

**AI-ENABLED INTELLIGENT ASSISTANT TO
IMPROVE READING AND COMPREHENSION
SKILLS IN ENGLISH LANGUAGE**

A.P. Ranaweera

(IT21182396)

B.Sc. (Hons) Degree in Information Technology Specialized in
Software Engineering

Department of Computer Science and Software
Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

April 2025

**AI-ENABLED INTELLIGENT ASSISTANT TO
IMPROVE READING AND COMPREHENSION
SKILLS IN ENGLISH LANGUAGE**

A.P. Ranaweera

(IT21182396)

Dissertation submitted in partial fulfillment of the requirements for the Special
Honor's Degree of Bachelor of Science in Information Technology Specializing in
Software Engineering


Department of Computer Science and Software Engineering

Sri Lanka Institute of Information Technology Sri Lanka

April 2025

DECLARATION

I declare that this is my own work and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
A.P. Ranaweera	IT21182396	

Signature of the Supervisor
(Dr. Dasuni Nawinna)

Date

.....

.....

ABSTRACT

This research presents an intelligent pronunciation coaching system designed to support English language learners in improving their speaking accuracy. The system integrates Automatic Speech Recognition (ASR) via Azure Cognitive Services to transcribe user speech and uses Grapheme-to-Phoneme (G2P) modeling to convert expected and spoken words into phoneme sequences. Dynamic Time Warping (DTW) is applied to align these sequences and compute similarity distances, enabling the system to identify precise mispronunciations at the phoneme level.

To enhance phoneme-level error analysis, the system incorporates articulatory feature-based comparison using the PanPhon library. By mapping phonemes to multi-dimensional articulatory vectors and measuring feature distances, the system offers more granular feedback. Fuzzy matching techniques are used to mitigate common speech recognition inaccuracies and ensure reliable alignment. When mispronunciations are detected, the system generates highlighted visual feedback, pinpointing erroneous phoneme positions and providing descriptive guidance on articulation corrections. A large language model (LLM) is used to dynamically generate similar-sounding practice words for each mispronounced phoneme, enhancing targeted learning. Users can interactively listen to these practice words via speech synthesis to reinforce correct pronunciation. The learning flow follows a gamified level-based progression, where users earn points for accurate pronunciation and unlock increasingly complex word sets based on phonetic and lexical difficulty. The entire system is deployed on Amazon Web Services using a microservice architecture, with each module containerized via Docker and orchestrated through Amazon ECS.

Keywords – Speech Recognition, Pronunciation feedback, ASR, Grapheme-to-Phoneme Model, Dynamic Time Warping, articulatory features, phoneme alignment, speech learning

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to my supervisor, Dr. Dasuni Nawinna, for her unwavering guidance, support, and encouragement throughout the course of this undergraduate research. Her insights and mentorship were instrumental in shaping the direction and quality of this work.

My sincere appreciation also goes to Mr. Jeewaka Perera, the co-supervisor of this project, for his valuable assistance and readiness to provide help whenever needed. I am especially thankful to Miss Ridmi Dinanjali, an external English language expert, whose contributions added significant practical depth and relevance to the pronunciation-focused aspects of the study.

I would also like to thank my fellow team members and friends for their collaboration, helpful feedback, and constant motivation throughout this journey. Their involvement greatly enriched the development of this research.

Finally, I am deeply thankful to my family for their endless patience, understanding, and encouragement. Their support was the foundation that allowed me to complete this work with confidence and focus. This achievement would not have been possible without the contributions of all the individuals mentioned above.

Table of Contents

DECLARATION	1
ABSTRACT	2
ACKNOWLEDGEMENT.....	3
1 INTRODUCTION.....	9
1.1 Background literature.....	9
1.1.1 Common Pronunciation Issues in English Language Learning.....	11
1.1.2 Traditional Vocabulary Learning Methods and Their Limitations.....	12
1.1.3 The Role of Computer-Aided Pronunciation Training (CAPT).....	13
1.1.4 Current Approaches to Pronunciation Error Detection and Feedback Mechanisms.....	14
1.1.5 Phoneme Recognition and Error Identification in English Pronunciation ...	15
1.2 Research Gap.....	16
1.2.1 Current Trends and Their Limitations in Pronunciation Learning Tools	16
1.3 Research Problem.....	19
1.4 Research Objectives	22
1.4.1 Main Objectives	22
1.4.2 Specific Objectives.....	22
2 METHODOLOGY	24
2.1 Methodology	24
2.1.1 Requirements Gathering and Analyzing	24
2.1.2 Feasibility Study.....	25
2.1.3 Problem Statement	26
2.1.4 System Designs	27
2.1.5 Speech Capturing and Audio Preprocessing Techniques	31
2.1.6 Speech Recognition and Transcription Pipeline.....	32
2.1.7 Phoneme Conversion and Representation Approaches	32
2.1.8 Phoneme Comparison Algorithm and Alignment Techniques.....	35
2.1.9 Error Highlighting and Feedback Generation	37
2.1.10 AI-Powered Personalized Practice Word Generation Based on Mispronounced Phonemes.	38
2.2 Commercialization aspects of the product	40
3 Testing & Implementation.....	42
3.1 Implementation.....	42
3.1.1 Implementation of Audio Capturing and Preprocessing	42
3.1.2 Implementation of Speech Recognition and Phoneme Analysis.....	45
3.1.3 Implementation of Similar Sounding Word Generation.....	47
3.1.4 Front-End Implementation	48
3.2 Testing.....	53
3.2.1 Test Plan and Test Strategy	53
3.2.2 Test Case Design.....	54
4 RESULTS AND DISCUSSIONS	57
4.1 Results	57
4.2 Research Findings	60

4.3	Discussion	61
5	CONCLUSION	65
6	REFERENCES.....	67

LIST OF FIGURES

Figure 1.1 User Confidence in English word pronunciation.....	19
Figure 1.2 Challenges facing English pronunciation	20
Figure 1.3 Pronunciation Feedback Types.....	20
Figure 1.4 User Selected Features for Pronunciation Tool	21
Figure 2.1 Overall system diagram	27
Figure 2.2 Use case Diagram of the component	28
Figure 2.3 Component System Diagram.....	29
Figure 2.4 Phoneme-Level Pronunciation Analysis and Feedback Workflow	39
Figure 3.1 Audio capturing process.	43
Figure 3.2 Audio Blob Conversion to WAV Format.....	43
Figure 3.3 Manually Constructing a WAV Header	44
Figure 3.4 Audio transcription using Microsoft Azure	45
Figure 3.5 Phoneme Extraction Using G2P Conversion.....	45
Figure 3.6 Making DTW more consistent	45
Figure 3.7 Phoneme Sequence Comparison Using DTW	45
Figure 3.8 Mismatch Detection and Feedback Highlighting	46
Figure 3.9 Mismatch Detection and Feedback Highlighting	46
Figure 3.10 Backend API Integration Endpoint.....	46
Figure 3.11 Similar Sounding Word Generation (Using Gemini LLM).....	48
Figure 3.12 Pronunciation Starter Guide	49
Figure 3.13 The Pronunciation Start Page	49
Figure 3.14 The Pronunciation Start Page	50
Figure 3.15 Pronunciation correct feedback.	51
Figure 3.16 Pronunciation incorrect feedback	51
Figure 3.17 Pronunciation incorrect feedback	52
Figure 3.18 Similar word generation based on the user's mispronounced phonemes.	52

LIST OF TABLES

Table 1:1:An Overview of Traditional Pronunciation Training Techniques	12
Table 1:2 Vocabulary Learning Tools Comparison.....	18
Table 2:1 Sample English words and their corresponding phoneme sequences.....	34
Table 3:1 Test Case to Record button triggers.....	54
Table 3:2 Test Case to audio file recording function.....	54
Table 3:3 Test Case to input voiceless audio	55
Table 3:4 Test Case to valid pronunciation	55
Table 3:5 Test Case to incorrect pronounce.....	55
Table 3:6 Test Case to practice word generation.....	56
Table 3:7 Test Case to listen generated sound.....	56
Table 3:8 Test Case to failure of Gemini Ai.....	56
Table 4:1 Accuracy Analysis Table.....	58
Table 4:2 Evaluation with Student-Level English Learners	59

LIST OF ABBREVIATIONS

Abbreviation	Full Term
IPA	International Phonetic Alphabet
CAPT	Computer-Aided Pronunciation Training
ASR	Automatic Speech Recognition
G2P	Grapheme-to-Phoneme
DTW	Dynamic Time Warping
LLM	Large Language Model
SDK	Software Development Kit
OOV	out-of-vocabulary
UI	User Interface
UX	User Experience
ECS	Elastic Container Service
DB	Database
API	Application Programming Interface
UAT	User Acceptance Testing
AWS	Amazon Web Services,
AI	Artificial Intelligence
CSS	Cascading Style Sheets

1 INTRODUCTION

1.1 Background literature

Even in English has become an essential global language not only for business and technology, but also for education, science and international communication. As a result, proficiency in English, especially proficiency in spoken communication, is increasingly seen as a key skill for personal and professional development. However, achieving correct pronunciation remains a significant obstacle for many non-native English speakers.

Mispronunciations can lead to misunderstandings, impacting both the speaker's confidence and the listener's comprehension. This challenge is particularly evident in regions where English is taught as a second or foreign language, where learners may have limited exposure to native pronunciation patterns and may be influenced by the phonetic structure of their mother tongue.

The importance of correct pronunciation in mastering English cannot be underestimated. Pronunciation involves the correct production of phonemes, the smallest units of sound that distinguish one word from another [1]. For example, the difference in pronunciation between "ship" and "sheep" hinges on the accurate articulation of the vowel phonemes /ɪ/ and /i:/ [2]. Mispronouncing phonemes can alter the meaning of a word entirely, leading to communication breakdowns. Therefore, there is a growing recognition of the need for educational tools that can provide learners with targeted pronunciation training, focusing on the correct articulation of phonemes.

Traditionally, language teachers have made use of the phonetic alphabet, and activities, such as transcription practice, diagnostic passages, detailed description of the articulatory systems, recognition/discrimination tasks, developmental approximation drills, focused production tasks (e.g., minimal pair drills, contextualized sentence practice, reading of short passages or dialogues, reading aloud/recitation), tongue twisters, and games (e.g., Pronunciation Bingo). Other trendy methods are listening

and imitating, visual aids, practice of vowel shifts, stress shifts related by affixation, and recordings of learner's production. All these techniques are based on teachers having their students learn each sound and then apply them in real speech. Some students benefit from these techniques; however, others do not learn the pronunciation of the other language easily from them. For this reason, new techniques are being developed to supplement the learning of English pronunciation [3].

Computer-Aided Pronunciation Training (CAPT) systems have emerged as a promising solution to these challenges. By leveraging advances in speech recognition technology, CAPT systems can analyze a learner's speech in real-time and provide immediate feedback on their pronunciation. However, many existing CAPT systems are limited in their ability to provide detailed, phoneme-level feedback. Instead, they often focus on broader aspects of pronunciation, such as stress, rhythm, and intonation, without addressing the specific phoneme-level errors that are critical to achieving native-like pronunciation.

The proposed research seeks to address these limitations by developing a CAPT system that integrates advanced speech recognition technologies with phoneme level analysis. The system will recognize spoken words into a sequence of phonemes and identify pronunciation errors at the phoneme level. To achieve this, the system will use the grapheme-to-phoneme model and a suite of speech recognition tools, among other approaches.

In addition to phoneme-level analysis, the proposed CAPT system will incorporate large language models (LLMs) to generate personalized training words for learners, according to the mispronounced phoneme set. In the context of the proposed CAPT system, LLMs will be used to create practice words that are similar in phonetic structure to the words that the learner has difficulty pronouncing. This approach ensures that learners receive targeted practice on the specific phonemes they struggle with, thereby improving their pronunciation over time.

1.1.1 Common Pronunciation Issues in English Language Learning

English language learners commonly encounter several significant pronunciation challenges that can affect their communication effectiveness. One of the most fundamental issues is the accurate production of phonemes, which are the smallest units of sound that distinguish one word from another. For example, the difference between "ship" and "sheep" hinges entirely on the precise articulation of the vowel phonemes /ɪ/ and /i:/ [4]. This level of phonetic precision is particularly challenging for non-native speakers, as mispronunciations can completely alter word meanings and lead to communication breakdowns.

Another challenge is that English spelling does not always match how words are pronounced. Words like "though," "through," and "cough" all have different pronunciations even though they look similar. This inconsistency makes it harder for learners to know how to pronounce new words just by looking at them [5]. In addition to individual sounds, learners also struggle with things like stress, intonation, and rhythm. English is a stress-timed language, meaning that some syllables are said with more force or emphasis than others. If a learner puts the stress on the wrong syllable for example, saying *preSENT* (verb) instead of *PREsent* (noun) it can change the meaning of the word or make it harder to understand [6].

Besides these language-related difficulties, there are also emotional challenges. Many learners feel shy or anxious about speaking in English, especially when they know their pronunciation isn't perfect. This fear can prevent them from practicing, which makes it even harder to improve [7]. Finally, in most classrooms, pronunciation is not given enough focus. Teachers may not have time to provide feedback to each student, or they may not be trained to identify and correct detailed pronunciation errors. As a result, students often do not receive the specific guidance they need to improve [8]. These problems show why it's important to have better tools that help learners improve their pronunciation. Specifically, learners need systems that can detect errors in individual sounds (phonemes) and give helpful, personalized feedback so they can practice and improve their spoken English more effectively.

1.1.2 Traditional Vocabulary Learning Methods and Their Limitations

Traditional pronunciation training has historically emphasized achieving “native-like” articulation, with a strong focus on accuracy and clarity. It relied on structured methods such as minimal pair drills, phonetic transcription, and contextualized exercises to help learners identify and produce target sounds effectively. Commonly, this instruction followed one of three pedagogical models. The Analytic-Linguistic Approach, which used tools like the International Phonetic Alphabet (IPA) to break down and study individual sounds. The Intuitive-Imitative Approach, which prioritized listening to native models and imitating them and the Integrative Approach, which combined pronunciation practice with other language skills through real-life communication tasks. Core techniques included phonetic training, listening and imitation, reading aloud, and the use of tongue twisters and visual aids. These methods provided foundational support for pronunciation learning [9].

However, these approaches also had limitations. They often prioritized accuracy over fluency, focused narrowly on segmental features (individual phonemes) rather than suprasegmentals like stress and intonation, and provided limited opportunities for real-world or spontaneous practice. Additionally, they lacked personalized feedback, particularly at the phoneme level, making it difficult for learners to recognize and address their specific pronunciation challenges [10]. Table 1 summarizes these techniques and their focus areas.

