CITS5503 Voice Verification

James Hercock, 21308586

Requirements:

This project fulfills the functional requirement of being able to register a user with a user provided name and to create a profile and voice enrolment associated with that name. It also fulfills the requirement of being able to verify a user that provides their name by using their voice.

To run this project:

1. Open a command prompt (preferably with admin privileges).
2. Navigate to this folder.
3. If this is the first time running the project, run "pip install -r requirements.txt" in the command prompt.
4. Run "python Project.py"
5. A GUI should appear, this is the project.

How to use:

On the entry screen there are 2 options, "Enrol New User" and "Verify Enrolled User".
Both are fairly self explanatory;

- "Enrol New User" creates a new user profile and attempts to enrol their voice.
- "Verify Enrolled User" takes a name and checks it against the database, then allows the user to attempt to verify their voice before then allowing them to vote if they successfully verify.

NOTES:

- For enrollment, duplicate names are not allowed. The project will verbally warn the user that the name entered already exists if attempted.
- The project requires internet connectivity to function.
- Since the speech client uses the internet to talk to the user, the speed of the internet connection will affect how responsive the project is.
- While the program is talking to the user, no other actions can be taken. This includes the recording of the user's voice, so the user must wait for the program to finish before responding with their passphrase.
- Currently the implementation uses a locally stored file to store both votes and profile IDs, this functions when running on only one machine, but full deployment will require storing votes and profile IDs to a cloud provider’s storage service.

References:

Recording code from: https://github.com/ayush1794/wave_mixer/blob/master/recording.py
Code handling API requests from:
https://github.com/Microsoft/Cognitive-SpeakerRecognition-Python/tree/master/Verification
(With some modifications to add return statements)