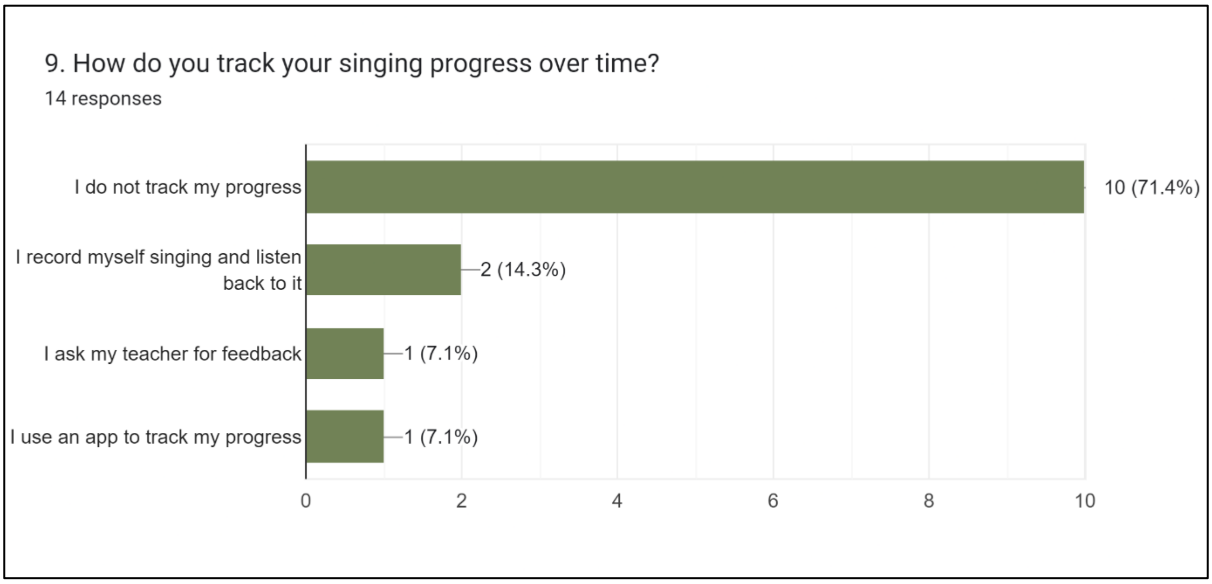


Figure 14 USER REQUIREMENTS QUESTIONNAIRE – QUESTION 9

Question nine shows that currently, 71.4% of the respondents do not track their progress, with 14.3% tracking progress through past recordings and 7.1% asking their teacher for feedback. This can be due to the lack of resources available for singing progress tracking and this project could provide this feature to the users.

10. If you were to design an app that helps people improve their singing at home, what features would you include?
• A visual of how in-tune a person's vocal cords are with the music or tone of vocalist
I want to be able to record myself and listen to progress, also do exercises live
AI Vocal Coach, Breath Control Training, Vocal Range Finder, Pitch & Tone Analyzer, Warm-Up Exercises
Voice warm ups that guide users through different vocal exercises and breathing techniques that focuses on breath control
Lessons with a teachers online
I don't know
Being able to see progress
Daily exercises to improve my singing range
Karaoke
Voice Control
Specific scales to make sure your in key
Exercises
Not sure
Be able to identify all the instruments playing in the songs and high pitches and lows a graph to show you when to go high and when to go low

Figure 15 USER REQUIREMENTS QUESTIONNAIRE – QUESTION 10

The last question was "*If you were to design an app that helps people improve their singing at home, what features would you include?*" There were a lot of different ideas with half of the responses being about exercises with live feedback, specifically on pitch accuracy, and breathing and three responses being about wanting to be able to track progress. This outlines the importance of including live-feedback exercises and progress tracking capabilities to Aria.

2.5. Vocal Pedagogy Research

A lot of this project's research has been focused on finding out how singing was taught in the past, how it is taught now, and how it can be taught using technology. To find the answer we need to first understand what singing is or more specifically how the human voice is produced in the body.

To put it briefly, there are three main components to generating voice; respiration, phonation, and resonance.

Respiration provides the necessary air pressure for voice production. Proper control of breath is critical in singing, as it figures out the stability and volume of the voice. Without sufficient air pressure, phonation (sound production) cannot occur effectively [23].

Phonation is the process of producing sound by the vibration of the vocal folds (vocal cords). The vocal folds, found in the larynx, chop the steady airflow into a series of quasi-periodic air pulses. This produces the fundamental tone of the voice, characterised by a spectrum of harmonic partials, which are multiples of the fundamental frequency. As frequency rises, the amplitudes of these partials decrease monotonically, creating a rich, harmonic structure [23].

Resonance shapes the sound in the vocal tract. When the airflow pulses from the vocal folds pass through this tract, they are changed by its shape and size. The vocal tract has resonances known as formants, which amplify certain frequencies [23].

<u>Formants and Harmonics</u>

Formants are peaks in the vocal tract's resonances that enhance nearby partials and shape vowel quality. The two lowest formants, F1 and F2, are critical for vowel differentiation and can be changed over a range of two octaves or more while higher formants cannot be varied as much and do not contribute to vowel quality, they signify personal voice timbre [23].

Pitch (F0) is decided by the fundamental frequency of the voice, which stays stable even if higher harmonics are altered [5]. H1-H2 is a parameter that measures the difference in amplitude between the first and second harmonics. It reflects the degree of glottal adduction, influencing voice quality from breathy to pressed tones [23].

Vocal Registers like vocal fry, modal, and falsetto in men, or chest, middle, head, and whistle in women, reflect different modes of vocal fold vibration and cover various parts of the pitch range [5]. Transitioning between registers is often accompanied by a change in pitch, and trained singers aim to minimise the timbral differences between registers [23].

Posture and Alignment

Posture is known to affect singing quality. Some studies suggest that how posture affects singing is dependent on the individual [21], with an overall general good posture helping with vocal efficiency, resonance, and breath support [22].

To achieve good posture and alignment, the spine should maintain its natural curves, head shoulders, pelvis, ankles, and feet should be lined up, and lastly the muscles of the trunk, shoulders, and neck should balance in a way that is not completely relaxed nor tense [24].

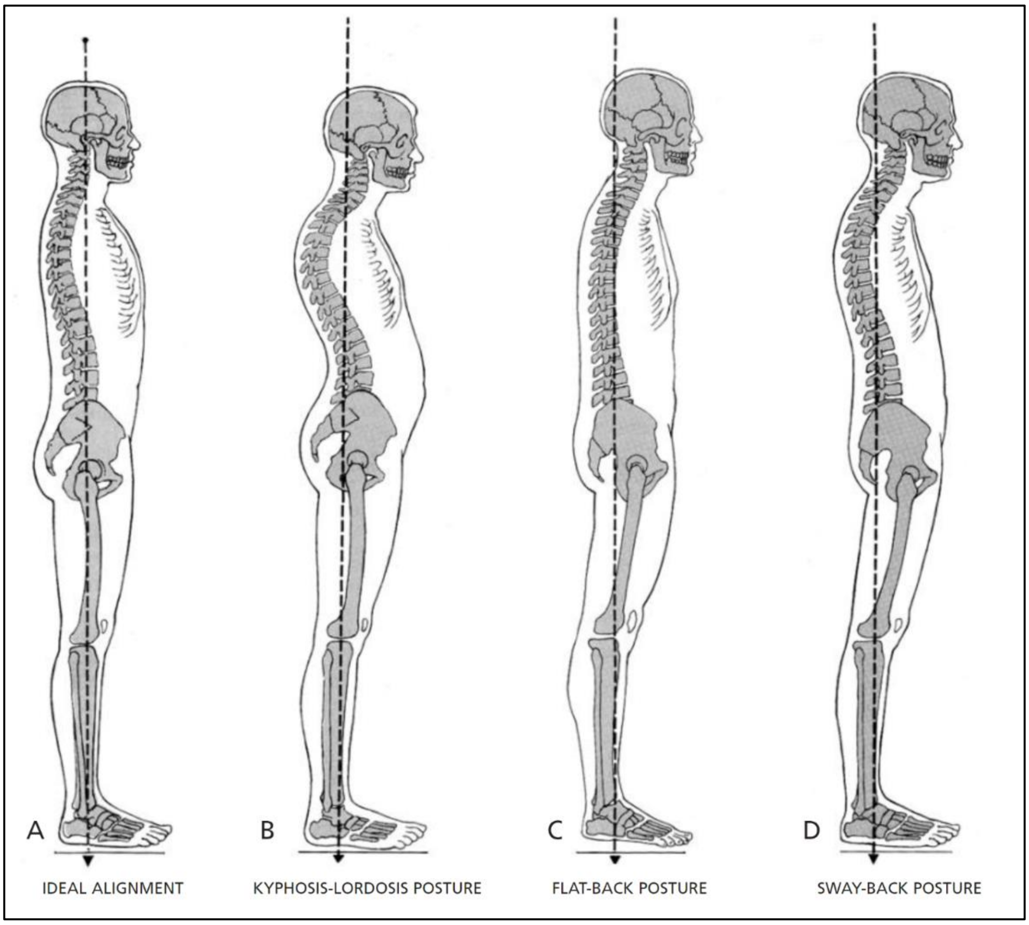


Figure 16 FOUR TYPES OF POSTURAL ALIGNMENT[4]

[4] "Mat/Ref Support Materials Sample by Merrithew™ - Issuu." Accessed: Nov. 14, 2024. [Online]. Available: https://issuu.com/merrithewhealthandfitness/docs/4486l_mr_support

Some useful imagery to help people align better; “Stand with feet underneath the hip joint.,” “Feel the entire foot on the floor,” “Bend the knees almost into a squat. Notice the crease in the hips. Stand and maintain a feeling of softness across the front of the hips and behind the knee.” [24].

There are also exercises that aim to help with better posture by strengthening supporting muscles and making the hips more flexible such as Pelvic Floor Exercises, Pelvic Tilts and Rolls, Cat Stretch and Rolling the Spine described more in detail in the journal “EFFECTING POSITIVE CHANGE: A MANUAL FOR TEACHERS OF SINGING” [24].

<u>Vocal Timbre</u>

Timbre is the soul of the voice (i.e. the identifying characteristics of one’s voice, for example, when the same note is played on two different instruments e.g. Piano and Trumpet, we can tell which one is which). It cannot be changed as it is related to the physical structure of a person’s body.

A study looking at diverse ways to describe vocal timber found that vocal timbre can be described through various parameters that capture both the physiological production of sound and its perceptual qualities [37]. Key descriptors include the phonation onset, which can range from “breathy” or “creaky” to “smooth” or “glottal,” and the position and shape of the vocal tract, influencing resonance and formant frequencies [37].

Adjustments in the vocal tract such as the constriction or expansion of the pharynx, larynx positioning, and velum movement, lead to distinct timbres like “oral twang", "nasal twang,” or “sob,” each characterised by specific acoustic and physiological traits [37].

Lastly, vocal timbre can be modulated by breath support and muscular anchoring, where high airflow and muscular engagement often accompany powerful timbres like belting, while softer sounds like falsetto and breathy voices involve lower pressure and airflow. [37]

A machine learning model can be trained on a set of labelled data that includes a natural speech sample, descriptors, and higher formant frequency ranges to be able to predict the user’s vocal timbre. The frequency data can be captured using FFT and added as a descriptive feature to the predictive model.

2.6. Vocal Exercises Research

In this section I will go through my singing exercises research, describing what each exercises category is trying to achieve and the exercises found that can be implemented into the project.

2.6.1. Warmups and Cooldowns

Warming up the voice before singing has been shown to be beneficial in creating a more stable sound when singing and cooldowns are beneficial in releasing tension in the vocal folds after the singing session has been completed [28].

Some common warmups include singing scales, hums and thrills and body stretching [29, 30].

2.6.2. Pitch Accuracy and Range

Pitch matching directly trains the user to recognise and sing pitches. It is done by playing a note on the piano and then trying to match it with your voice.

Another way to improve pitch range and accuracy is to practice scales and intervals [35]. Scales are sets of sequential notes that follow a pattern and sound good together. There are several types of scales such as major and minor scales, chromatic scales, pentatonic scales etc. [36]

2.6.3. Breath Control and Support

Breath control training has been shown to help singers use abdominal-diaphragmatic-costal breathing which in turn proves their singing [38].

For better breath support, diaphragmatic breathing can be performed. It is done by sitting comfortably, placing one hand on the upper chest and the other below the rib cage and breathing in slowly through the nose, tightening the abdominals and then exhaling trough pursed lips [39].

Another exercise is the 'hissing' exercise which enhances breath control and capacity by regulating airflow. It is performed by inhaling deeply and exhaling slowly while making a "ssss" sound, sustaining it as long as possible [40].

Lastly to help with the ability to take in a good breath quickly, an exercise involving breathing in quarters is helpful. "On the count of 1 breathe in 1/4 full, on the count of two breathe in 1/2 full, on the count of three breathe in 3/4 full and on the count of four breath in the final quarter. Repeat as you exhale, again in four stages. Sing:1, 121, 12321, 1234321, 123454321, 12345654321, 1234567654321, 123456787654321 and silently snatch a breath at each comma. "[41]

2.6.4. Resonance and Tone Quality

Exercises focusing on tone quality have been found to be beneficial in improving students' tone [32].

In this section I will describe some of the common exercises found focusing on improving tone quality and resonance. Some of these are used in choirs while others are recommended by singing coaches to their students.

Repeating a vowel sequence "Ah-ai-ee" on a single pitch [31]. This exercise is performed by singing the vowels consecutively on the same comfortable pitch. As the transition between the vowels is happening, the abdominal muscles should be engaged in a gentle swinging or pulsing motion to support breath and vocal production. This exercise promotes better resonance and richness in tone by practicing vowel transitions and strengthens diaphragmatic support for better vocal endurance.

Humming exercises promote forward resonance, improving tone clarity and richness [33]. To perform humming exercises effectively, start by sitting or standing in a comfortable and relaxed position, ensuring good posture with your back straight and shoulders relaxed. Begin by breathing in through your nose and out through your mouth silently until you establish a smooth rhythm. Once comfortable, inhale through your nose and exhale on a gentle "hmm" sound with your lips closed, feeling the vibrations on your lips. Maintain a comfortable pitch, avoiding strain, and stop if you experience discomfort or tension.

The process is repeated 10 times and progressively adding vowel sounds then 'mmm' at the beginning of words and finally moving on to full sentences with 'mmm' as the person becomes more comfortable with the exercise [33].

The last exercise found is lip thrills, they help with balancing airflow and vocal cord tension, promoting a resonant tone.

"Sit or stand with good posture and take a deep breath in through your nose, allowing your stomach to expand (imagine that you are filling up a balloon). As you exhale through pursed lips, make an "r" sound (think of how horses snort!). The goal is to make a continuous "rrrr" sound without pausing for breath—you should feel vibrations throughout your entire body! Start with 10 trills and work up to 20 as you get more comfortable with the exercise. "[34]

2.7. Conclusions

In this chapter, we delved into the existing solutions available for vocal training, evaluated various technologies for implementation, and conducted additional research to inform the development of Aria.

The analysis of existing applications like Vocal Image, Yousician, and Sing Sharp revealed a gap in the market for a comprehensive, personalized singing training app that focuses on vocal health and provides detailed feedback across multiple aspects of singing. These apps work well for basic training and entertainment but lack the depth and personalization required for significant vocal improvement.

Next.js was selected for the client-side due to its performance benefits and strong community support. FastAPI was chosen for the server-side because of its high performance and compatibility with Python's rich ecosystem of audio processing and machine learning libraries. Firebase was selected as the database technology for its real-time capabilities and ease of integration, which are crucial for Aria's features like live feedback and progress tracking.

The requirements research highlighted the needs of amateur singers, emphasizing the demand for live feedback, personalised exercises, and progress tracking. Vocal pedagogy research provided a foundation for developing effective training methods and understanding the complexities of voice analysis. The exploration of vocal analysis techniques informed the selection of appropriate algorithms and tools for accurate assessment.

This research has established a solid foundation for Aria's development. By understanding the limitations of existing solutions and carefully selecting technologies that align with this project’s objectives, an application that effectively supports users in improving their singing abilities through personalised, comprehensive training can be created.

3. System Design

3.1. Introduction

In this chapter, the project's software methodology and system architecture will be explained in detail including a complete overview of the system, the compiled user requirements as well as the UI/UX designs. The implementation plan for each feature is also discussed.

3.2. Software Methodology

There are multiple software methodologies that are popular in the software industry.

The waterfall method is one of the oldest and most traditional ones. It is a linear sequential process, going through stages such as requirements gathering, design, system testing, implementation, verification, and deployment. It has a clear structure and is easy to manage however similarly to a real waterfall, as it flows through one stage to another, there is no time to reflect on what has been done and it could end up leading to the creation of a project that does not fulfil the user requirements. As this project is heavily end-user focused, and requirements will need to be constantly reviewed to make sure the product truly improves the users singing, the waterfall approach will not offer the best structure.

Scrum is an agile project management framework that provides structure for teams to manage their work through a set of values, principles, and practices [20]. It works by organising the tasks needed to complete a project into a scrum artefact called a Product Backlog. Then some of the tasks from the Product Backlog are selected to be developed during a period (i.e. two weeks), called a Sprint. After a Sprint is completed, the results are tested, and the next Sprint gets planned. This process is repeated until the full Product Backlog is completed.

Scrum is based on empiricism and lean thinking and employs an iterative and incremental approach to complete work which allows for the constant review of work completed makes sure that all user requirements are accurately met [10].

This project is individual work therefore some of the Scrum figures and practices will be excluded. For example, the Scrum Master, Scrum Product Owner, and Scrum development team will all be the author. Sprint reviews (i.e. a meeting at the end of a sprint to show and discuss the work done during the sprint [20]) and Sprint planning (i.e. a meeting held at the beginning of a sprint to identify which tasks should be completed during the sprint [20]) will be conducted in collaboration with the supervisor.

Daily stand-ups (i.e. a short daily meeting of the scrum development team, scrum master and product owner to discuss progress [20]) will be omitted.

3.3. System Architecture Approaches

There are dozens of unique design patterns and architectures when it comes to developing software applications. I will be exploring three of the most suitable ones for this project.

3.3.1. Client–server pattern

The client-server model is simple to understand. The user (e.g. client) interacts with the user interface and sends requests to the server that processes the request and sends back a response.

As the server is handling most of the logic, it allows for easy resource sharing, scalability opportunities and easier administration and maintenance. However, it also introduces issues such as single point of failure and network dependency.

As this project will be dealing with a lot of data processing (i.e. Evaluating the user's voice during initial vocal assessment and exercises) the use of a server looks like a clever idea. However, due to the nature of the app needing real time feedback for some of the features (i.e. During singing exercises), having all the logic on the server and being network dependent might not be the most optimal solution.