Table 1:1:An Overview of Traditional Pronunciation Training Techniques

Technique	Description	Approach Type	Focus Area
Phonetic Training (IPA)	Learning and applying the International Phonetic Alphabet to identify and produce English sounds.	Analytic-Linguistic	Segmental (individual phonemes)
Minimal Pair Drills	Contrasting words that differ by one sound (e.g., “ship” vs. “sheep”)	Intuitive-Imitative	Sound discrimination and accuracy

Listening and Imitation	Repeating words or phrases after native speaker models	Intuitive-Imitative	Overall pronunciation fluency
Tongue Twisters	Practicing difficult sound sequences for better articulation and fluency.	Integrative	Articulation and clarity
Reading Aloud/Recitation	Practicing pronunciation through poetry or passages.	Integrative	Natural speech rhythm and stress
Teacher/Peer Feedback	Direct correction and suggestions provided during class activities.	All approaches	General articulation and fluency

1.1.3 The Role of Computer-Aided Pronunciation Training (CAPT)

Introduction Computer-Aided Pronunciation Training (CAPT) plays a significant role in helping language learners improve their pronunciation through interactive, technology-driven methods. CAPT systems provide individualized, instant feedback using advanced technologies like automated speech recognition (ASR), allowing learners to focus on phonemes, intonation, and stress patterns effectively. CAPT has proven particularly beneficial for non-native speakers, as it offers a cost-effective and scalable way to enhance pronunciation skills compared to traditional methods [11].

Research shows that CAPT improves learners' pronunciation skills by helping them practice repeatedly and receive real-time corrections. Moreover, these systems are found to be highly motivating, especially when integrated into personalized learning environments [12]. CAPT tools also emphasize the importance of intelligibility over perfection, aiding learners in achieving effective communication [13]. The proposed research seeks to develop a CAPT system that addresses these limitations by integrating phoneme-level analysis and real-time feedback. This system will utilize advanced speech recognition technologies to convert spoken words into their constituent phonemes, identify pronunciation errors, and provide immediate feedback on the specific sounds that need improvement. By focusing on the phoneme level, learners can target the precise areas where they struggle, rather than receiving generalized feedback.

1.1.4 Current Approaches to Pronunciation Error Detection and Feedback Mechanisms

Modern pronunciation training systems, particularly those embedded in popular language learning platforms have made strides in integrating speech technologies for user interaction. However, their error detection and feedback mechanisms remain largely superficial. Most current approaches rely on automatic speech recognition (ASR) to compare a user's spoken input against a predefined word or sentence model. If the utterance deviates beyond a certain threshold, the system typically provides a binary result labeling the pronunciation as either "correct" or "incorrect" without indicating what went wrong or how to improve [14]. While this kind of feedback may be easy to interpret, it lacks the granularity required to support meaningful learning, especially for learners struggling with specific phonemes.

These systems often do not identify the specific phoneme-level deviations, nor do they provide insights into articulatory issues such as stress misplacement, intonation errors, or substitution of individual sounds [15]. As a result, learners may receive a negative assessment without understanding whether the problem was a missed vowel quality, a misarticulated consonant, or a rhythm issue. This lack of diagnostic feedback limits the learner's ability to correct errors effectively. Moreover, the absence of targeted practice recommendations or personalized support makes it difficult for users to improve over time. While some platforms attempt to offer pronunciation guides or native recordings for comparison. They fall short of providing interactive, corrective guidance at the phoneme level. Which research shows is critical for long-term pronunciation development [16].

These limitations have highlighted the need for next-generation systems capable of offering detailed, real-time, and phoneme-specific feedback. The goal is to move beyond judgment-based outputs and toward constructive, educational feedback that identifies exact mispronunciations.

1.1.5 Phoneme Recognition and Error Identification in English Pronunciation

Phoneme recognition is a critical aspect of pronunciation training. Existing CAPT systems that use Automatic Speech Recognition (ASR) technology typically compare a learner's spoken input to a predefined word model, delivering feedback at the word level. While this is useful for general language learning, it does not offer the granularity needed to correct specific sound errors. This limitation is particularly significant for English learners, where even small phoneme-level mistakes can change the meaning of a word [17].

The proposed system will provide advanced phoneme recognition algorithms to break down words into individual phonemes and identify errors at this level. The system will detect errors and provide corrective feedback. This precise, phoneme-level feedback allows learners to focus on correcting individual sounds rather than grappling with entire words or sentences.

In this research, the proposed system adopts a fine-grained approach to pronunciation evaluation by segmenting spoken input into phonemes and comparing them directly with the expected phonemic output of a given word. The process begins with automatic speech recognition (ASR), which transcribes user speech. This transcription is then analyzed using a Grapheme-to-Phoneme (G2P) model to generate both the expected and actual phoneme sequences. By comparing these two sequences, the system can identify mismatches that correspond to phoneme-level pronunciation errors. To measure the similarity between the spoken and expected phoneme strings, the system uses Dynamic Time Warping (DTW) an algorithm well-suited for aligning sequences with temporal variation [18], [19]. DTW not only identifies insertions, deletions, and substitutions of phonemes, but also quantifies the overall pronunciation distance, offering a metric for learner progress.

Furthermore, this research incorporates articulatory-aware analysis using tools such as PanPhon, which compares the feature-based representation of phonemes rather than just their symbolic labels. This allows the system to determine if an error was due to a change in voicing (e.g., /t/ \rightarrow /d/), place of articulation (e.g., /t/ \rightarrow /k/), or manner (e.g.,

/s/ → /ʃ/), offering learners detailed insights into the nature of their mistakes [20]. By identifying which exact sound was mispronounced and explaining how it differs from the target sound in terms of articulatory features, the system delivers constructive, phoneme-specific feedback.

This level of detail is essential for developing effective pronunciation habits, particularly in English where phonemic precision is vital for intelligibility. For example, failing to differentiate between /i:/ and /ɪ/ in “sheep” and “ship” can drastically alter meaning. By focusing on phoneme-level recognition and correction, the system not only enhances learners' awareness of subtle sound distinctions but also empowers them to self-correct through guided feedback and practice. This is further strengthened by the integration of Large Language Models (LLMs), which dynamically generate phonetically similar practice words based on the user's mispronounced phonemes, facilitating targeted repetition and reinforcing proper articulation. In essence, this phoneme recognition and error identification framework transforms pronunciation training from passive correction to interactive diagnosis and guided improvement, closing the feedback gap that exists in many traditional and commercial language learning systems.

1.2 Research Gap

1.2.1 Current Trends and Their Limitations in Pronunciation Learning Tools

Despite the availability of several digital tools aimed at improving pronunciation, few focus specifically on phoneme-level feedback. Popular language-learning platforms such as Duolingo, Rosetta Stone, BoldVoice and Linguacoach emphasize overall language skills such as vocabulary, grammar, and comprehension, while offering limited support for detailed pronunciation training [21]-[24]. These platforms often evaluate spoken input at the word or sentence level, indicating whether the pronunciation was correct or incorrect, without identifying the specific phonemes that were mispronounced.

The paper [25] considering pronunciation errors as divided into accent and lexical errors and a methodology for detecting each is presented and evaluated. The paper is

investigated in the context of three corpora, two on which humans were asked to annotate pronunciation errors, and one where they were asked to transcribe actual pronunciation. Results are consistent with accent and lexical errors being defined as distinct categories of error that can be detected separately. The system was successfully able to detect word-level accent and lexical errors on the latter corpus but not the former two. It was, however, able to diagnose lexical and general and specific accent error tendency with satisfactory performance across all three datasets. Analysis suggested that the annotators of the first two corpora were systematically under-annotating accent errors and that therefore the phonetic transcription technique is a superior method of annotation for error detection tasks [25].

Furthermore, existing platforms often offer generic feedback and do not adapt to individual learners' needs. For instance, while tools may provide users with an indication of mispronounced words, they fail to offer targeted exercises that focus on the specific sounds that need improvement. The absence of adaptive learning mechanisms, such as generating similar words with the same phonetic structure for additional practice, leaves learners without sufficient resources to effectively address their pronunciation challenges.

In addition, most of the widely used pronunciation platforms rely on fixed content and do not leverage advanced technologies like Large Language Models (LLMs) to dynamically generate new practice material based on user performance. As a result, learners may not be exposed to diverse words that challenge their weak phonemes, leading to slower progress in mastering accurate pronunciation.

The proposed system seeks to address these research gaps by introducing a phoneme-level feedback mechanism, real-time error detection, and adaptive learning. The integration of LLMs allows for the generation of similar words that target specific phoneme errors [26], creating a personalized and dynamic learning path. This system provides a more efficient way for users to identify, understand, and correct their pronunciation errors, something that is largely missing from the current landscape of pronunciation tools.

Table 1:2 Vocabulary Learning Tools Comparison

Research paper / Tools	Pronounce error detection	Phoneme level feedback	Similar words generation for practice	Focus on phoneme level practice	Use of large language models (LLMs)
Duolingo	✓	×	×	×	×
Rosetta Stone	✓	×	×	×	×
BoldVoice	✓	×	×	×	×
Linguacoach	✓	×	×	✓	×
Research paper A	✓	×	×	×	×
Research paper B	✓	✓	×	×	×
Research paper C	✓	×	×	×	×
Purposed system	✓	✓	✓	✓	✓

The comparison table clearly illustrates the differences between existing systems and the proposed platform. They focused on providing feedback at the word or sentence level. While these platforms help users understand whether a word is pronounced correctly, they do not identify the specific sound (phoneme) that causes the mistake. In contrast, the proposed system offers phoneme-level feedback, enabling users to pinpoint and correct errors with greater precision.

Another key gap addressed by the proposed system is real-time feedback. Unlike current platforms that only provide feedback after the user completes speaking a word or sentence, the proposed solution gives immediate corrections, allowing users to adjust their pronunciation on the spot.

1.3 Research Problem

The challenge of mastering English pronunciation remains a significant barrier for non-native speakers, particularly in regions where English is not the primary language. Despite advances in language learning technologies, most current tools primarily focus on general language acquisition, offering limited emphasis on phonetic precision and individualized feedback. This lack of tailored pronunciation guidance often leaves learners struggling with specific sounds, resulting in persistent errors that hinder effective communication and reduce learner confidence. The problem is further compounded by the scarcity of accessible, user-friendly systems capable of delivering detailed phoneme-level analysis and adaptive feedback, especially those that can evolve with a learner's progress and needs.

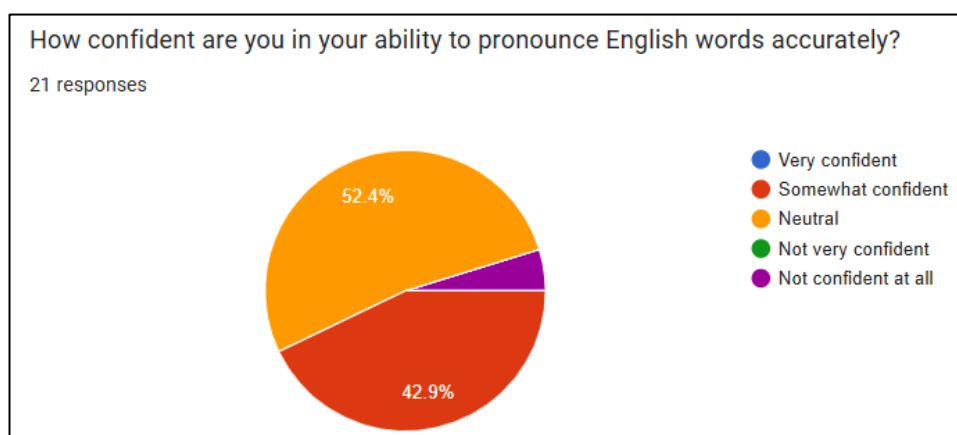


Figure 1.1 User Confidence in English word pronunciation.

The responses (Figure 1.1) indicate that most participants do not feel fully confident in their ability to pronounce English words accurately. A significant portion expressed only moderate confidence, while others remained neutral or unsure about their pronunciation skills. This suggests that learners are aware of their limitations and may benefit from tools that offer more focused support. It highlights the need for pronunciation training solutions that build user confidence through clear, phoneme-specific feedback and guided practice, rather than relying solely on generic or binary evaluations.

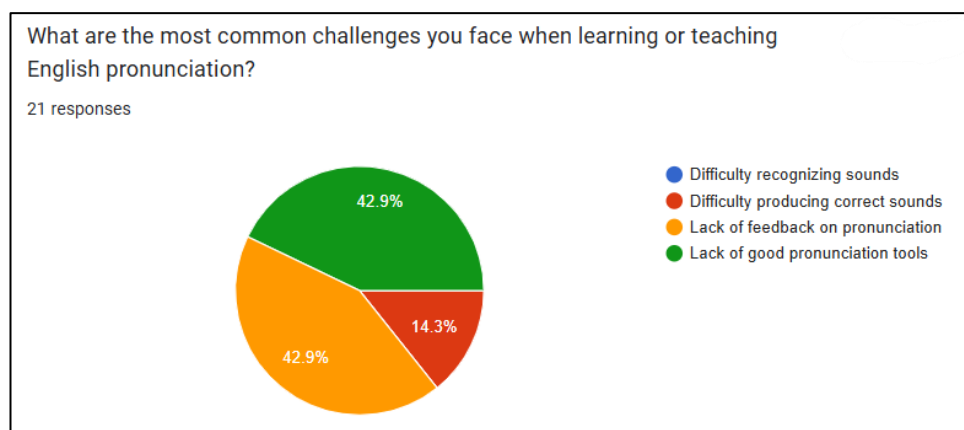


Figure 1.2 Challenges facing English pronunciation

The responses (Figure 1.2) highlight two major challenges learners face in mastering English pronunciation: the lack of effective pronunciation tools and the lack of useful feedback on their pronunciation efforts. These issues are just as common as the difficulty in producing correct sounds, showing that learners are not only struggling with speech itself but also with the quality of support available to them. This emphasizes the need for advanced learning systems that provide meaningful, targeted feedback and offer practical tools designed to address individual pronunciation challenges.

