**Overview**

This project is about building a speaker recognition software that will help in identifying any interview frauds. So basically the process is as follows:

1. This contains two steps, the first is enrollment and the second one is verification.
2. The user first enrolls by providing his name and his email ID and say the phrase that is provided at runtime.
3. If the user is already enrolled, he has to provide his enrolled email ID and then speak out the phrase provided on screen.

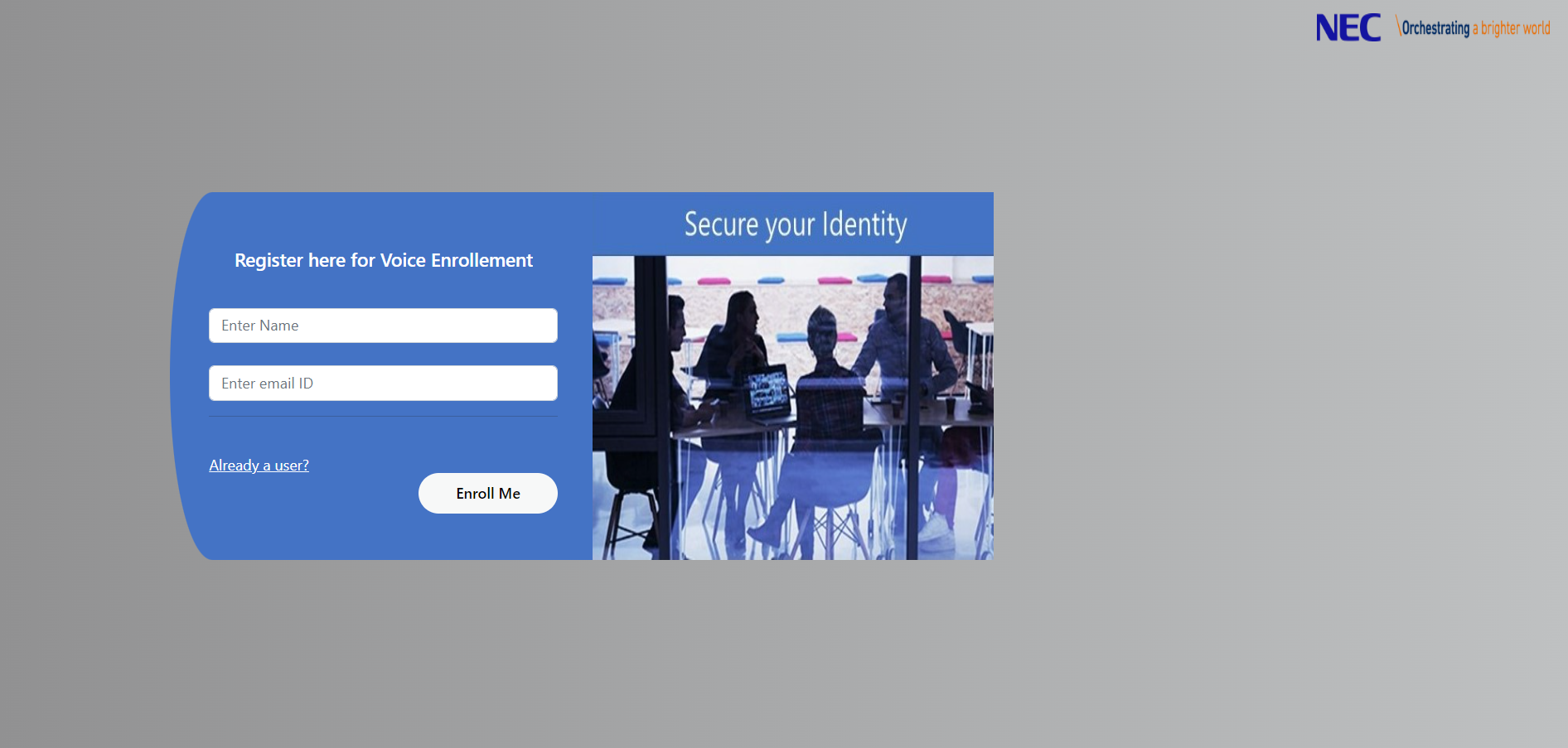
**Framework & Technology**

The framework used here is flask that acts as local server and runs python. Also, in combination to that I have used HTML, CSS to build the front-end UI. For the functioning of the webpage I have used JavaScript along with AJAX to run python scripts and communicate with the server. For storage of data I am using mySQL database. I am using Microsoft Azure Speaker Recognition API for this project.

