SpeechRacer: Enhancing English Pronunciation Through Gamified Practice and Automated Speech Recognition

Saw Jing Wen, Izz Hafeez Bin Zek Hazley, Lam Chun Yu

Abstract

In this paper, we present *SpeechRacer*, an interactive, multiplayer platform that gamifies English pronunciation practice using automated speech recognition (ASR). The platform employs the browser's Web Speech API [8] to evaluate spoken inputs and integrates real-time feedback into competitive game mechanics. By creating a dynamic and engaging environment, *SpeechRacer* aims to boost learners' motivation and consistency. We detail the design considerations, system architecture, and compare its effectiveness to existing applications. Preliminary results suggest that *SpeechRacer* offers a compelling alternative to traditional methods, enhancing user engagement and language learning outcomes.

1 Introduction

Pronunciation is a fundamental aspect of mastering a language, playing a critical role in communication and mutual understanding. However, many learners of English as a second language (ESL) struggle with pronunciation due to a lack of opportunities for consistent and engaging practice [6]. Traditional methods, such as rote repetition or instructor-led drills, can be effective but are often perceived as monotonous and fail to maintain long-term learner motivation [7]. In addition, the need for a human instructor to provide feedback and monitor progress can make these methods resource-intensive and inaccessible to many learners, especially those with limited access to language learning resources or trained educators.

Recent advancements in browser-based technologies, particularly the Web Speech API [8], a World Wide Web Consortium supported specification that most modern browsers comply with, have opened new possibilities for pronunciation practice. The Web Speech API, available in modern browsers, allows developers to integrate speech recognition capabilities directly into web applications, making it a scalable and accessible tool for language learning.

To address challenges in pronunciation and fluency, we introduce *SpeechRacer*, a gamified platform designed to enhance English pronunciation through an engaging and interactive approach. *SpeechRacer* emphasises clarity and speed—two essential elements of coherent speech.

Clarity ensures that speech is easily understood, reducing ambiguity in communication, while speed reflects natural fluency, helping learners sound more conversational and confident. However, these elements are deeply interconnected: excessive focus on clarity can lead to unnaturally slow speech, while prioritising speed without clarity risks producing incomprehensible output. *SpeechRacer* addresses this balance by integrating both aspects into its gameplay mechanics.

To evaluate clarity, the platform uses the Web Speech API to determine whether spoken words are correctly recognised, providing real-time feedback to guide learners. On the other hand, to evaluate speed, *SpeechRacer* compares a player's pronunciation pace to that of other participants in real-time. Players are challenged to pronounce words or sentences as quickly as possible without sacrificing clarity, with their speed tracked against the performance of other competitors. This competitive element encourages learners to improve their fluency while maintaining accuracy, leading to more natural and effective communication.

The primary goal of *SpeechRacer* is to combine the benefits of ASR technology with the motivational power of gamification to create a platform that is both effective and enjoyable for learners. In this paper, we detail the design considerations and architecture of *SpeechRacer*, compare its functionality to existing solutions, and discuss its potential impact on language learning outcomes. Preliminary evaluations suggest that the combination of gamification and ASR can provide a valuable alternative to traditional methods, increasing user engagement and fostering consistent practice.

2 Related Work

The field of language learning applications has seen significant advancements in recent years. Existing platforms have attempted to address pronunciation challenges in different ways, with varying degrees of success. Below, we discuss notable platforms and their approaches, highlighting how *SpeechRacer* builds upon and differentiates itself from these solutions.

2.1 ChatterFox

ChatterFox focuses on improving American English pronunciation through video lessons and ASR feedback. It provides users with targeted exercises and progress tracking [2]. However, the platform relies heavily on passive learning methods, such as watching videos, which can limit engagement. While ASR feedback is incorporated, it lacks real-time competitive elements, which are central to *SpeechRacer*. By introducing multiplayer gameplay and immediate feedback, *SpeechRacer* creates a more dynamic and interactive experience.

2.2 BoldVoice

BoldVoice offers pronunciation coaching specifically tailored for non-native English speakers, featuring personalised feedback from ASR technology [1]. Its strength lies in providing detailed phoneme-level analysis, enabling learners to refine specific aspects of pronunciation. However, this granular focus often prioritises clarity over natural fluency and does not address the motivational benefits of gamified practice. *SpeechRacer* aims to bridge this gap by balancing clarity and speed in a gamified context, encouraging a holistic approach to fluent speech.

