Year 12 Software Engineering Major Project

Écoute-Moi

French Pronunciation Examiner

Student Name: Akshat Khurana

Course: Year 12 Software Engineering Stage 6

GitHub URL:

https://github.com/akshatk-khurana/French-Pronunciation-Examiner

Table of Contents

1. Identifying and Defining..... 3
1.1 Problem Statement..... 3
1.2 Project Purpose and Boundaries..... 4
1.3 Stakeholder Requirements..... 4
1.4 Functional Requirements..... 4
1.5 Non-Functional Requirements..... 4
1.6 System Requirements..... 5
1.7 Constraints..... 5
2. Research and Planning..... 5
2.1 Development Methodology..... 5
2.2 Tools and Technology..... 6
2.3 Gantt Chart Timeline..... 6
2.4 Communication Plan..... 7
3. System Design..... 7
3.1 Context Diagram..... 7
3.2 Data Flow Diagrams..... 7
3.3 Structure Chart..... 10
3.4 Data Dictionary..... 10
3.5 UML Class Diagram..... 11
4. Producing and Implementing..... 12
4.1 Development Process..... 12
4.2 Key Features Developed..... 12
4.3 Screenshots of Interface..... 12
4.4 Version Control Summary..... 15
5. Testing and Evaluation..... 16
5.1 Testing Methods Used..... 16
5.2 Test Cases and Results..... 17
5.3 Evaluation Against Requirements..... 20
5.4 Improvements and Future Work..... 20
6. Feedback and Reflection..... 20
6.1 Summary of Client Feedback..... 20
6.2 Personal Reflection..... 21
7. Appendices..... 22
7.1 Full Gantt Chart (based on actual implementation)..... 22
7.2 Errors Encountered..... 22

1. Identifying and Defining

1.1 Problem Statement

Speaking examinations are a major aspect of the French Beginners and Continuers courses and hence a mastery of one's responses to questions within supplied speaking portfolios is key to succeeding in these examinations. A common issue faced by students in these courses is that there is no means of consolidating both their speaking portfolio and critiques of the accuracy of their pronunciation. Students must practice their speaking portfolio questions on their own, and can only receive feedback from their class teacher when practice speaking examinations are conducted in class.

1.2 Project Purpose and Boundaries

My proposed software engineering solution aims to allow French Beginners and Continuers to store their speaking portfolio responses and students to practice their pronunciation of these responses with real-time feedback on how they pronounce French words, making any time spent practicing for these examinations incredibly efficient and beneficial to students. It does not aim to serve as a speaking coach of any kind, but simply allow students to view their pronunciation accuracy using colour-coded feedback

1.3 Stakeholder Requirements

The stakeholders of this proposed solution will be students studying either of the two French courses and they will need an intuitive and user-friendly application that allows them to practice for future speaking examinations.

1.4 Functional Requirements

- The user should be able to create an account with a username and password and log in using these credentials.
- Additionally, the user should be able to add new questions and edit existing questions in their speaking portfolio.
- The user should be able to begin practice sessions with a chosen question and hear a question being asked, followed by seeing visual indications (red or green font) of their level of pronunciation for each of the words in their response to said question.
- On incorrectly pronouncing a word, users should be able to hover over the word and hear the correct pronunciation.

1.5 Non-Functional Requirements

Performance:

- The system should respond within 2 seconds for all operations, including pronunciation scoring.

- There should be minimal lag when the user accesses speaking portfolios and the questions within them.

Security:

- Confirmation prompts must be in place for critical actions (e.g., deleting questions or portfolios)
- User passwords should be hashed for storage in the database.

Reliability:

- Individual portfolio questions must be saved and loaded without errors or data loss.
- Edge cases should be handled in the deletion of questions from the portfolio, regardless of the position of the question within the portfolio.

Usability:

- An accessible and intuitive interface must be created using HTML and CSS.
- The system should be compatible with browsers commonly used by students. (Chrome and its variations, or browsers built on top of Chromium)

1.6 System Requirements

- At least 2GB RAM, 1GHz CPU, 500MB storage.
- Python 3.13 or later installed.
- Django and Pytorch frameworks installed.

1.7 Constraints

The main constraint of this project is the time which can be dedicated to it, and this will vary depending on how quickly the data required to train the model is received. In terms of technical constraints, the largest and only such constraint is the knowledge of training the neural network that will be used for grading pronunciations, as this will require significant learning and research for someone (me) who has rarely trained models in the past.

2. Research and Planning

2.1 Development Methodology

Agile is a highly efficient and iterative approach chosen for this project to allow sufficient time to complete its two major aspects; a web application where users can add to their speaking portfolio and commence practice sessions, and a trained custom Siamese Neural Network model that scores the user's pronunciations. Pertaining to the Agile approach of sprints - development and testing stages of quick succession - two sprints were allotted to this project, one for each of the above-mentioned aspects.