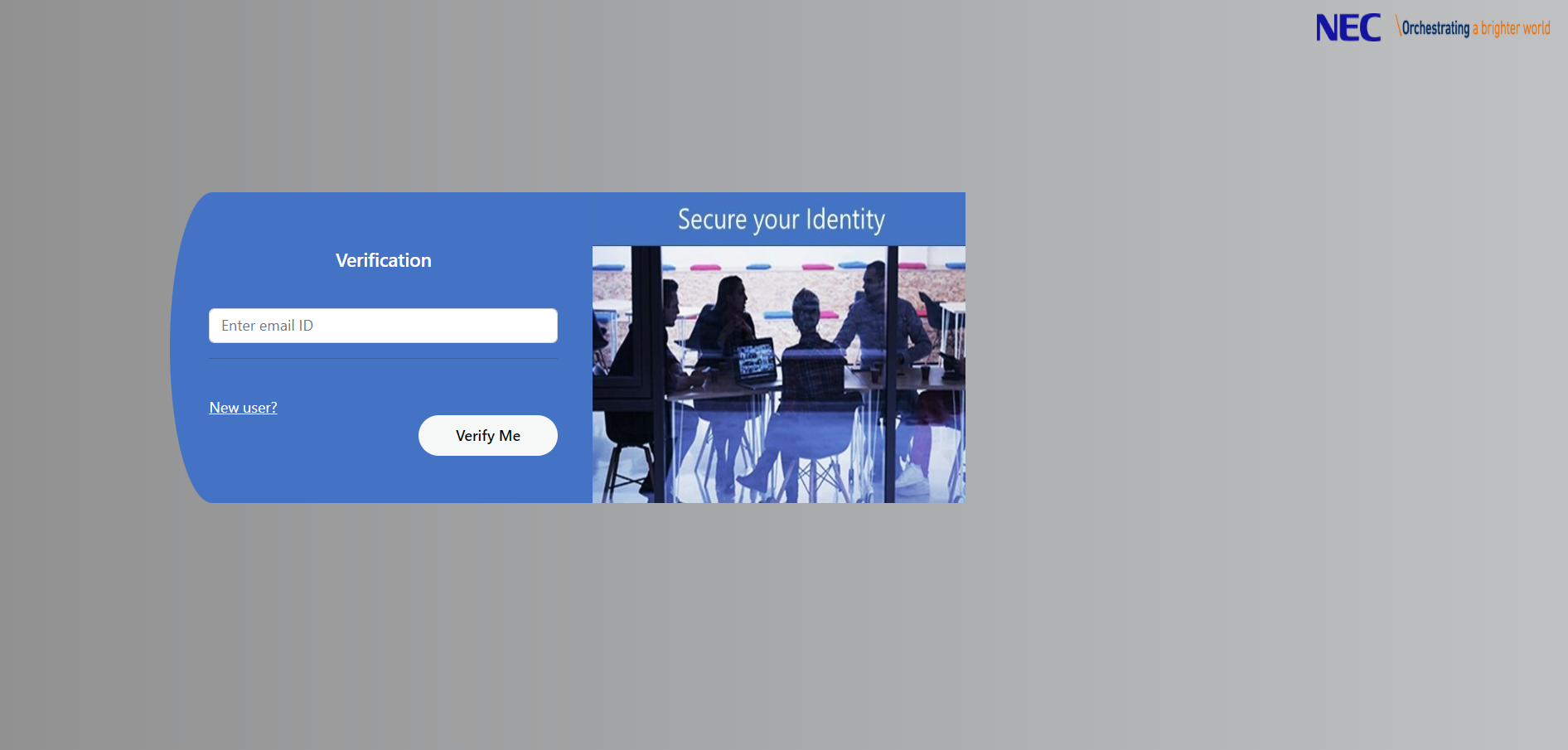
**UI Design**

The UI design is fairly simple and can be seen below

UI for enrollment -

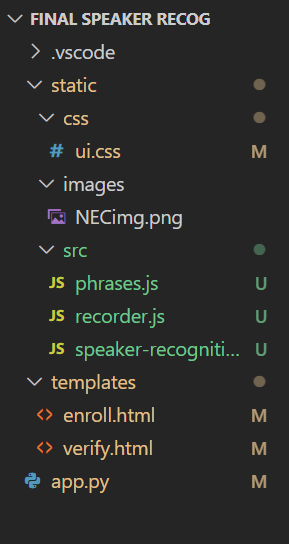


UI for verification-



**Code Segments**

The following is the structure of the entire project



1. static folder three folders that are css(ui.css that is used for styling), images(contains NEC logo) and src(which contains three js files that are recorder.js which is used to record audio, phrases.js , speaker-recognition.js).
2. templates folder contains two html files that are enroll.html, verify.html.
3. app.py is the python file that runs the flask local server as well have perform all database operations.

