Design document

Lokesh Agnihotri

# Version Table

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Changes | Author |
| 1.0 | 6/14/2023 | Initial Draft | Lokesh Agnihotri |
|  |  |  |  |
|  |  |  |  |

Contents

[Version Table 1](#_Toc137670956)

[Introduction 2](#_Toc137670957)

[Design 3](#_Toc137670958)

[Sequence Diagram 3](#_Toc137670959)

[C4 Model: 5](#_Toc137670960)

[Test Strategy: 7](#_Toc137670961)

# Introduction

The Pronunciation Trainer Application is an innovative tool designed to assist students of MBO in improving their English pronunciation skills. The application aims to provide a user-friendly and interactive learning environment where students can practice and refine their pronunciation through various exercises and activities. By leveraging the power of technology, the application aims to enhance students’ confidence and proficiency in spoken English. The application has been developed using a combination of HTML, CSS, JavaScript, and Python Flask framework. This design document outlines the key components, features, and architecture of the Pronunciation Trainer Application.

# Design

## Sequence Diagram

A screenshot of a computer screen

Description automatically generated with low confidence

**User Interaction:**

The User enters text on the HTML front-end interface.

**Text Processing:**

The entered text is sent from the HTML front-end to the Python backend through JavaScript.

**In the Python backend:**

The received text is converted to audio using a text-to-speech conversion process.

The entered text is processed to derive the correct phonetic pronunciation using phonetic conversion techniques.

**Voice Recording:**

The User has the option to record their voice for pronunciation checking.

The recorded voice is sent back to the Python backend.

**Voice-to-Text Conversion:**

**In the Python backend:**

* The recorded voice is converted to text using speech-to-text conversion methods.
* The converted text is used to derive the IPA (International Phonetic Alphabet) phonetic pronunciation.

**Pronunciation Comparison:**

* The IPA pronunciation derived from the recorded sound is compared against the IPA pronunciation derived from the reference text (which the user is supposed to practice).
* Based on the comparison, an accuracy score is determined:
* If the phonetics match 100%, the accuracy score is considered good.
* If there are mismatches, the accuracy score is lower.

**Visual Feedback:**

* The matched phonetics are color-coded as green to indicate correct pronunciation.
* The unmatched phonetics are color-coded as red to indicate incorrect pronunciation.

**User Iteration:**

* The user can repeat the process by recording their voice again to check their accuracy and improve their pronunciation.
* This elaboration provides a clearer understanding of each step in the sequence diagram. It highlights the flow of information and actions between the user, front-end, back-end, and the processes involved in pronunciation analysis and feedback.

## C4 Model:

A picture containing text, screenshot, diagram, font

Description automatically generated

The C1 diagram provides an overview of the context in which the Pronunciation Trainer Application functions. It illustrates the key components and their interactions:

**User**: The User is an individual interested in improving their pronunciation skills. They interact with the application by recording their voice and seeking feedback on their pronunciation accuracy.

**Web Browser**: The User utilizes a web browser as their interface to access and interact with the Pronunciation Trainer Application. The browser provides a user-friendly environment for the User to navigate the application and meet their needs.

**Pronunciation Trainer Application**: At the core of the system, the Pronunciation Trainer Application processes the User's voice recordings and provides feedback on their pronunciation accuracy. It acts as a bridge between the User and the backend components.

**Backend**: The Backend serves as the engine of the application, handling the processing and computation tasks. It utilizes multiple libraries that perform various computations and analysis on the User's voice recordings. These libraries enable functionalities such as speech recognition, phonetic conversion, and accuracy assessment.

A picture containing text, screenshot, font, diagram

Description automatically generated

The C2 diagram provides a container-level view of the Pronunciation Trainer Application, showcasing the key components and their relationships: Pronunciation Trainer Application: The Pronunciation Trainer Application represents the main container. It encompasses the different components responsible for delivering the pronunciation training experience.

**Web Browser (Front end):** The Web Browser component serves as the user interface for the Pronunciation Trainer Application. Users interact with the application through the browser, accessing its features and functionalities.

**Server**: The Server component acts as the backend of the application, handling user requests and processing the necessary logic. It serves as the intermediary between the Web Browser and the underlying components.

The C2 diagram provides a high-level perspective of the system architecture, focusing on the major containers and their relationships. It outlines the flow of information and interactions between the Web Browser, Server, and the Pronunciation Trainer Application. This summary helps to understand the overall structure and communication patterns within the application.