3.3.2. Model–view–controller

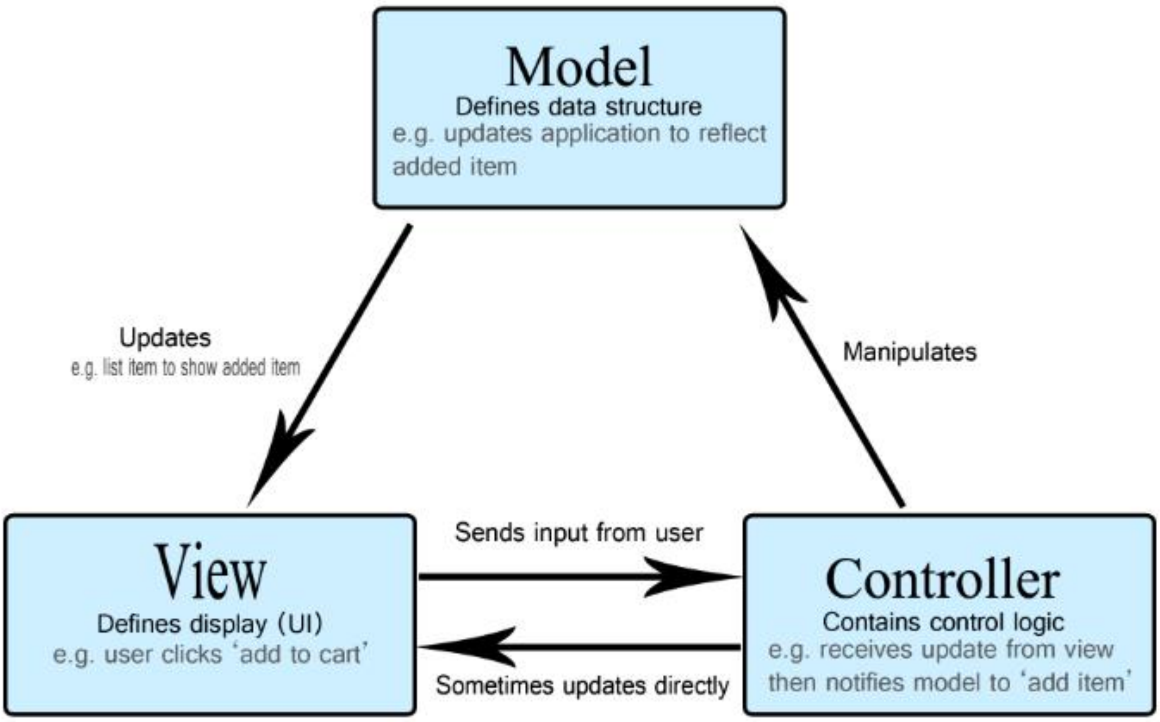


Figure 17 MVC FROM MDN [19]

The model-view-controller pattern involves separating the project's logic into three interconnected elements. The model typically manages data and business logic, the view deals with the layout and display, and the controller updates the model or view in response to input from the user [19].

In the case of this project, the model would store data about the user, define the exercises, and store the theory lessons, it would also do the evaluations and processing, the view would display the app's features such as the analysis test interface, the exercises screen, and daily recommendations dashboard.

MVC can provide a good structure and helps organise code, however since Aria has real-time features, the latency created by this model will not provide for a viable solution.

3.3.3 Microservices and Monolithic architectures

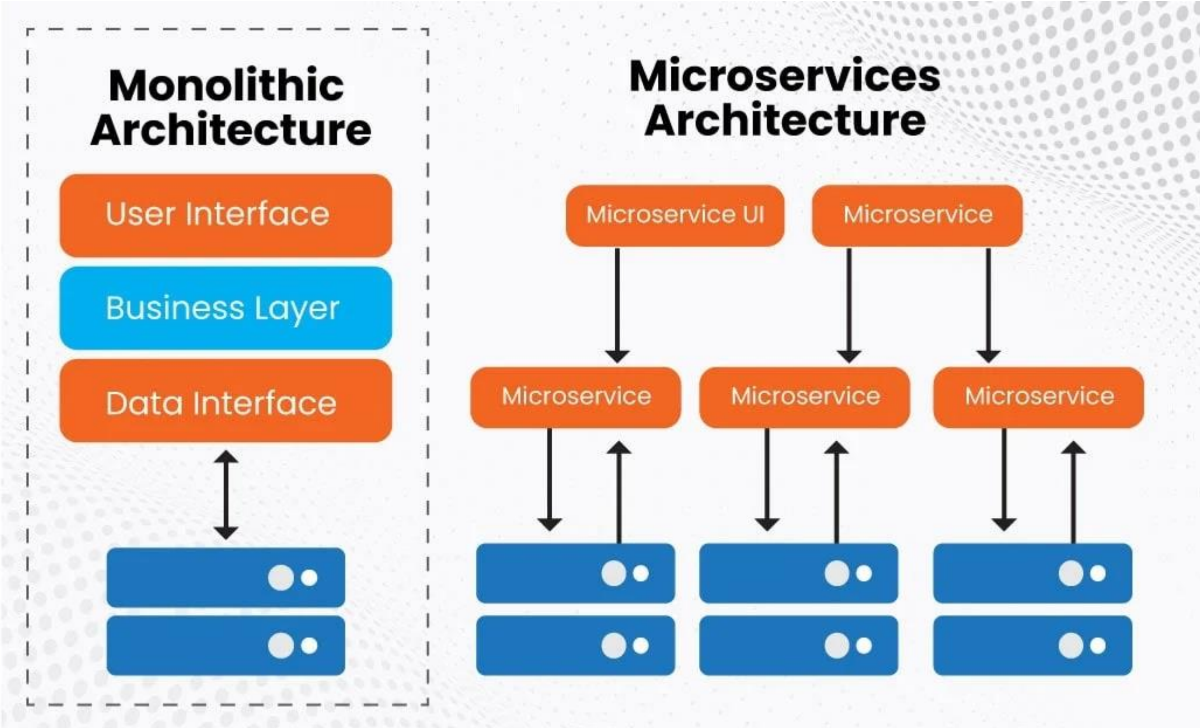


Figure 18 : MONOLITHIC ARCHITECTURE VS. MICROSERVICES

In a monolithic architecture, the application is developed as a single, unified unit, with interconnected and interdependent components. Typically, it encapsulates all business functions such as data access, user interface, and business logic into one codebase and deployable artifact. This provides simplicity and better performance as all processes happen within the same application. However, it does cause scalability limitations due to only being able to scale horizontally and it slows down development for bigger project due to the creation of phenomenon called "spaghetti code." What this means for this project is that with a monolithic architecture, due to the complexity of it, the code will become hard to debug and test.

The microservices architecture is a modular approach where an application is structured as a suite of small, independently deployable services, each focusing on a particular function or capability. These services are organised based on the project's requirements and are independently deployable, each having its own interface, logic, and database. They communicate with each other using lightweight protocols, usually HTTP or message queues. This architecture offers better scalability, fault-tolerance, and ease of development as each component is independent of one another. The down sides being increased complexity due to the need for careful data management and latency issues and often requiring the use of tools like Docker or Kubernetes for monitoring.

Microservices are fantastic for large, complicated, and commercial applications such as Netflix, where there are multiple teams working on the project, however for this project, as it is developed by one person, it would introduce unnecessary complexity and difficulties with deployment.

3.4. Overview of System

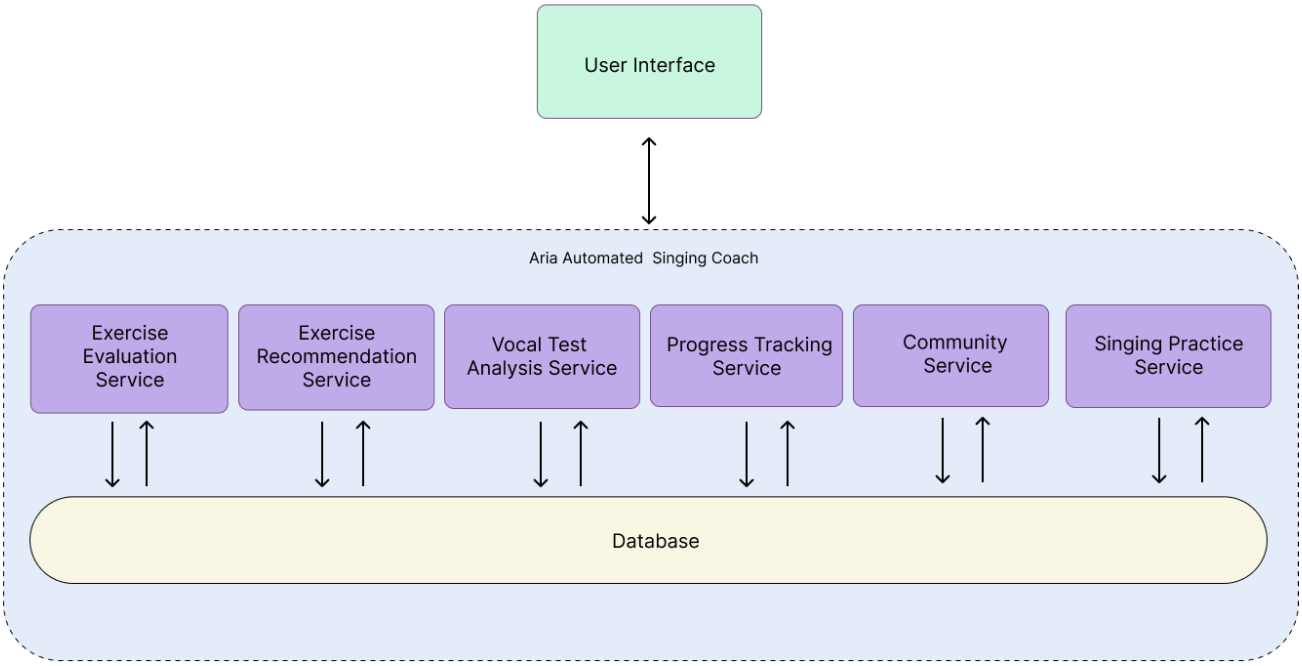


Figure 19 ARIA SYSTEM ARCHITECTURE

The architecture picked for this project is a Hybrid Mono-Microservices Architecture. With this architecture the project gains the ability to use modularised components while still using a centralised database and being deployed as a single application.

The entry point of the application is the displayed user interface on the browser. Each of the services can be called using an API from the UI. Each service has access to the database and can call other services if needed.

3.5. User Requirements – Product Backlog

Based on questionnaire responses and the research outlined in the previous chapters, the following requirements were deducted. These will also serve as the Product Backlog for the application.

User Interface

ID	Task Name	PRIORITY	SPRINT	STATUS
1110	Landing Page	Medium		Not Started
1111	Sing In and Sing up Page	Medium		Not Started
2220	Vocal Analysis Test Start Page	High		Not Started
2221	Vocal Analysis Test "Correct Posture" Page	High		Not Started
2223	Vocal Analysis Test "Speak In Natural Voice" Page	High		Not Started
2224	Vocal Analysis Test "Sing Scale" Page	High		Not Started
2225	Vocal Analysis Test "Inhale and Exhale" Page	High		Not Started
2226	Vocal Analysis Test "Results" Page	High		Not Started
2227	Vocal Analysis Test "Recommended plan" Page	High		Not Started
3330	Home Page - Dashboard, Navigation Menu	High		Not Started
4440	Daily Exercise Start Page	High		Not Started
4441	Daily Exercises Warmup Page	High		Not Started
4442	Daily Exercises Exercise Page	High		Not Started
4443	Daily Exercises Exercise Finish and Feedback Page	High		Not Started
5550	Theory Lessons Page	Medium		Not Started
5551	Theory Lessons: Breathing Techniques Page	Medium		Not Started
5552	Theory Lessons: Vocal Anatomy Page	Medium		Not Started
5553	Theory Lessons: Pitch and Intonation Page	Medium		Not Started
5554	Theory Lessons: Vocal Registers	Medium		Not Started
5555	Theory Lessons: Tone and Timbre	Medium		Not Started
5556	Theory Lessons: Vocal Health	Medium		Not Started
5557	Theory Lessons: Performance Techniques	Medium		Not Started
6660	Progress Tracking Page - Dashboard, Streak	High		Not Started
7770	Community - Streak Leaderboard	Low		Not Started

Exercise Recommendations Service

ID	Task Name	PRIORITY	SPRINT	STATUS
4442220	Data: Create 3 sets of exercises for Pitch Accuracy and Range	High		Not Started
4442221	Data: Create 3 sets of exercises for Breath Control and Support	High		Not Started
4442222	Data: Create 3 sets of exercises for Resonance and Tone Quality	High		Not Started
4442223	Data: Create 3 sets of exercises for Register Blending	Medium		Not Started
4442224	Data: Create 3 sets of exercises for Dynamic Control and Expression	Medium		Not Started
4442225	Recieves the Vocal Test results and creates a recomendation for the exercises that should be done today.	High		Not Started
4442226	Creates a recommendation for the exercises that should be done today based on the user's current progress.	High		Not Started

Voice Test Analysis Service

ID	Task Name	PRIORITY	SPRINT	STATUS
2221110	Interpret vocal analysis to provide detailed feedback using an LLM	High	1	Not Started

Progress Tracking Service

ID	Task Name	PRIORITY	SPRINT	STATUS
6661110	Recieve exercise evaluation data, process it and store it in the database - update progress, update streak	High		Not Started

Community Service

ID	Task Name	PRIORITY	SPRINT	STATUS
7771111	Get leaderboard data and send it to UI	Medium		Not Started

Singing Practice Service

ID	Task Name	PRIORITY	SPRINT	STATUS
8881110	Data: Create 5 songs - lyrics, pitch, techniques	Medium		Not Started
8881111	GET song data and send to UI for display	Medium		Not Started
8881112	Recieve user recording and pereform an indepth analysis to provide detailed feedback	Medium		Not Started

3.6 UI/UX Design

The first step of this project's development is the UI/UX designs. Figma was used to create a general layout and flow of the application. The designs serve as an inspiration for the development phase and elements will vary in the final product. The designs can be seen in Appendices A.

As the website was being developed during Christmas break, a complete overhaul of these designs was made to make the application look more user-friendly and professional, see Appendices B.

4. Software Development

4.1. Introduction

This chapter outlines the structure and implementation of the application, beginning with the user onboarding process and initial voice analysis, and continuing through daily practice exercises, theoretical lesson modules, and karaoke engagement.

4.2. Main Structure

4.2.1. Voice Analysis Test and Sign Up

An essential component of personalised vocal training is the ability to evaluate a user's current vocal capabilities objectively. The Voice Analysis Test was developed as an interactive and data-driven assessment tool to capture key metrics of a user's voice. These metrics serve as a foundational baseline, enabling the system to monitor progress over time and tailor subsequent exercises according to each individual's vocal profile.

The test is a part of the Sign-Up process for new users and measures the average vocal pitch of the user, determines the user's singing range (lowest and highest sustainable notes), assess pitch accuracy in relation to a reference musical scale and evaluates the user's lung capacity and breath control ability.

It is implemented using a combination of FastAPI (Python) on the backend and Next.js (React) on the frontend, ensuring smooth and responsive interaction between the user and the application. The backend processes the audio data, performs analysis using Librosa [43] for pitch detection and FFmpeg for audio format conversion, and delivers structured results in a standardised format.

The frontend provides an interactive, step-by-step interface guiding the user through the different phases of the test. These steps include:

Average Pitch Detection:
The user records a short vocal sample. The system calculates the average pitch of the sample by extracting the high-confidence pitch values and computing their mean frequency.

Pitch Range and Accuracy Analysis:
The user is prompted to sing a predefined sequence of notes (a musical scale). The backend analyses the sample, identifies the most prominent notes, and compares them to the target scale. Each note's cents deviation (how far off the note is from the reference) is calculated to assess pitch accuracy. The system also determines the user's overall singing range, identifying the lowest and highest notes sung.

Lung Capacity Measurement:
The user performs a controlled breathing exercise involving inhalation, breath-holding, and exhalation phases. The application records the duration of each phase to estimate the user's breath control and lung capacity.

All results are then stored securely in a Firebase database, as a collection linked to the user's account.

4.2.2. Vocal Exercises and Daily Practice

The vocal training system developed in this project is structured around three principal categories of exercises: Breath Control, Dynamic Control, and Pitch Accuracy and Range. Each category corresponds to a dedicated folder in the codebase, encapsulating the logic, assets, and visual components required for that specific training focus.

Breath Control Exercises

Breath control is often the least varied category in terms of traditional vocal training exercises, which posed an opportunity for innovation through gamification. To address this, I implemented three interactive exercises that transform standard breath work into playful and visually engaging experiences.

The first of these is the Balloon Game, where users are tasked with inflating virtual balloons by exhaling steadily into the microphone. Each successful exhale creates a balloon, with its size proportional to the duration of the breath. This seemingly simple interaction presented notable technical challenges. Capturing continuous exhalation using standard microphone input is inherently noisy; most systems cannot distinguish between exhalation and speech purely based on volume. In the current implementation, breath is inferred using real-time audio amplitude analysis via the Web Audio API's AnalyserNode, treating any non-silent, sustained input above a volume threshold as breath. This is a heuristic approximation and, while effective for the gamified setting, remains an open area for improvement.

Visually, the balloon is animated using Framer Motion with physics-inspired transitions to create a natural inflation effect. Careful attention was paid to screen geometry—ensuring that the balloon is centred, visually appealing, and remains within safe bounds of screen size, especially on devices with smaller resolutions. Balloons float away once completed, and stats such as average exhale duration, stability (calculated from standard deviation of durations), and progress towards set goals are tracked and stored.

The second exercise, Bird Scroller, is inspired by Flappy Bird [44]. Here, the bird's vertical position is mapped directly to the user's breath volume: steady exhaling causes the bird to rise, while stopping allows it to fall. The goal is to maintain flight for a set duration, with occasional breaks granted for inhalation. The continuous scrolling environment and randomly generated platforms introduce a gentle difficulty curve. The game loop is handled through requestAnimationFrame, and collision logic is based on bounding box intersection with platforms—requiring precise breath control to maintain altitude.

The third breath control activity is the Candle Game, where a row of candles must be extinguished through sustained exhalation. Each candle takes one second to extinguish, requiring the user to maintain breath without pause. The key challenge in this exercise was temporal segmentation of breath and synchronisation with animated feedback. A soft real-time mechanism is employed to measure the duration of breath segments and to control the progressive extinguishing of candles. Smooth UI updates are achieved by decoupling rendering from logical updates using separate useEffect loops and requestAnimationFrame for the visual timers. The game records breath duration, stability, and consistency, again using amplitude-based analysis. A cooldown is introduced between rounds to simulate natural breath cycles and avoid hyperventilation.