2.3 Preply

Preply connects learners with live tutors for personalised language instruction, including pronunciation practice [9]. While effective, this model is resource-intensive, cost-intensive and dependent on the availability of qualified tutors. It also lacks scalability, making it less accessible for learners with budget or scheduling constraints. In contrast, *SpeechRacer* leverages browser-based ASR to provide scalable and automated feedback, reducing reliance on human instructors while maintaining an engaging learning environment.

2.4 Duolingo

Duolingo is one of the most widely used language learning platforms, incorporating gamification to keep users engaged [3]. While its lessons include basic pronunciation exercises, these are often limited in depth and do not prioritise advanced fluency or real-time feedback. Additionally, Duolingo's gamification focuses on rewards for individual progress, lacking the competitive multiplayer dynamics found in *SpeechRacer.* By integrating competitive gameplay with advanced ASR technology, *SpeechRacer* offers a more targeted and immersive approach to pronunciation practice.

3 Differentiation of *SpeechRacer*

While these platforms have made significant strides in language learning, *SpeechRacer* distinguishes itself by addressing specific gaps:

- **Gamification with Multiplayer Focus:** Unlike traditional gamification methods that reward individual milestones, *SpeechRacer* introduces multiplayer competitions to drive engagement and motivation.
- **Integrated Clarity and Speed Metrics:** By evaluating both clarity and speed in pronunciation, *SpeechRacer* encourages a balanced approach to fluent communication, which existing solutions often neglect.
- **Scalability through Browser-Based ASR:** Leveraging the Web Speech API allows *SpeechRacer* to provide accessible and cost-effective pronunciation practice without requiring additional software or hardware, both on the user and developer-side.

4 Design Considerations

When designing *SpeechRacer*, several core principles guided our approach to ensure an engaging and effective user experience. These principles were rooted in providing simplicity, enjoyment, reliability, and fostering a competitive spirit.

4.1 Simplicity: "Get On and Play"

A key design philosophy behind *SpeechRacer* is its accessibility. The platform is designed for learners of all technical proficiencies, ensuring that users can start practicing immediately without navigating through complex setups. Thus, the choice of a browser based game was chosen over a native or mobile app that the user would have to download.

Consideration was also made for the game to be able to be enjoyed both in single-player and multiplayer contexts, where the user can just hop into a game from the landing page with only one click. Consequently, the decision was made to forgo user accounts and sign-ins, as these additional steps might deter users from engaging with the application.

4.2 Fun

Players participate in fast-paced, real-time multiplayer pronunciation races, competing to complete challenges accurately and quickly. The aim of this is to provide a compelling gameplay loop, transforming language practice to an enjoyable activity.

4.3 Useful

While fun and engagement are critical, *SpeechRacer* is ultimately designed as an educational tool to improve English pronunciation. Metrics such as words per minute (WPM), accuracy scores, and comparative feedback allow learners to track their progress over time. By focusing on practical aspects of pronunciation, such as clarity and fluency, *SpeechRacer* ensures that gameplay translates into tangible language learning outcomes.

Our choice of prompts to use also play a large role in our gameplay mechanics.

4.4 Potential for Competition

Competition is a driving force behind *SpeechRacer*, making it an engaging platform for learners. Players can compete against friends or strangers in multiplayer games, aiming to pronounce words or sentences faster and more accurately than their opponents.

5 System Design

The architecture of *SpeechRacer* is designed to balance real-time interactivity, scalability, and accessibility. The platform employs a client-server model, leveraging browser-native technologies for the client-side while utilising efficient server-side frameworks to handle real-time communication and data processing.

5.1 Architectural Design

5.1.1 Frontend. The frontend of *SpeechRacer* is developed using React [11], emphasising key design considerations to enhance reusability, maintainability, and user experience. Modularity is a primary focus, achieved by decomposing the user interface into reusable components. This modular approach not only facilitates code reuse but also simplifies maintenance and scalability, allowing developers to update or extend functionalities with minimal impact on the overall system.

