Advice

Lokesh Agnihotri

Contents

[Abstract 1](#_Toc137309438)

[Introduction 3](#_Toc137309439)

[1. Front-end Technologies 3](#_Toc137309440)

[Available technologies 3](#_Toc137309441)

[Comparison 3](#_Toc137309442)

[Conclusion 4](#_Toc137309443)

[Back-end Technologies 4](#_Toc137309444)

[Available technologies 4](#_Toc137309445)

[Comparison 5](#_Toc137309446)

[Conclusion 5](#_Toc137309447)

[Programming Language 6](#_Toc137309448)

[Languages: 6](#_Toc137309449)

[Comparison of Programming Languages 6](#_Toc137309450)

[Conclusion: 6](#_Toc137309451)

[Conclusion 7](#_Toc137309452)

Abstract:

This document provides a comprehensive analysis of the technologies and software frameworks for developing a Pronunciation Assessment Application. It explores various options and considers essential factors such as availability, performance, security, and scalability. Based on the evaluation, this document offers recommendations for selecting the most suitable technologies. It covers front-end technologies, back-end frameworks, programming languages, and their comparisons, ensuring informed decision-making to optimize the functionality and effectiveness of the application.

Introduction:

The Pronunciation Assessment Application is designed to assist users in improving their pronunciation skills. This document provides an overview of the technological aspects of the project,

* including front-end development
* back-end infrastructure
* programming languages
* AI for following activities:
  + Speech to text conversion
  + Text to Speech conversion
  + Speech or text comparison.

## Front-end Technologies

### Available technologies

* **HTML** (Hypertext Markup Language): A standard markup language for creating the structure and content of web pages.
* **Angular**: A TypeScript-based open-source framework developed by Google for building dynamic web applications.
* **React**: A JavaScript library developed by Facebook for building user interfaces, focusing on component-based development.
* **Vue.js:** A progressive JavaScript framework for building user interfaces, known for its simplicity and ease of integration.

### Comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Technology** | **Cost** | **Availability** | **Performance** | **Security** | **Scalability** |
| **HTML, CSS, JavaScript** | Free and open source | Widely | Light and fast rendering | Secure | Scalable for small projects |
| **Angular** | Free and open source | Wide community Support | Optimized for high performance | Strong Security Features | Highly |
| **React** | Free and open source | Wide community Support | Efficient rendering | Strong Security Features | Highly |
| **Vue.js** | Free and open source | Wide community Support | Fast rendering | Strong Security Features | Highly |

### Conclusion

Though, Angular, React, and Vue.js offer their own advantages in terms of performance, community support, and scalability. However, they may introduce additional complexity and a steeper learning curve, making them more suitable for larger-scale projects where the benefits outweigh the associated costs.

Considering the cost aspects, availability, performance, security, and scalability, HTML, CSS, and JavaScript are recommended for the front-end development, ensuring a cost-effective, widely supported, performant, and secure solution that can be scaled for small to medium-sized projects.

## Back-end Technologies

### Available technologies

**Flask**: A lightweight and flexible Python-based web framework that offers simplicity, compatibility, and rapid development capabilities.

**Django**: A high-level Python web framework that provides a robust set of tools and features for building web applications.

**Express.js**: A fast and minimalist web application framework for Node.js, known for its simplicity and flexibility.

**Ruby on Rails**: A full-stack web application framework written in Ruby that emphasizes convention over configuration and promotes rapid development.

**Laravel**: A PHP-based web framework that offers a clean and elegant syntax, along with powerful tools and features for building scalable applications.

**ASP.NET:** A web framework developed by Microsoft that allows developers to build dynamic web applications using .NET programming languages such as C#.

**Node.js**: A runtime environment that allows developers to run JavaScript on the server-side, providing a scalable and efficient backend solution.