All three exercises share common architectural patterns: real-time audio sampling, breath detection through FFT analysis, cooldown logic, user feedback through animation, and structured stat tracking. The exercises culminate in a performance report stored in Firebase, which includes metrics like exhale duration, stability, and any achievements earned.

Dynamic Control

The implemented exercise, Crescendo/Decrescendo, instructs the user to sustain a pitch while increasing their volume over four seconds (crescendo) and then decreasing it over the next four seconds (decrescendo).

The technical implementation involves capturing live audio data, computing root-mean-square (RMS) volume, and detecting pitch using the YIN algorithm from the pitchfinder library [45]. Volume data is normalised based on an empirically determined maximum input amplitude, ensuring meaningful feedback regardless of microphone sensitivity.

The interface provides real-time visualisation of both volume and pitch, with a volume bar and pitch display updated on each animation frame. The input is filtered to only accept pitch within the human vocal range (50–2000 Hz) to avoid artefacts caused by ambient noise or system hum. At the conclusion of the exercise, statistics such as maximum and minimum volume, average pitch deviation from the target, and overall performance are calculated and stored.

The challenge here was achieving responsive, low-latency pitch and volume analysis in the browser without relying on native plugins. The entire pipeline—from microphone access to real-time display—operates entirely on the frontend, with careful optimisation of FFT window sizes and smoothing parameters.

Pitch Accuracy and Range

The final category focuses on pitch accuracy and vocal range extension through Scale Training. The system first fetches the user's vocal range—previously calculated and stored in Firebase—and generates a major scale within this range. Notes are synthesised using soundfont-player [46], and the scale is played back to the user, accompanied by a visual interface that resembles a karaoke-style note-following timeline.

The canvas-based visualisation maps each note to a segment of the timeline, and a moving line helps the user track timing. During the user's attempt to sing the scale, pitch is continuously detected using YIN, and a red marker is drawn in real time to indicate pitch accuracy. Each note's duration, pitch deviation (in cents), and timing alignment are recorded.

Results from this exercise include percentage accuracy (based on how long the user stays within tolerance range) and average deviation in cents across the entire scale. These metrics are not only displayed immediately but also stored to inform future scale generation, adapting the training difficulty as the user's range improves.

Daily Practice Routine

To guide consistent user engagement, a daily practice routine has been implemented. It consists of a warm-up phase followed by one exercise from each category—selected either manually or based on previous performance trends. This routine ensures balanced vocal development and provides a familiar structure for users.

4.2.3. Theory Lessons Generation

The development of the theory lessons was guided by both educational considerations and technical implementation, with an aim to provide a structured yet interactive way for users to engage with foundational vocal concepts. To produce the lesson content, ChatGPT was used as a generative language model. Specific prompts were crafted to request pedagogically sound overviews on topics such as Articulation Basics or Vocal Health. For example, prompts like "Generate the information needed to provide a lesson on the topic of Articulation Basics in singing practice" enabled rapid content prototyping, which could then be edited and modularised for delivery within the application.

The platform tracks user progress by storing the identifiers of completed lessons within a completedLessons array in each user's document in the Firestore database. This list is fetched and compared to the locally defined lesson set on page load. This dynamic enables real-time user progress tracking and visual representation through a progress bar, shown on both individual category cards and at an overall course level. This progress metric is calculated by iterating through all defined lessons and comparing them with the user's completion data retrieved from Firestore.

The application listens for authentication state changes and redirects unauthenticated users to a login page. Upon successful login, the app retrieves the user's lesson completion data and uses this to dynamically update lesson cards and category progress indicators.

Each theory lesson is structured into thematic categories such as Vocal Basics, Breath Control, and Advanced Techniques, with associated durations and completion flags. The lesson data is stored locally in a structured object (initialCategories) and rendered using reusable card components. These cards include icons from the react-icons library and are styled using TailwindCSS with gradients and transitions that make the interface more visually engaging.

The user interface dynamically reflects the user's learning journey. When a lesson is completed—either by finishing a quiz or manually marking it complete—this change is persisted in the Firestore database. The updated lesson status is reloaded when the user returns to the theory section, ensuring state consistency across sessions.

One particularly interesting implementation detail is the way lesson progress is handled. The application does not rely on server-side rendering for user-specific data but rather performs all operations client-side after authentication, maintaining responsiveness while reducing backend complexity. This also ensures a separation between static content (lesson metadata) and dynamic user-specific data (completion status).

The architecture also supports a gamified quiz module. Each lesson may optionally conclude with a multiple-choice quiz, also stored on the client, with the results tracked via Firebase. For instance, the lesson on Articulation Basics includes a quiz with five targeted questions, designed to reinforce the material through retrieval practice. When all questions are answered correctly, the application triggers a visual celebration using react-confetti, and the lesson is marked as completed. This interactive feedback loop fosters user engagement and serves as a formative assessment mechanism.

4.2.4. Karaoke Practice

The karaoke practice is designed to bridge the gap between guided vocal training and recreational singing. At its core, the karaoke is structured around two major components: the song dashboard interface, which offers users a curated and filterable list of available tracks, and the real-time lyric and pitch-matching interface, where active vocal engagement takes place. These are implemented in two respective React components: the SingingPage and RealTimeLyrics.

The SingingPage is primarily concerned with user experience and data presentation. It leverages Framer Motion for smooth animations and @formkit/auto-animate for dynamic DOM updates, enabling a visually responsive environment as users filter songs by genre or keyword. Songs are stored locally in an array but can be expanded to fetch from Firebase Firestore. Each song entry includes metadata such as title, genre, duration, album art, and importantly, links to its corresponding lyrics and audio files. User preferences are supported through toggling of the "like" state, visually represented with animated heart icons. To maintain engagement, dynamic UI elements such as confetti hearts and genre-coloured buttons further personalise the experience.

Upon selecting a song, the user is routed to the RealTimeLyrics component, which serves as the performance and evaluation environment. This view integrates real-time audio input, lyric synchronisation, and pitch accuracy visualisation. The implementation combines pitchfinder (specifically, the YIN algorithm), the Web Audio API, and @tonaljs/tonal to convert live microphone input into quantised MIDI values. These values are then compared against the expected notes for a given lyric line.

An innovative aspect of this implementation is its micro-sustain smoothing algorithm. To avoid flickering pitch data and false negatives, the detected MIDI values are buffered in a rolling window and averaged using a weighted linear interpolation. This produces a smoothed, representative pitch, which is then mapped back to a musical note using tonal theory. The output is displayed to the user as a comparison between the "target" note—parsed from the lyric JSON structure—and the live-detected pitch. This comparison is colour-coded to show pitch accuracy, employing a tiered system that ranges from cool blue (precise) to red (inaccurate), offering immediate visual feedback.

Each lyric line is tied to time-anchored segments with optional note annotations. These are parsed at runtime from structured JSON files that contain not only the text but a nested array of expected pitch-time mappings. As the audio plays—managed through a controlled audioRef—the current time is tracked and used to determine the active lyric line. The UI updates using Framer Motion transitions, providing both a smooth fade-in and an animated progress bar that visualises the line's current playback status. This visual reinforcement further aids temporal coordination between visual input and vocal output.

User exits and settings are managed with modal overlays, ensuring session data is respected and user choices are deliberate. The inclusion of a microphone calibration modal (MicCheckModal) is particularly crucial from a UX perspective, as it ensures a functional audio stream before the karaoke session begins, mitigating frustration due to browser permission errors or muted devices.

Beyond the front-end visualisation, the karaoke module also accommodates extensibility. The underlying architecture is modular and clearly demarcated, allowing for future integration of backend features such as pitch performance logging, personalised feedback, or even machine learning-based singing assessment. Furthermore, the system supports a variety of audio formats and lyric structures, allowing non-Western musical styles (e.g., K-pop, J-pop) to be incorporated just as easily as classical Western ballads.

4.3. Song Details Generation

A separate Python programme was developed to extract and structure lyric segments and pitch information from the a cappella versions of each song.

The required a cappella files were sourced and manually downloaded from YouTube in WAV format.

The audio is then processed using the Librosa [43] library to detect pitches over time, with any noise or irrelevant pitches filtered out. These pitch results are subsequently converted into MIDI notes for a clear digital representation.

In parallel, the Whisper model [42] is employed to transcribe the lyrics, producing start and end timestamps for each segment. By aligning these timestamps with the pitch data, the system constructs a synchronised map of each lyric line and its corresponding notes. The data are then serialised into JSON, enabling easy retrieval and display within a karaoke interface.

Once the JSON file is loaded on the frontend, each lyric line is displayed in time with the background instrumental, while the pitch data provide visual cues for real-time singing accuracy.

During testing, the system performed well on English songs such as Sabrina Carpenter's "Please Please Please" and Adele's "Someone Like You," accurately capturing the lyrics and aligning pitch transitions. For Korean and Japanese songs—like IVE's "HEYA" and Ado's "Usseewa"—the transcriptions were reasonably accurate, though some slight inaccuracies emerged, likely owing to limited training data and language-specific quirks of the model. Nonetheless, this framework convincingly demonstrates how extracted notes and timestamps can be used to deliver a true karaoke experience, and it can be readily extended to build a larger song library with minimal additional effort.

4.4. Conclusions

The development of Aria demonstrates the feasibility and pedagogical value of a web-based, data-driven singing trainer that blends audio signal processing with gamified learning. By offering an ecosystem of tailored vocal exercises, real-time pitch feedback, and engaging karaoke practice, the application successfully supports both structured training and recreational singing in a unified platform.

The project tackled a variety of technical challenges—from reliable browser-based pitch and volume analysis to the accurate alignment of lyrics and notes in multiple languages. Key innovations, such as the smoothing of live pitch input and the generation of synchronised lyric-note mappings, were central to enhancing user experience and training accuracy. Moreover, the use of modern frontend frameworks and serverless architecture facilitated the creation of a responsive and scalable system that can be easily extended in future iterations.

Importantly, the system not only captures performance data but also leverages it to adapt training materials to individual users, offering the potential for personalised feedback and long-term skill tracking. With all user data securely stored in Firebase and core logic decoupled across modular components, Aria provides a robust foundation for future developments, including machine learning-driven analysis, expanded lesson content, and a community-based progression model.

5. Testing and Evaluation

5.1. Introduction

A critical part of the development lifecycle for any software system is the implementation of a rigorous and structured testing plan. In the context of Aria, testing had to account not only for standard user interface reliability and system integration but also for more complex, domain-specific challenges such as real-time audio analysis accuracy and synchronised visual feedback. This chapter outlines the comprehensive testing approach adopted throughout the project lifecycle, including unit testing, integration testing, and user acceptance testing. It also presents the outcomes and learnings gathered from this process, as well as the evolution of the testing plan in response to iterative development.

5.2. System Testing

System testing for Aria encompassed a broad range of both functional and non-functional assessments, including responsiveness, cross-platform compatibility, audio synchronisation accuracy, and vocal pitch detection validation.

A particular focus was placed on verifying the accuracy of the pitch detection module, a core functionality in both the vocal exercises and karaoke practice features. To validate pitch accuracy, the YIN-based pitch detector used in the browser was benchmarked against known reference tones available on YouTube and other publicly available pitch-matching videos. In these tests, sustained notes from reference videos were played into the microphone while the application ran in detection mode. The output MIDI note and detected frequency were recorded and compared against the expected musical pitch.

This process was repeated across a broad frequency spectrum, including pitches from both male and female vocal ranges (approximately MIDI 48–85). The results demonstrated that the system consistently identified pitch mostly correctly. These results fell well within the acceptable range for the purposes of real-time vocal feedback in a learning environment. Furthermore, the micro-sustain algorithm introduced in the karaoke component proved essential in eliminating false pitch flickering, particularly during natural vibrato or tonal decay.

The lyrics synchronisation system was also validated by comparing it against the official lyrics of each song. Playback testing ensured that each lyric line appeared and transitioned at the correct moment relative to the audio track, with edge cases (e.g., very short or overlapping lines) tested through both automation and manual review.

5.3. System Evaluation

To evaluate the usability and effectiveness of Aria in a real-world setting, user testing was conducted with a sample group of vocal learners. These users were asked to engage with all major components of the system—voice analysis, daily exercises, theory lessons, and karaoke practice—over multiple sessions. The evaluation process was structured into three stages: initial observation and feedback collection via questionnaire. The user questionnaire included both Likert-scale and open-ended questions, targeting various aspects such as interface clarity, visual feedback usefulness, ease of navigation, and perceived learning benefit as seen in Appendix C.

Overall, responses were positive, with most users citing the karaoke feature and animated feedback during exercises as particularly helpful and engaging.

In addition to survey data, user sessions were observed with consent to identify any usability issues or errors. These sessions revealed minor inconsistencies in the behaviour of breath detection under high ambient noise conditions, prompting adjustments to the amplitude threshold values and FFT smoothing settings.

The feedback loop led to multiple improvements throughout the project, including adjusting the pitch tolerance ranges for more realistic feedback, adding cooldown logic in breath games to let the user take a break in between and simplified navigation in the karaoke module for better accessibility on touch devices.

5.4. Conclusions

Through targeted system testing including pitch detection validation against external references the project ensured the functional integrity of its most complex features.

User evaluation, via questionnaire and live interaction, provided critical insights into the perceived value and effectiveness of the system. Not only did this process confirm the pedagogical merit of the application, but it also identified key areas for refinement that were addressed before final deployment.

The testing plan evolved iteratively throughout the development lifecycle, embracing a continuous improvement philosophy. Each new feature introduced during development was accompanied by targeted unit and system-level tests, and all feedback received was systematically incorporated into future design sprints.

6. Conclusions and Future Work

6.1. Introduction

This chapter reflects on the progress made throughout the development of Aria, summarising the key accomplishments and outlining areas for future enhancement. The work completed thus far demonstrates the viability and potential of the application, while also identifying opportunities to expand its functionality and reach.

6.2. Plans and Future Work

While Aria already provides a solid foundation for accessible vocal training, there remain several avenues for future development. Expanding the song library is a priority, both in terms of quantity and the accuracy of lyrics and timing synchronisation. A more refined and extensive repertoire would significantly enhance the user experience and allow for greater variety in practice sessions.

Further personalisation is also a key goal. Tailoring exercises and song recommendations to the user's specific vocal profile and preferences would increase engagement and the effectiveness of training. Additionally, incorporating more exercise types could address a broader range of vocal challenges and techniques.

The social dimension of learning has yet to be fully explored. Introducing features that enable users to collaborate, communicate with friends, or even participate in challenges could foster a sense of community and motivation.

Two planned features that could not be implemented within the scope of this phase are the leaderboard system and detailed progress tracking. Both elements would provide valuable feedback loops and gamification, encouraging consistent practice and celebrating users' achievements. Their implementation remains an important objective for future iterations.

6.3. Conclusions

Aria has demonstrated strong potential as an inclusive and practical platform for vocal training, particularly for individuals without access to traditional coaching. The current implementation showcases a well-structured architecture, key pedagogical features, and an intuitive user interface that collectively support musical development.

By integrating real-time vocal analysis, structured exercises, and foundational music theory, the application offers a comprehensive solution for beginner and intermediate users alike. While there is room for growth, especially in terms of personalisation, community features, and advanced tracking, the groundwork laid thus far provides a promising base for continued innovation.

With further development, Aria can evolve into a robust, community-driven platform that empowers users to explore and expand their vocal potential. The project affirms the value of merging modern technology with the principles of vocal pedagogy to foster creativity, discipline, and artistic expression in an accessible digital environment.

Bibliography

[1] H. Moss, J. Lynch, and J. O'Donoghue, "Exploring the perceived health benefits of singing in a choir: an international cross-sectional mixed-methods study," *Perspect Public Health*, vol. 138, no. 3, pp. 160–168, May 2018, doi: 10.1177/1757913917739652.

[2] A. Lewis *et al.*, "Singing for Lung Health—a systematic review of the literature and consensus statement," *npj Prim Care Resp Med*, vol. 26, no. 1, pp. 1–8, Dec. 2016, doi: 10.1038/npjpcrm.2016.80.

[3] S. Lee, D. O'Neill, and H. Moss, "Promoting well-being among people with early-stage dementia and their family carers through community-based group singing: a phenomenological study," *Arts Health*, vol. 14, no. 1, pp. 85–101, Feb. 2022, doi: 10.1080/17533015.2020.1839776.

[4] N. Hendry, D. S. Lynam, and C. Lafarge, "Singing for Wellbeing: Formulating a Model for Community Group Singing Interventions," *Qualitative Health Research*, vol. 32, no. 8–9, p. 1399, May 2022, doi: 10.1177/10497323221104718.

[5] E. Helitzer and H. Moss, "Group singing for health and wellbeing in the Republic of Ireland: the first national map," *Perspect Public Health*, vol. 142, no. 2, pp. 102–116, Mar. 2022, doi: 10.1177/17579139221081400.

[6] M. Garzón García, J. Muñoz López, and E. Y Mendoza Lara, "Voice Habits and Behaviors: Voice Care Among Flamenco Singers," *Journal of Voice*, vol. 31, no. 2, p. 246.e11-246.e19, Mar. 2017, doi: 10.1016/j.jvoice.2016.08.007.

[7] S. S. Santos, T. Montagner, G. R. Bastilha, L. F. Frigo, and C. A. Cielo, "Singing Style, Vocal Habits, and General Health of Professional Singers," *Int. Arch. Otorhinolaryngol.*, vol. 23, pp. 445–450, Nov. 2019, doi: 10.1055/s-0039-1693140.

[8] E. Craigo and E. Dolan, "Neural Network-Aided Audio Processing for Automated Vocal Coaching."

[9] R. Ware, "The Use of Science and Imagery in the Voice Studio: A Survey of Voice Teachers in the United States and Canada," *Journal of Singing - The Official Journal of the National Association of Teachers of Singing*, vol. 69, no. 4, pp. 413–417, Mar. 2013.

[10] S. Doganyigit and N. Yigit, "The Perspective of Trainers on the Use of Imagination in Singing Training," *Journal of Pedagogical Research*, vol. 7, no. 1, pp. 174–186, 2023.

[11] J. B. Hausknecht, K. M. Murdaugh, E. Nagl, and C. T. Herbst, "Global Inventory and Similarity Rating of Singing Voice Assessment Terms Used at English Speaking Academic Institutions," *Journal of Voice*, vol. 38, no. 2, pp. 284–292, Mar. 2024, doi: 10.1016/j.jvoice.2021.08.020.

[12] G. F. Welch, "Singing and Vocal Development," in *The Oxford Handbook of Singing*, G. F. Welch, D. M. Howard, and J. Nix, Eds., Oxford University Press, 2019, p. 0. doi: 10.1093/oxfordhb/9780199660773.013.52.

[13] H. Ma, "The Historical Evolution of the Art of Singing," *International Journal of Education and Humanities*, vol. 15, no. 2, Art. no. 2, Jul. 2024, doi: 10.54097/wxbjkh09.