2.2 Tools and Technology

Visual Studio Code was deemed appropriate as it supports files of a range of different formats and programming languages. This will be crucial to training the pronunciation scoring model and the web interface. Python 3.13 is the language of choice for this solution due to its support for many of the frameworks that are to be used in this project, while being great to use with the object oriented programming (OOP) paradigm. HTML and CSS will also be key to structuring and styling the application's frontend to make it accessible and intuitive for students. Python's Django framework has been chosen due to its inbuilt SQL and DBMS support, as well as its ability to render web pages and designate API endpoints within an application. As opposed to a simpler framework like Flask, Django offers for the streamlining of both database storage and user authentication, with its built in hashing and retrieval modules.

2.3 Gantt Chart Timeline

The following is an estimated timeline of the project, not accounting for any future issues or hindrances.

Stage	5th - 11th May	12th - 18th May	19th - 25th May	26th - 1st June	2nd - 8th June	9th - 15th June	16th - 22nd June	23th - 29th June	30th June - 4th July
	W2	W3	W4	W5	W6	W7	W8	W9	W10
Planning									
Design									
Development Sprint 1									

1.1 Training Pronunciation Model									
1.2 Quantitative Model Testing									
1.3 Qualitative Model Testing (teachers)									
Development Sprint 2									
2.1 User Authentication									
2.2 Managing Portfolio Questions									
2.3 Practice Sessions									
Testing									
Functional Testing									
Non-Functional Testing									
Evaluation									

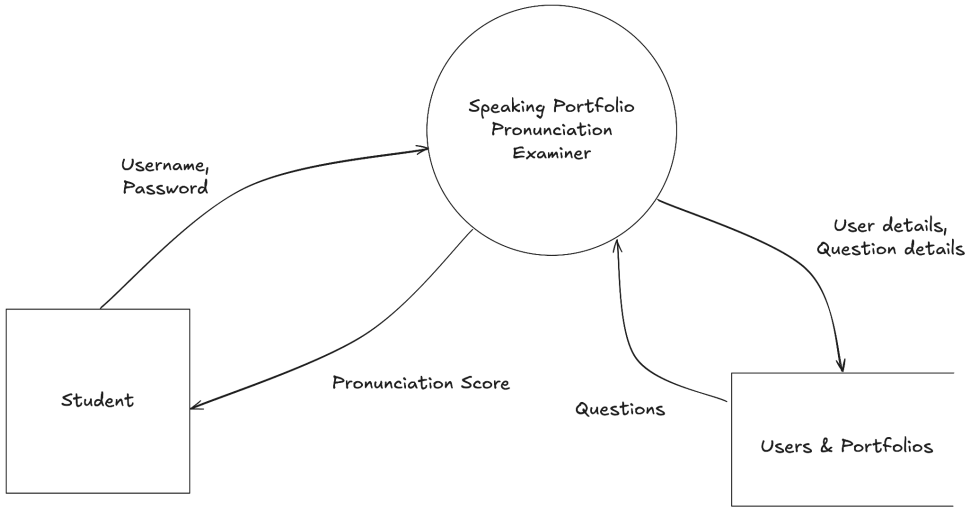
2.4 Communication Plan

The clients (students studying an HSC French course) will be contacted for their feedback in French lessons and these results will be part of the Acceptance testing undertaken at the end of the development.

3. System Design

3.1 Context Diagram

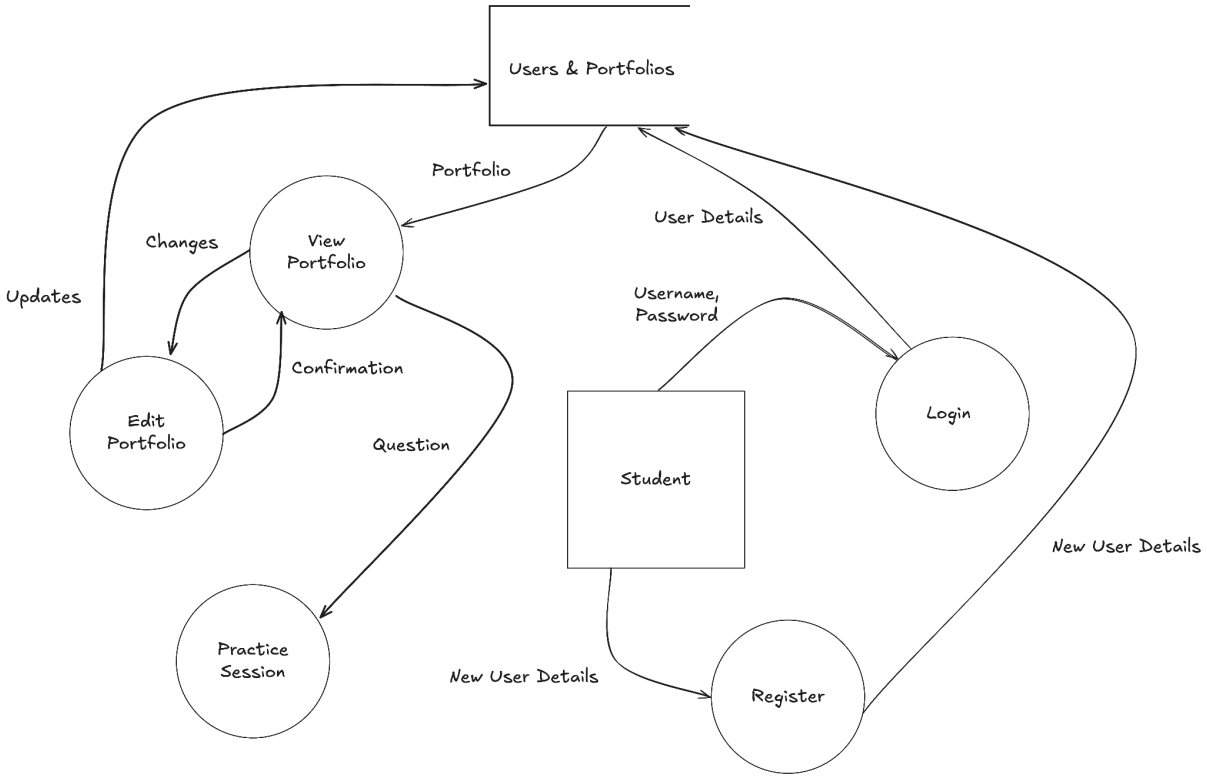
While very simple, the context diagram allows for an overall view of the system and the entities - in this case, these are just the student and the backend system.