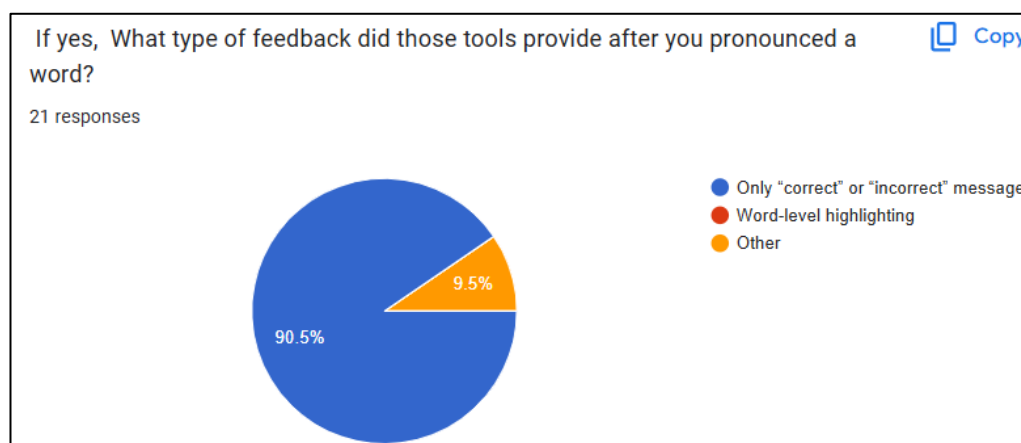


Figure 1.3 Pronunciation Feedback Types

The chart (Figure 1.3) shows that most learners using pronunciation tools receive very basic feedback, typically limited to a simple "correct" or "incorrect" message. This kind of evaluation lacks depth and fails to inform the user about what specifically went

wrong. Only a few reported receiving more detailed responses such as word-level highlighting. This clearly indicates a gap in the quality of feedback provided by existing tools and highlights the need for systems that deliver more informative, phoneme-specific, and actionable feedback to help learners truly improve.

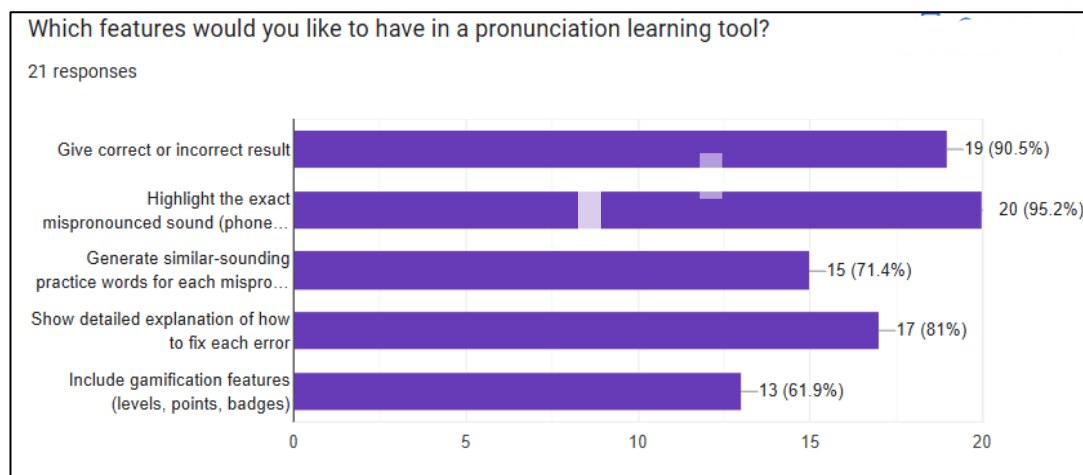


Figure 1.4 User Selected Features for Pronunciation Tool

The figure 1.4 responses clearly show that learners are not just looking for basic feedback they expect a more advanced, supportive, and personalized pronunciation learning experience. Most participants expressed interest in tools that can highlight the exact mispronounced sound within a word, explain how to fix that error, and provide similar-sounding words for further practice. Additionally, many also favor having motivational elements like gamification to keep the learning engaging. These preferences confirm a strong demand for a pronunciation system that offers phoneme-level analysis, real-time correction, and intelligent guidance, far beyond the capabilities of most existing tools.

1.4 Research Objectives

1.4.1 Main Objectives

To design and implement an intelligent speech error detection and feedback system that operates at the phoneme level. The goal is to enhance English pronunciation learning by analyzing the learner's spoken input, accurately identifying mispronounced phonemes, and delivering personalized corrective feedback. This system aims to provide real-time, targeted guidance that helps learners understand their specific pronunciation errors and improve through focused practice. By leveraging phoneme-level analysis, the tool will offer a more precise and effective learning experience, especially for non-native speakers seeking to improve their spoken English clarity and confidence.

1.4.2 Specific Objectives

Phoneme-Level Speech Analysis

- The first objective focuses on implementing a precise phoneme-level analysis mechanism by utilizing grapheme-to-phoneme (G2P) modeling and Dynamic Time Warping (DTW). This mechanism compares the phoneme sequence of a learner's spoken input with the expected phoneme sequence of the target word. DTW helps identify where deviations occur, including insertions, deletions, and substitutions of phonemes. This step is crucial in determining the accuracy of pronunciation at a granular level, enabling the system to detect exactly which sounds are mispronounced.

Real-Time Feedback System Design

- Design an interactive feedback system that highlights specific pronunciation errors. Rather than giving generic feedback, the system aims to visually and contextually point out the mispronounced phonemes, offering learners clear insights into their pronunciation mistakes. By supporting real-time correction,

this feedback mechanism enhances the learner's ability to recognize and self-correct errors immediately, promoting faster and more effective pronunciation improvement.

Personalized Practice Word Generation

- Generating personalized word suggestions based on the mispronounced phonemes detected. Using these targeted words, the learner can practice sounds they struggle with in different contexts. This personalized practice reinforces correct pronunciation patterns and helps prevent recurring errors. The inclusion of similar-sounding words also enhances sound differentiation skills, making this feature a practical and learner-specific support mechanism.

2 METHODOLOGY

2.1 Methodology

2.1.1 Requirements Gathering and Analyzing

The initial phase of this research involved systematically identifying and analyzing the requirements necessary to develop an effective phoneme-level pronunciation training system. The goal was to ensure that the proposed solution would address real user challenges and align with best practices in pronunciation pedagogy and intelligent feedback delivery.

To gather relevant insights, a survey was conducted among English learners and instructors, focusing on their experience with existing pronunciation tools, the type of feedback they receive, and their expectations for improvement. Most participants indicated that current tools offer limited support, typically providing only a “correct” or “incorrect” response without identifying the specific phonemes that were mispronounced. Many users also expressed a need for personalized practice, visual feedback, and real-time correction guidance, confirming a significant gap in available learning solutions.

In parallel, a literature review was conducted to examine current technologies used in pronunciation training, such as Automatic Speech Recognition (ASR), Grapheme-to-Phoneme (G2P) models, and Dynamic Time Warping (DTW). The analysis highlighted that while some research systems offer phoneme-level evaluation, few provide it in a user-friendly, adaptive, and accessible form suitable for regular learner use.

The requirement analysis process resulted in the identification of both functional and non-functional requirements. Functional requirements include speech input processing, phoneme extraction, error detection, visual feedback presentation, and practice word generation. Non-functional requirements focus on system usability, real-time response, scalability, and adaptability to user progress. These clearly defined

requirements provided the foundation for the design and development of a comprehensive Computer-Aided Pronunciation Training (CAPT) system capable of delivering intelligent, corrective, and personalized feedback.

System Requirements of the Phoneme-Level Speech Error Detection Module

- The system should allow users to speak or upload a word they are trying to pronounce.
- It must accurately recognize the word the user pronounced and compare it with the expected word.
- The system should analyze the spoken word to identify which parts (sounds) were pronounced incorrectly.
- The user should receive clear feedback showing where the pronunciation went wrong and how it can be improved.
- The system should offer similar-sounding words for the user to practice and improve their pronunciation over time.

2.1.2 Feasibility Study

- **Technical Feasibility**

The system is technically feasible due to the availability of reliable open-source tools and frameworks for speech recognition and phoneme processing. Technologies such as Automatic Speech Recognition (ASR), Grapheme-to-Phoneme module (G2P), and Dynamic Time Warping (DTW) for phoneme comparison are well-supported and integrable. Additionally, tools like PanPhon and speech synthesis APIs can be incorporated for enhanced feedback. The system architecture is designed with modularity in mind, making it compatible with modern development stacks and easily deployable using containerized environments like Docker and platforms such as AWS ECS.

- **Economic Feasibility**

The system can be developed and deployed with cost-effective tools and

infrastructure. Most of the libraries and frameworks used in the project are open-source or offer free academic usage tiers. Cloud services such as AWS offer flexible pricing models that fit research and pilot deployment budgets.

- **Operational Feasibility**

The system is designed to be intuitive and easy to use, even for non-technical learners. The real-time feedback mechanism and clear phoneme-level error explanations ensure a smooth user experience. Educators and students alike can operate the system with minimal training. Furthermore, the interface is responsive and accessible on device. Making it adaptable to classroom and self-learning environments.

- **Scheduling Feasibility**

The project is manageable within the defined academic timeline. The system has been broken into functional components such as input processing, phoneme detection, feedback generation, and practice word suggestion, allowing parallel development and testing. Development progress is tracked through project management tools. Given the status of implementation, remaining work including optimization, UI, and deployment can be completed within the expected research project timeframe.

2.1.3 Problem Statement

Despite the growing use of digital language learning tools, most pronunciation systems offer only basic feedback typically indicating whether a word was pronounced correctly or not without identifying the specific phoneme-level errors that led to mispronunciation. This lack of detailed, actionable feedback prevents learners from understanding their exact mistakes and hinders effective improvement. Furthermore, existing tools rarely adapt to individual learner needs or provide targeted practice suggestions. As a result, many non-native English speakers continue to struggle with accurate pronunciation, leading to communication difficulties and reduced confidence. This research aims to address this gap by developing a system capable of detecting

phoneme level pronunciation errors and delivering real-time, personalized feedback, along with guided practice resources based on individual learner performance.

2.1.4 System Designs

2.1.4.1 Overall system diagram

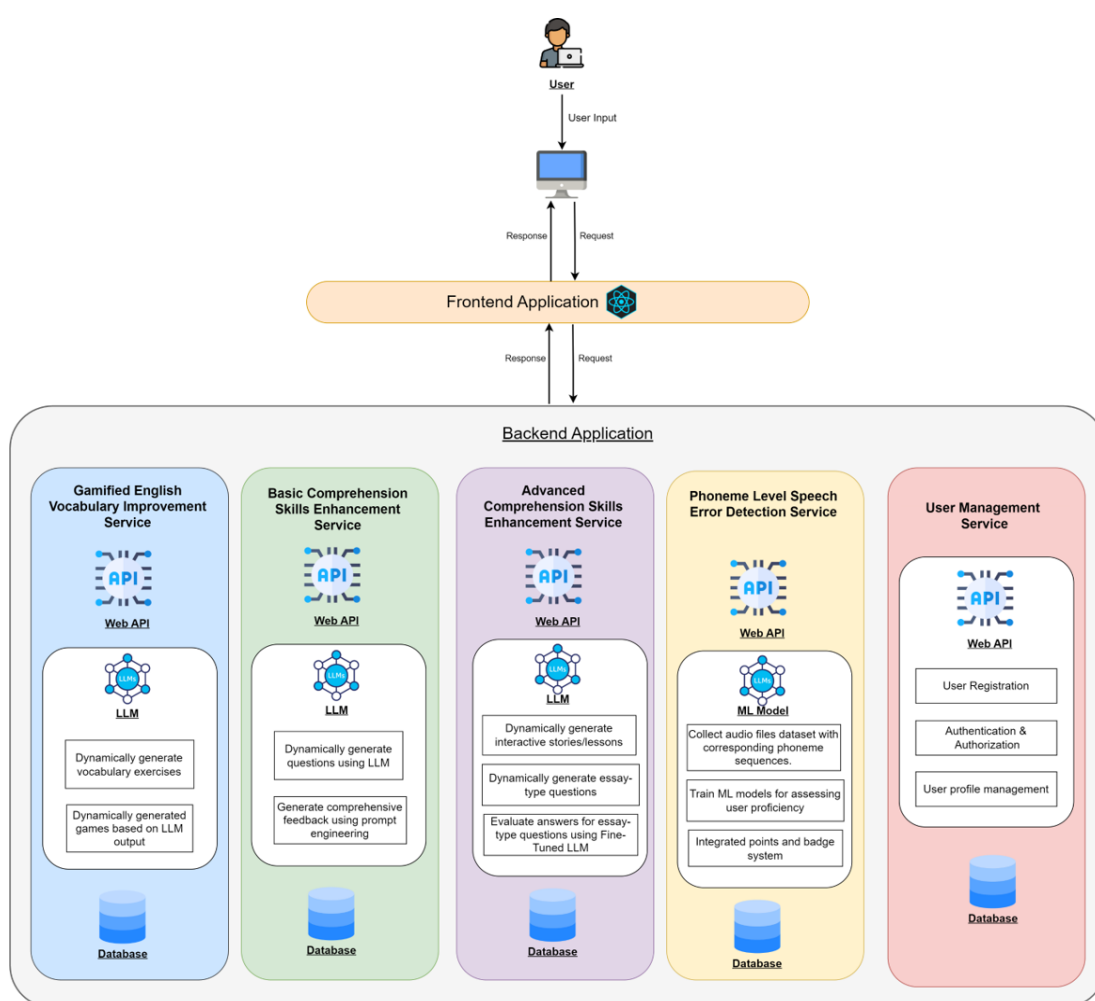


Figure 2.1 Overall system diagram

The Figure 2.1 application is designed using microservice architecture, where each major feature is implemented as an independent service that communicates with others through defined APIs. This modular approach enhances scalability, parallel development, and deployment flexibility. One of the core modules in this system is the Phoneme-Level Speech Error Detection Module, which operates as a standalone microservice within the broader learning platform.

