

# Orator.io

Joey Li (joeyli96, 1670375)  
William Bhot (willbhot, 1669655)

Public speaking is an important skill, yet current methods of improving public speaking are inaccessible and expensive, leaving room for automation.

## 1. Vision

Orator.io is a web-based public-speaking training tool. It is targeted at anyone who is looking to improve their public speaking, from presenters for a school project to competitive debaters. This tool aims to automate speech coaching and eliminate the costs associated with having a speech coach.

The main features of this project are:

1. Analysis of the pace at which the user is speaking and suggestions on better pacing of speeches.
2. Analysis of frequently used words and alternative synonyms that could be used.
3. Analysis of hesitations and mistakes -- this includes analysing the frequency of the usage of words like "um" as well as stutters and other filler words made in the speech.
4. Analysis of the general tone or impact conveyed through the speech.
5. An option for the user to upload a transcript for more detailed analysis, including pauses and breathing.
6. An optional account to track running totals, weaknesses, and skills to provide more personalized feedback.

Possible extended features include statistical moving average analysis of pace, feedback on facial and body expressions, and speaking drills.

The scope of this project is limited to a tool that analyses the delivery of a speech, but not the content of the speech and how it affects the impact.

Currently, alternatives include hiring a speech coach (in person, or online via video calling) or recording oneself speaking and performing self-diagnosis.

1. **Human Speech Coach (Online):** There are a variety of websites that offer online sessions with human speech coaches, who will listen to speeches and give feedback.  
Advantages: An experienced speech coach can provide feedback about specific areas of weakness. Furthermore, these speech coaches would be able to focus on content to provide a more holistic analysis of the impact of the speech.  
Disadvantages: The cost is the main disadvantage, which makes this option less accessible, especially to demographics like public high school debaters who do not have much funding. Another disadvantage is that these are usually scheduled appointments and the user cannot simply practice their speech according to their own schedule.
2. **Self-Recording:** This involves recording oneself and listening to the recording in order to try and find weaknesses and mistakes. Seeing as our tool would allow the user to listen to their recording there are no advantages of this method compared to our tool.  
Disadvantages: Single-sourced analysis that may be subjective, lack of professional feedback, and a lot of work required to perform self-analysis on a speech.

What Orator.io hopes to achieve is to eliminate the arduous task of self-analyzing a speech, and provide objective feedback to improve public speaking ability without a fee.

## 2. Software Architecture

This application will be developed for the web, with potential to be ported to mobile environments. It will have two main components: the front-end web application client and the back-end running on the server. The front-end will deal with recording speech audio and sending it to the server, where it will be analysed to calculate various statistics. Meaningful results are then sent back to the front-end where data is displayed to the user. To track progress and provide more personalized feedback, results can be stored in our database, allowing the user to get information about their past speeches, track progress, and focus on weaknesses in speaking.

The backend logic of the application will rely on speech-to-text APIs such as Google Speech-to-Text and IBM Watson. This allows for basic analysis, such as word usage and average pace. Speech mood API's such as Vokatari can analyse the general tone of a recorded speech, which can be used to calculate tone and impact.

What is technically interesting about this project is that more advanced features, such as intonation, emphasis, and expression will probably require patching the APIs used and developing algorithms possibly involving machine learning to learn different ways words are pronounced for emphasis.

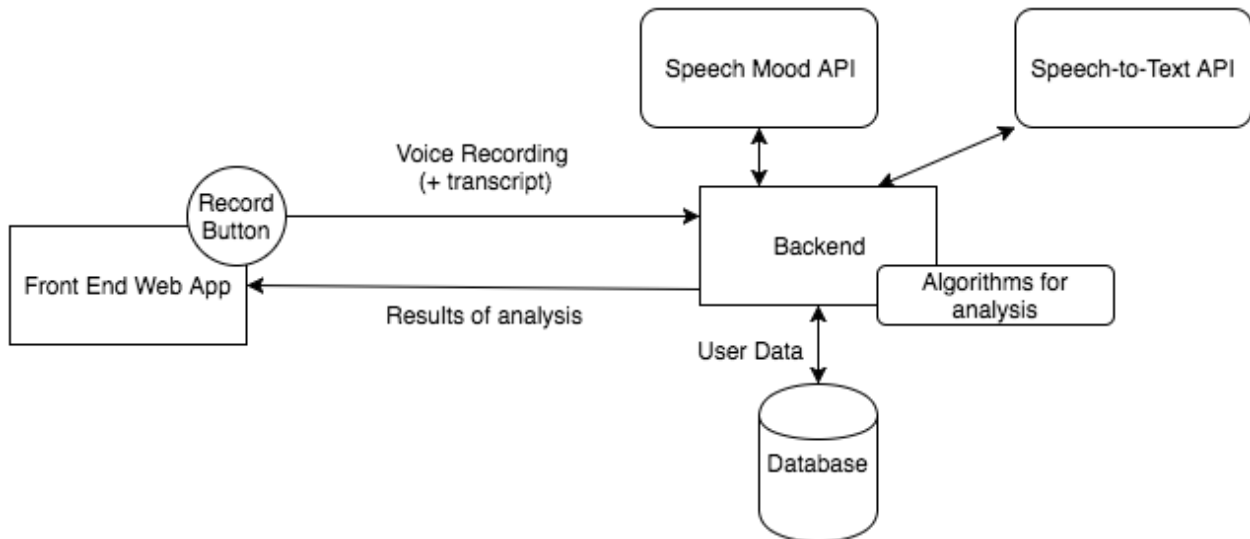


Figure 1 High-level component-model of Orator.io.

## 3. Challenges and Risks

The most challenging part of this project would be finding when each word is said in the speech to provide some of the more advanced functionality such as analysis of pauses and hesitations. This would require patching the existing API to include this functionality. In order to mitigate this risk, we will use the API that that is easiest to extend and try to tackle this problem early on in the development lifecycle.