State management is handled using React's built-in hooks, providing an efficient and intuitive way to manage component states and side effects. This method leverages React's functional programming capabilities, resulting in cleaner code and more predictable state transitions. The integration of WebSockets is pivotal for real-time game state updates, enabling instantaneous communication between the client and server. This ensures that players receive immediate feedback on game events, thereby enhancing the interactive experience.

Navigation between different game states is managed using React Router. This library allows for declarative routing in the application, providing seamless transitions and preserving the application state across different views. By managing routes effectively, React Router contributes to a cohesive user experience as players progress through various stages of the game.

ah

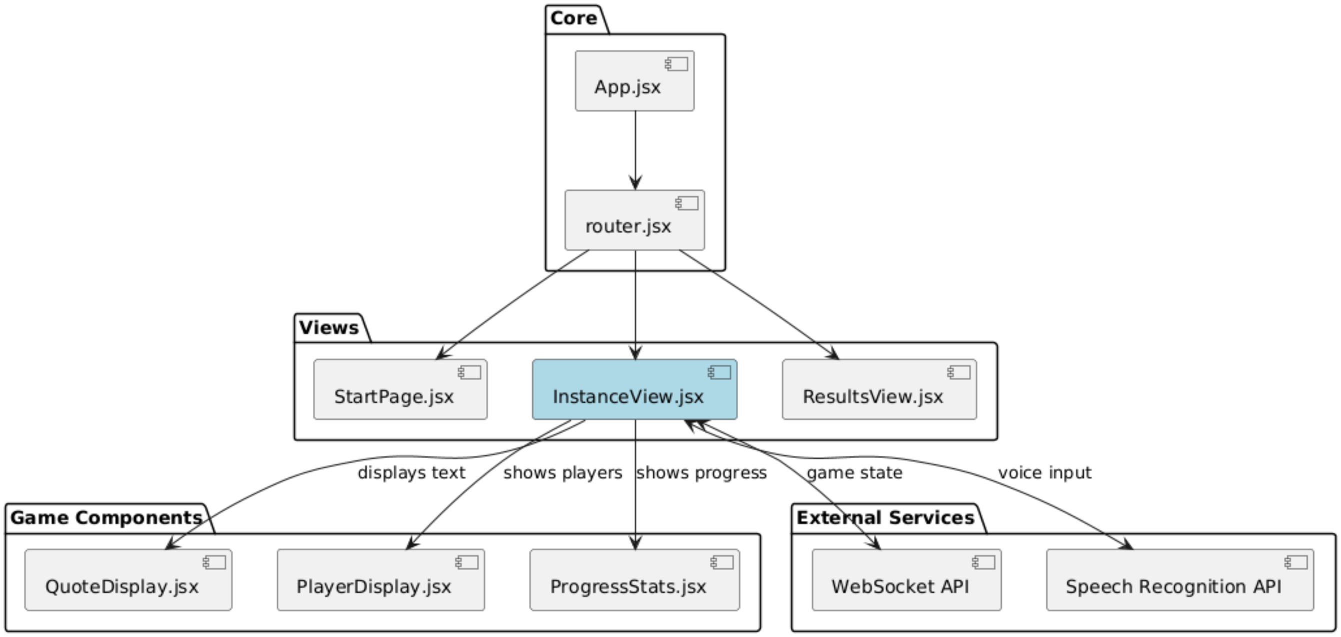

Figure 1: Frontend Architecture

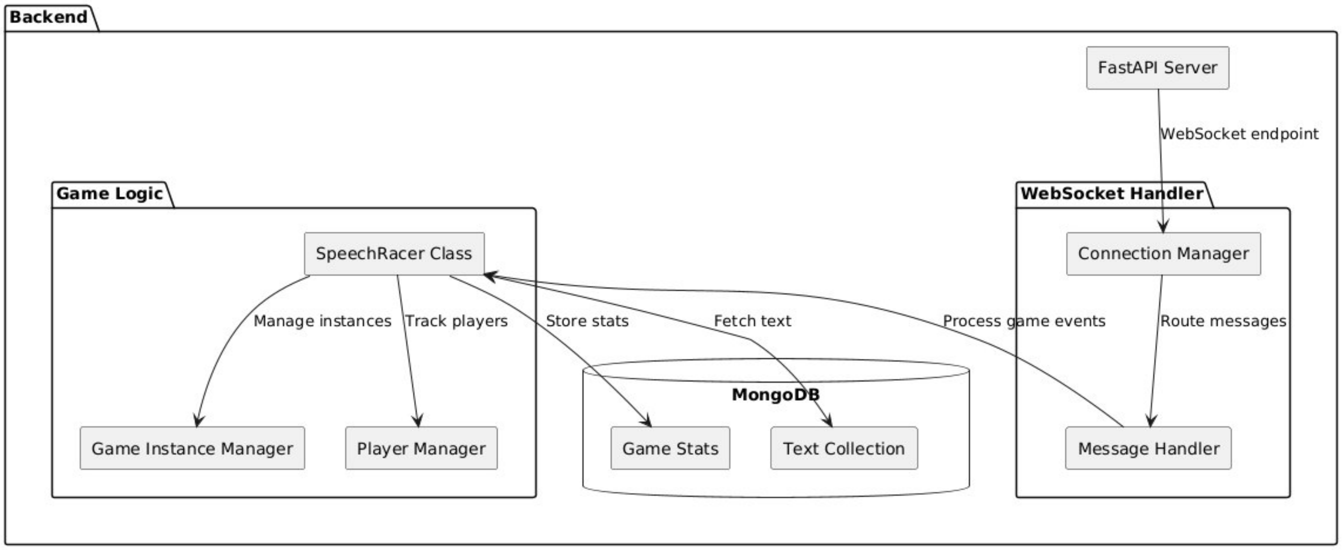

Figure 2: Backend Architecture

5.1.2 Backend. The backend is built using FastAPI, a modern, high-performance web framework for building APIs with Python [10]. FastAPI's support for asynchronous programming and WebSockets is crucial for the real-time communication requirements of Speech Racer. The backend implements a game instance system designed to group players by time slots, ensuring synchronised gameplay among participants who start at the same time.

Player connections and disconnections are handled gracefully, with the system dynamically managing the active game instances. An auto-cleanup mechanism is implemented to remove completed game instances, optimising server resources and maintaining system performance. This ensures that stale data does not accumulate, which could potentially degrade the application's responsiveness over time.

The bidirectional real-time communication facilitated by WebSockets between the server and clients is essential for the application's functionality. This connection manages player progress updates, synchronises game states, and handles player completion events. By providing a continuous exchange of information, the WebSocket integration ensures that all players have a consistent and engaging experience throughout the game.