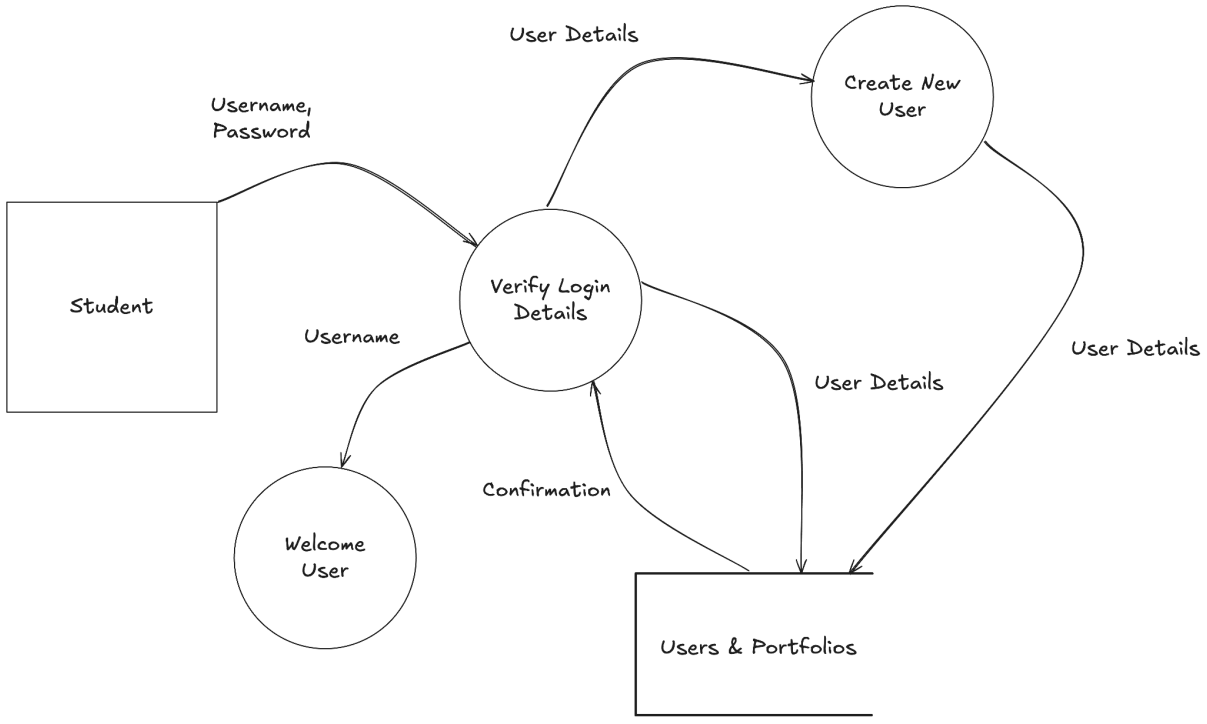
3.2 Data Flow Diagrams

The data flow diagrams used in this project were critical to designing the system as they dictated the route taken by data within processes and ultimately also highlighted the need for certain in-between processes.