[14] S. Sadie and G. Grove, *The New Grove Dictionary of Music and Musicians*. London: Macmillan Publishers Limited, 1980. Accessed: Nov. 05, 2024. [Online]. Available: http://www.oxfordmusiconline.com

[15] K. Ragan, "Defining Evidence-Based Voice Pedagogy: A New Framework."

[16] N. Henrich *et al.*, "Towards a Common Terminology to Describe Voice Quality in Western Lyrical Singing: Contribution of a Multidisciplinary Research Group."

[17] M. Hoch and M. J. Sandage, "Working Toward a Common Vocabulary: Reconciling the Terminology of Teachers of Singing, Voice Scientists, and Speech-Language Pathologists," *Journal of Voice*, vol. 31, no. 6, pp. 647–648, Nov. 2017, doi: 10.1016/j.jvoice.2016.12.014.

[18] G. Welch, D. Howard, and J. Nix, *The Oxford Handbook of Singing*. 2019.

[19] "MVC - MDN Web Docs Glossary: Definitions of Web-related terms | MDN." Accessed: Nov. 10, 2024. [Online]. Available: https://developer.mozilla.org/en-US/docs/Glossary/MVC

[20] "What is scrum and how to get started." Accessed: Nov. 11, 2024. [Online]. Available: https://www.atlassian.com/agile/scrum

[21] G. Luck and P. Toiviainen, "Ideal singing posture: evidence from behavioural studies and computational motion analysis," pp. 15–19, Sep. 2007.

[22] J. L. Chapman and R. Morris, *Singing and Teaching Singing: A Holistic Approach to Classical Voice, Fourth Edition*. Plural Publishing, 2021.

[23] D. Deutsch, *The Psychology of Music*. Elsevier Science, 2012.

[24] S. Mccarther, "EFFECTING POSITIVE CHANGE: A MANUAL FOR TEACHERS OF SINGING".

[25] "Vocal Image: AI Voice Coach. Speaking and Communication Lessons.," Vocal Image: AI Voice Coach. Speaking and Communication Lessons. Accessed: Nov. 23, 2024. [Online]. Available: https://www.vocalimage.app/en/

[26] "How to Sing | Singing Lessons | Learn to Sing," Yousician. Accessed: Nov. 23, 2024. [Online]. Available: https://yousician.com/singing

[27] "Sing Sharp - Singing Lessons, Vocal Warmups Trainings." Accessed: Nov. 23, 2024. [Online]. Available: https://www.singsharp.com/

[28] C. Mezzedimi, M. C. Spinosi, T. Massaro, F. Ferretti, and J. Cambi, "Singing voice: acoustic parameters after vocal warm-up and cool-down," *Logopedics Phoniatrics Vocology*, vol. 45, no. 2, pp. 57–65, Apr. 2020, doi: 10.1080/14015439.2018.1545865.

[29] "Warming up to sing," BBC Bitesize. Accessed: Nov. 24, 2024. [Online]. Available: https://www.bbc.co.uk/bitesize/articles/zfrb7nb

[30] P. S. Phillips, *Singing Exercises for Dummies*. John Wiley & Sons, 2012.

[31] C. B. Fry, “Exercises for the Development of Choral Tone,” *Bulletin of the Music Teachers National Association*, vol. 14, no. 3, pp. 23–25, 1949.

[32] “The daily implementation of sequential sustained tone exercises as a means of improving the ensemble intonation and tone quality of second-year middle school bands - ProQuest.” Accessed: Nov. 25, 2024. [Online]. Available: https://www.proquest.com/docview/304500580?pq-origsite=gscholar&fromopenview=true&sourcetype=Dissertations%20&%20Theses

[33] “Humming exercises,” Cambridge University Hospitals. Accessed: Nov. 25, 2024. [Online]. Available: https://www.cuh.nhs.uk/patient-information/humming-exercises/

[34] J. Henny, “Resonance Voice Exercises to Improve Tone Quality,” John Henny Vocal Studio. Accessed: Nov. 25, 2024. [Online]. Available: https://johnhenny.com/episode-215-finding-singing-resonance/

[35] J. Green, “3 Singing Exercises to Improve Your Vocal Pitching,” Musical U. Accessed: Nov. 25, 2024. [Online]. Available: https://www.musical-u.com/learn/3-singing-exercises-to-improve-your-vocal-pitching/

[36] True, “Singing Scales: The Ultimate Guide for Vocal Mastery,” KenTamplinVocalAcademy.com. Accessed: Nov. 25, 2024. [Online]. Available: https://kentamplinvocalacademy.com/how-to-sing/techniques/scales-for-singing/

[37] K. Heidemann, “A System for Describing Vocal Timbre in Popular Song,” *Music Theory Online*, vol. 22, no. 1, Mar. 2016, Accessed: Nov. 25, 2024. [Online]. Available: https://www.mtosmt.org/issues/mto.16.22.1/mto.16.22.1.heidemann.html

[38] K. H. Phillips, “The Effects of Group Breath-Control Training on the Singing Ability of Elementary Students,” *Journal of Research in Music Education*, vol. 33, no. 3, pp. 179–191, Oct. 1985, doi: 10.2307/3344805.

[39] “Diaphragmatic Breathing Exercises & Benefits,” Cleveland Clinic. Accessed: Nov. 25, 2024. [Online]. Available: https://my.clevelandclinic.org/health/articles/9445-diaphragmatic-breathing

[40] “Breathing Exercises for Singing - Diaphragm, Hissing, etc.” Accessed: Nov. 25, 2024. [Online]. Available: https://themusicambition.com/breathing-exercises-for-singing/

[41] “Vocal Technique for Singers: Breathing and Breath Control 101 « Emily Peasgood.” Accessed: Nov. 25, 2024. [Online]. Available: https://emilypeasgood.com/vocal-technique-for-singers-breathing-and-breath-control/

[42] A. Radford, J. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, “Robust Speech Recognition via Large-Scale Weak Supervision,” Dec. 2022. Available: https://cdn.openai.com/papers/whisper.pdf

[43] “librosa — librosa 0.10.1 documentation,” librosa.org. https://librosa.org/doc/latest/index.html#

[44] “Play Flappy Bird,” *flappybird.io*. https://flappybird.io/

[45] peterkhayes, “GitHub - peterkhayes/pitchfinder: A compilation of pitch detection algorithms for Javascript.,” *GitHub*, 2025. https://github.com/peterkhayes/pitchfinder (accessed Apr. 11, 2025).

[46] danigb, “GitHub - danigb/soundfont-player: Quick soundfont loader and player for browser,” *GitHub*, 2023. https://github.com/danigb/soundfont-player (accessed Apr. 11, 2025).