Design Diagrams for the component

2.1.4.1.1 Use Case Diagram

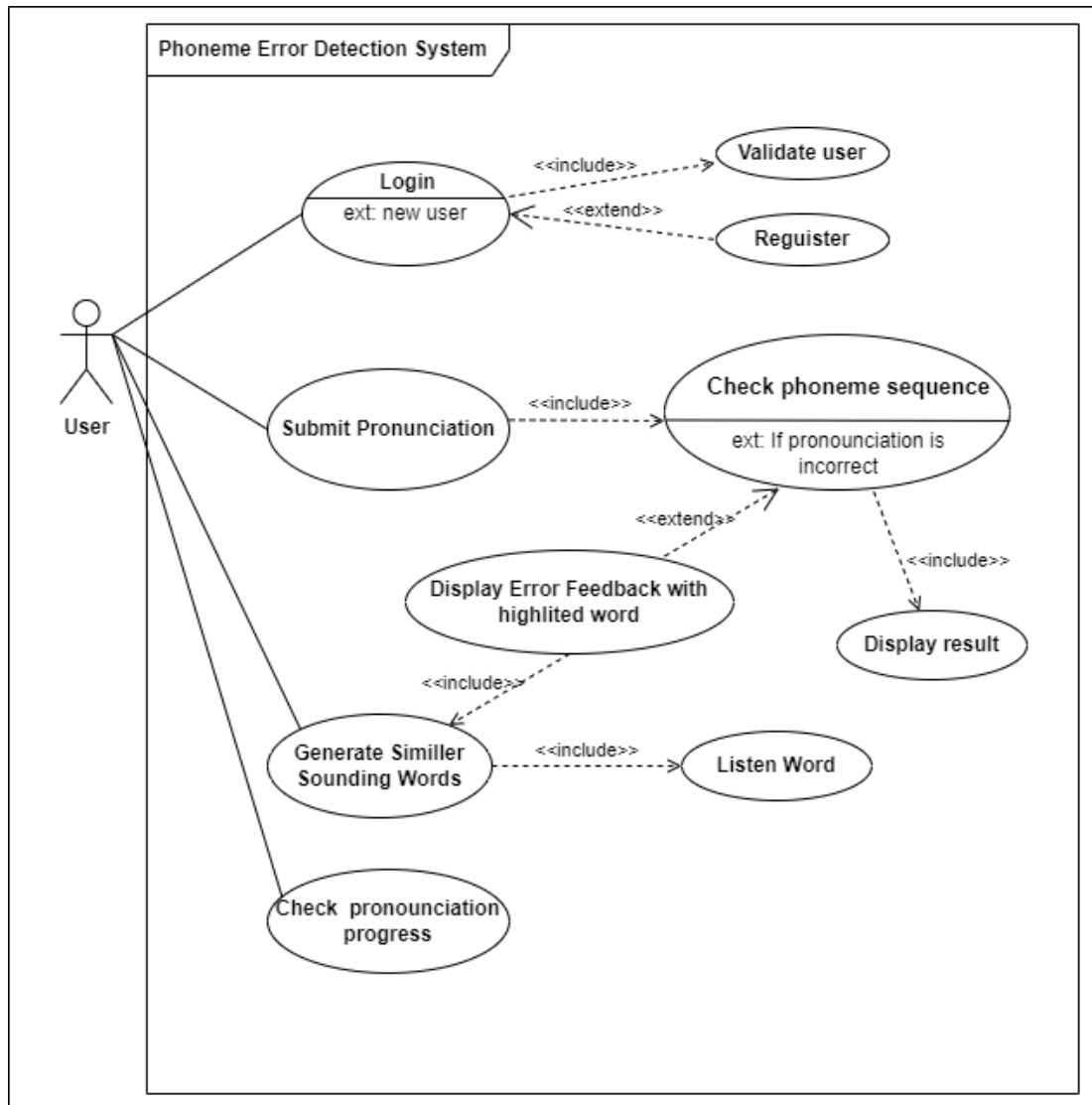


Figure 2.2 Use case Diagram of the component

2.1.4.1.2 Component System Diagram

The proposed system is composed of four major components, each responsible for a distinct function in supporting the learner's reading and comprehension development. Among these, the Phoneme-Level Speech Error Detection Module identifying mispronunciations at a fine-grained level and generating targeted corrective feedback. This component works alongside others such as vocabulary delivery, reading

comprehension assistance, and user progress tracking, all implemented as individual microservices. To represent the internal structure and interaction flow of the speech error detection module clearly, the system can be modeled through a component-level diagram as shown below.

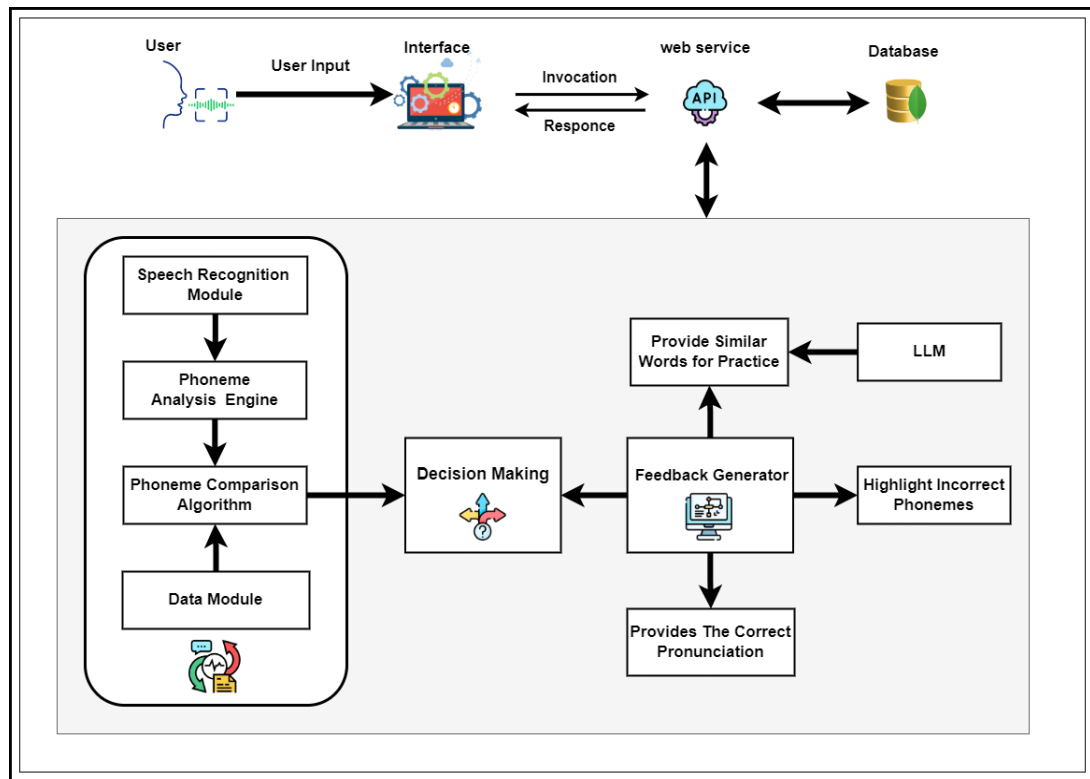


Figure 2.3 Component System Diagram

The proposed pronunciation training system is built as a figure 2.4 modular, microservice-based web application, comprising multiple independent components that work together to deliver real-time feedback on English pronunciation. Once the user accesses the application through the frontend interface built using React.js. They are presented with a word to pronounce, dynamically chosen based on their current learning level (beginner, intermediate, advanced). The user initiates the recording process via an intuitive UI, and their speech is captured and encoded into an audio blob using browser native Web APIs. This blob is then converted wave and sent to a backend service via a RESTful API call for further processing.

At the backend the process initiates with the Speech Recognition Module. Speech

recognition is performed using Azure Cognitive Services to transcribe the spoken input. The transcribed output is forwarded to the Phoneme Analysis Engine, which converts the recognized word and the expected target word into phoneme sequences using a Grapheme-to-Phoneme (G2P) model. This transformation allows the system to shift from surface-level text analysis to a more fine-grained phoneme-level comparison.

The resulting phoneme sequences are passed to the Phoneme Comparison Algorithm, which uses Dynamic Time Warping (DTW) to align the expected and spoken phoneme strings. This algorithm identifies key pronunciation errors. The comparison engine may also incorporate articulatory feature-based matching (PanPhon) to understand the nature of the mispronunciation. Throughout this process, a Data Module provides access to linguistic references such as pronunciation dictionaries and historical error patterns. Once the phoneme mismatches are detected, the results are passed to the decision-making section, which interprets the alignment data to determine the correctness of the pronunciation. It evaluates the severity of the errors and prepares structured information that can be transformed into feedback.

The processed result flows into the feedback generator, which performs several key functions. It first triggers the Highlight Incorrect Phonemes module to visually mark the problematic parts of the word for the user. And give detailed feedback according to user input. Then, it uses a Text-to-Speech function to output the correct pronunciation, allowing the learner to compare and make self-correct. Simultaneously, the feedback system connects with a Large Language Model (LLM) to generate a list of similar-sounding words for the mispronounced phoneme. These practice words are tailored to the learner's weak areas and serve as additional corrective input. The full set of feedback including highlighted phonemes, correct audio, and recommended practice words, and the option to listen to that word, is returned to the frontend for display.

2.1.5 Speech Capturing and Audio Preprocessing Techniques

Speech capturing and preprocessing are essential steps in building an effective pronunciation training system. These processes ensure that the spoken input from users is collected in a clear, standardized, and analyzable format suitable for downstream processing such as phoneme alignment and feedback generation.

The system captures speech using web-based MediaRecorder API technology, which allows real-time access to the user's microphone through their browser. Once the user initiates the pronunciation task, a short countdown timer provides a brief preparation period before recording begins. This ensures smoother user experience and more consistent speech input.

To ensure high-quality recordings, several preprocessing configurations are applied at the time of capture. These include:

- Mono channel audio capture, which simplifies phoneme stream alignment.
- Noise suppression and echo cancellation, which reduce background sounds and audio distortion.
- Standardized sampling rate (e.g., 44.1 kHz), which helps maintain clarity and compatibility across devices and models.

After recording, the audio is segmented into chunks, compiled into a single blob, and converted into a wav format. This conversion ensures that all inputs meet the same structural standards required for accurate analysis by speech recognition and phoneme comparison algorithms. This methodology ensures that every audio sample entering the analysis pipeline is reliable and comparable, directly contributing to the system's core goal. Such as providing real-time, accurate, and phoneme-level feedback on English pronunciation.

2.1.6 Speech Recognition and Transcription Pipeline

After audio is captured and preprocessed on the client side, it enters the Speech Recognition Pipeline, a crucial stage that transforms into machine-readable text. This transcription acts as the foundation for identifying phoneme-level pronunciation errors.

In the proposed system, transcription is performed using Azure Cognitive Services, a cloud-based automatic speech recognition (ASR) service that provides high-accuracy results across a wide range of accents and noise conditions. When the audio file reaches the backend, the Azure SDK is used to load the file into a speech recognizer that processes it in real time. The ASR engine returns the transcribed of what it interprets from the audio input. To ensure the analysis aligns with the learning objectives. The system isolates only the first word from the transcription, assuming a single-word input from the user per session. This word becomes the primary subject for phoneme-level analysis and comparison

2.1.7 Phoneme Conversion and Representation Approaches

After speech transcription, the next key stage in the pronunciation analysis pipeline involves converting both the expected and user-spoken words into phoneme sequences. This transformation is critical for enabling a fine-grained comparison between the correct pronunciation and the learner's spoken output. To achieve this, the system uses a Grapheme-to-Phoneme (G2P) conversion model, supported by auxiliary logic for handling non-standard or mispronounced inputs.

The system utilizes the g2p_en library, a neural network-based grapheme-to-phoneme converter that maps English words into their phonetic representations. This model is trained using the CMU Pronouncing Dictionary and handles context-sensitive word interpretation, effectively translating each word into a sequence of phonemes.

This phoneme mapping serves two key functions,

- **Expected Word Mapping:** Converts the correct word into its canonical

phoneme representation.

- **User Word Mapping:** Converts the ASR-transcribed version of what the user spoke into a corresponding phoneme stream.

Specifically, the g2p_en model, which can convert any given English word whether standard, uncommon, or user-defined into its corresponding sequence of phonemes. Unlike traditional static dictionaries that only support predefined vocabulary, this neural-based G2P model offers dynamic phoneme generation, making it suitable for handling out-of-vocabulary (OOV) words, regional word variations, and potentially invented pronunciations from language learners. The following are the main methodological improvements

Neural G2P Modeling

The G2P model used is based on Transformer-based architecture, trained on linguistic datasets like CMUdict. It can infer phonetic transcriptions even for phonetically ambiguous or previously unseen words by learning generalizable grapheme-sound patterns. This provides high flexibility and resilience in real-world usage where learners may pronounce unfamiliar or newly coined words.

Handling Non-Standard or Uncommon Words

When the ASR system transcribes an uncommon or incorrectly pronounced word, the backend verifies if the word is valid. If the word is uncommon or incorrectly spelled, the system applies fuzzy string matching to suggest the most probable correct equivalent from a standard vocabulary list. This allows the system to create a reliable phoneme sequence even from error-prone input.

Parallel Phoneme Mapping:

Both the expected word and the recognized word are converted into phoneme sequences using the same G2P logic. This maintains consistency and comparability in

downstream phoneme alignment. The phoneme arrays are then passed into comparison modules for alignment and mismatch detection.

Phoneme-Level Normalization and Stress Marker Retention:

To ensure high-quality phoneme comparison, the system performs phoneme-level normalization that prepares the generated sequences for reliable alignment. Unlike simplistic approaches that discard stress indicators, this system intentionally retains stress markers (e.g., 1 for primary stress, 0 for unstressed) in phonemes like AH0, AE1, etc. These markers are essential in English pronunciation because they influence not just the sound of individual phonemes but also the intonation and rhythmic pattern of words, which are crucial for meaning and naturalness.

For instance, distinguishing between the noun and verb forms of words like record or conduct depends entirely on where the stress is placed. Misplacing stress can lead to misunderstandings, even if the phonemes themselves are correctly articulated. By retaining stress annotations, the system enables suprasegmental error detection, allowing it to give feedback not only on sound substitution, insertion, or deletion, but also on stress misplacement.

Below table 2.1 shows the sample word list with corresponding phoneme sequence.

Table 2:1 Sample English words and their corresponding phoneme sequences

Word	Phoneme Sequence
elephant	EH1 L AH0 F AH0 N T
university	Y UW2 N AH0 V ER1 S AH0 T IY0
butterfly	B AH1 T ER0 F L AY2
memory	M EH1 M ER0 IY0
umbrella	AH0 M B R EH1 L AH0

Support for Dialectal and Learner Variants:

By using dynamic phoneme generation, the system can better interpret user pronunciations that may deviate from standard norms. For instance, users might say "aks" instead of "ask" or "library" instead of "library." Instead of rejecting these, the system translates them into approximate phoneme sequences for comparison.

2.1.8 Phoneme Comparison Algorithm and Alignment Techniques

The phoneme comparison process is one of the core elements of the pronunciation evaluation module, enabling the system to analyze the spoken phonemes against the expected phoneme sequence and identify mispronunciations with precision. This stage builds on the normalized phoneme outputs generated by the G2P model for both the expected word and the transcribed speech. To achieve accurate and interpretable alignment between these sequences. Even in cases of timing, insertions, deletions, or substitutions. This system employs the Fast Dynamic Time Warping (FastDTW) algorithm combined with custom feature-based distance metrics.