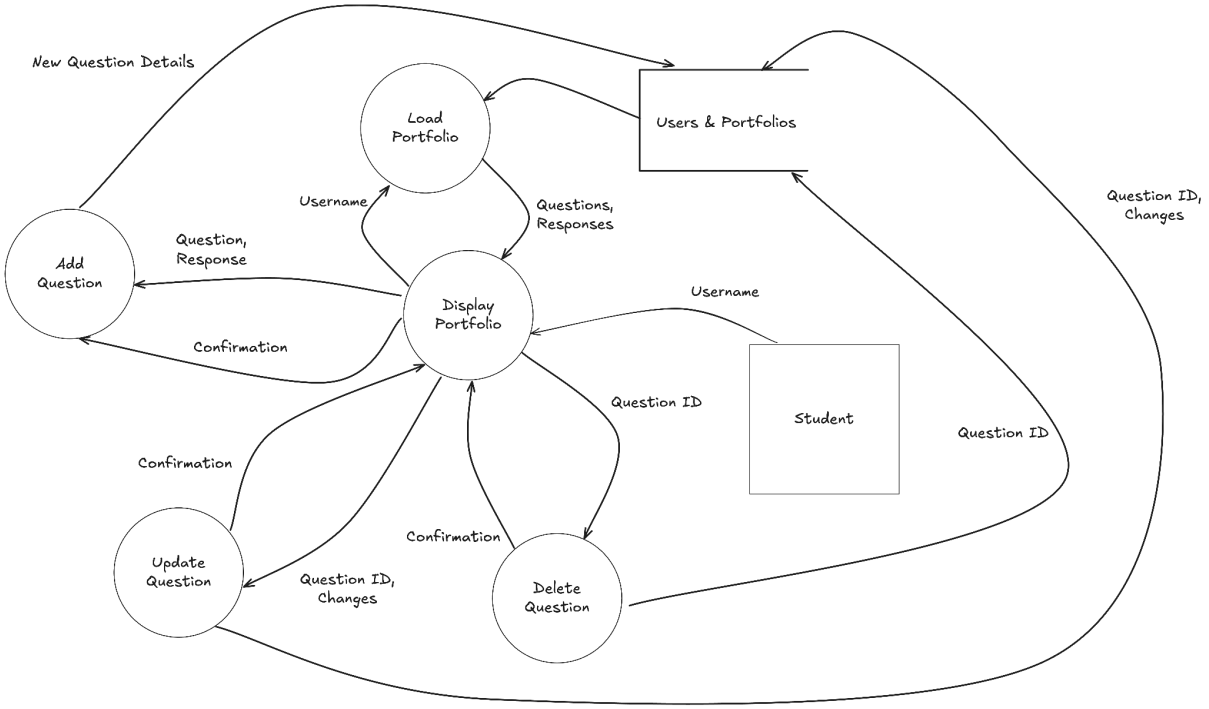
Level 1



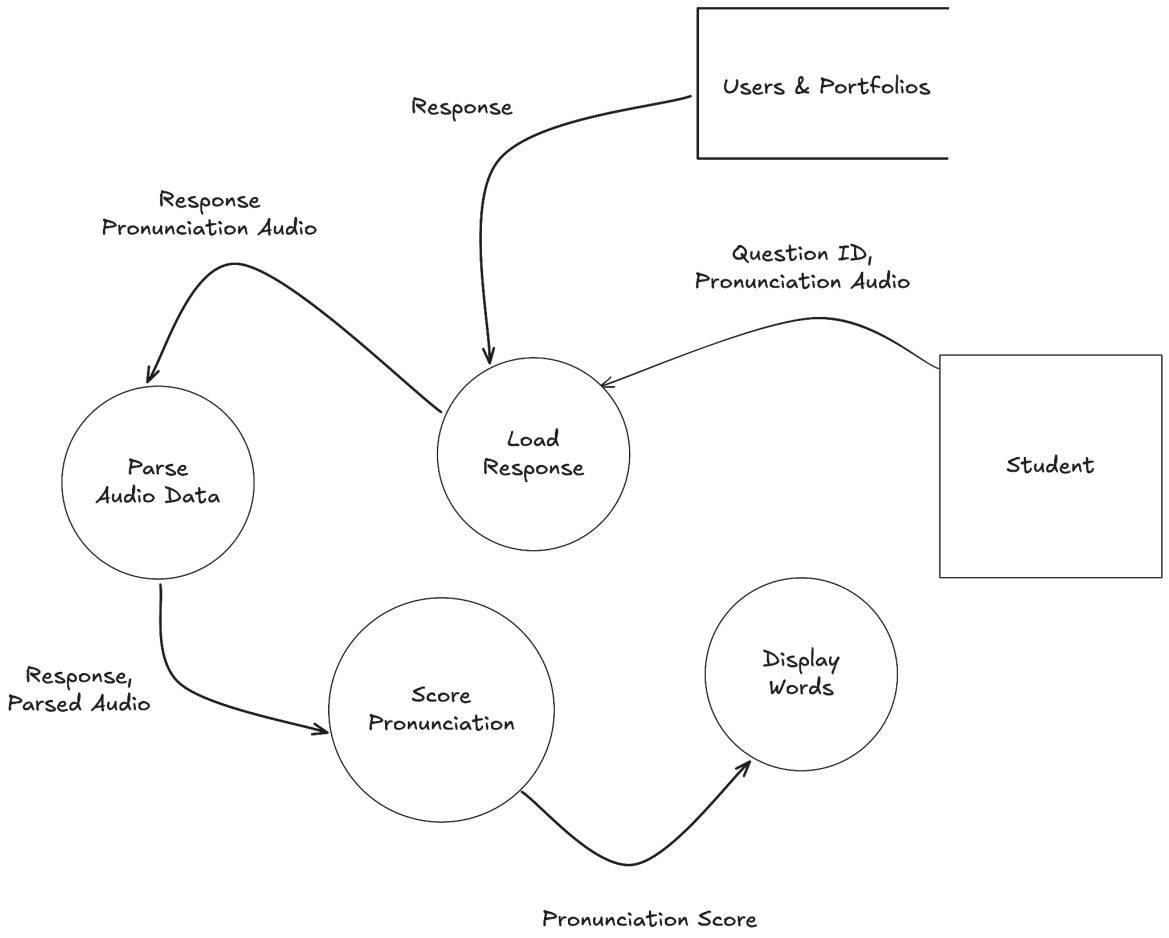
Level 2 - Authentication



Level 2 - Portfolio Modification

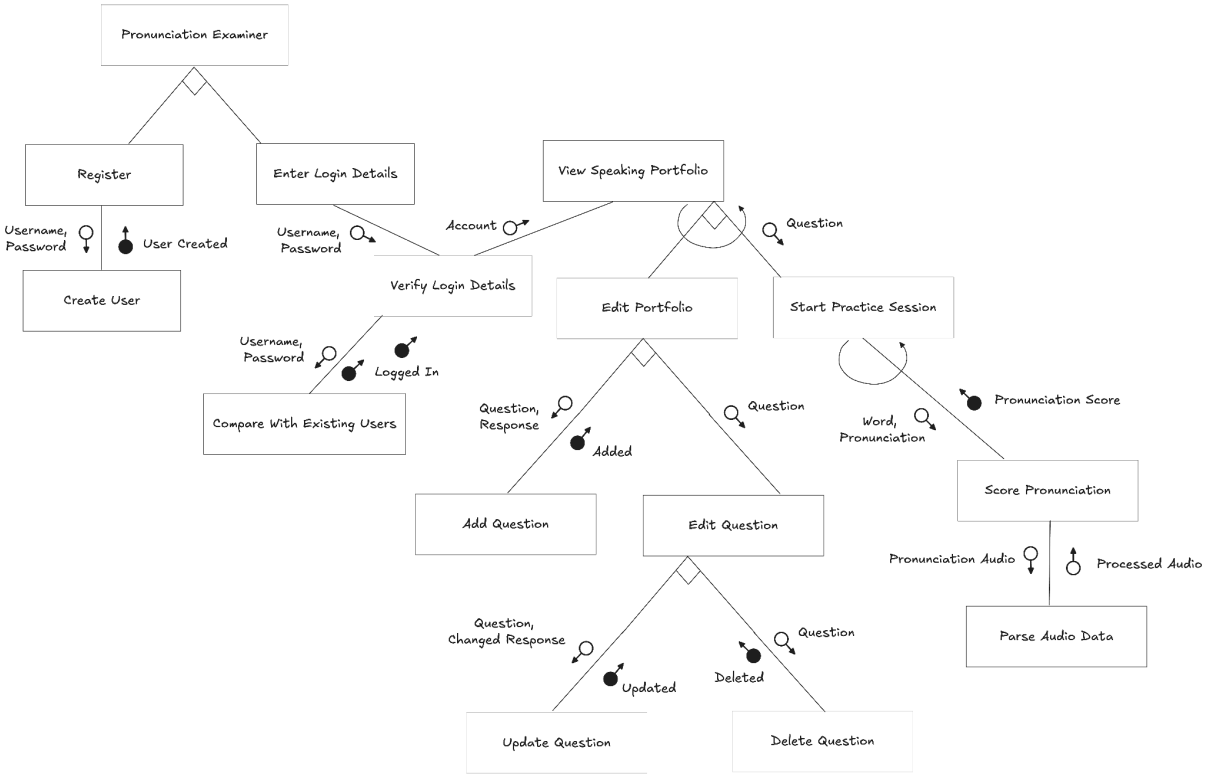


Level 2 - Practice Sessions



3.3 Structure Chart

The structure chart allows for a view of the hierarchy between all the processes that make up this solution, as well as the flags and data exchanged between processes. Specifically, the hierarchy includes the authentication system (on the left) with the rest of the application (on the same level as the authentication system, but on the right) and shows the modules that will make together allow the user to modify their speaking portfolio and also get their pronunciation scored.



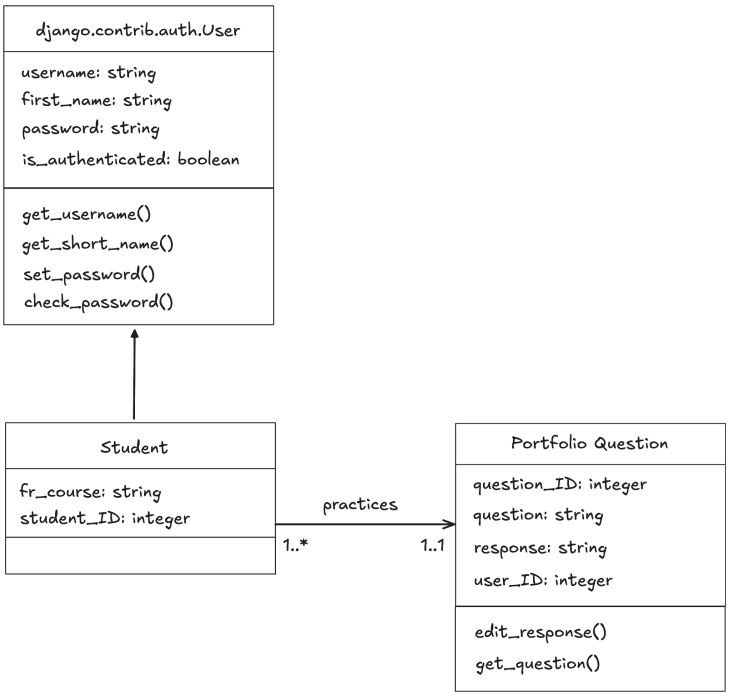
3.4 Data Dictionary

Variable	Data Type	Size	Description	Example	Validation
score	Boolean	0 or 1	Pronunciation Level	1 (Good)	NA
username	String	32 chars	Chosen username.	redwallaby	Text only, not blank.
first_name	String	32 chars	Given first name.	Kobe	Text only, not blank