Appendice A – UI Design Plan

3.6.1 Sign Up and Login Designs

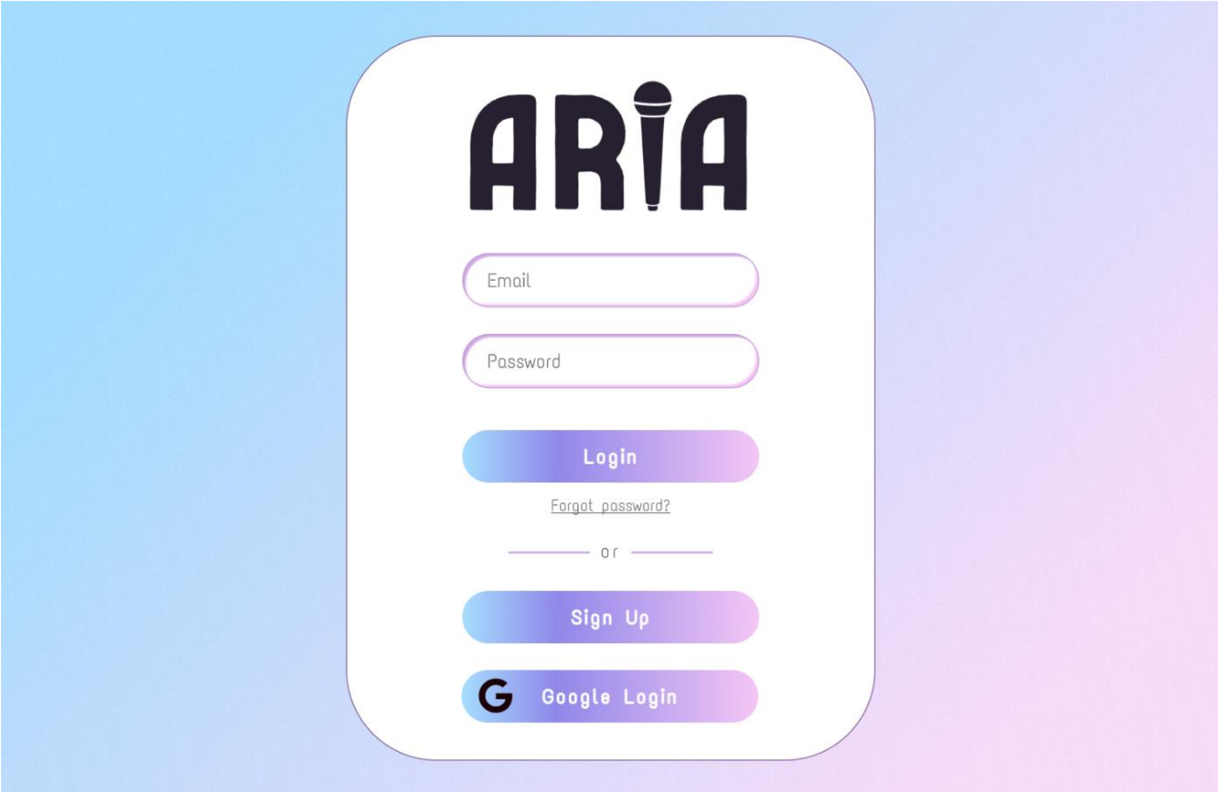


Figure 20 ARIA SING IN PAGE DESIGN

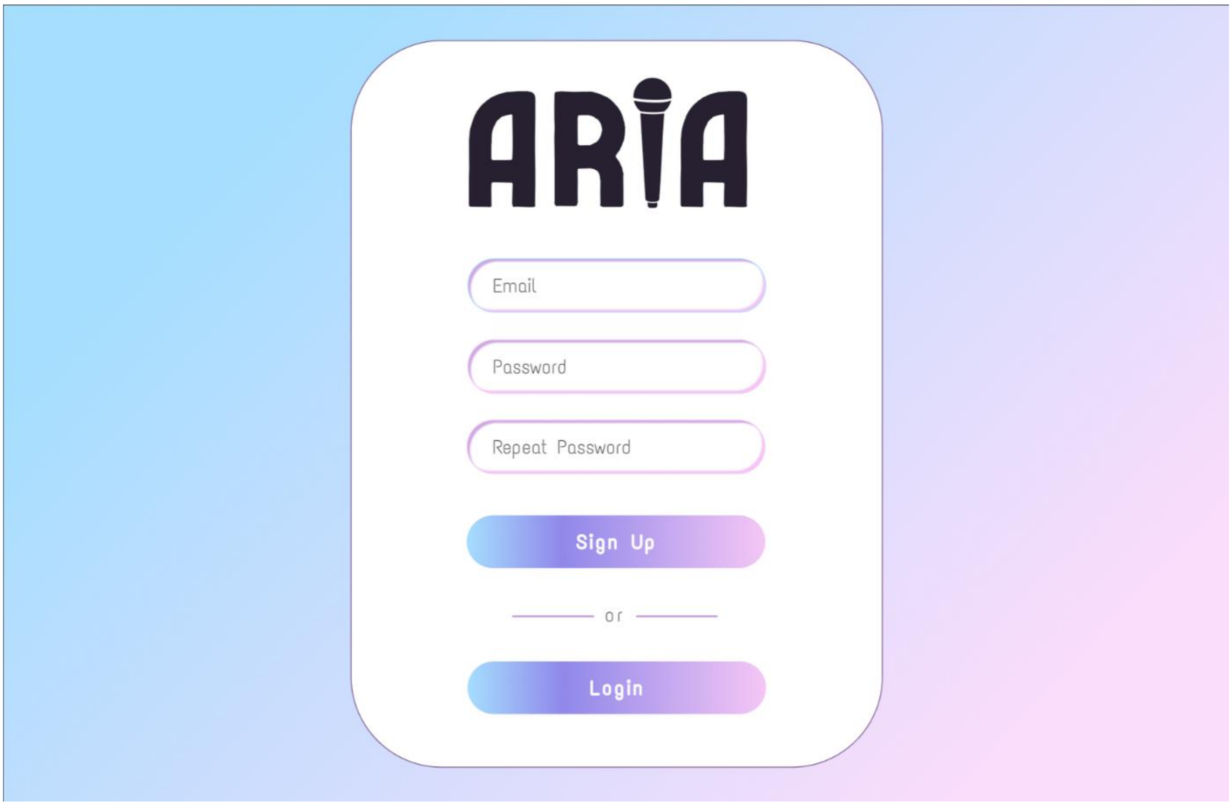


Figure 21 Aria Sing Up Page Design

3.6.2 Vocal Analysis Test Design

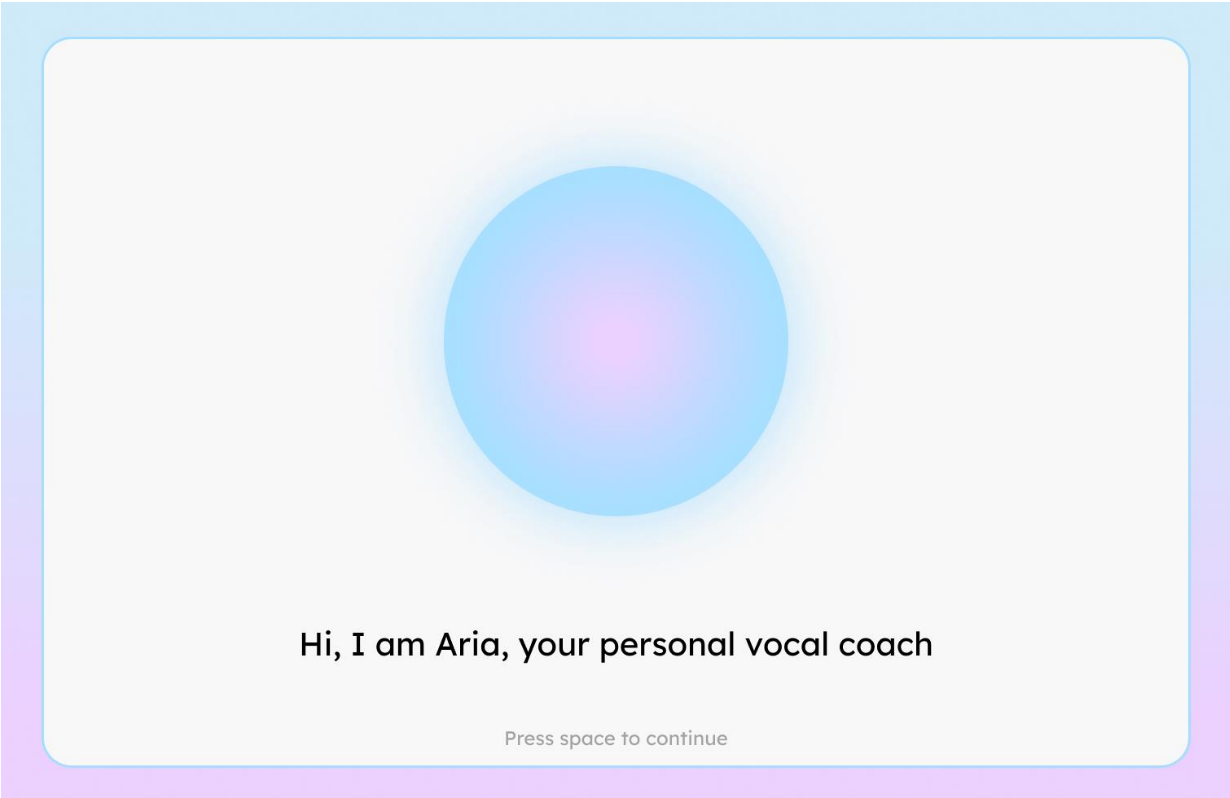


Figure 22 ARIA VOCAL ANALYSIS TEST START DESIGN

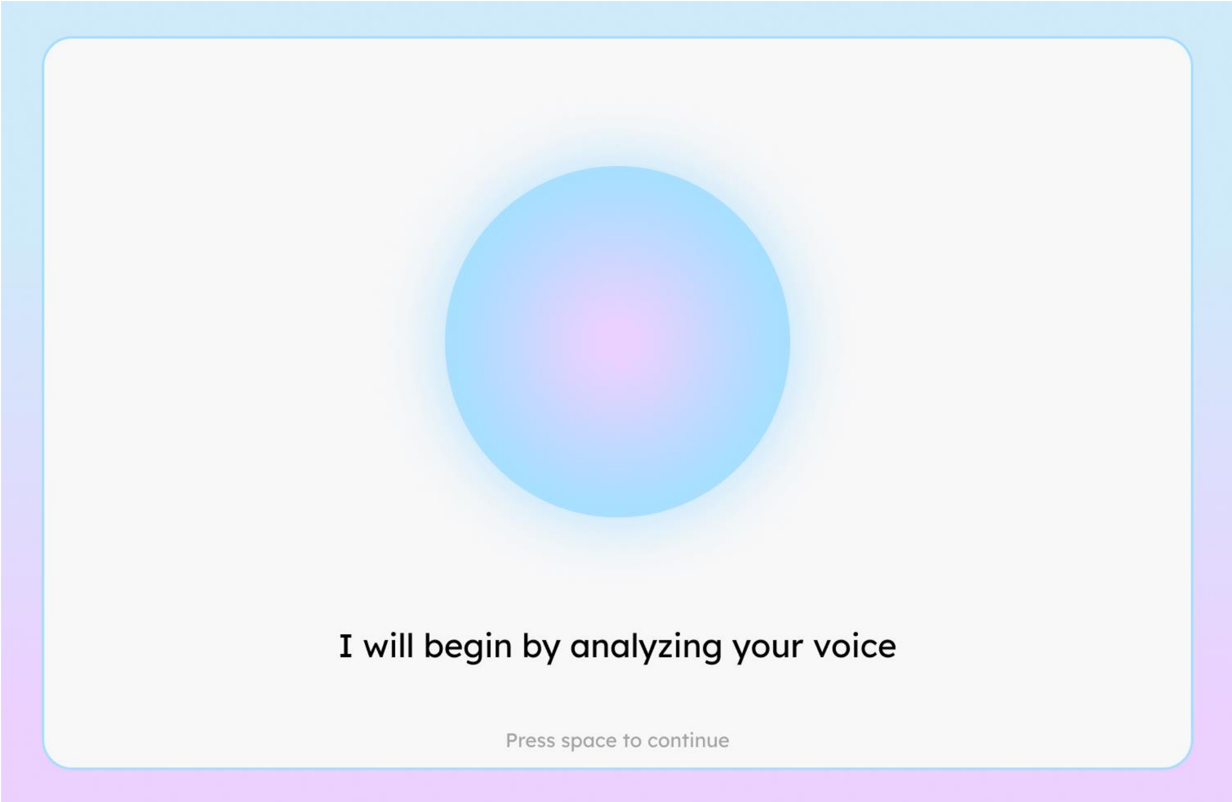


Figure 23 ARIA VOCAL ANALYSIS TEST START DESIGN

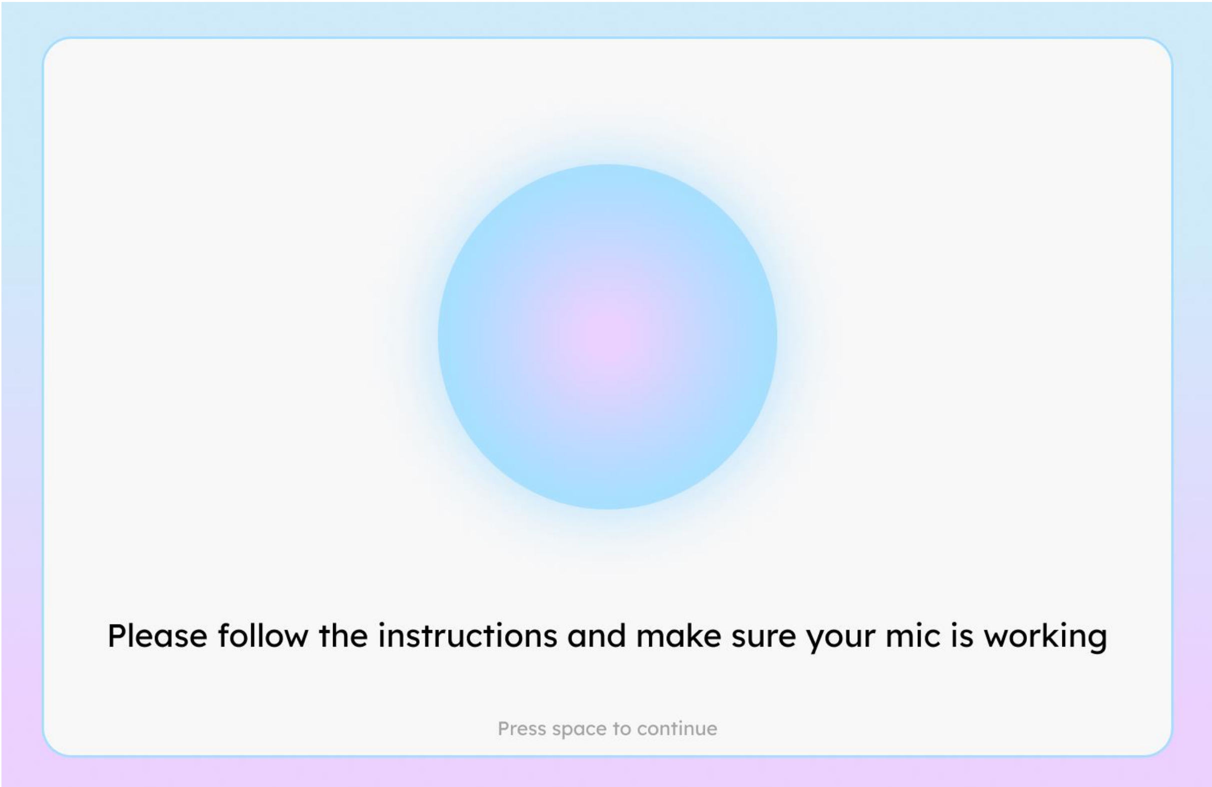


Figure 24 ARIA VOCAL ANALYSIS TEST START DESIGN

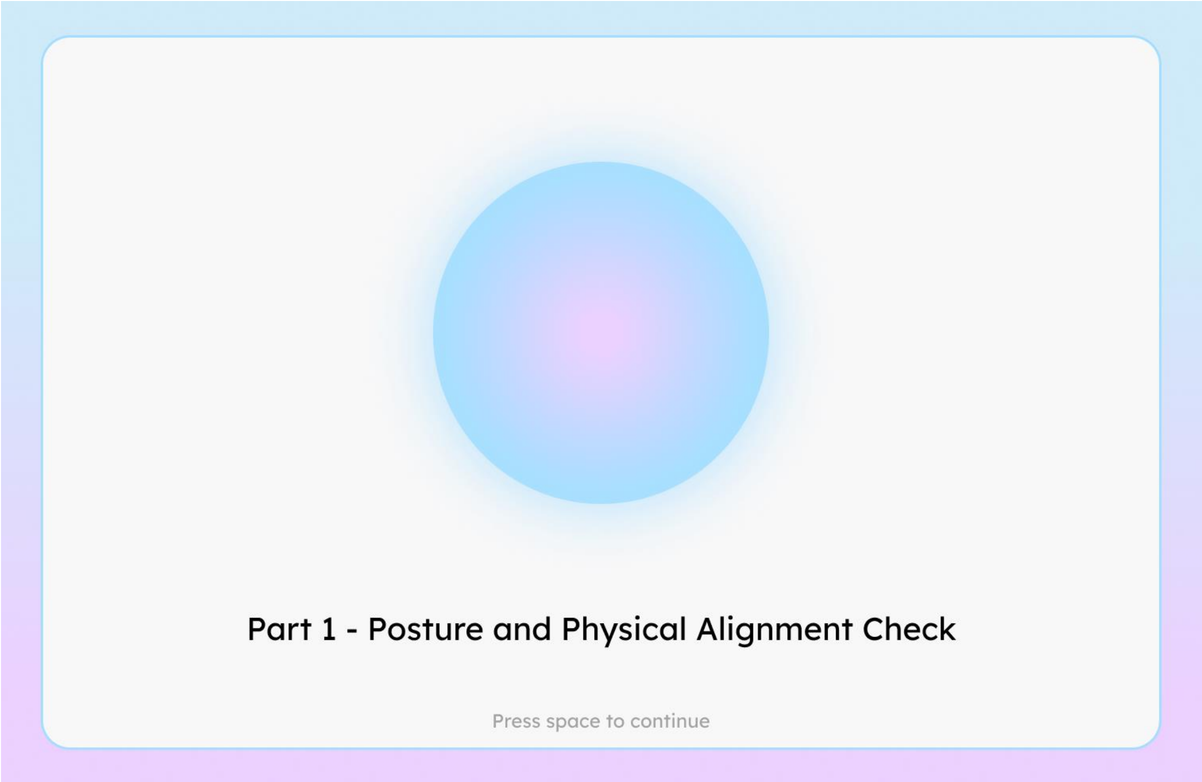


Figure 25 ARIA VOCAL ANALYSIS TEST POSTURE ALIGNMENT START DESIGN

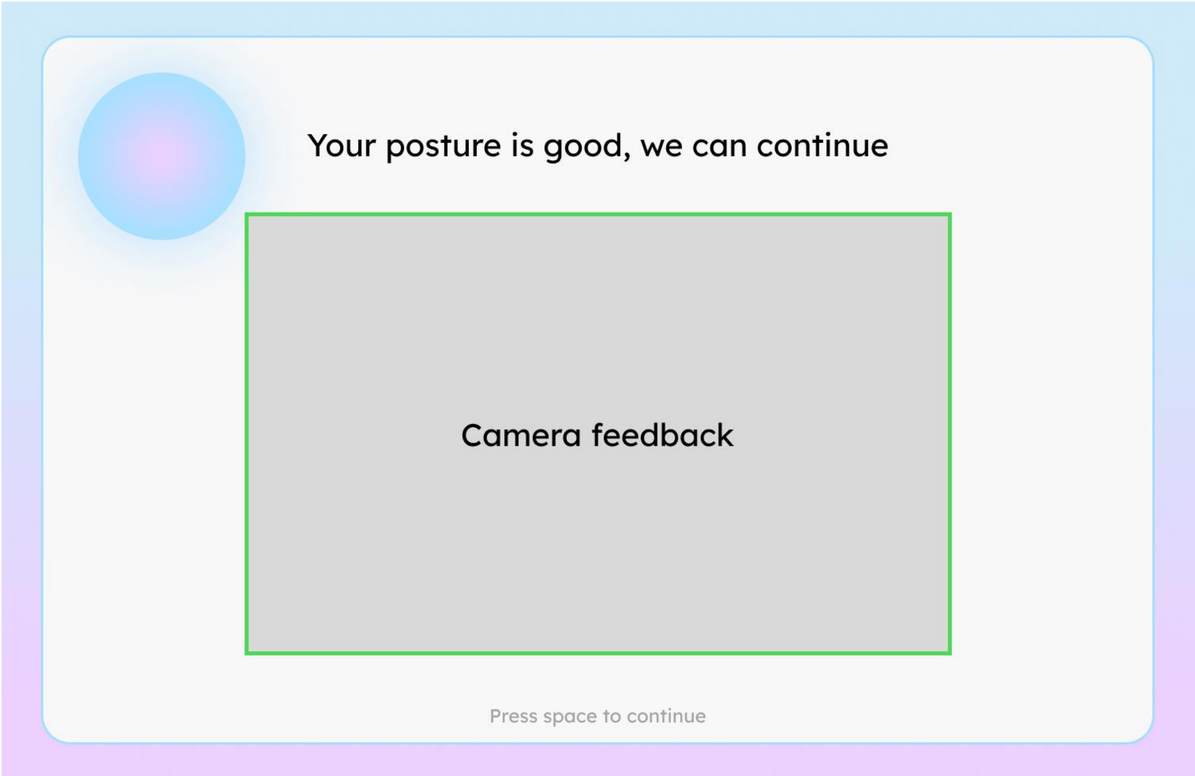


Figure 26 ARIA VOCAL ANALYSIS TEST POSTURE ALIGNMENT DESIGN

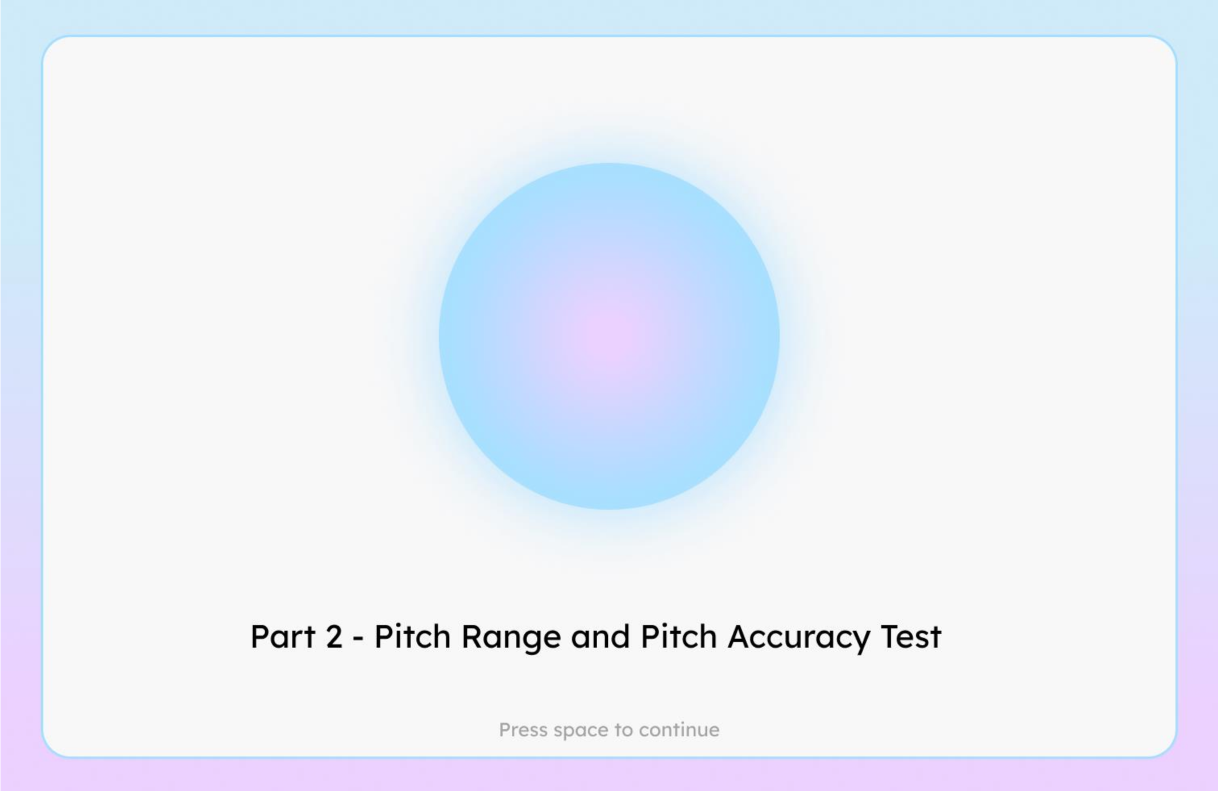


Figure 27 ARIA VOCAL ANALYSIS TEST PITCH RANGE START DESIGN

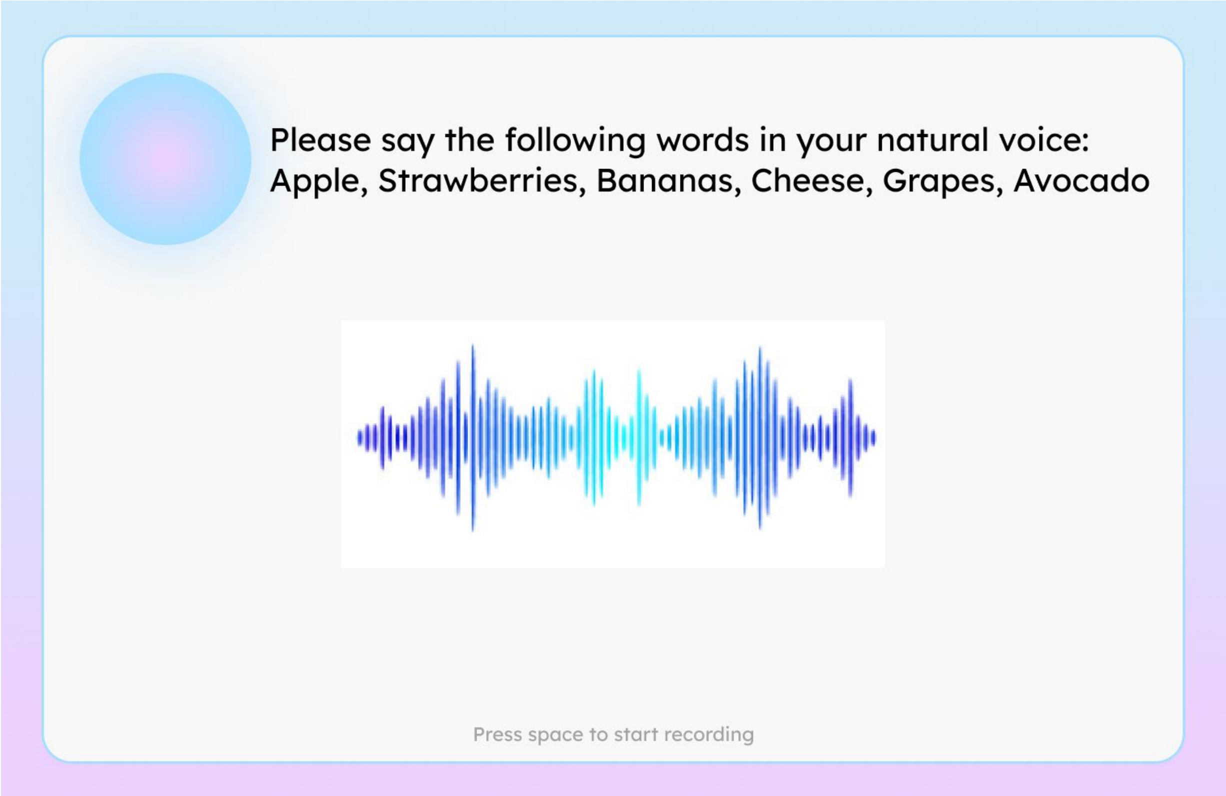


Figure 28 ARIA VOCAL ANALYSIS TEST NATURAL PITCH DESIGN

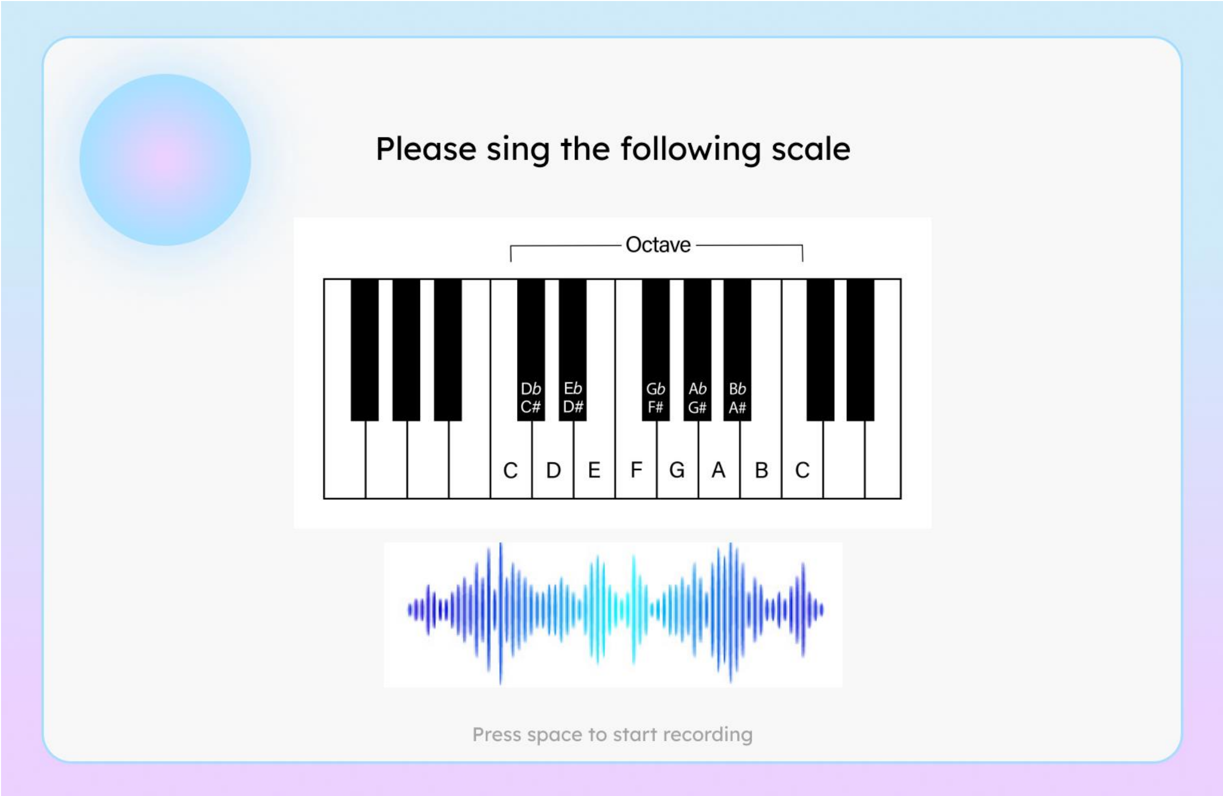


Figure 29 ARIA VOCAL ANALYSIS TEST SING A SCALE DESIGN

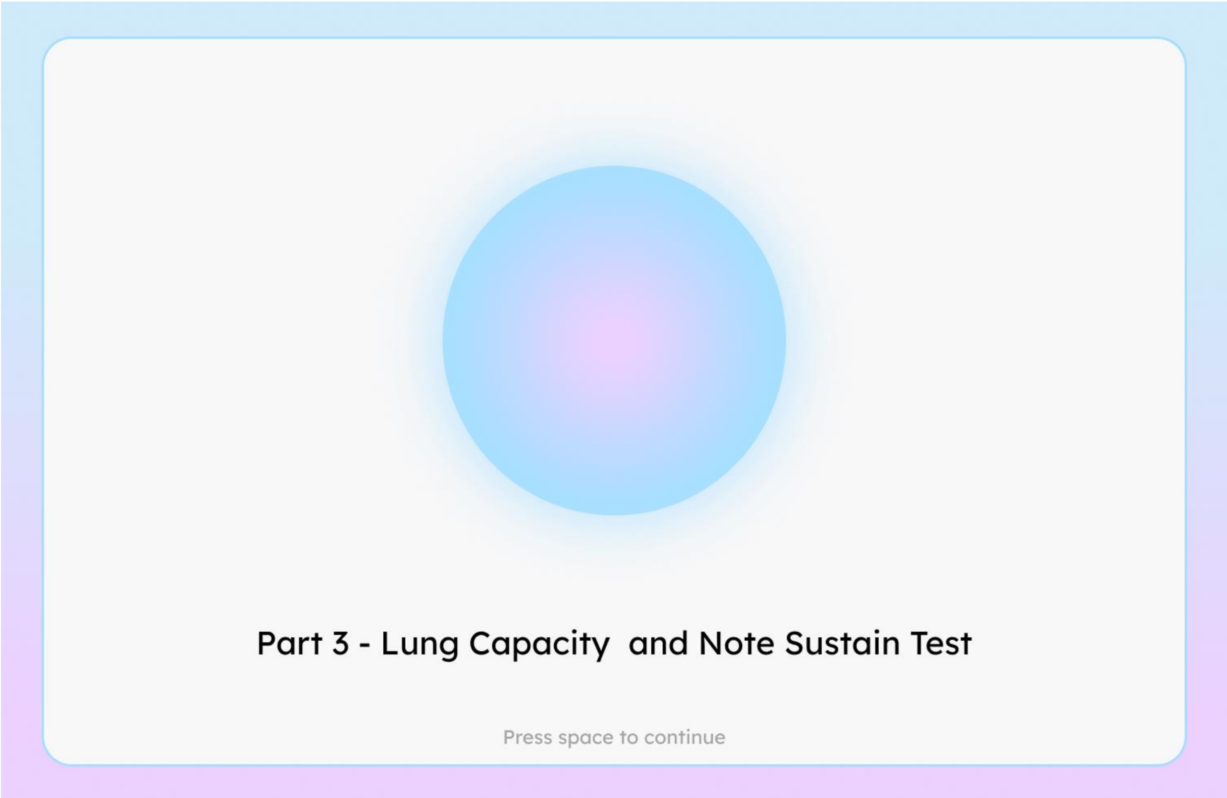


Figure 30 ARIA VOCAL ANALYSIS TEST LUNG CAPACITY START DESIGN

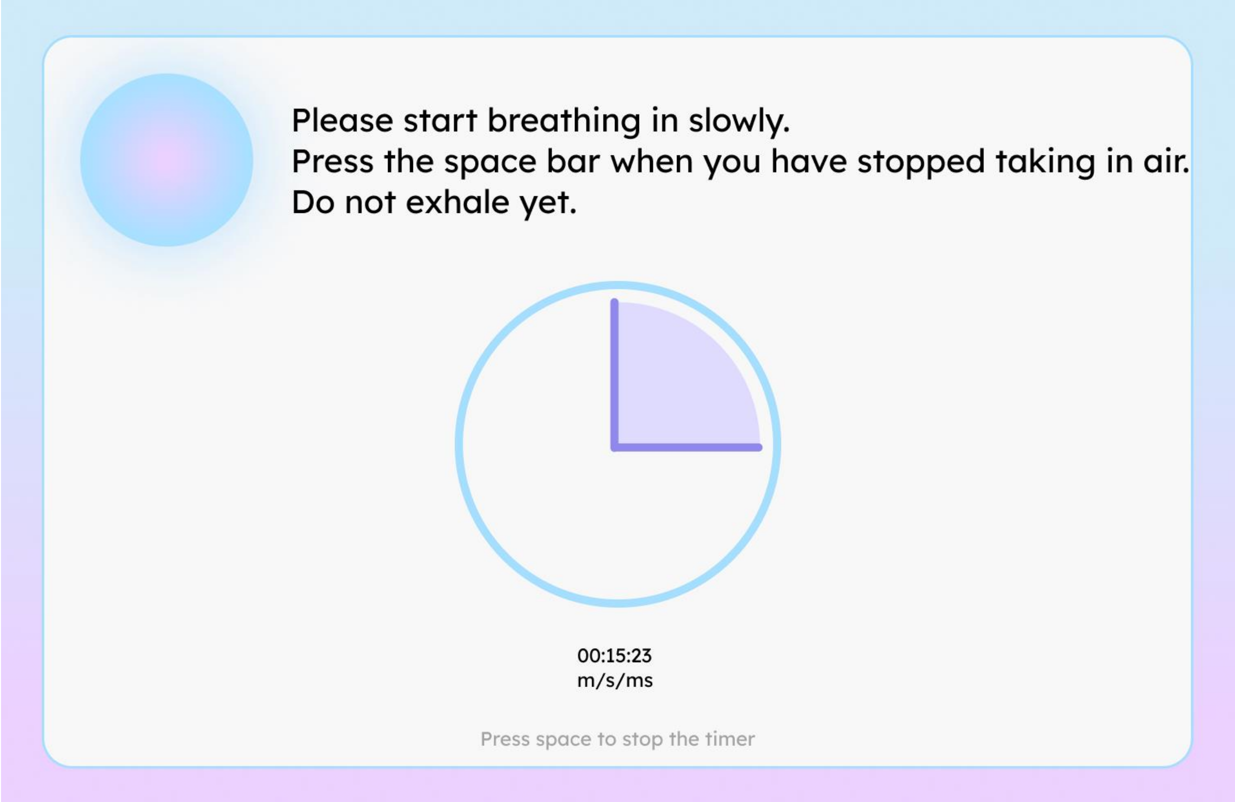