These architectural choices in both the frontend and backend are instrumental in delivering a responsive and scalable application. By leveraging modern technologies and design principles, Speech Racer offers a robust platform for users to engage in real-time, speech-based racing challenges.

The WebSocket connection between the server and the player gives a bidirectional real-time communication, handling player progress updates, game state synchronisation, and player completion events.

5.2 Technical Implementation Details

In developing *SpeechRacer*, several critical technical design decisions were made to enhance performance and user experience.

Firstly, the application utilises the browser-native Speech Recognition API. This choice leverages built-in browser capabilities for speech-to-text conversion, eliminating the need for external libraries or plugins. It ensures broad compatibility across different platforms and devices while reducing latency in speech processing.

Secondly, the implementation of a continuous listening mode with real-time text matching is central to the application's functionality. This feature enables the system to continuously capture and process spoken input from users, matching it against the provided prompts instantaneously. Real-time text matching not only improves the responsiveness of the application but also provides immediate feedback to users, which is crucial in a racing context where speed and accuracy are paramount.

A significant challenge encountered was the registration of matched words, as the Web Speech API continuously refines its transcript based on new context to improve accuracy, potentially overwriting previously recognised words or causing previously unrecognised words to become recognised. This requires a robust mechanism to track word matches dynamically.

We wrote the algorithm in Algorithm 1 to address this issue by updating `transcriptProg` and `progress`, which are integer indices representing the current position being evaluated in `transcriptWords` (words from the transcript) and `quoteWords` (words from the reference quote), respectively.