password	String	8 chars	Chosen password	french1234	Only alphanumeric, no symbols. At least 1 number.
is_authenticated	Boolean	True/False	User authentication status.	True	NA
student_ID	Integer	2 digits	Unique ID of each student.	01	Primary key.
fr_course	String	NA	Course student is enrolled in.	Beginners	From a dropdown list.
question_ID	Integer	4 digits	Unique ID of each question.	0001	Primary key.
question	String	1024 chars	Question from portfolio.	Parle moi de toi.	Text only, not blank.
response	String	1024 chars	Response to portfolio question.	…	Text only, not blank.

3.5 UML Class Diagram

While this application does not explicitly follow the object oriented paradigm, a UML diagram was chosen as it best represented the database structure for storing users and associated portfolio questions. It clearly shows the important attributes and methods of each of the classes that are defined in the models.py file in the Django project, so Django can turn them into SQLite database tables.



4. Producing and Implementing

4.1 Development Process

Before commencing development, I talked to the relevant persons to ensure I could get a variety of different pronunciation audio files to train the model that would power the practice session functionality. Due to legal reasons, I couldn't use unconsented existing recordings and hence had to create a Google Form to collect consented pronunciation recordings from French students in Years 7-11. As this took some time, I started by building out the web application in Django, making sure everything, except the practice session functionality was fully functional.

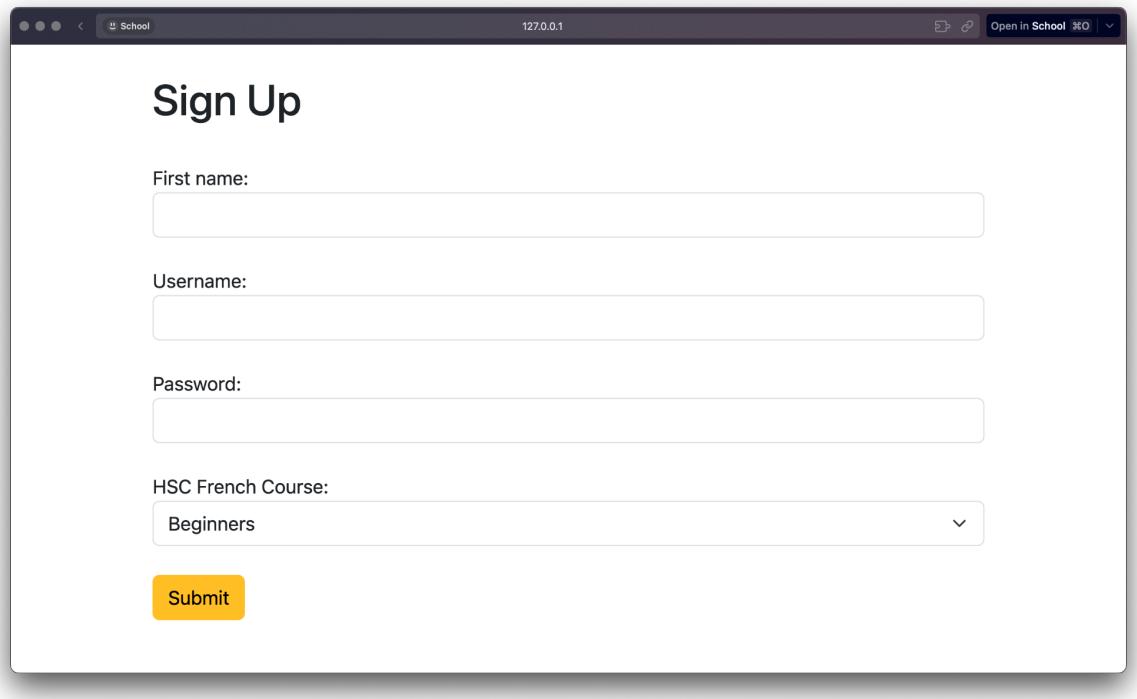
Nearing the end of the sprint, I also began working on the JavaScript that would allow the application to record audio in the browser and send it back to the backend for processing. The model was trained near the end of the development process as this was when sufficient pronunciation audio data was received. Unfortunately, due to the unforeseen time it took to receive the data, a lot of development time was lost and hence the application's practice session feature remains incomplete.