Figure 31 ARIA VOCAL ANALYSIS TEST LUNG CAPACITY TIMER DESIGN

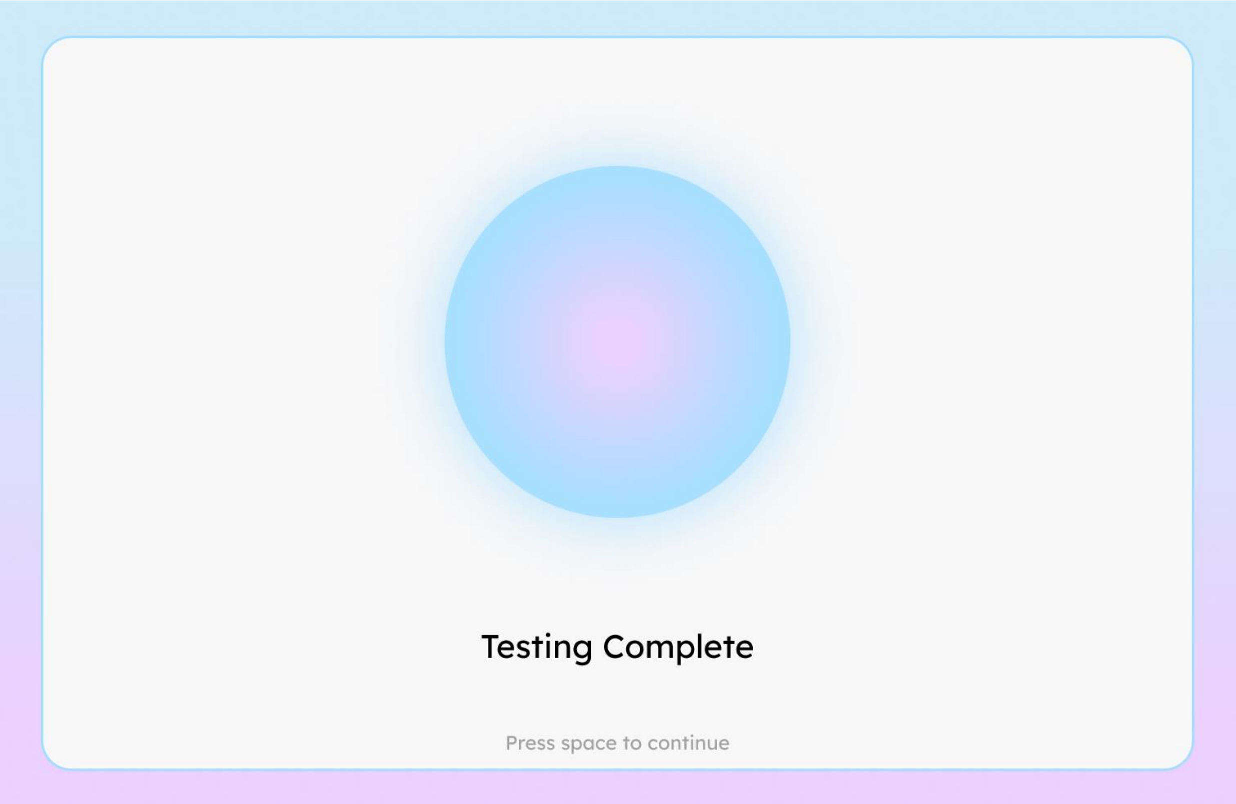


Figure 32 ARIA VOCAL ANALYSIS TEST COMPLETE DESIGN

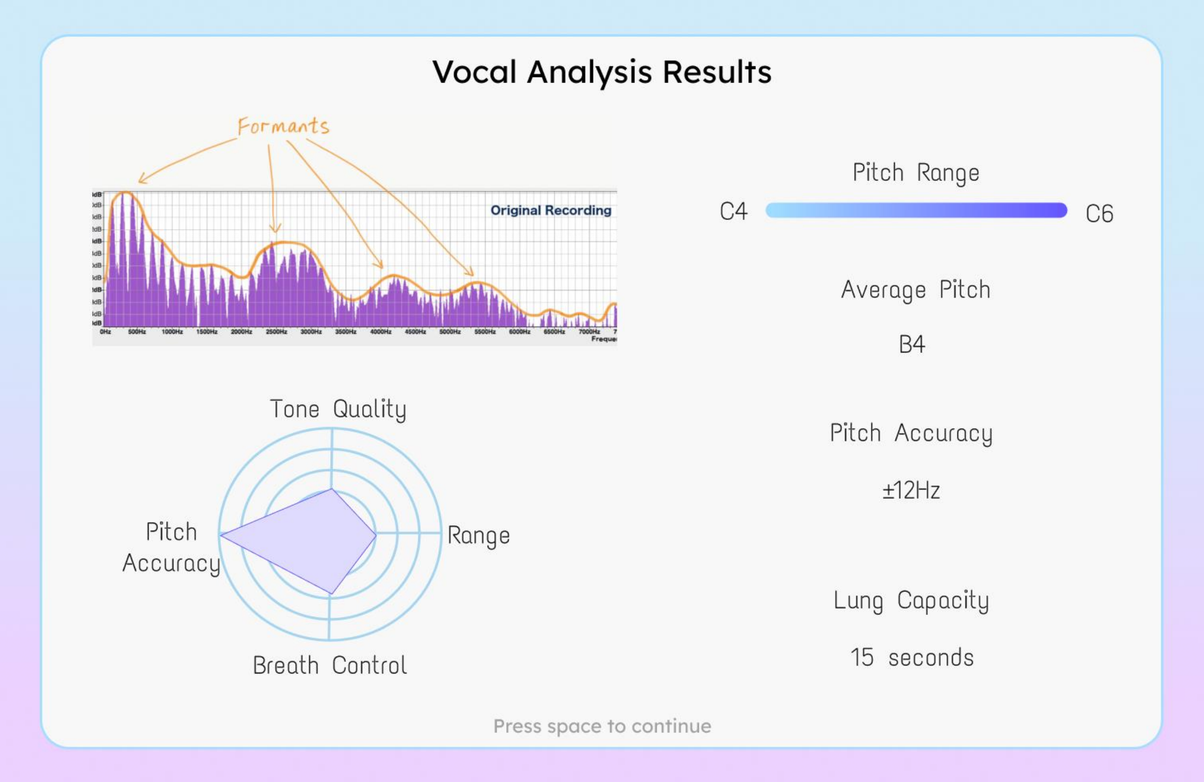


Figure 33 ARIA VOCAL ANALYSIS TEST RESULTS DESIGN

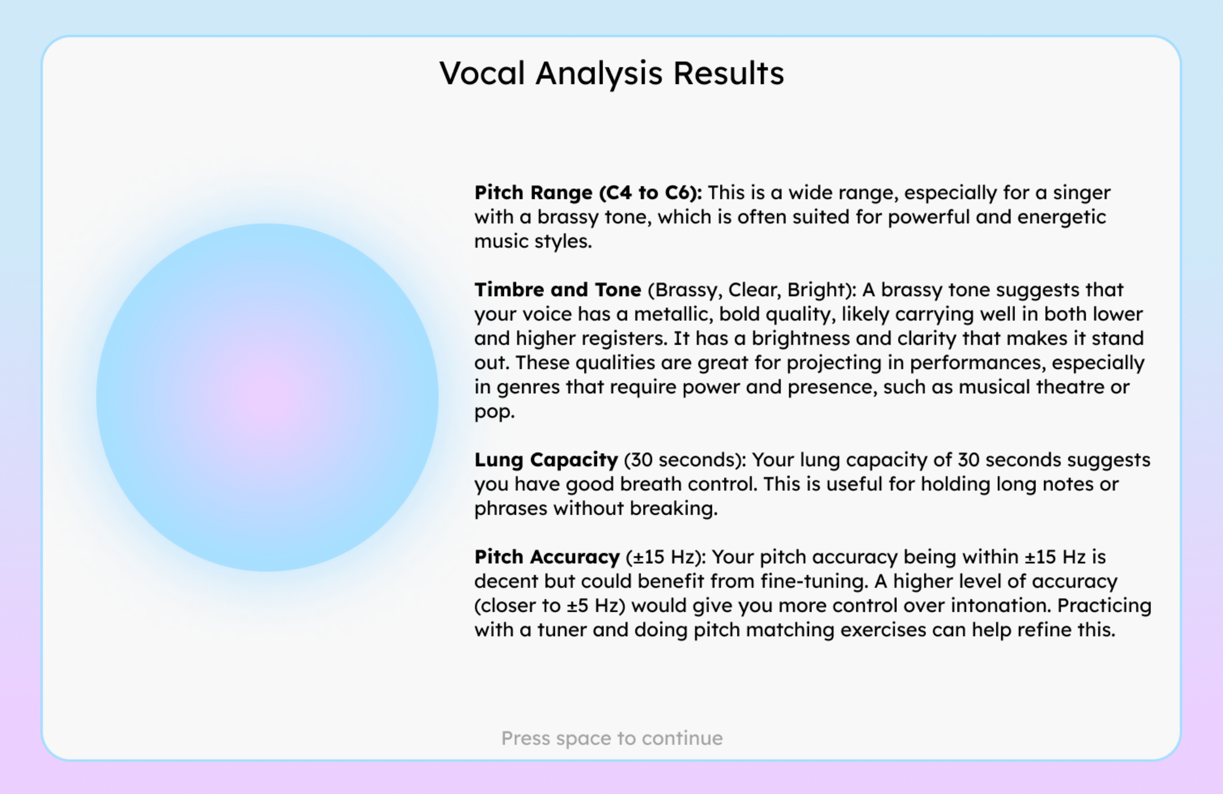


Figure 34 ARIA VOCAL ANALYSIS TEST FEEDBACK DESIGN

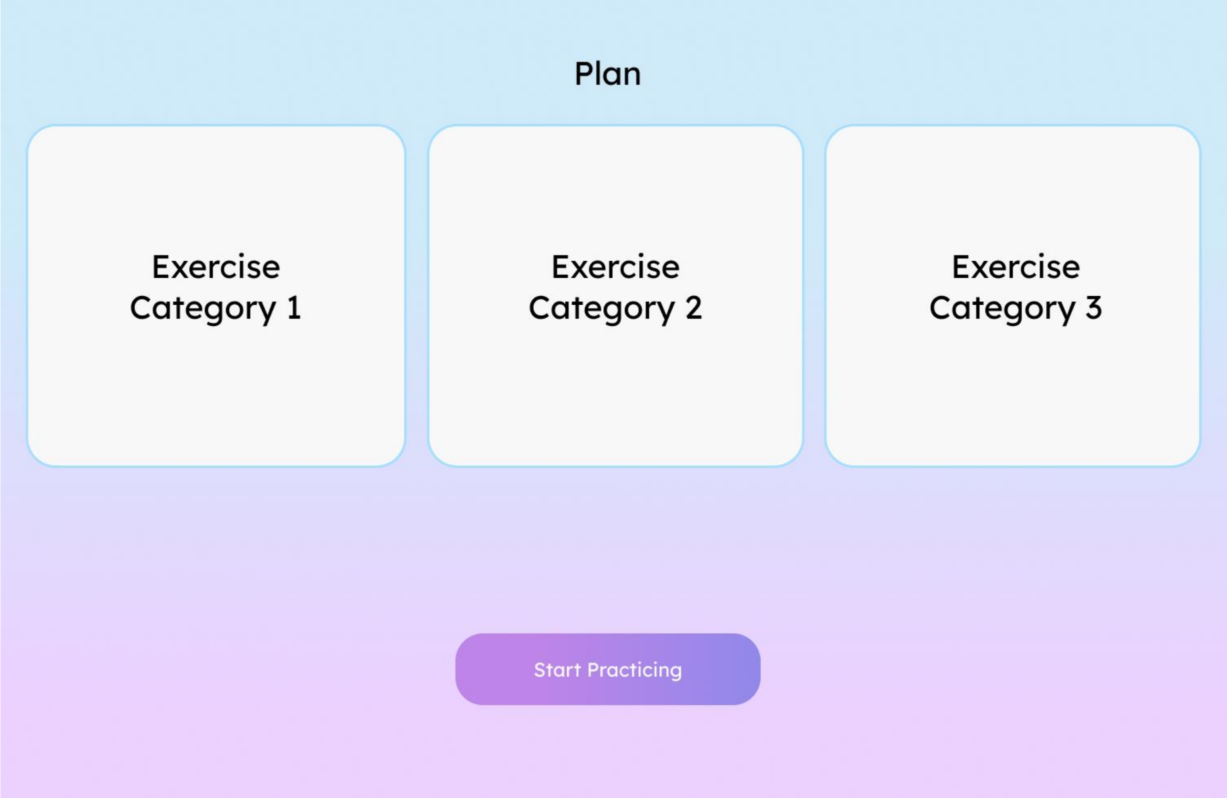


Figure 35 ARIA VOCAL ANALYSIS TEST RECOMMENDED PLAN DESIGN

3.6.3 Home Page Design

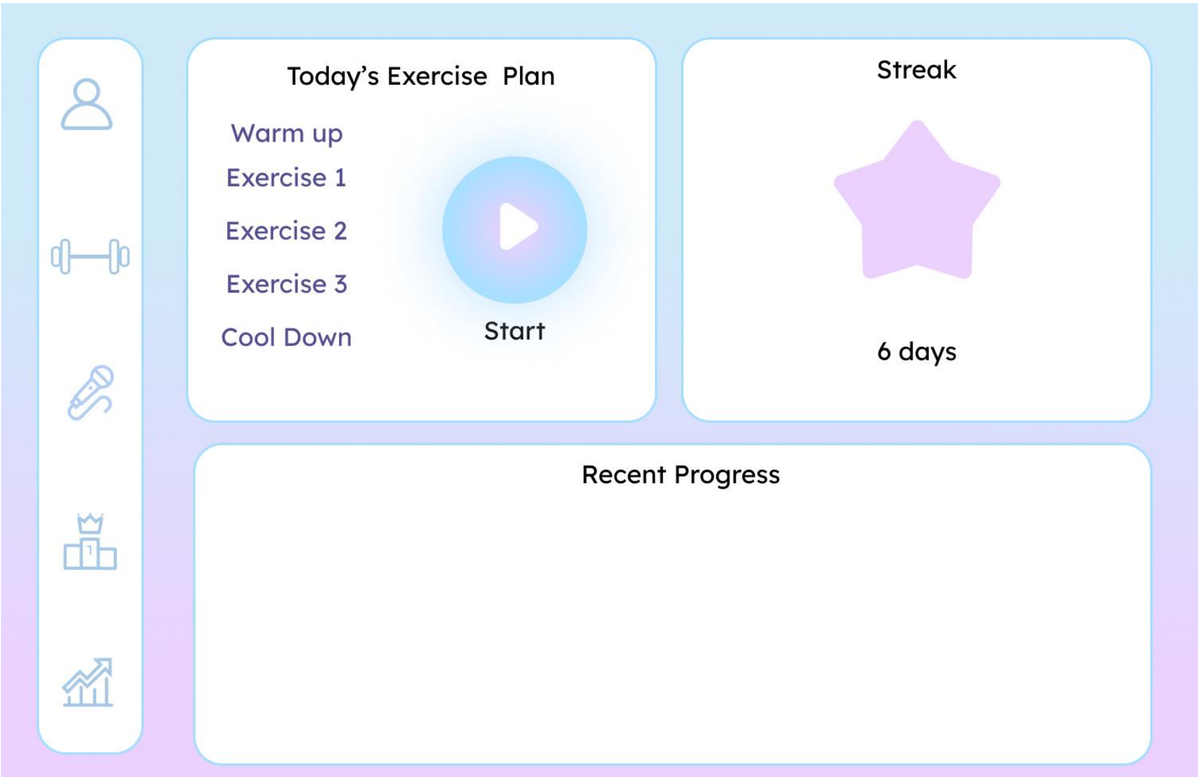


Figure 36 : ARIA HOME PAGE DESIGN

3.6.4 Singing Practice Design

Figure 37 ARIA SINGING SECTION DESIGN

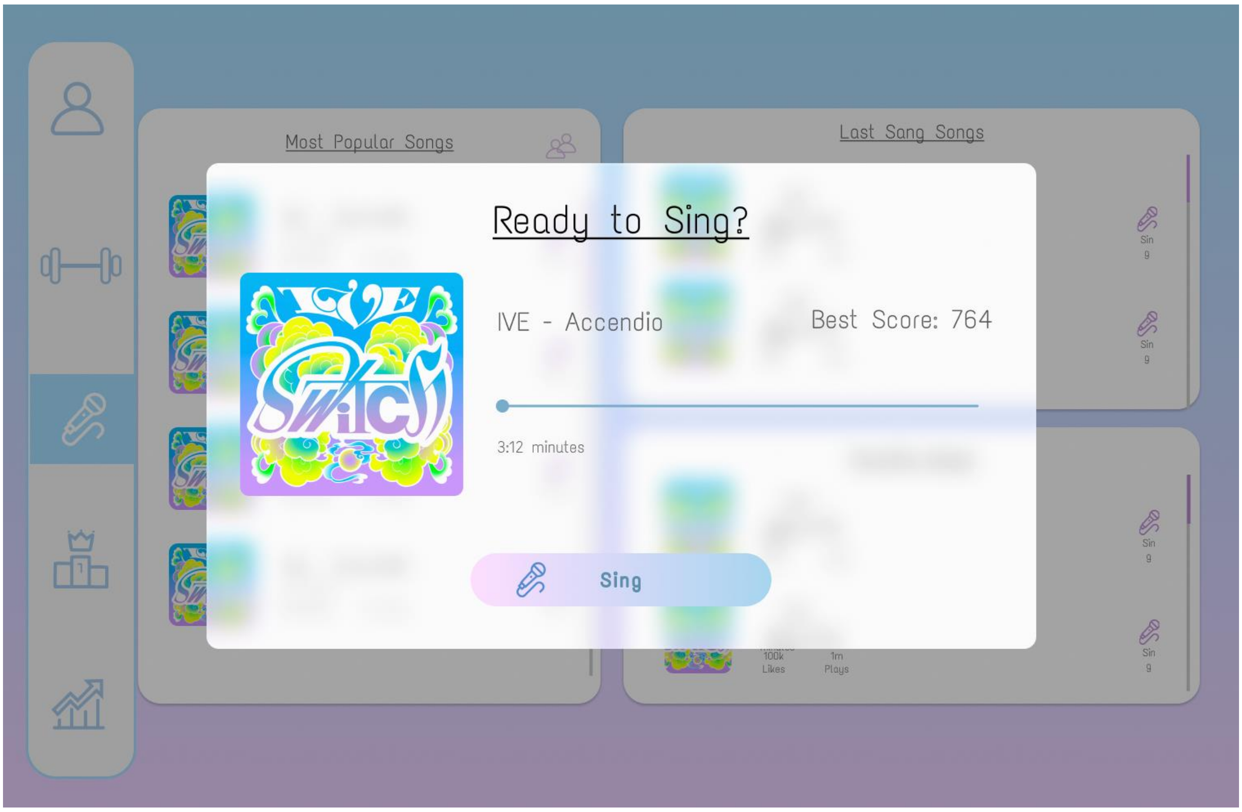


Figure 38 ARIA SONG SELECT DESIGN

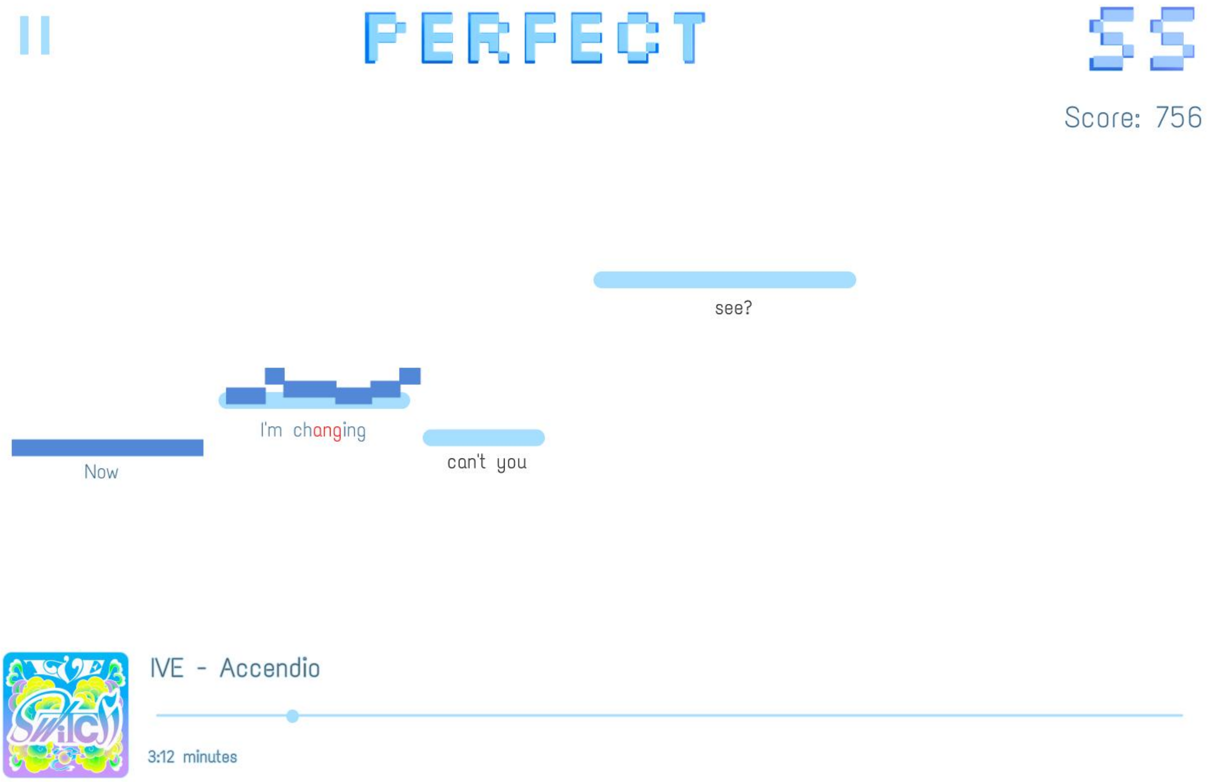


Figure 39 ARIA LIVE SINGING SCREEN DESIGN

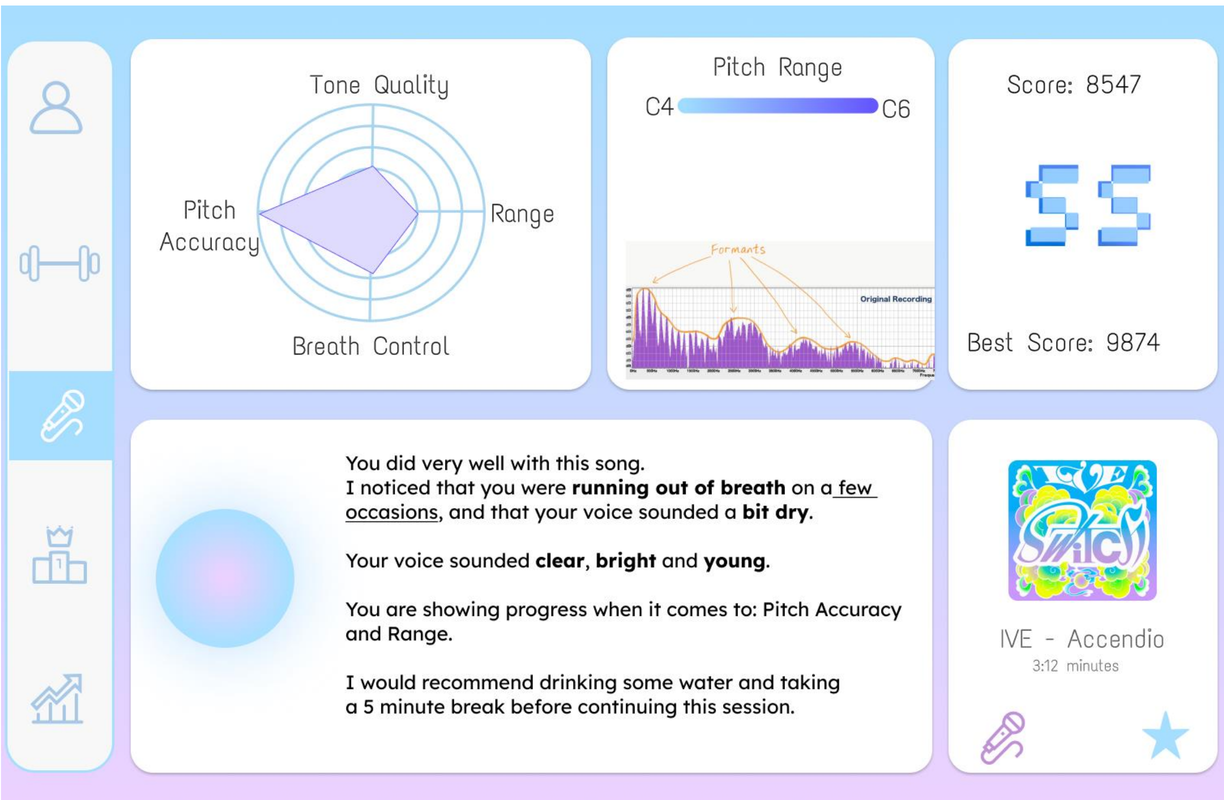


Figure 40 ARIA LIVE SINGING SCREEN RESULTS DESIGN

Appendice B – Final UI

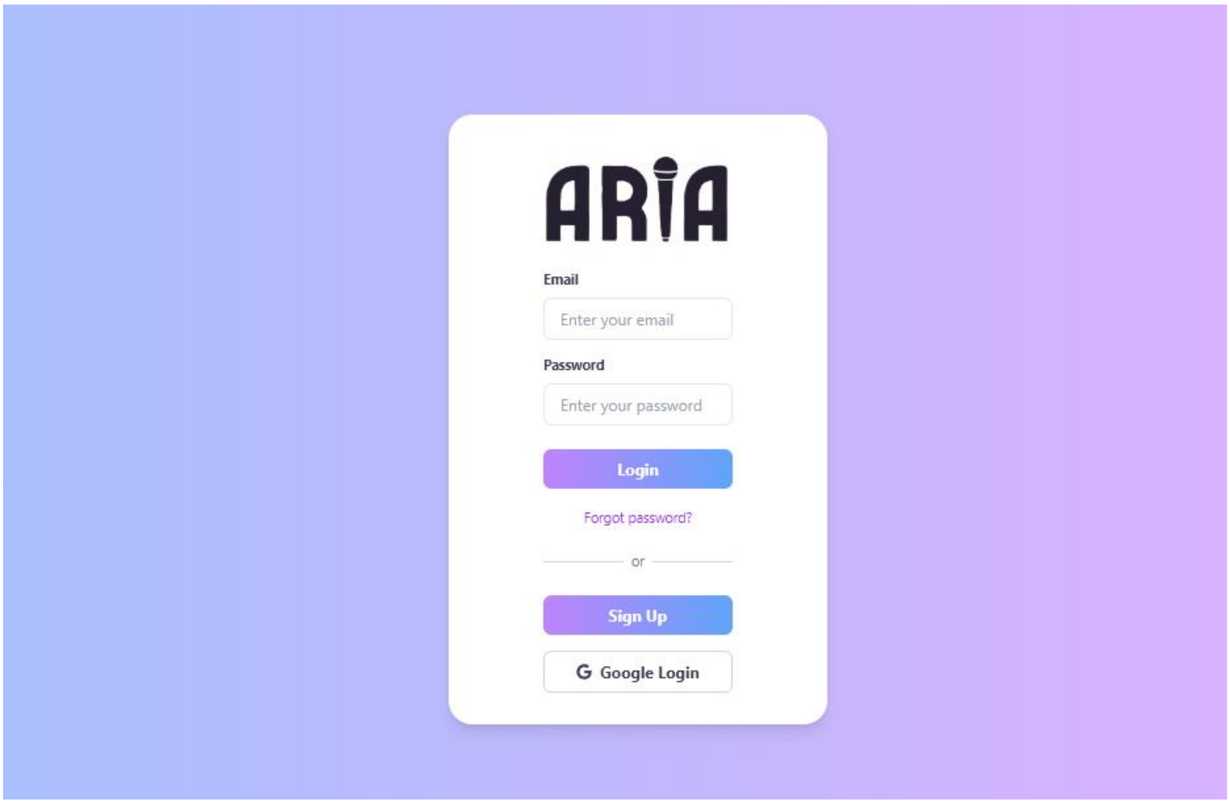


Figure 41 Login Page

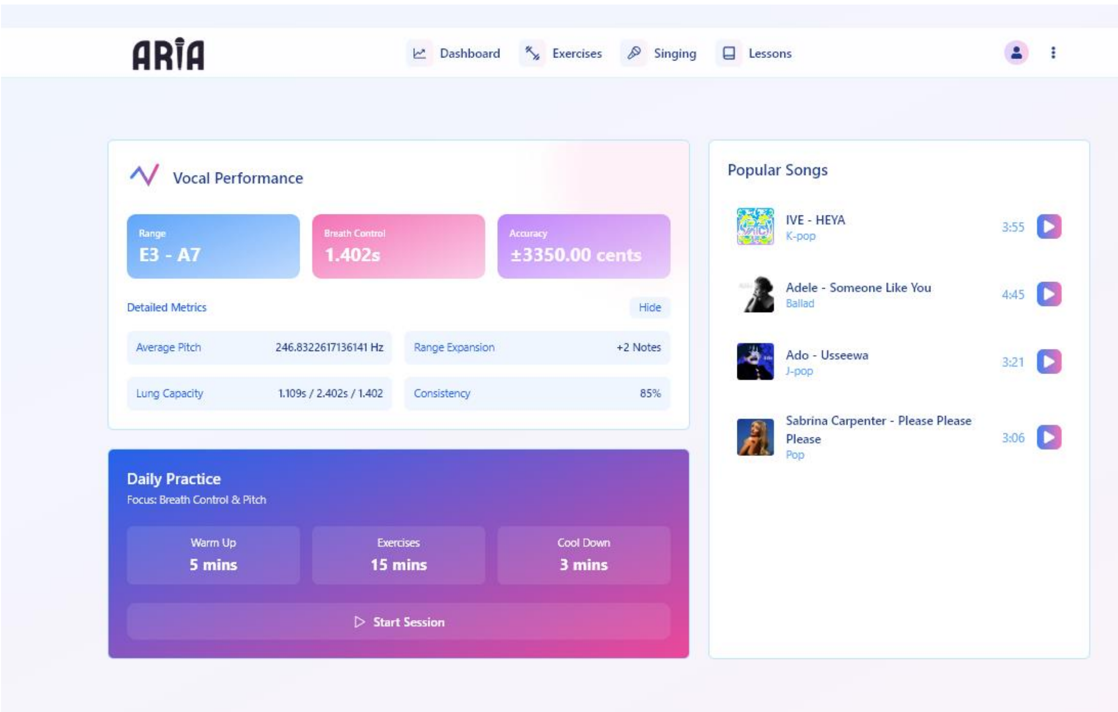


Figure 42 Dashboard Page

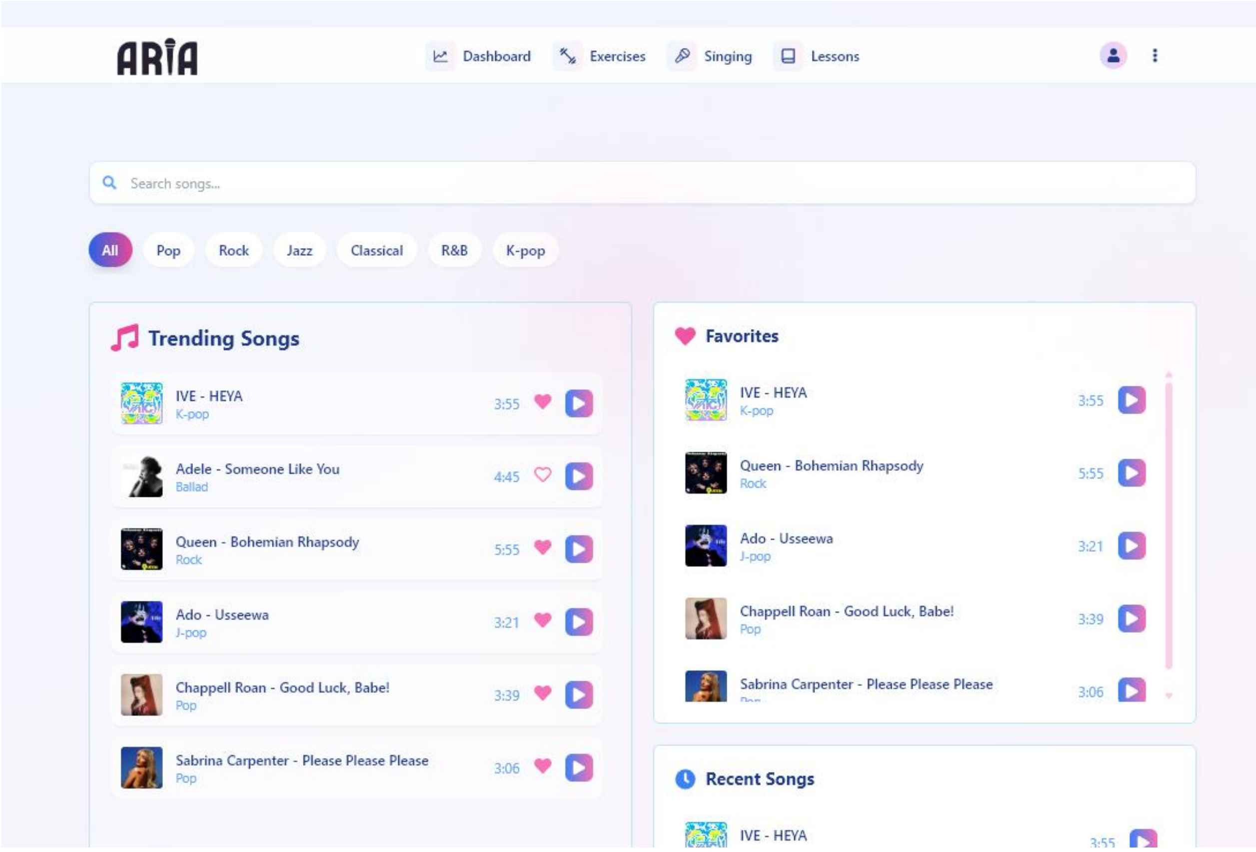


Figure 43 Singing Page

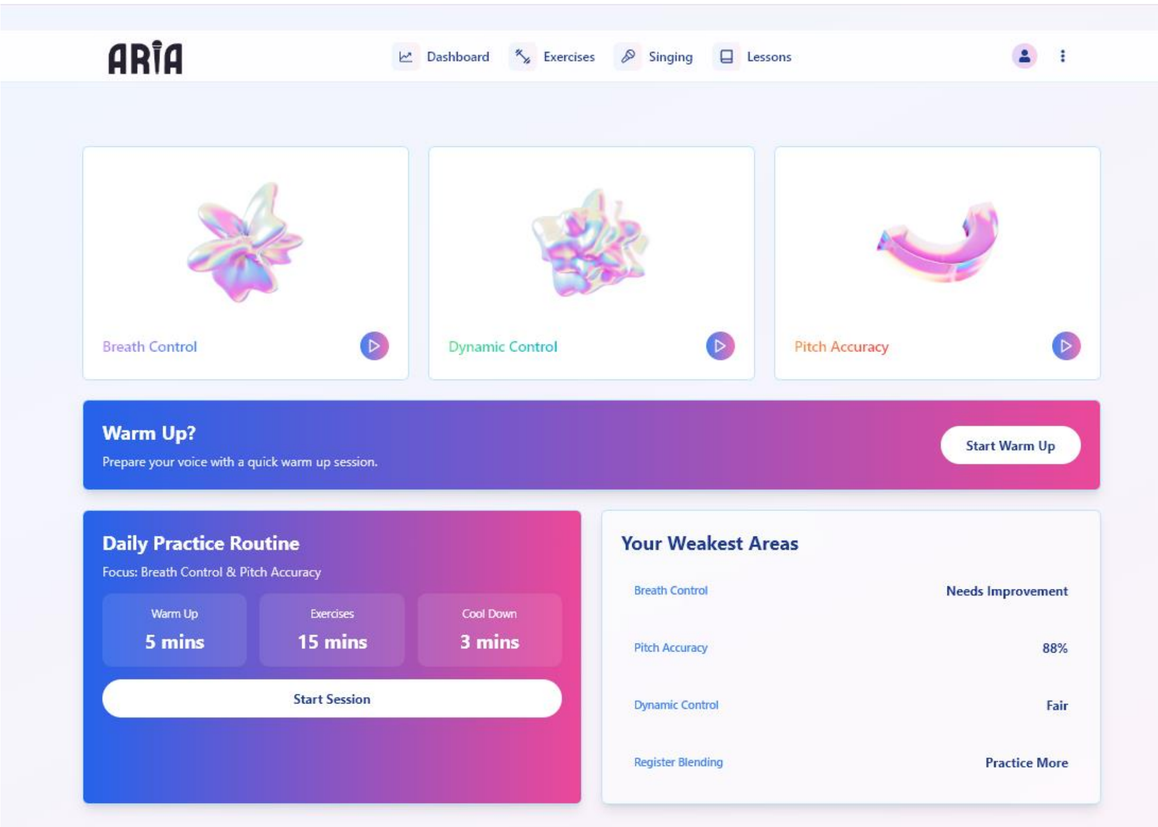


Figure 44 Exercises Page

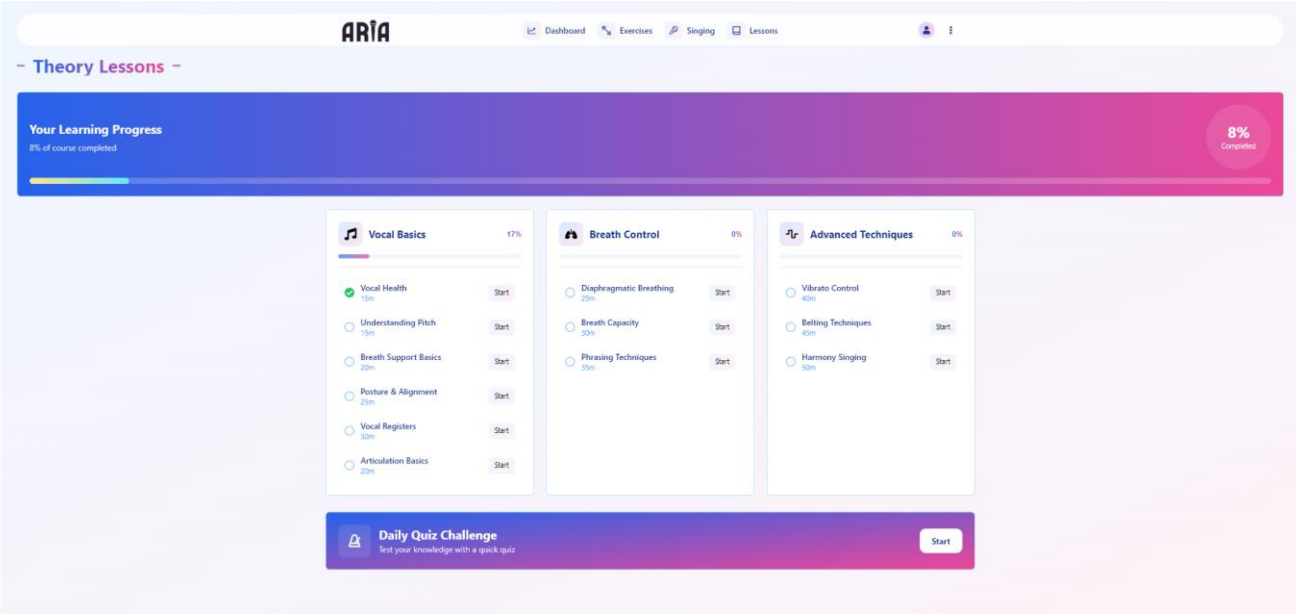


Figure 45 Theory Lessons Page