Lastly, *SpeechRacer* incorporates real-time calculation of performance statistics, including accuracy and words per minute (WPM). By processing speech input dynamically, the application can calculate these metrics on-the-fly, offering users immediate insights into their performance. This real-time analysis enhances the competitive aspect of the game by allowing users to adjust their strategies promptly based on their ongoing performance.

These technical design decisions collectively contribute to a seamless and engaging user experience, positioning *SpeechRacer* as an effective tool for both entertainment and the improvement of speech skills.

6 Game Design

6.1 Level Selection

The level selection mechanism in SpeechRacer implements a difficulty-based progression system. Players can select from multiple difficulty tiers, each corresponding to increasingly complex text passages. This granular difficulty selection ensures appropriate challenge levels for users with varying speech recognition proficiencies. The system employs a WebSocket-based matchmaking protocol, allowing players to join specific difficulty brackets while maintaining competitive balance.

Algorithm 1 Matching Transcript Words with Quote Words

```
matchCount ← 0
skip ← 0                    ▷ Tracks latest transcript word to be in consideration
for i ← 0 to length(quoteWords) −1 do
    found ← false
    for j ← skip to length(transcriptWords) −1 do
        if quoteWords[i] == transcriptWords[j] then
            matchCount ← matchCount + 1
            skip ← j + 1
            transcriptProg ← transcriptProg + j + 1
            found ← true
            break
        end if
    end for
    if found == false then          ▷ No word match found
        isNextWordAnError ← true
        break
    end if
end for
progress ← progress + matchCount
```

6.2 Game Instance

The game instance represents the core gameplay loop, implemented through a sophisticated real-time speech recognition system. The architecture comprises several key components:

(1) **Speech Recognition Engine**: Utilizes the Web Speech API through the `react-speech-recognition` interface, providing continuous speech-to-text conversion with minimal latency. While each browser is provided freedom in how this API is implemented [8], the browser that the app was primarily designed for is the Chromium family (e.g. Chrome, Edge, Opera, etc.), which uses Google's Text-to-Speech AI service [4]. This service is based on Google's proprietary in-house 2B-parameter Speech-to-text (STT) model called Chirp [5].

(2) **Word Validation System**: Implements a progressive word matching algorithm that:
- Processes raw transcript data through the `onlyWords` utility
- Validates spoken words against the target text
- Maintains a timeout mechanism (configurable through settings) for each word
- Tracks errors and progress in real-time

(3) **Multiplayer Synchronization**: Employs WebSocket connections to:
- Broadcast player progress updates
- Synchronize game states across multiple clients
- Handle player joining and leaving events

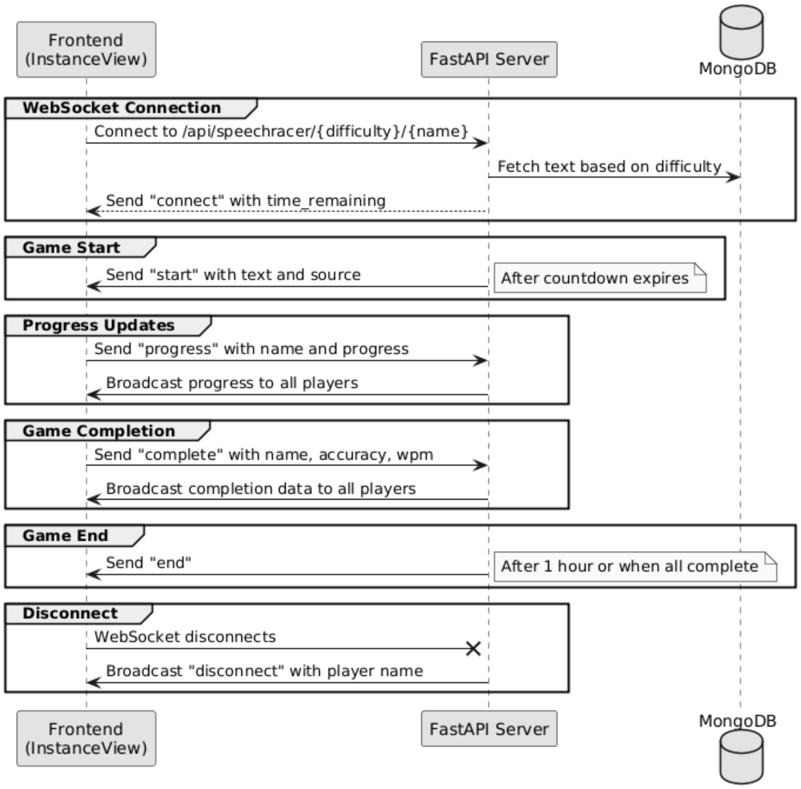
Frontend (InstanceView)
FastAPI Server
MongoDB
WebSocket Connection
Connect to /api/speechracer/{difficulty}/{name}
Fetch text based on difficulty
Send "connect" with time_remaining
Game Start
Send "start" with text and source
After countdown expires
Progress Updates
Send "progress" with name and progress
Broadcast progress to all players
Game Completion
Send "complete" with name, accuracy, wpm
Broadcast completion data to all players
Game End
Send "end"
After 1 hour or when all complete
Disconnect
WebSocket disconnects
Broadcast "disconnect" with player name
Frontend (InstanceView)
FastAPI Server
MongoDB