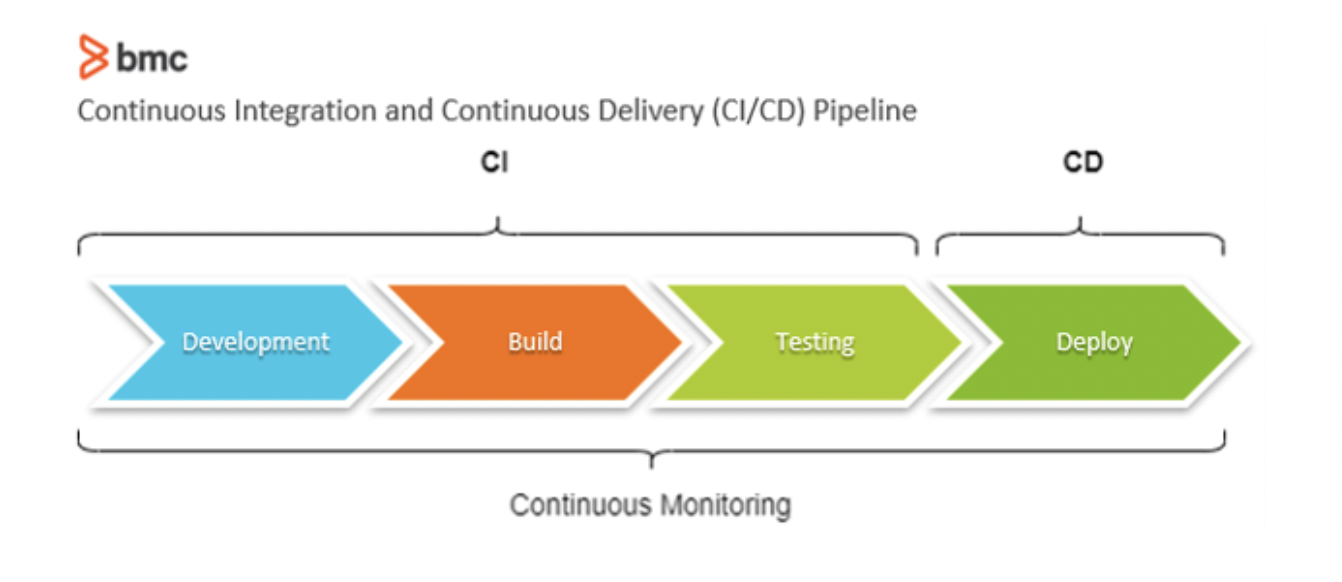
A picture containing text, screenshot, diagram, font

Description automatically generated

The C3 diagram provides a component-level view of the Pronunciation Trainer Application, highlighting the key components and their relationships. The C3 diagram showcases the major components and their relationships within the Pronunciation Trainer Application. It highlights the role of the User Interface, JavaScript, Flask, eng\_to\_ipa, Speech Recognition, and Text to Speech in delivering the pronunciation training functionality to the user. This summary provides a clear understanding of the component structure and interactions within the application.

# Test Strategy:

The development process for the Pronunciation Trainer application will be carried out using a test-driven approach, ensuring high quality and reliable functionality. To achieve this, a comprehensive set of unit tests and system tests will be implemented, which will be automatically executed each time a commit is made in the GitHub repository. Unit tests will be employed to validate the correctness of individual components and functions within the application. These tests will target specific units of code, such as methods, to ensure their expected behavior and functionality. By thoroughly testing these isolated units, any potential bugs or errors can be identified early in the development cycle, allowing for prompt debugging and maintenance.  
In addition to unit tests, system tests will be implemented to validate the integration and interaction between different modules and components of the application. These tests will focus on the overall system behavior, testing various scenarios and user interactions to ensure the application functions as intended. System tests will cover a wider scope, verifying not only individual units but also their collaboration and the application's overall performance and usability.



To streamline the testing process and ensure consistent execution, an automated testing framework will be utilized, integrated with the GitHub repository. This setup will enable the tests to be automatically triggered upon each commit, providing immediate feedback on the code changes.

By adopting a test-driven development approach, the Pronunciation Trainer application will benefit from increased reliability, stability, and maintainability. The rigorous testing practices will help catch and address issues early in the development process, reducing the likelihood of critical bugs reaching the production stage. This approach also fosters a robust development environment, promoting collaboration, code quality, and efficient debugging.  
In conclusion, the development of the Pronunciation Trainer application will employ a test-driven approach, utilizing unit tests and system tests. Through automated execution upon each commit in the GitHub repository, the application's functionality and reliability will be thoroughly assessed, ensuring a high-quality product.