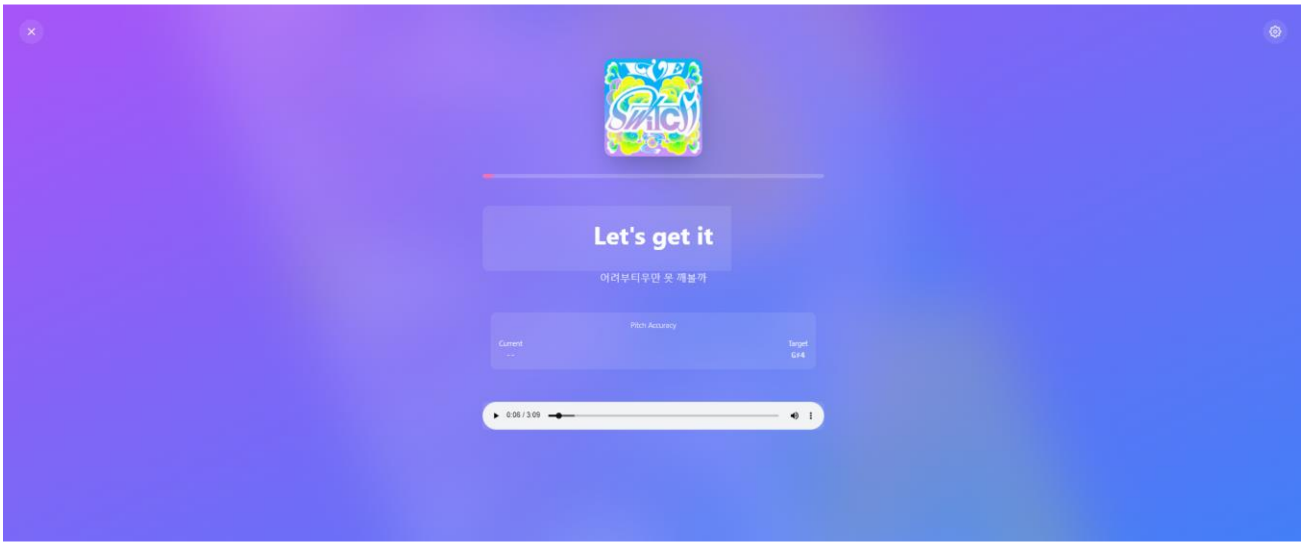


Figure 46 Singing a song Page

Figure 47 Pitch Accuracy Exercise

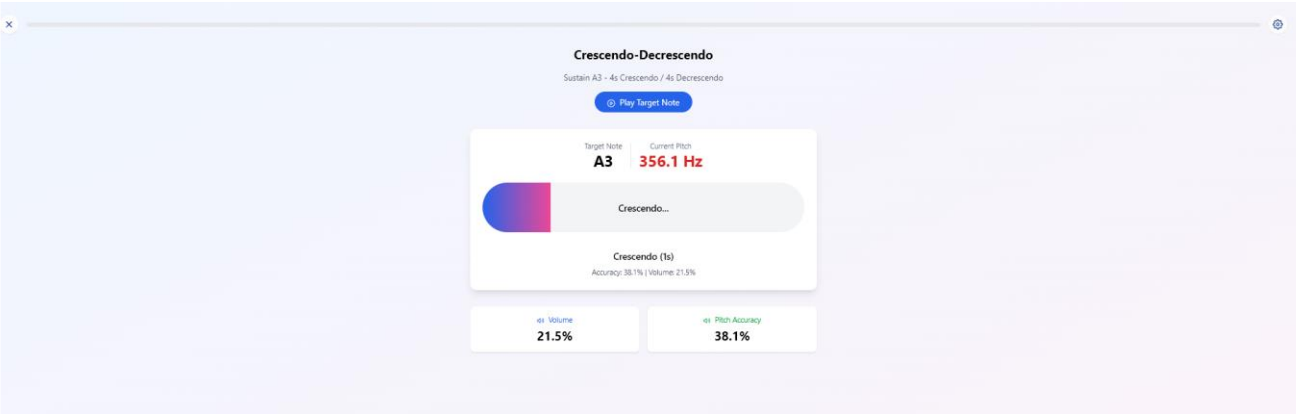


Figure 48 Dynamic Control Exercise

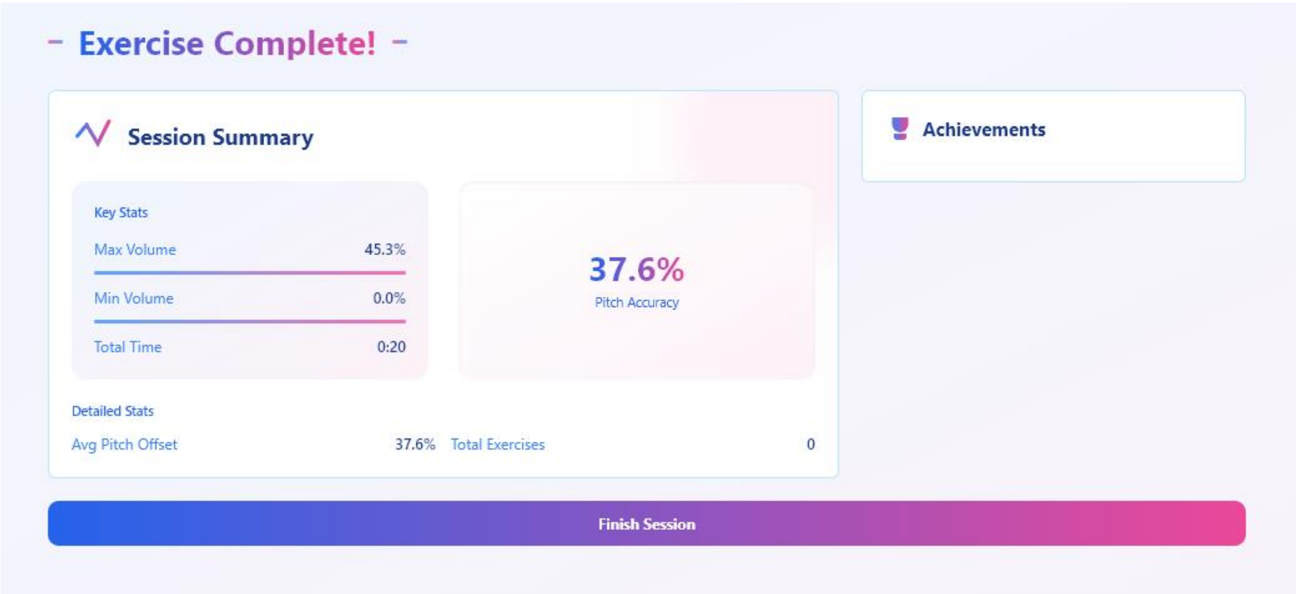


Figure 49 Dynamic Control Exercise Complete Screen

Appendice C – User Evaluation Questionnaire

User Evaluation Questionnaire – Aria Singing Trainer App

* Indicates required question

Section A: General Use and Interface

How easy was it to navigate the Aria application? *

- ◯ Very difficult
- ◯ Difficult
- ◯ Neutral
- ◯ Easy
- ◯ Very easy

The design and layout were visually appealing. *

- ◯ Strongly disagree
- ◯ Disagree
- ◯ Neutral
- ◯ Agree
- ◯ Strongly Agree

Did you encounter any bugs or technical issues? *

- ◯ Yes
- ◯ No

If yes, please describe briefly:

Your answer

On what device did you primarily use Aria?

- ◯ Desktop/Laptop
- ◯ Tablet
- ◯ Mobile

Section B: Karaoke Module

Did the karaoke feature feel accurate and responsive? *

- ◯ Not at all
- ◯ Slightly
- ◯ Moderately
- ◯ Very
- ◯ Extremely

Was the pitch feedback clear and useful? *

- ◯ Strongly disagree
- ◯ Disagree
- ◯ Neutral
- ◯ Agree
- ◯ Strongly Agree

Did the lyrics and music stay in sync? *

- ◯ Never