### Comparison

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Technology** | **Language** | **Framework** | **Cost** | **Availability** | **Performance** | **Security** | **Scalability** |
| **Flask** | Python | Light Weight | Low | High | Good | Good | Good |
| **Django** | Python | Full Featured | Medium | High | Excellent | Excellent | Excellent |
| **Express.js** | JavaScript | Minimalist | Low | High | Good | Good | Good |
| **Ruby On Rails** | Ruby | Full featured | Medium | High | Good | Good | Good |
| **Laravel** | PHP | Full featured | Low | High | Good | Good | Good |
| **ASP.NET** | C#J | Full Featured | High | High | Excellent | Excellent | Excellent |
| **Node.js** | JavaScript | Runtime | Low | High | Excellent | Good | Excellent |

In this comparison table, the technologies are evaluated based on the following parameters:

(Deed, 2023): The ability of the technology to handle increased workloads and scale with growing demands.

## Conclusion

Flask is chosen as the backend technology for the Pronunciation Trainer App due to its cost-effectiveness, availability, good performance, security features, and scalability capabilities. Despite other options like Django, Express.js, Node.js, Ruby on Rails, and Laravel offering excellent features, Flask proves to be a more suitable choice. It aligns well with the project's budget, provides extensive resources and community support, exhibits satisfactory performance, and offers adequate security measures. Flask's lightweight nature, simplicity, and compatibility with Python make it an optimal choice for the Pronunciation Trainer App's development, outweighing the higher costs and complexities associated with other technologies.

## Programming Language

## Languages:

**Python**: is a versatile and powerful language known for its simplicity, readability, extensive library support, and strong community, making it ideal for rapid application development and integration of various functionalities.

**Ruby**: is a dynamic, object-oriented language known for its simplicity and developer-friendly syntax, often used in web development with frameworks like Ruby on Rails.

**C**# is a versatile language developed by Microsoft, commonly used for building Windows applications, web services, and game development.

**Go**: also known as Golang, is a statically typed language designed for efficiency and scalability, with built-in support for concurrent programming, making it suitable for networked and distributed systems.

**Swift** is a programming language developed by Apple, designed for building applications for iOS, macOS, watchOS, and tvOS, known for its safety, performance, and modern syntax.

### Comparison of Programming Languages

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **Python** | **Ruby** | **C#** | **GO** | **Swift** |
| **Simplicity** | High | High | Moderate | Moderate | High |
| **Library Support** | Extensive | Moderate | Extensive | Moderate | Moderate |
| **Community** | Strong | Active | Active | Active | Active |
| **Integration** | Excellent | Good | Good | Excellent | Excellent |
| **Scalability** | Good | Moderate | Good | Excellent | Excellent |

### Conclusion:

Based on the comparison above, Python is chosen as the primary programming language for the Pronunciation Trainer App. Python's simplicity, extensive library ecosystem, developer productivity, and strong community support make it an excellent choice for rapid application development. Its integration capabilities align well with the requirements of the app, allowing for easy incorporation of speech recognition libraries and other necessary functionalities.

While other programming languages like Ruby, PHP, C#, Go, and Swift have their merits, they may not offer the same level of simplicity, extensive library support, and community engagement as Python. Each language has its specific use cases and strengths, but for the Pronunciation Trainer App, Python proves to be the optimal choice, striking a balance between functionality, development efficiency, and integration capabilities.

## AI For working with Text and Speech

Introduction:

Before delving into the AI part and categorizing the activities, it is essential to explore the different methods of checking pronunciations. By understanding these approaches, we can select the most suitable one and then proceed with choosing the appropriate libraries, models, and implementation strategies to achieve our goals effectively.

## Exploring Pronunciation Assessment Methods:

**Human Evaluation**: Traditional method involving human experts who assess and provide feedback on pronunciation accuracy manually. This approach offers subjective analysis but can be time-consuming and resource intensive.

**Automated Speech Recognition** (ASR): Utilizes algorithms and models to convert spoken language into written text. ASR systems can be trained to evaluate pronunciation accuracy by comparing input speech with reference pronunciations. This method provides objective assessments but may require extensive training data and fine-tuning.

**Acoustic Analysis**: Involves analyzing audio signals to extract acoustic features and measure pronunciation quality based on various metrics such as pitch, intensity, and formants. This method provides quantitative insights into pronunciation but may require advanced signal processing techniques and domain-specific knowledge.