Dynamic Time Warping (DTW) for Sequence Alignment

Dynamic Time Warping (DTW) is a time series alignment algorithm that enables the matching of two sequences of unequal length or pace. In pronunciation comparison, this flexibility is essential as users may speak faster or slower than reference pronunciation or may alter phoneme timing. To optimize computational efficiency, the system adopts FastDTW, a linear-complexity approximation of classic DTW that maintains near-optimal accuracy. The DTW algorithm aligns each phoneme in the expected sequence to the best-matching phoneme in the transcribed sequence, creating a path that minimizes the total distance (cost) of transformation from expected to spoke phonemes.

Phoneme-to-Index Mapping for Comparison

Before applying DTW, both the expected and spoken phoneme sequences are encoded into index-based representations, where each unique phoneme is assigned a numerical index. This uniform mapping allows DTW to compute a distance matrix over simple integer sequences instead of raw text.

For example:

- Expected phonemes: ['B', 'AH0', 'N', 'AE1', 'N', 'AH0'] → [1, 2, 3, 4, 3, 2]
- Transcribed phonemes: ['B', 'AA0', 'N', 'AE1', 'AH0'] → [1, 5, 3, 4, 2]

This abstraction also makes it easy to substitute the raw distance function with more sophisticated alternatives, such as articulatory or acoustic feature distances.

Articulatory Feature-Based Distance (PanPhon)

To go beyond index-based symbolic comparison, the system can be extended to use PanPhon, a library that maps IPA phonemes to articulatory feature vectors. Each phoneme is described as a 22-dimensional binary or trinary vector capturing articulatory features such as Voicing, Place of articulation, Manner of articulation, Nasality, Stress presence. Using this, the system can calculate the cosine distance or Hamming distance between phoneme feature vectors instead of symbolic mismatches. This results in more linguistically meaningful alignment and enables gradient feedback (e.g., “close but incorrect”) rather than binary classification.

Mismatch Detection and Error Classification

After alignment, the system iterates through the DTW path and classifies each pair into the following categories.

- Substitution – incorrect phoneme produced (example: B instead of P)
- Insertion – extra phoneme added by speaker
- Deletion – expected phoneme skipped
- Stress Mismatch – correct phoneme spoken but with incorrect stress marker

These are aggregated into structured feedback which highlights error positions, error types, and phonemes requiring practice. The feedback is further enriched using visual highlights in the UI and suggestions for improvement. The system also produces a distance score, representing how close the user's pronunciation is to the expected sequence. This score was used to quantify pronunciation improvement over time, set adaptive thresholds for passing/failing and visualize pronunciation quality progression per user

2.1.9 Error Highlighting and Feedback Generation

Following the phoneme alignment and comparison stage, the system transitions into a feedback generation phase that transforms raw error data into actionable and visually intuitive guidance for learners. This component plays a pivotal role in bridging the gap between computational phoneme analysis and effective language learning by presenting results in a pedagogically meaningful format

Visual Pronunciation Feedback Using Highlighted Word Errors

After comparing the expected and spoken phoneme sequences using DTW, the system reconstructs the word and embeds visual markers at the specific locations of pronunciation mistakes. The incorrect segments are rendered in red to help the user instantly locate where their articulation deviates. This is achieved by mapping the aligned DTW path, Identifying mismatched phoneme indices between expected and transcribed sequences and wrapping the corresponding characters in the word string with HTML/CSS-based annotations. This approach allows learners to immediately locate which part of the word was incorrectly articulated, supporting targeted repetition.

Structured Feedback Breakdown

Each mismatch is converted into a structured feedback object. which includes,

- Expected phoneme vs. spoken phoneme.

- Position index in the phoneme sequence.
- Error type (e.g., substitution, insertion, deletion, stress mismatch).
- Error severity score (based on DTW distance or articulatory feature dissimilarity).
- Timestamp range (from audio segmentation).

Stress Marker Analysis and Feedback

Using stress information embedded in phoneme tags (1, 2, 0), the system can detect prosodic errors, which are especially important for intelligibility. For example, misplacing stress in multi-syllable words often leads to confusion.

- The system checks if the correct syllable is stressed (primary or secondary).
- If not, feedback is provided to emphasize the right syllable.

2.1.10 AI-Powered Personalized Practice Word Generation Based on Mispronounced Phonemes.

One of the distinguishing features of the proposed pronunciation coaching system is its ability to generate individualized word-level practice based on a learner's specific pronunciation errors. This is achieved through the integration of an AI-powered language model (LLM) Google's Gemini which dynamically produces tailored vocabulary lists in response to phoneme-level feedback.

When a user mispronounces a word, the backend component analyzes the discrepancy between the expected phoneme sequence and the transcribed sequence using alignment algorithms like DTW. The resulting list of missed dictionaries is sent as input to a Gemini-powered query. (Sample prompt: "Generate 5 commonly used English words for each of the following phonemes: [list of phonemes]. Return the output in JSON format where each phoneme is mapped to its corresponding word list.")

Gemini processes this prompt using its extensive linguistic knowledge and returns a clean, structured list of relevant, phonetically aligned vocabulary. These practice words are displayed under categorized sections corresponding to each phoneme. To

further reinforce learning, each word is paired with a text-to-speech playback feature using the Web Speech API. This allows learners to hear the correct pronunciation of each suggested word and repeat it aloud. By hearing and mimicking the target phonemes in varied word contexts, learners develop stronger phonological awareness and sound articulation skills.

The following figure 2.4 shows the end-to-end workflow of the phoneme-level pronunciation analysis module.

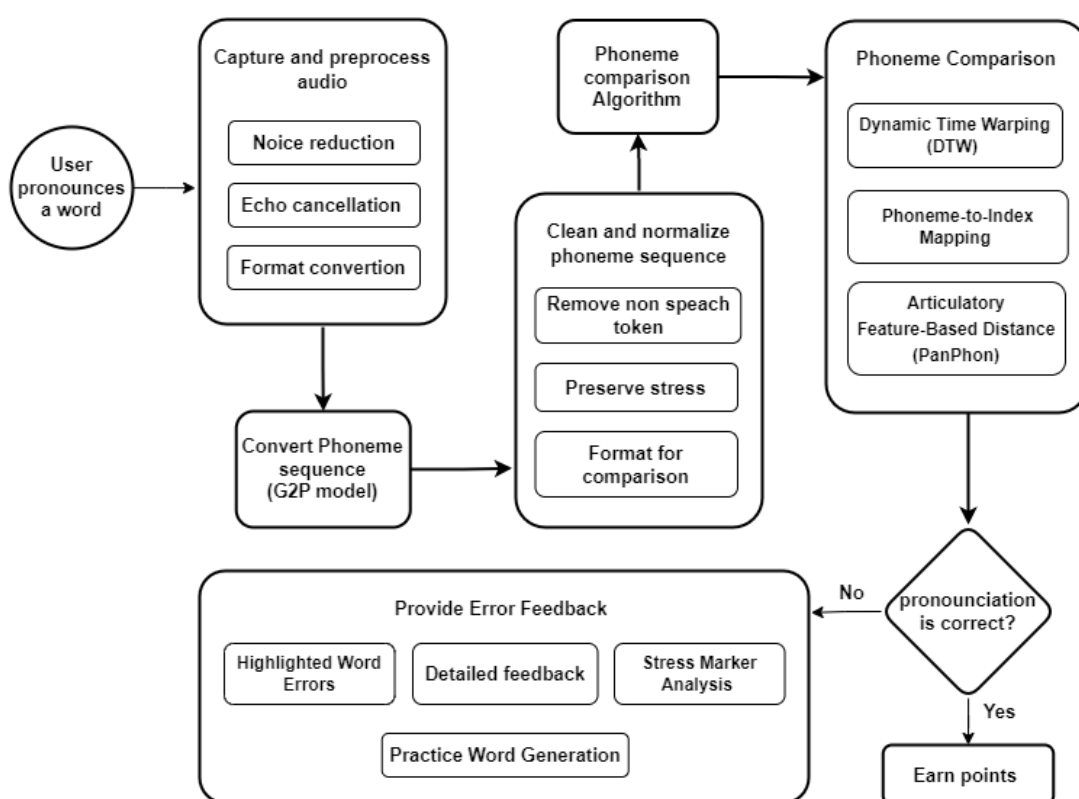


Figure 2.4 Phoneme-Level Pronunciation Analysis and Feedback Workflow

2.2 Commercialization aspects of the product

The proposed phoneme-level pronunciation error detection and feedback system presents strong potential for commercialization, particularly in the growing field of EdTech and AI-powered language learning. As global demand for English language proficiency continues to rise especially in non-native English-speaking learners are increasingly seeking tools that offer personalized, accessible, and intelligent language training without the high costs of human tutoring. This system addresses that demand by providing a scalable, interactive solution that delivers real-time, phoneme-specific feedback, helping users master pronunciation through corrective guidance and targeted practice.

The system is designed to serve a broad and high-demand user base, that includes

- Parents and schools seek guided pronunciation tools for children.
- English tutors and training institutes looking to integrate AI tools into their curriculum.
- School and university students in non-English-speaking countries.
- Adult learners aiming to improve communication skills for work or immigration
- Teachers and language coaches are looking for feedback tools to use with students.
- Professionals in global industries requiring clear spoken English.

The demand is particularly high in countries where English education is widespread, but exposure to native-like pronunciation is limited. In these markets, the product could serve as a cost-effective and self-paced alternative to language institutes.

The system is developed with a non-technical user base in mind. Users do not need to install complex software or have advanced digital skills. The web-based platform offers the following things. A clean, intuitive UI with guided instructions, Instant highlighted feedback, Simple voice recording features, Gamified progress tracking, Access to audio playback and practice words for each mistake.

From a business perspective, the product offers several viable revenue models. These

include a freemium structure where basic features are free, and premium tiers offer advanced tracking, analytics, and unlimited practice access. institutional licensing for schools, language centers, and universities. And direct B2C subscriptions.

Marketing strategies for this component could include targeted social media outreach to students, YouTube tutorial campaigns, partnerships with English language tutors or influencers. And keyword-optimized blog content that positions the product as a smart, practical tool for improving spoken English. In addition, collaboration with English language preparation centers offers a promising channel to reach learners actively seeking pronunciation support. As user data accumulates, the platform can also support learner analytics and long-term progress tracking. Furthermore, the modular backend and AI integration make the system highly scalable, opening pathways for multilingual support or expansion into other skill domains, such as reading fluency or accent training.

In essence, this system is not just a technical solution, but a learner-first product with strong educational and commercial potential. It transforms pronunciation correction into a personalized, engaging, and effective process making it appealing and accessible to a wide audience from casual learners to academic institutions. With its AI capabilities, real-time interaction, and web-based convenience, this product stands well-positioned to serve global learners and evolve into a commercially viable digital language learning tool.

3 Testing & Implementation

3.1 Implementation

The implementation phase of the phoneme level speech error detection module was guided by the goal of providing users with real-time, accurate, and personalized pronunciation feedback. This module forms a critical part of the Readify web application and was developed within a modular, service-oriented architecture to ensure maintainability, scalability, and integration with other components of the system.

To accommodate both technical complexity and usability, the development environment consisted of a React.js frontend enhanced with Tailwind CSS for UI styling, and a FastAPI backend written in Python. The entire system was developed and tested in a dockerized local development environment, enabling consistent deployment across different stages. Real-time database management and user authentication were supported using Firebase, while cloud deployment was handled via AWS ECS (Elastic Container Service) for production use. This section elaborates on the technical implementation of the speech pronunciation feedback system component. The implementation process is structured into several core functional segments, each addressing a specific aspect of the user flow from speech capture to phoneme-based feedback.

3.1.1 Implementation of Audio Capturing and Preprocessing

To initiate recording, the system uses the `navigator.mediaDevices.getUserMedia()`, which prompts the user for microphone access and streams audio input. The stream is configured with specific audio constraints including a sample rate of 44100 Hz, mono channel recording (1 channel), and audio cleanup options like echo cancellation and noise suppression to enhance quality.

```
const stream = await navigator.mediaDevices.getUserMedia({
  audio: {
    sampleRate: 44100,
    channelCount: 1,
    echoCancellation: true,
    noiseSuppression: true
  }
});
streamRef.current = stream;
```

Figure 3.1 Audio capturing process.

This ensures the audio captured is clean and conforms to the expected technical specifications for speech processing. The system then begins recording using the MediaRecorder API. Recorded audio data is stored in chunks and later assembled into a single Blob object for processing.

Browsers may default to recording in formats like WebM, which aren't always compatible with backend speech recognition systems. Therefore, the application implements a conversion routine to ensure audio is in standard wav format a lossless, widely accepted format for speech analysis

```
const convertToWav = async (blob) => {
  return new Promise((resolve) => {
    const audioContext = new (window.AudioContext || window.webkitAudioContext)();
    const fileReader = new FileReader();

    fileReader.onload = async (e) => {
      try {
        const arrayBuffer = e.target.result;
        const audioBuffer = await audioContext.decodeAudioData(arrayBuffer);

        // Convert to WAV
        const wavBuffer = audioBufferToWav(audioBuffer);
        const wavBlob = new Blob([wavBuffer], { type: 'audio/wav' });
        resolve(wavBlob);
      } catch (error) {
        console.log('Using original blob format');
        resolve(blob);
      }
    };

    fileReader.readAsArrayBuffer(blob);
  });
};
```

Figure 3.2 Audio Blob Conversion to WAV Format

Above figure 3.2 method ensures that regardless of the original format, the output is always in a WAV structure suitable for phoneme extraction and accurate speech recognition. The conversion process utilizes the Web Audio API to decode and re-encode the waveform.

Below figure 3.3 function `audioBufferToWav()` builds the WAV file by writing the RIFF (Resource Interchange File Format) header and appending 16-bit PCM samples. This low-level control over the WAV structure ensures compatibility with virtually all downstream audio analysis tools and APIs.

```
const audioBufferToWav = (buffer) => {      You, 2 weeks ago • pronounce-update
  const length = buffer.length;
  const arrayBuffer = new ArrayBuffer(44 + length * 2);
  const view = new DataView(arrayBuffer);

  Windsurf: Refactor | Explain | Generate JSDoc | X
  const writeString = (offset, string) => {
    for (let i = 0; i < string.length; i++) {
      view.setUint8(offset + i, string.charCodeAt(i));
    }
  };

  writeString(0, 'RIFF');
  view.setUint32(4, 36 + length * 2, true);
  writeString(8, 'WAVE');
  writeString(12, 'fmt ');
  view.setUint32(16, 16, true);
  view.setUint16(20, 1, true);
  view.setUint16(22, 1, true);
  view.setUint32(24, 44100, true);
  view.setUint32(28, 44100 * 2, true);
  view.setUint16(32, 2, true);
  view.setUint16(34, 16, true);
  writeString(36, 'data');
  view.setUint32(40, length, true);

  const channelData = buffer.getChannelData(0);
  let offset = 44;
  for (let i = 0; i < length; i++) {
    const sample = Math.max(-1, Math.min(1, channelData[i]));
    view.setInt16(offset, sample < 0 ? sample * 0x8000 : sample * 0x7FFF, true);
    offset += 2;
  }

  return arrayBuffer;
};
```

Figure 3.3 Manually Constructing a WAV Header

This step finalizes the WAV file format using correct metadata. It adds RIFF tags, sample rate, bit depth, and PCM data to ensure the file is properly structured. This precision supports consistency across various browsers and backend APIs.