- ◯ Rarely
- ◯ Sometimes
- ◯ Often
- ◯ Always

Section C: Vocal Exercises

How engaging did you find the exercises? *

- ◯ Not engaging
- ◯ Slightly engaging
- ◯ Moderately engaging
- ◯ Very engaging
- ◯ Extremely engaging

Which exercise was your favourite? *

Your answer

Do you feel these exercises helped you improve your singing? *

- ◯ Yes
- ◯ No
- ◯ Maybe

Section D: Learning Content

Were the theory lessons clear and informative? *

- Yes
- No
- Neutral

Would you prefer more interactivity (e.g., quizzes, videos) in the lessons? *

- Yes
- No
- Maybe
- Not Sure

How helpful was the progress tracking feature in keeping you motivated? *

- Not helpful
- Slightly helpful
- Moderately helpful
- Very helpful
- Extremely helpful

Section E: Overall Experience

How likely are you to use Aria regularly in your vocal training routine? *

- ◯ Very unlikely
- ◯ Unlikely
- ◯ Not sure
- ◯ Likely
- ◯ Very likely

Do you feel that Aria helped improve your vocal skills in any way? *

- ◯ Not at all
- ◯ A little
- ◯ Moderately
- ◯ Significantly
- ◯ Greatly

What features would you like to see improved or added? *

Your answer

Any additional feedback, suggestions, or comments? *

Your answer

Submit Clear form