**Phonetics-based Approaches**: Focuses on phonetic aspects of pronunciation, examining phoneme-level accuracy, stress patterns, intonation, and rhythm. This method requires phonetic expertise and can provide detailed feedback on specific pronunciation aspects.

From the above-mentioned method, it will be best to opt for the Phonetics based methods as other two methods require extensive fine tuning and advance signal processing techniques, both of which are time consuming.

### Choosing the Preferred Method:

To determine the most suitable method for our Pronunciation Assessment Application, we need to consider factors such as accuracy, scalability, resource requirements, and implementation complexity. Each method has its strengths and limitations, but considering our project goals and available resources, an automated approach like Automated Speech Recognition (ASR) proves to be a promising choice.

## Phonetics based Approaches:

Now that we have chosen our approach, it’s best to understand what phonetics-based approach is and if there are various phonetics-based approaches.  
namely there are three phonetics-based approaches.

1. Phoneme-to-Phoneme (P2P)
2. Speech-to-Speech (S2S)
3. Text-to-Speech (TTS)

Kxsvkldfgdfkdk More to addd here

Conclusion:

In conclusion, before embarking on the AI-related activities for our Pronunciation Assessment Application, it is crucial to explore the different methods of checking pronunciations. By selecting the most suitable method, such as Automated Speech Recognition (ASR), we can ensure accurate and objective assessments. Furthermore, by leveraging libraries and models specifically designed for ASR, we can implement this functionality effectively. Considering factors like accuracy, scalability, resource requirements, and implementation complexity, we can make informed decisions to achieve our goals successfully and deliver a robust Pronunciation Assessment Application.

Conclusion: Based on the analysis and comparisons, it is recommended to use HTML, CSS, and JavaScript for the front-end development of the Pronunciation Assessment Application. Flask is suggested as the back-end framework, and Python as the primary programming language. These choices ensure simplicity, compatibility, rapid development, extensive library support, and integration capabilities. By considering factors such as availability, performance, security, and scalability, these technologies provide a solid foundation for creating a user-friendly, feature-rich, and efficient Pronunciation Assessment Application.

# Bibliography

Deed, I. (2023, 05 08). *Highest Performing Web Framework Benchmarks.* Retrieved from pangea.ai: https://www.pangea.ai/resources/web/

Fabien, M. (n.d.). *Introduction to ASR.* Retrieved from https://maelfabien.github.io/machinelearning/speech\_reco/#evaluation-metrics

Gandhi, Y. (2022). Automatic Speech Recognition: Types and Examples. Analytic Steps. Retrieved from https://www.analyticssteps.com/blogs/automatic-speech-recognition-types-and-examples

Karczewski, D. (2022, December 27). Retrieved from Ideamotive: https://www.ideamotive.co/blog/best-frontend-frameworks

Klatt, D. H. (1978). *Speech perception: a model of acoustic-phonetic.* Retrieved from https://pdf.sciencedirectassets.com/272464/1-s2.0-S0095447019X34002/1-s2.0-S0095447019310599/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEGUaCXVzLWVhc3QtMSJIMEYCIQD9busBokYN22OEBa5Yb8JKXwhV2Q3H6lv%2BBDrqNsbgcQIhAIdi%2BmzYmUCDEO4Ppx%2BBFIH%2FXGeBkstL8bpY

Papastratis, I. (2022, 07 12). *Speech Recognition: a review of the different deep learning approaches.* Retrieved from https://theaisummer.com/speech-recognition/

Parabattina, B. &. (2016). Acoustic Phonetic Approach for Speech Recognition: A Review. .

Tiobe. (2023, 06). *Tibe Index.* Retrieved from Tiobe Index: https://www.tiobe.com/tiobe-index/

Veerarraghavan, S. (2023, 06 1). Retrieved from https://www.simplilearn.com/best-programming-languages-start-learning-today-article

# Give advice concerning the choice of software architecture or existing software frameworks whereby cost aspects and quality properties such as availability, performance, security, and scalability play a role.

# Provide advice on the organization of a software development process, including the test process.

(*Agile. Waterfall,*

*Testing.,. user testing and n*)

# Provide advice about the approach to take during the processing and consultation of large quantities of data with attention for privacy.