3.1.2 Implementation of Speech Recognition and Phoneme Analysis

```
def transcribe_audio_azure(file_path):
    audio_config = speechsdk.audio.AudioConfig(filename=file_path)
    recognizer = speechsdk.SpeechRecognizer(speech_config=speech_config, audio_config=audio_config)
    result = recognizer.recognize_once()
    return result.text.lower()
```

Figure 3.4 Audio transcription using Microsoft Azure

Above figure 3.4 `audio_config` parameter links the saved audio file, and the recognizer component processes it to return the transcribed string. The result is normalized to lowercase to simplify further comparison operations. Using Azure ensures high accuracy and real-time transcription capabilities, even under diverse accents.

```
def get_phonemes_g2p(word):
    phonemes = g2p(word)
    return [p for p in phonemes if p != ' ']
```

Figure 3.5 Phoneme Extraction Using G2P Conversion

After completing the phoneme extraction logic figure 3.5 shown function uses the `g2p_en` model to convert any word whether standard or non-standard into its constituent phonemes. This phoneme sequence is later used for precise comparison. The G2P model is crucial in handling out-of-vocabulary words, user-defined inputs, or mispronunciations, supporting robustness in learner interaction.

```
def flatten_phonemes(phonemes):
    """Flatten the phoneme list if it is nested."""
    if isinstance(phonemes[0], list): # Check if the phonemes are nested
        return [item for sublist in phonemes for item in sublist] # Flatten the list
    return phonemes
```

Figure 3.6 Making DTW more consistent

```
def phoneme_alignment_dtw(expected_phonemes, transcribed_phonemes):
    phoneme_dict = {phoneme: idx for idx, phoneme in enumerate(set(expected_phonemes + transcribed_phonemes))}
    expected_indices = [phoneme_dict[phoneme] for phoneme in expected_phonemes]
    transcribed_indices = [phoneme_dict[phoneme] for phoneme in transcribed_phonemes]
    distance, path = fastdtw(expected_indices, transcribed_indices)
    return distance, path, expected_phonemes, transcribed_phonemes
```

Figure 3.7 Phoneme Sequence Comparison Using DTW

While implementing phoneme comparison algorithm Here figure 3.7 DTW (Dynamic Time Warping) is used to align phoneme sequences and calculate the similarity distance. This allows tolerance for differences in speech rate and fluency. The returned distance helps determine pronunciation accuracy, while the path maps which phonemes matched or misaligned.

```
def highlight_word(expected_word, path, expected_phonemes, transcribed_phonemes):
    highlighted_word = list(expected_word)
    mismatches = []
    for (i, j) in path:
        if i < len(expected_phonemes) and j < len(transcribed_phonemes):
            if expected_phonemes[i] != transcribed_phonemes[j]:
                if j < len(highlighted_word):
                    highlighted_word[j] = f"<span class='text-red-500'>{highlighted_word[j]}</span>"
                mismatches.append((i, expected_phonemes[i], transcribed_phonemes[j]))
    return "".join(highlighted_word), mismatches
```

Figure 3.8 Mismatch Detection and Feedback Highlighting

This figure 3.8 function uses the phoneme comparison algorithm output and DTW path to visually highlight mismatched phoneme positions in the expected word. By wrapping incorrect letters in red, it offers intuitive feedback. The mismatches list is also used to generate structured error feedback in the response.

```
{
  "result": "incorrect",
  "transcribed_word": transcribed_word,
  "expected_word": expected_word,
  "full_transcription": transcribed_text,
  "highlighted_word": highlighted,
  "Expected Phonemes": expected_phonemes,
  "Transcribed Phonemes": transcribed_phonemes,
  "missed_phonemes": missed_phonemes,
  "phoneme_feedback": [
    {
      "position": i,
      "expected": expected,
      "transcribed": transcribed
    }
    for i, expected, transcribed in mismatches
  ],
  "distance": distance
}
```

Figure 3.9 Mismatch Detection and Feedback Highlighting

```
@app.post("/compare")
async def compare_audio(audio: UploadFile = File(...), expected_word: str = Form(...)):
```

Figure 3.10 Backend API Integration Endpoint

The figure 3.10, “/compare” endpoint accepts an audio file and the expected word. It transcribes the audio, converts both expected and transcribed words into phonemes, aligns them, detects mismatches, and sends back structured feedback highlighted text, error positions, and practice recommendations.

3.1.3 Implementation of Similar Sounding Word Generation

```
const generateSimilarWordsWithGemini = async (phonemes) => {
  setIsLoadingSimilarWords(true);
  try {
    const phonemeList = phonemes.join(', ');
    const prompt = `Generate 5 simple English words for each of these phonemes/sounds: ${phonemeList}.

    For each phoneme, provide exactly 5 common English words that contain that specific sound.
    Format your response as a JSON object where each phoneme is a key and the value is an array of 5 words.

    Example format:
    {
      "AH": ["about", "cup", "love", "come", "done"],
      "P": ["pat", "pop", "paper", "apple", "happy"]
    }

    Only return the JSON object, no additional text.`;

    const requestBody = {
      contents: [{
        parts: [{ text: prompt }]
      }]
    };

    const response = await fetch(GEMINI_API_URL, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify(requestBody)
    });

    if (!response.ok) {
      throw new Error(`HTTP error! status: ${response.status}`);
    }

    const data = await response.json();
    const generatedText = data.candidates[0].content.parts[0].text;

    // Try to parse the JSON response
    try {
      const cleanedText = generatedText.replace(/```json\n?|\n?```/g, '').trim();
      const parsedWords = JSON.parse(cleanedText);
      setGeneratedSimilarWords(parsedWords);
    } catch (parseError) {
      console.error('Error parsing Gemini response:', parseError);
      // Fallback to default similar words
      const fallbackWords = {};
      phonemes.forEach(phoneme => {
        fallbackWords[phoneme] = similarWords[phoneme] || ['practice', 'more', 'words', 'with', 'sound'];
      });
      setGeneratedSimilarWords(fallbackWords);
    }
  }
}
```

```

    } finally {
      setIsLoadingSimilarWords(false);
    }
  };

  // Modified function to handle showing similar words
  Windsurf: Refactor | Explain | X
  const handleShowSimilarWords = async () => {
    if (!showSimilarWords && result?.missed_phonemes) {
      // If we're about to show similar words and don't have them generated yet
      if (Object.keys(generatedSimilarWords).length === 0) {
        await generateSimilarWordsWithGemini(result.missed_phonemes);
      }
    }
    setShowSimilarWords(!showSimilarWords);
  };

```

Figure 3.11 Similar Sounding Word Generation (Using Gemini LLM)

Figure 3.11 shows the key to this functionality lies in leveraging Gemini AI (Google's LLM) to dynamically generate phoneme-specific practice words. Instead of relying on a static dictionary, the application sends a structured prompt to Gemini with a list of mispronounced phonemes and requests a JSON-formatted response containing 5 relevant English words for each. The prompt is carefully designed using a few shot examples and strict formatting instructions to ensure that the model returns the output in a machine-readable JSON object. This is essential for easy parsing and integration on the frontend.

3.1.4 Front-End Implementation

The frontend of the pronunciation training system is built using React.js with Vite for efficient rendering and fast development. It leverages Tailwind CSS to create a responsive and clean user interface. This part of the system is responsible for interacting with users, allowing them to record audio, view pronunciation feedback, track their progress, and receive suggestions. The frontend communicates with backend services via API calls and is designed to be user-friendly, ensuring accessibility even for those with minimal technical background.

- **Implement the Pronunciation Starter Guide Page**

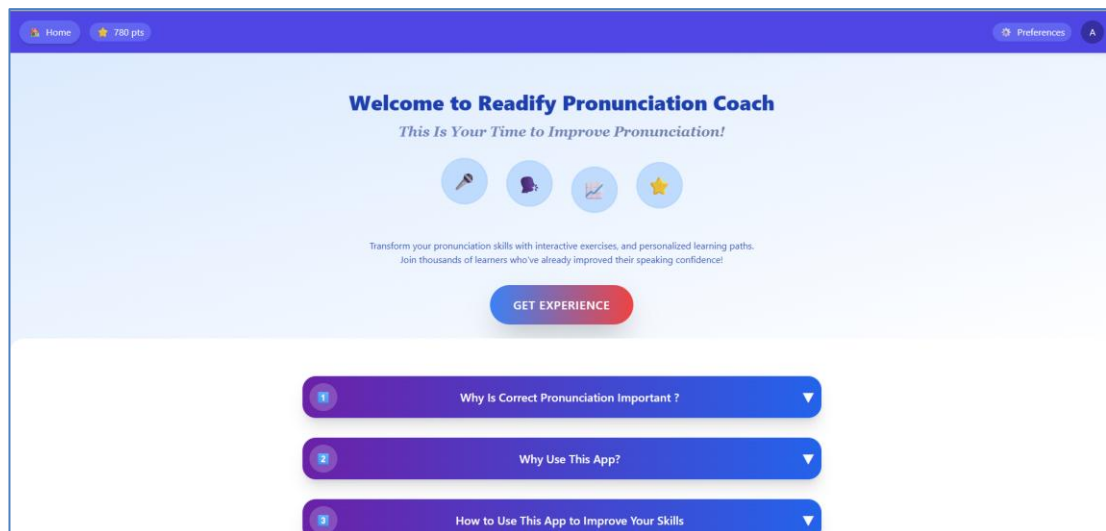


Figure 3.12 Pronunciation Starter Guide

Figure 3.12 provides practical tips on speaking clearly, selecting a quiet environment, and adjusting microphone input ensuring the system captures quality audio for accurate feedback. By simplifying key system features and providing friendly guidance, this section helps users, especially non-technical learners, feel confident and informed as they begin their pronunciation improvement journey.

- **Implement the Pronunciation Start Page**

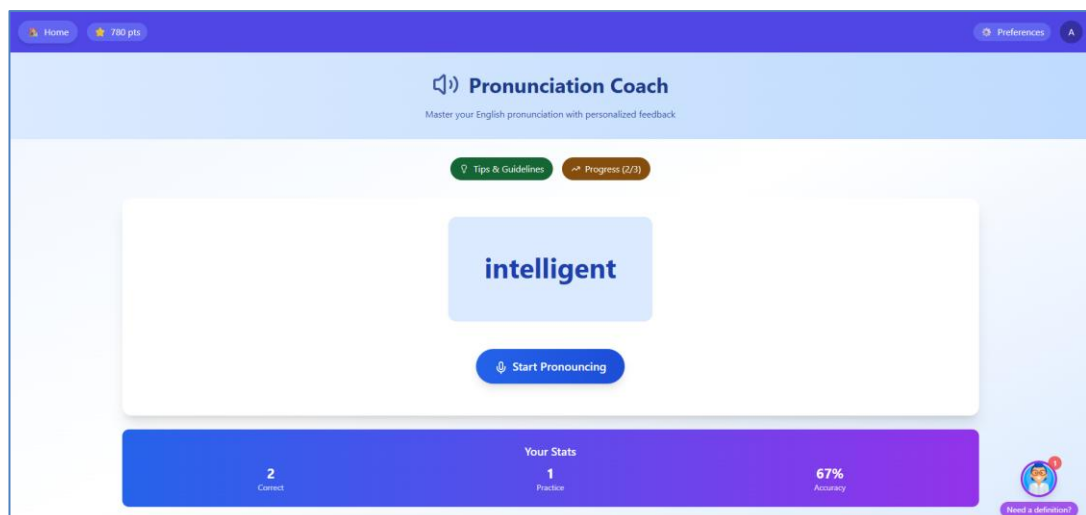


Figure 3.13 The Pronunciation Start Page



Figure 3.14 The Pronunciation Start Page

Figure 3.13 and figure 3.14 show the Pronunciation Start Page is where users begin their pronunciation practice journey. A word is dynamically generated through a Large Language Model (LLM) based on the user’s proficiency level, which is retrieved from the Firebase database. This ensures that the content is appropriately tailored and challenging for each individual user. After the word appears, a short countdown timer prepares the user to speak. The system then records and analyzes their pronunciation in real time. Additionally, a Progress Section visually tracks the user’s correct and incorrect attempts, offering insight into their learning progress and encouraging continued improvement.

▪ Implement the Pronunciation Feedback display

The pronunciation feedback generation is handled reactively using React hooks and dynamically updated component state. Once the user finishes recording their pronunciation, the audio is processed and sent via an HTTP POST request to a FastAPI backend. This is managed through the `checkPronunciation()` function, which constructs a `FormData` object including the user's audio and expected word, then sends it to the “/compare” endpoint. Upon receiving the backend response, the result is stored in the result state using `setResult(data)`. This triggers a re-render of the UI and conditionally displays feedback based on whether the pronunciation was correct or not. If correct, a success message and positive UI cues such as score update. If incorrect, a

comprehensive feedback section is rendered.

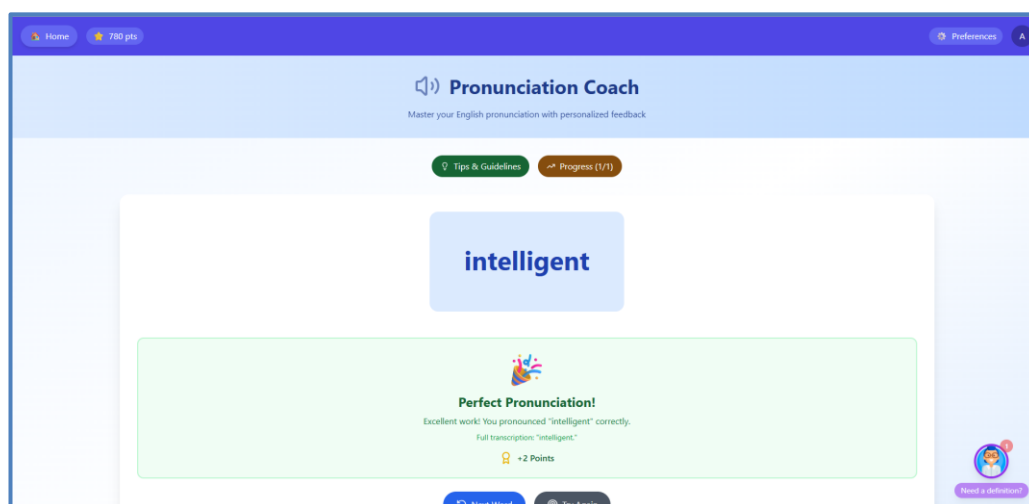


Figure 3.15 Pronunciation correct feedback.