Figure 3: WebSocket Interaction with the client

- Coordinate game start and end conditions

6.3 Results View

The results interface provides comprehensive performance analytics through a dedicated routing system. Key metrics include:

- Words Per Minute (WPM):

$$\frac{\text{total words} - \text{errors}}{\text{time elapsed (in seconds)}} \times 60$$

- Accuracy percentage:

$$\frac{\text{total words} - \text{errors}}{\text{total words}} \times 100\%$$

- Time elapsed:

$$\text{end time} - \text{start time}$$

The system persists these statistics through WebSocket messages to the server, enabling cross-session performance tracking and potential leaderboard implementation. The results view serves as both a performance feedback mechanism and a competitive comparison tool in multiplayer scenarios.

This architecture ensures a seamless integration between real-time speech processing, multiplayer functionality, and performance analytics, creating an engaging and competitive typing experience.

7 Comparative Analysis of Existing Solutions

A comparative analysis between the platforms brought in the Related Work section and *SpeechRacer* are provided in Table 1.

8 User Testing

Scheduling conflicts posed challenges for in-person testing among the developers. To overcome this, nine friends participated in dedicated in-person testing sessions. These sessions were instrumental in uncovering and resolving numerous bugs in the code. In particular Impressively, eight out of the nine testers expressed that they would gladly play the game again.

9 Further Work

The work presented in *SpeechRacer* demonstrates leveraging automated speech recognition (ASR) technology to improve English pronunciation through gamification. However, there are several opportunities for further development and enhancement to broaden its applicability, deepen the analysis, and sustain user engagement over time.

9.1 Phoneme-level Detection

One area for enhancement involves integrating phoneme-level detection capabilities. While the current implementation evaluates spoken input primarily at the word level due to our desire to leverage the Web Speech API (which is inherently word-level),

Table 1: Comparison of Language Learning Platforms

Feature	ChatterFox	BoldVoice	Preply	Duolingo	SpeechRacer
Pronunciation Unit	Phoneme-level	Phoneme-level	Word-level	Word-level	Word-level
Gamification	Minimal	Minimal	None	Individual Progress	Real-time Multiplayer
Account Required	Yes	Yes	Yes	Yes	No
Customizable Difficulty	No	No	Tutor-dependent	Partial	Yes
Multiplayer Interaction	None	None	None	Limited (Friend Quests)	Yes
Scalability Limitation	ASR deployment	ASR deployment	Tutor Availability	None	None (Browser-based)
Cost Accessibility	Subscription	Subscription	High (Tutor Fees)	Free (Ads/Subscription)	Free

phoneme-level detection can provide more granular feedback on pronunciation accuracy. This feature could:

- Identify specific phonetic inaccuracies, such as incorrect stress or vowel sounds.
- Offer tailored suggestions for improvement, such as highlighting which phonemes to adjust.
- Facilitate advanced training modules for users focusing on nuanced aspects of pronunciation or accents.