**Description of Code Segments**

1. **app.py** – used to run flask web server and also used to connect to database. This file has functions like params(), fetch(), executeQuery() which perform several database operations. It also runs HTML files using render\_template.
2. **enroll.html** – It’s the HTML webpage used for enrollment of user.
3. **verify.html** – It’s the HTML webpage used for verification.
4. **phrases**.**js** – This JavaScript file contains the phrases used for enrollment and verification.
5. **recorder**.**js** – This JavaScript file is used to record audio and format it so that azure API accepts it.
6. **speaker-recognition.js** – This file is used for enrollment as well as verification.
7. **ui**.**css** – This file is used for styling of both HTML files.

**API Documentation**

Speaker verification can be either text-dependent or text-independent. Text-dependent verification means that speakers need to choose the same passphrase to use during both enrollment and verification phases. Text-independent verification means that speakers can speak in everyday language in the enrollment and verification phrases.

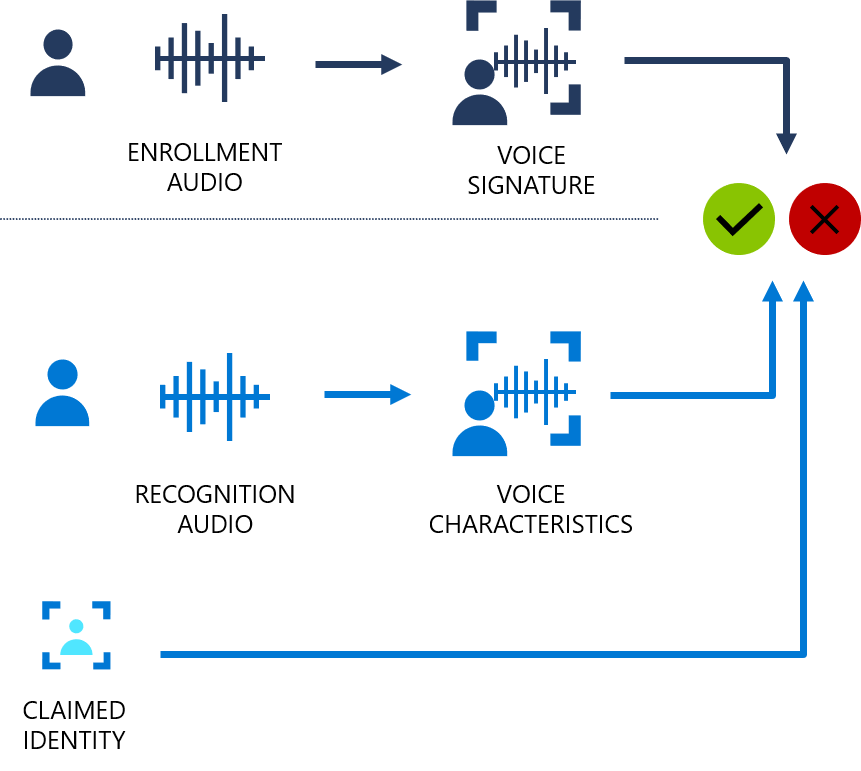
For text-dependent verification, the speaker's voice is enrolled by saying a passphrase from a set of predefined phrases. Voice features are extracted from the audio recording to form a unique voice signature, and the chosen passphrase is also recognized. Together, the voice signature and the passphrase are used to verify the speaker.

Text-independent verification has no restrictions on what the speaker says during enrollment, besides the initial activation phrase to activate the enrollment. It doesn't have any restrictions on the audio sample to be verified, because it only extracts voice features to score similarity.

The APIs aren't intended to determine whether the audio is from a live person, or from an imitation or recording of an enrolled speaker.

Speaker identification helps you determine an unknown speaker’s identity within a group of enrolled speakers. Speaker identification enables you to attribute speech to individual speakers.

Enrollment for speaker identification is text-independent. There are no restrictions on what the speaker says in the audio, besides the initial activation phrase to activate the enrollment. Similar to speaker verification, the speaker's voice is recorded in the enrollment phase, and the voice features are extracted to form a unique voice signature. In the identification phase, the input voice sample is compared to a specified list of enrolled voices.



Endpoint – <https://westus.api.cognitive.microsoft.com/speaker/identification/v2.0>

Key – c686b7d236e04a4fab81f95f6e148202

**Summary**

I have successfully made speaker recognition software which will help in detecting interview frauds.