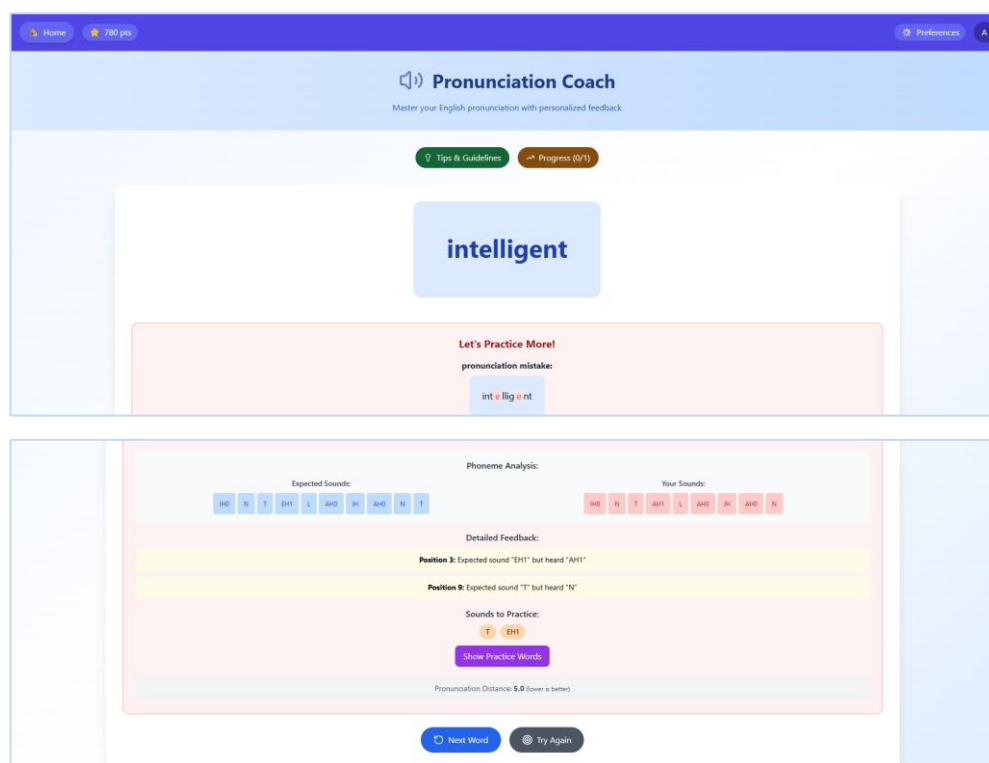


Figure 3.16 Pronunciation incorrect feedback

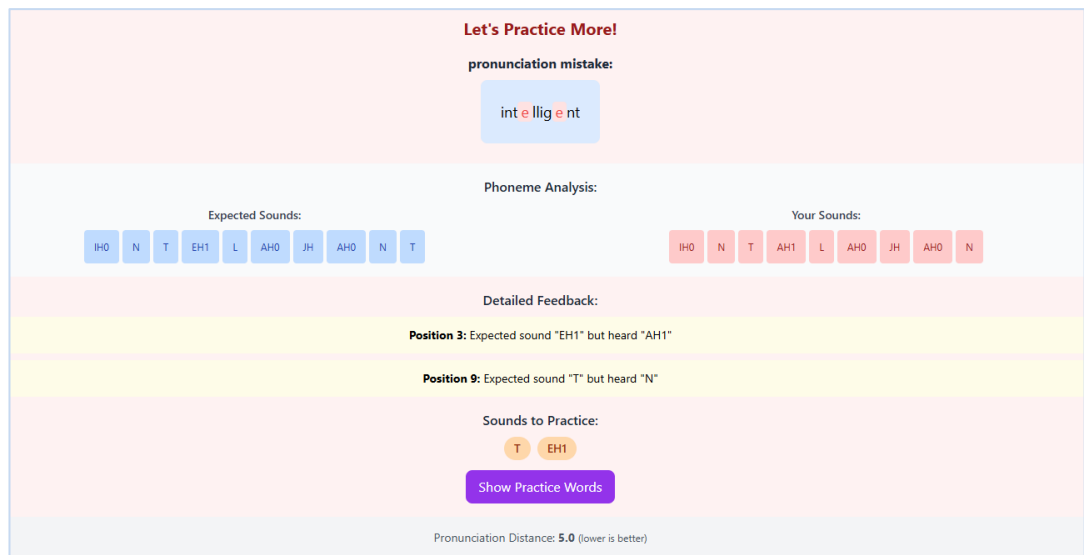


Figure 3.17 Pronunciation incorrect feedback

Figure 3.15 illustrates the result when the user correctly pronounces the given word. In this case, the system recognizes the pronunciation as accurate, awards points to the user, and stores this progress in the database using the Context API for user-specific tracking. If the pronunciation is incorrect, the system displays a result like Figure 3.16. For example, as shown in Figure 3.17, the displayed word was "intelligent", but the user pronounced it as "intahllijant". Based on this discrepancy, the system provides targeted feedback highlighting the mispronounced phonemes.

▪ Implement the Pronunciation Feedback display

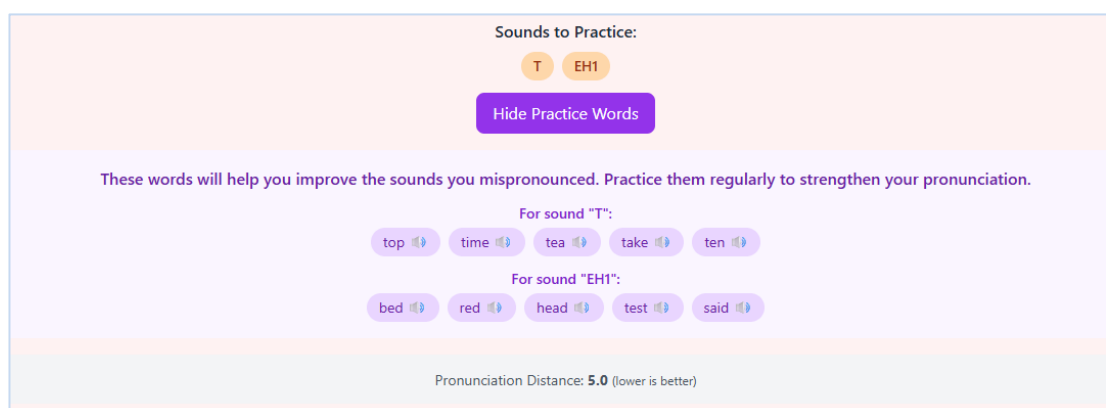


Figure 3.18 Similar word generation based on the user's mispronounced phonemes.

Figure 3.18 demonstrates the functionality of similar word generation based on the user's mispronounced phonemes. After analyzing the pronunciation errors, the system identifies the specific phonemes that were incorrectly articulated. These phonemes are then used to query the Gemini AI model through a prompt designed to retrieve five simple and commonly used English words that share similar phonetic components. The output from Gemini is returned in JSON format (as a stringified response), where each mispronounced phoneme serves as a key, and the value is an array of related practice words. The system parses this response and dynamically displays the suggested words to the user. Each generated word is rendered with a text-to-speech button, allowing the user to listen and familiarize themselves with the correct pronunciation.

3.2 Testing

The testing phase plays a vital role in ensuring the reliability, correctness, and performance of the proposed phoneme-level pronunciation feedback system. Given that this component directly handles user audio input, speech recognition, phoneme comparison, and phoneme feedback. It is crucial to rigorously validate each module's behavior under a variety of real-world conditions. This system includes both frontend and backend components and integrates external services such as Azure Cognitive Speech Services and Gemini AI. Therefore, the testing approach must cover end-to-end scenarios, including audio capturing, file uploading, backend response validation, phoneme-level mismatch detection, and personalized feedback generation.

3.2.1 Test Plan and Test Strategy

To ensure a comprehensive evaluation, the testing process adopted a combination of unit testing, integration testing, and manual exploration testing, covering all the critical functionalities of the system. The testing objectives include verifying correct audio capture, reliable API communication, accurate phoneme comparison, and contextual feedback display.

The test strategy identifies below key objective:

- Validate correctness of phoneme extraction and alignment

- Ensure accurate recognition of mispronunciations
- Verify real-time feedback generation is contextually appropriate
- Evaluate UI responsiveness and UX quality
- Confirm Gemini word generation logic produces relevant results

Testing Types Covered,

- Unit Tests: Focused on individual logic blocks like phoneme mapping, JSON parsing, and word comparison functions.
- Integration Tests: Validated full workflows such as audio upload → phoneme analysis → feedback display.
- Manual UI Tests: Confirm button triggers, checked recording triggers, feedback visibility, and pronunciation replay features.
- Stress Tests: Use multiple audio inputs with varying environments.
- Cross-Platform Tests: Run across browsers and devices to ensure accessibility and consistent rendering.
- Test Group: 20 participants with varied English proficiency levels.

3.2.2 Test Case Design

Structured test cases were designed to target each component and expected scenario. Below is a representation,

Table 3:1 Test Case to Record button triggers

Test Case ID	TC-001
Scenario	Record button triggers countdown
Input	Click on "Start Pronouncing"
Expected Output	Countdown begins from 3
Status	Pass

Table 3:2 Test Case to audio file recording function

Test Case ID	TC-002
Scenario	Audio file successfully recorded and saved

Input	Speak during 4-second window
Expected Output	Audio blob created
Status	Pass

Table 3:3 Test Case to input voiceless audio

Test Case ID	TC-002
Scenario	Check system behavior when user submits silent input
Input	User presses the record but remains silent, submits background noise only
Expected Output	System responds with an error alert: “No words transcribed from the audio.”
Status	Pass

Table 3:4 Test Case to valid pronunciation

Test Case ID	TC-003
Scenario	Backend accepts audio & returns phoneme comparison
Input	Upload recorded audio with valid word
Expected Output	JSON with phoneme arrays and feedback
Actual Output	Correct response generated
Status	Pass

Table 3:5 Test Case to incorrect pronounce

Test Case ID	TC-004
Scenario	Incorrect pronunciation detected accurately
Input	Say invalid input as a pronounce
Expected Output	Feedback highlights wrong phonemes
Status	Pass

Table 3:6 Test Case to practice word generation

Test Case ID	TC-005
Scenario	Gemini generates correct practice words
Input	Missed phoneme = “AH”, “EH1”
Expected Output	AI returns 5 related practice words accurately.
Actual Output	AH - cup, luck, mud, done, come EH1 – bed, head, red, said, dead
Status	Pass

Table 3:7 Test Case to listen generated sound

Test Case ID	TC-06
Scenario	Speech synthesis plays similar word
Input	Click speaker next to word
Expected Output	Audio plays corresponding sound
Status	Pass

Table 3:8 Test Case to failure of Gemini Ai

Test Case ID	TC-07
Scenario	Fallback triggered on Gemini error
Input	Internet disabled or malformed response
Expected Output	Show try again message as an alert.
Status	Pass

4 RESULTS AND DISCUSSIONS

4.1 Results

The phoneme-level speech error detection system developed in this project aims to provide accurate, real-time feedback on pronunciation mistakes made by users. To evaluate the effectiveness of the system, a series of manual and system-generated tests were conducted focusing on the correctness of transcription, phoneme extraction, comparison accuracy, and feedback generation. The evaluation process involved simulating different pronunciation patterns including correct, near correct, and clearly mispronounced forms and observing how the system responded to each input.

The system was then evaluated on its ability to:

- Correctly transcribe the spoken word,
- Convert both the expected and spoken words to phoneme sequences,
- Compare these phoneme sequences using a DTW-based alignment algorithm,
- Identify mismatches and generate relevant feedback and accurate similar sound words.

To validate the accuracy of this approach, a manual testing process was carried out with a group of selected users representing different proficiency levels. Each user was prompted with a list of target words to pronounce, and the system-generated feedback was compared against expected results. The comparison included checking Whether the system correctly transcribed the spoken word. Whether the expected and transcribed phoneme sequences were accurately generated. Whether the system highlighted the correct mispronounced phonemes. Whether the feedback, including similar sounding practice words, was relevant and useful

To validate the system's performance, we conducted a structured pronunciation test using a sample group of English-proficient speakers. Participants were instructed to pronounce a curated list of words. The system then evaluated each utterance, and the percentage of correctly identified pronunciations was recorded.

This evaluation method allowed us to test not just the backend accuracy of speech transcription and phoneme analysis, but also the alignment between human

pronunciation and system expectations. The results demonstrated the system’s high capability in recognizing correct pronunciation and distinguishing subtle phoneme errors.

Table 4:1 Accuracy Analysis Table

Sample Word List	Correct Pronunciation Detection Rate
entrepreneur	98%
ambiguity	96%
phenomenon	94%
necessarily	98%
sophisticated	96%

These table 4.1 results reflect the high precision of the phoneme alignment and recognition algorithms. The system consistently identified correctly pronounced words with an accuracy rate above 94%, indicating a strong correlation between expected and detected phoneme sequences. The analysis also involved calculating the overall system accuracy across the test set using the formula:

$$\text{Accuracy (\%)} = (\text{Number of correct identifications} / \text{Total attempts}) \times 100$$

This practical validation shows that the system can deliver reliable feedback and can be confidently used as a supportive tool in pronunciation training. The consistent performance across different users and words suggests robustness in real-world learning environments.

To evaluate the effectiveness of the system under practical user conditions, we conducted a test with a group of student-level English learners. Each student was asked to pronounce the same five challenging words. After two metrics were measured,

- Speech Accuracy: how accurately users pronounce each word.
- Feedback Accuracy: how correctly the system identified pronunciation mistakes and delivered feedback (highlighting errors, phoneme mismatches, and suggestions).

Table 4:2 Evaluation with Student-Level English Learners

word	Speech accuracy of group	Feedback accuracy
entrepreneur	43.33%	90%
ambiguity	36.66%	93.33%
phenomenon	46.66%	90%
necessarily	53.33%	96.66%
sophisticated	40%	96.66%

These results highlight two key insights:

- **Practical Demand:** The relatively low speech accuracy confirms the widespread difficulty learners face when pronouncing complex English words, validating the need for an intelligent coaching tool.
- **System Reliability:** The high feedback accuracy ($\geq 90\%$) demonstrates the robustness of the system in identifying pronunciation errors even when users make significant speech mistakes. The reliable highlighting of mispronounced phonemes and the generation of relevant practice suggestions ensure that users receive actionable guidance.