9.2 More Detailed Analysis

SpeechRacer could incorporate a deeper analysis of user performance to provide actionable insights. Potential features include:

- **Pronunciation Heatmaps** : Visualize areas where the user struggled most (e.g., specific phonemes or words).
- **Speech Dynamics Metrics** : Analyze pitch, intonation, and stress patterns for prosody assessment.
- **Longitudinal Tracking** : Introduce user accounts to enable progress tracking over multiple sessions.

9.3 Seasonal Challenges

Gamification is central to SpeechRacer's appeal, and introducing seasonal challenges could sustain long-term user engagement:

- **Themed Competitions** : Introduce challenges tied to holidays, cultural events, or themes (e.g., "Winter Wonderland Words" or "Spring Speech Sprint").
- **Collaborative Events** : Organize multiplayer cooperative events where players work together to achieve a collective goal, such as pronouncing a set number of words in a global challenge.
- **Prizes** : Offer rewards for completing these challenges, such as digital badges or other forms of digital collectibles.

10 Conclusion

In this paper, we presented *SpeechRacer*, a gamified, browser-based platform that leverages automated speech recognition (ASR) to enhance English pronunciation practice. By combining real-time multiplayer gameplay with metrics evaluating clarity and speed, SpeechRacer addresses key challenges in traditional language learning approaches, including monotony, resource dependency, and a lack of engagement.

Our system design prioritizes accessibility, scalability, and user enjoyment, achieved through the integration of the Web Speech API, React-based frontend, and a FastAPI backend. Through innovative features such as real-time feedback, competitive gameplay, and customizable difficulty levels, SpeechRacer delivers an interactive and educational experience.

A comparative analysis highlights SpeechRacer's distinct position in the market, offering unique advantages such as cost accessibility, immediate usability, and multiplayer engagement. Preliminary evaluations suggest promising outcomes for user engagement and language learning effectiveness.

Moving forward, SpeechRacer can further enhance its utility and appeal through the introduction of phoneme-level analysis, advanced performance metrics, and seasonal challenges. By continuously evolving and leveraging cutting-edge technologies, SpeechRacer has the potential to redefine the way pronunciation is taught, making language learning more effective, scalable, and enjoyable for English learners around the world.

References

[1] BoldVoice. [n. d.]. BoldVoice: Improve Your English Accent. https://www.boldvoice.com/. Accessed: 2024-11-22.

[2] ChatterFox. [n. d.]. ChatterFox: English Pronunciation and Fluency Training. https://chatterfox.com/. Accessed: 2024-11-22.

[3] Duolingo. [n. d.]. Duolingo: Learn a Language for Free. https://www.duolingo.com/. Accessed: 2024-11-22.

[4] Google Cloud. [n. d.]. Speech-to-Text: Automatic Speech Recognition. https://cloud.google.com/speech-to-text. Accessed: 2024-11-22.

[5] Google Cloud. 2023. Bringing the Power of Large Models to Google Cloud's Speech API. https://cloud.google.com/blog/products/ai-machine-learning/bringing-power-large-models-google-clouds-speech-api. Accessed: 2024-11-22.

[6] Thuan Phan Kim. 2023. Reviewing the Significance of Practice in Learning English as a Second Language: Challenges, Impacts, and Strategies. *Educational Research and Reviews* (2023). https://www.researchgate.net/publication/374177228_Reviewing_the_Significance_of_Practice_in_Learning_English_as_a_Second_Language_Challenges_Impacts_and_Strategies

[7] Xiuping Liu. 2005. Arousing the College Students' Motivation in Speaking English through Role-Play. *Asian EFL Journal* (2005). https://www.asian-efl-journal.com/xiuping_11-05_thesis.pdf

[8] Mozilla Developer Network. [n. d.]. Web Speech API. https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API. Accessed: 2024-11-22.

[9] Preply. [n. d.]. Preply: Online Language Tutors & Teachers. https://preply.com/. Accessed: 2024-11-22.

[10] Sebastián Ramírez. 2018. FastAPI is a modern, fast (high-performance), web framework for building APIs with Python based on standard Python type hints. *FastAPI Documentation* (2018). https://fastapi.tiangolo.com/

[11] Jordan Walke. 2013. Introducing React: A JavaScript Library for Building User Interfaces. *Facebook Engineering* (2013). https://reactjs.org/blog/2013/06/05/why-react.html