Additionally, the phoneme-level breakdown of errors gave learners actionable insights into where their pronunciation diverged from the expected norms. For instance, a student mispronouncing the correct word as incorrect word was provided with clear visual cues highlighting the specific phonemes that were incorrectly pronounced, along with suggested corrections and similar-sounding practice words. This individualized approach to feedback significantly enhances the learner's ability to focus on and improve specific weaknesses. To further verify usability and system comprehension, we tracked real-time interactions using the frontend progress indicators. These metrics provided a seamless visualization of user improvement over time, reinforcing motivation through gamified scoring and word accuracy feedback.

4.2 Research Findings

The research conducted throughout this project has yielded significant findings in the field of automated pronunciation assessment particularly at the phoneme level. By integrating advanced speech processing, grapheme-to-phoneme (G2P) modeling, and intelligent feedback generation the system demonstrated both technical efficacy and educational relevance. The following summarizes the key findings based on experimental results and user evaluations.

Phoneme-Level Analysis Improves Feedback Precision

Traditional pronunciation assessment tools typically rely on word-level correctness or simple binary scoring systems. In contrast, this system performs phoneme-level error detection using DTW alignment algorithms and G2P-based phoneme sequence generation. This allows it to identify exact locations and types of pronunciation mistakes enabling learners to focus on individual sound corrections, not just entire words.

G2P Flexibility Enables Broad Vocabulary Support

The integration of a Transformer-based G2P model enabled dynamic phoneme generation for both standard and out-of-vocabulary (OOV) words. This was especially useful in accommodating learner-specific pronunciations or creative mispronunciations, which often deviate from canonical English forms.

This research finding is significant because it demonstrates that the system can function effectively without relying solely on fixed pronunciation dictionaries a known limitation in many commercial solutions. It reinforces the model's capability to generalize phoneme predictions even for uncommon, invented, or accented inputs, thereby increasing the inclusiveness and robustness of the tool.

Speech-to-Phoneme Pipeline Maintains Accuracy in Variable Conditions

This robustness was further supported by pre-processing techniques such as echo cancellation, noise suppression, and signal normalization, ensuring that phoneme extraction remained stable across diverse environments. This robustness is essential in practical settings especially in schools, at home, or mobile learning scenarios where audio conditions are often suboptimal. The findings affirm that the backend design supports real-time, noise-resilient speech capture and processing.

Learner Engagement Is Enhanced Through Personalized Feedback

One of the standout features is the generation of similar-sounding practice words based on mispronounced phonemes. Using Gemini LLM, the system crafts contextual vocabulary tailored to the user's mistake, turning feedback into a focused practice opportunity. The ability to click and hear each suggested word makes the learning experience interactive and engaging.

These research findings confirm the effectiveness of combining deep linguistic modeling, AI-based feedback generation, and modern front-end-backend technologies. The system not only detects phoneme-level pronunciation issues with high precision but also translates them into meaningful feedback, driving both self-improvement and learner engagement.

4.3 Discussion

The development and evaluation of the phoneme-level pronunciation feedback system have revealed critical insights into both its technical strengths and its pedagogical significance. This section discusses the implications of the results, how the system addresses key challenges in pronunciation training, and areas where it can be further improved or extended.

Value of Phoneme-Level Feedback

One of the most significant contributions of the system is its ability to provide pronunciation evaluation not just at the word level but at the phoneme level. Unlike conventional language learning tools that only tell the user if they were correct or not, this system identifies exactly which sounds were mispronounced and where in the word they occurred. This level of specificity is crucial for effective language learning, particularly for learners struggling with certain sound patterns.

This helps learners internalize the difference between what they intended to say and what they produce, making feedback more actionable.

System Performance Across Proficiency Levels

The system was evaluated in two phases: using proficient English speakers and student-level learners. The results from proficient speakers, with detection rates consistently above 94% (related to table 4.1), validated the technical robustness of the backend components,

- Azure Cognitive Services provided reliable transcription.
- The Grapheme-to-Phoneme model handled both standard and out-of-vocabulary (OOV) words effectively.
- The Dynamic Time Warping (DTW) algorithm accurately identified phoneme-level mismatches.

In the second phase, student-level users were evaluated using the same test words. Although speech accuracy was lower (ranging from ~36% to ~53%), the feedback accuracy remained high, often above 93% (related to 4.2 table). This contrast proves that while learners struggled, the system consistently provided useful and precise guidance on their mistakes.

Frontend Experience and User Engagement

The frontend, built using React.js and styled with Tailwind CSS, offers a responsive and intuitive interface. The user can interact with the system in real-time: pronounce

words, receive visual and audio feedback, and view highlighted mispronunciations. The state management system efficiently handles transitions between recording, processing, and feedback views, creating a smooth user experience. More importantly, even non-technical users can operate the system without training. This supports the system's use in educational institutions, home learning environments, and mobile platforms.

Educational Value and User Feedback

During trials, users indicated that the immediate phoneme-level feedback, combined with visual highlights and audio replays, significantly improved their awareness of pronunciation errors. Many reported that they had never received such specific guidance from traditional learning methods or teachers. Moreover, tracking progress with metrics such as pronunciation score and correct vs. incorrect word ratio helped maintain learner motivation especially when combined with a point-based reward system integrated into Firebase.

Challenges and Future Improvements

Despite the promising results, certain limitations were identified:

- **Accent handling:** Users with regional accents occasionally received lower accuracy scores due to mismatches in phoneme expectation. Incorporating accent-specific ASR models may help address this.
- **Sentence-level pronunciation:** The current system is optimized for single-word feedback. Extending to phrases or sentences would allow a richer analysis of rhythm, stress, and intonation.
- **User feedback loop:** Adding features where users can confirm or reject feedback will help further refine the system's intelligence through reinforcement learning.

As users interact with the system in real-world scenarios, whether in academic, self-learning, or professional environments their experiences and feedback become crucial

to the system's continued refinement. During preliminary testing phases, users provided valuable suggestions such as incorporating support for sentence-level pronunciation, adding localized accent calibration, and expanding the practice word set for each phoneme. These insights will inform the next stage of development, guiding enhancements to the feedback engine, UI/UX improvements, and deeper personalization features. By continuously gathering user feedback and analyzing behavioral usage data, we aim to evolve the system into a more adaptive, intelligent, and inclusive learning platform that can serve a broader and more diverse user base.

5 CONCLUSION

The phoneme-level speech error detection system developed in this project addresses a growing need for intelligent, personalized pronunciation training tools especially for learners of English as a second language. Through the integration of advanced technologies such as Azure Speech Services, a Transformer-based G2P model, and Dynamic Time Warping (DTW) phoneme alignment algorithms, the system enables users to receive precise and context-aware feedback on their spoken English.

The core strength of this solution lies in its ability to compare expected and actual pronunciation at a granular, phoneme level. Unlike generic speech recognition tools, this system isolates specific pronunciation mistakes, highlights errors within the word and recommends corrective practice using AI-generated similar-sounding words. These features support self-directed learning and allow users to refine their speech patterns iteratively.

The implementation framework comprising a FastAPI-based backend, a responsive React.js frontend with Tailwind CSS, and Firebase for user state management ensures scalability, maintainability, and real-time interactivity. This modular, microservice-oriented architecture also supports future enhancements without disrupting the overall system.

The results gathered from both technically proficient speakers and student-level users showed strong performance, with high feedback accuracy across different test words. User feedback indicated that the tool was intuitive, engaging, and informative, making it an asset for pronunciation improvement regardless of the user's initial proficiency level.

In summary, this system not only provides effective pronunciation assistance but also demonstrates the practical application of AI in language education. It bridges the gap between machine accuracy and human learning by offering tailored guidance that evolves with the user's progress. As English proficiency becomes more essential in academic, professional, and global communication contexts, tools like this will be key to empowering learners with accessible, adaptive, and intelligent support.

Future versions of the system can explore more sophisticated error classifications, stress and intonation feedback, multilingual support, and broader integration into learning management systems. By continuing to gather user data and suggestions, the system can be improved and expanded to meet even more diverse learning needs.

6 REFERENCES

- [1] A. Noiray, K. Iskarous, and D. H. Whalen, "Variability in English vowels is comparable in articulation and acoustics," *Lab. Phonol. J. Assoc. Lab. Phonol.*, vol. 5, no. 2, Jan. 2014, doi: 10.1515/lp-2014-0010.
- [2] Peter, "AE 466 – Ship or Sheep? | English Pronunciation of /i:/ vs /ɪ/," *Aussie English*, Aug. 25, 2022. [Online]. Available: <https://aussieenglish.com.au/ae-466-ship-or-sheep-english-pronunciation-of-i-vs-%C9%AA/>
- [3] M. Hismanoglu and S. Hismanoglu, "Language teachers' preferences of pronunciation teaching techniques: traditional or modern?," *Procedia - Social and Behavioral Sciences*, vol. 2, no. 2, pp. 983–989, Jan. 2010, doi: 10.1016/j.sbspro.2010.03.138.
- [4] N. S. Jayasundara and A. H. A. Farook, "Difficulties encountered by English as a second language learners in using stress and intonation," *Int. J. Innov. Sci. Eng. Technol.*, vol. 7, no. 9, pp. 86–87, 2020. [Online]. Available: https://ijiset.com/vol7/v7s9/IJISSET_V7_I9_10.pdf
- [5] V. Cook, *Second Language Learning and Language Teaching*, 4th ed. Routledge, 2016. doi: 10.4324/9781315883113.
- [6] S. Gandhioke and C. Singh, "Learner Awareness of the 'Music' of Spoken English—Focus on Intonation—And Its Impact on Communicative Competence," *Creative Educ.*, vol. 14, pp. 454–468, 2023, doi: 10.4236/ce.2023.143031.
- [7] Y. Lan and M. Wu, "Application of Form-Focused Instruction in English Pronunciation: Examples from Mandarin Learners," *Creative Educ.*, vol. 4, pp. 29–34, 2013, doi: 10.4236/ce.2013.49B007.
- [8] V. Cook, *Second Language Learning and Language Teaching*, 4th ed. Routledge, 2016. doi: 10.4324/9781315883113.
- [9] M. Hismanoglu and S. Hismanoglu, "Language teachers' preferences of pronunciation teaching techniques: traditional or modern?," *Procedia - Social and Behavioral Sciences*, vol. 2, no. 2, pp. 983–989, 2010, doi: 10.1016/j.sbspro.2010.03.138.
- [10] K. Alshalaan, "A Comparison between English and Arabic Sound Systems Regarding Places of Articulation," *OALib*, vol. 7, no. 5, pp. 1–7, 2020, doi: 10.4236/oalib.1105679.
- [11] P. M. Rogerson-Revell, "Computer-Assisted Pronunciation Training (CAPT): Current Issues and Future Directions," *RELC J.*, vol. 52, no. 1, pp. 189–205, 2021, doi: 10.1177/0033688220977406.
- [12] D. Korzekwa, J. Lorenzo-Trueba, T. Drugman, and B. Kostek, "Computer-assisted pronunciation training—Speech synthesis is almost all you need," *Speech Commun.*, vol. 142, pp. 22–33, Jun. 2022, doi: 10.1016/j.specom.2022.06.003.

- [13] S. Coulange, "Computer-aided pronunciation training in 2022: When pedagogy struggles to catch up," *HAL (Le Centre Pour La Communication Scientifique Directe)*, Jan. 2023, doi: 10.5281/zenodo.8137754.
- [14] J. Levis, "L2 pronunciation research and teaching," *J. Second Lang. Pronunciation*, vol. 7, no. 2, pp. 141–153, 2021, doi: 10.1075/jslp.21037.lev.
- [15] D. Liakin, W. Cardoso, and N. Liakina, "Learning L2 pronunciation with a mobile speech recognizer: French /y/," *CALICO J.*, vol. 32, no. 1, pp. 1–25, 2014, doi: 10.1558/cj.v32i1.25962.
- [16] A. Neri, C. Cucchiaroni, and H. Strik, "Selecting segmental errors in non-native Dutch for optimal pronunciation training," *Int. Rev. Appl. Linguist. Lang. Teach.*, vol. 44, no. 4, pp. 357–404, 2006.
- [17] R. Ai, "Automatic Pronunciation Error Detection and Feedback Generation for CALL Applications," in *Lecture Notes in Computer Science*, vol. 9288, pp. 175–186, 2015, doi: 10.1007/978-3-319-20609-7_17.
- [18] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 26, no. 1, pp. 43–49, Feb. 1978.
- [19] A. Neri, C. Cucchiaroni, and H. Strik, "Automatic speech recognition for second language learning: How and why it actually works," *Proc. Int. Conf. Speech Lang. Technol. Educ.*, pp. 34–37, 2006.
- [20] J. Mortensen, J. D. Dautenhahn, and M. Littell, "PanPhon: A resource for mapping IPA segments to articulatory feature vectors," *Proc. COLING 2016*, pp. 3475–3484, 2016.
- [21] S. Nita, E. R. N. Sari, K. Sussolaikah, and S. M. F. Risky, "The Implementation of Duolingo Application to Enhance English Learning for Millennials," *J. Int. Lingua Technol.*, vol. 2, no. 1, pp. 1–9, Jun. 2023, doi: 10.55849/jiltech.v2i1.215.
- [22] R. N. Hermana, "Rosetta Stone Application on Students' Pronunciation," *J. English Teach. Linguist. Stud. (JET Li)*, vol. 5, no. 2, pp. 92–102, Oct. 2023, doi: 10.55215/jetli.v5i2.8779.
- [23] E. Miller, "BoldVoice Review: Despite its usefulness, it has a BIG flaw," *Medium*, Mar. 05, 2024. [Online]. Available: <https://medium.com/@emmamillerw1990/boildvoice-review-despite-its-usefulness-it-has-a-big-flaw-33c944b1150c>
- [24] "LinguaCoach an English Phonetics Approach The best place to master English pronunciation." [Online]. Available: <https://linguacoach.appspot.com>
- [25] K. Kyriakopoulos, K. M. Knill, and M. J. F. Gales, "Automatic Detection of Accent and

Lexical Pronunciation Errors in Spontaneous Non-Native English Speech,” *INTERSPEECH 2020*, Oct. 2020, doi: 10.21437/interspeech.2020-2881.

[26] Q. Ai et al., “Information Retrieval meets Large Language Models: A strategic report from Chinese IR community,” *AI Open*, vol. 4, pp. 80–90, Jan. 2023, doi: 10.1016/j.aiopen.2023.08.001.