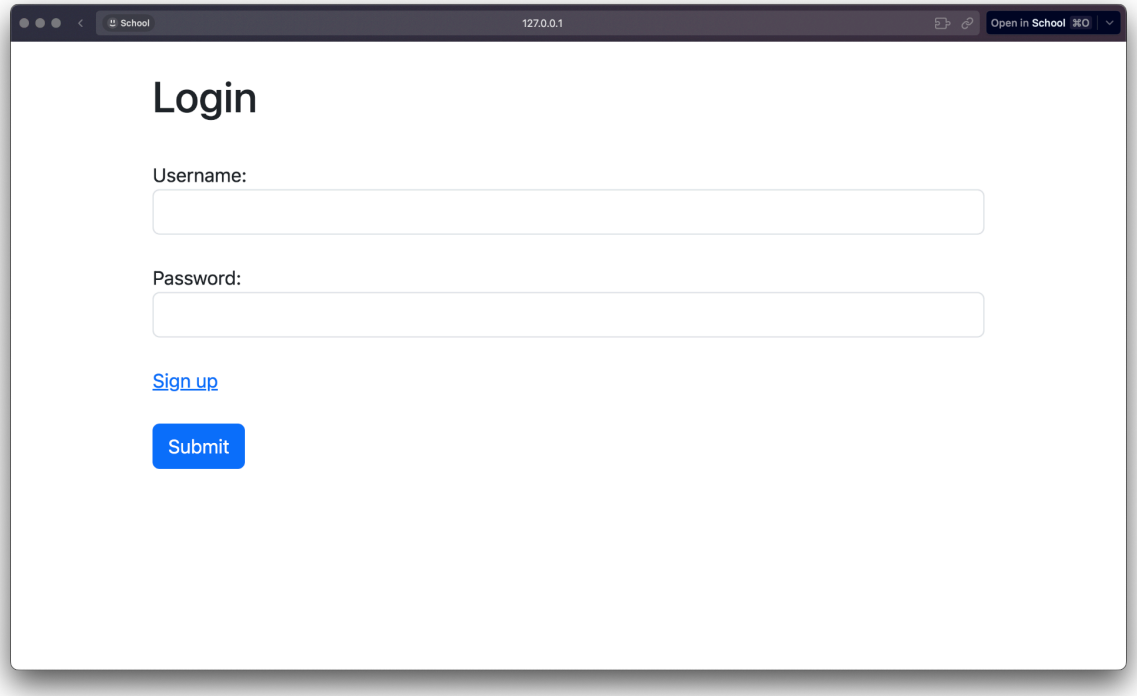
4.2 Key Features Developed

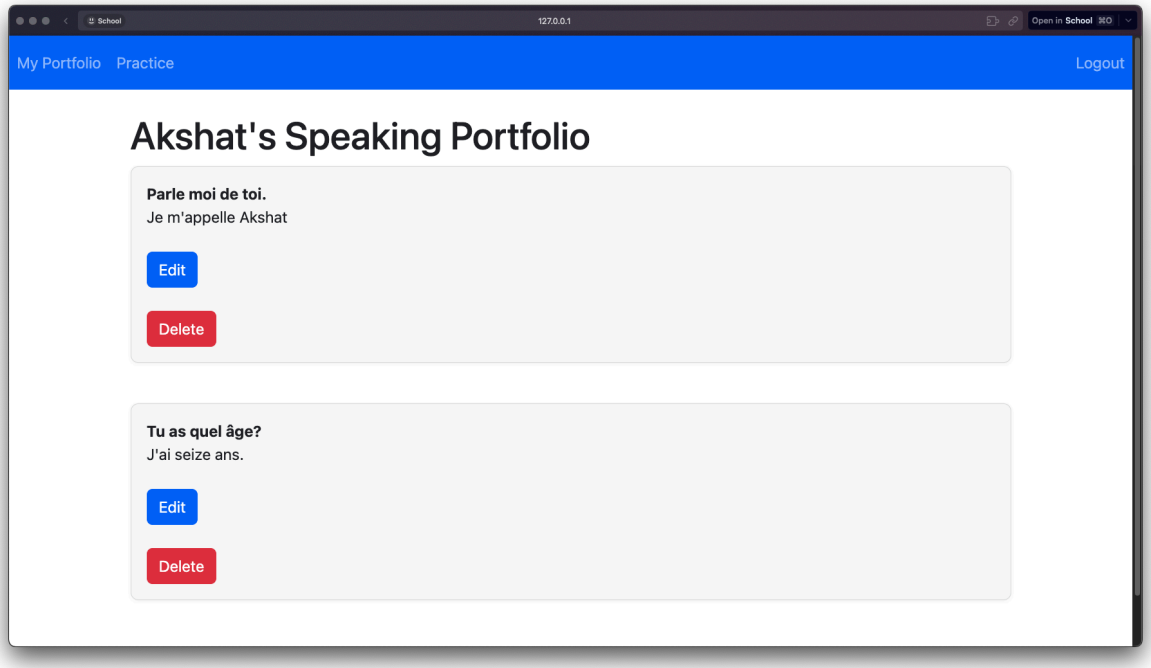
The core features of this application are:

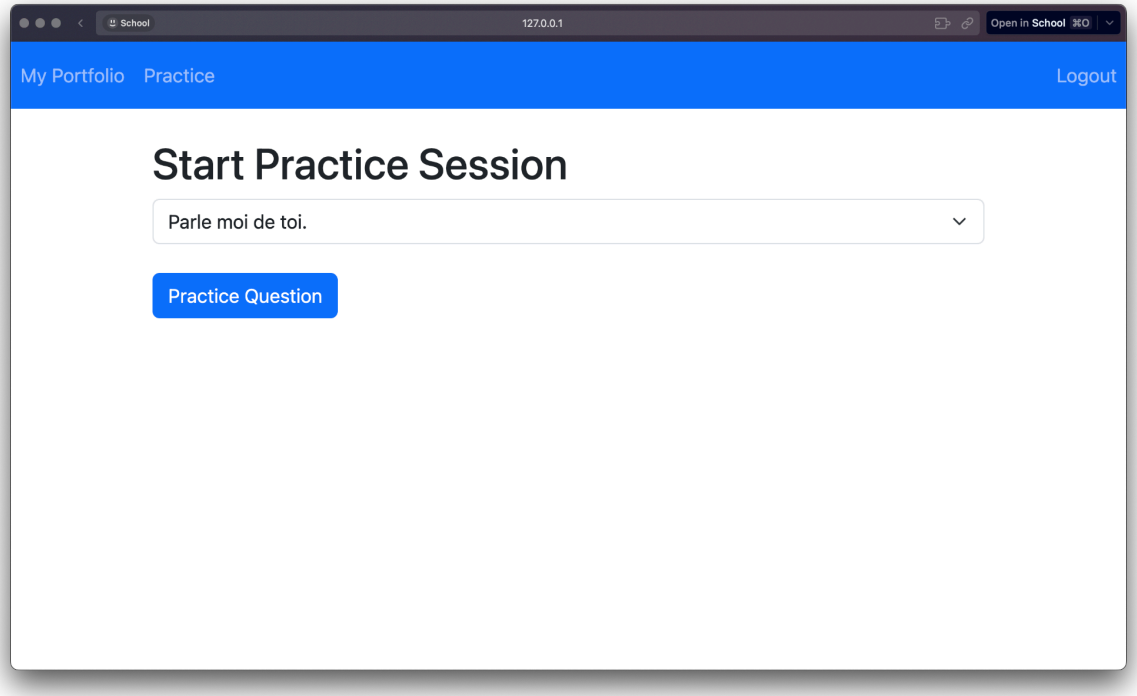
- *An authentication system* that allows users to securely access their speaking portfolio and logout once finished.
- *A portfolio page* which gives users the ability to add, edit and delete existing questions, while viewing their speaking portfolio in its entirety.
- *A practice session page* that allows users to choose a question they want to practice and directs them to a page with the question and their pre-defined response. On this page, users can click on the question to hear it being spoken (simulating an actual speaking examination) and hover any of the words in their response which they are unsure about pronouncing to hear a correct and accurate pronunciation (from Google Text-To-Speech).

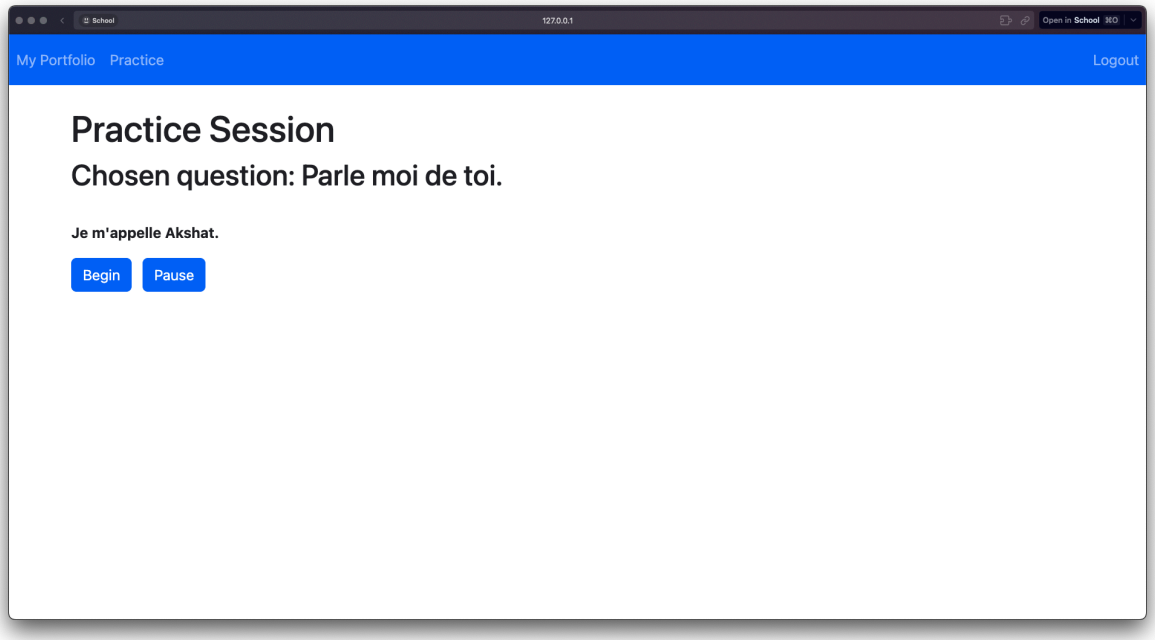
4.3 Screenshots of Interface




School
127.0.0.1
Open in School ⌘O
Login
Username:
Password:
Sign up
Submit

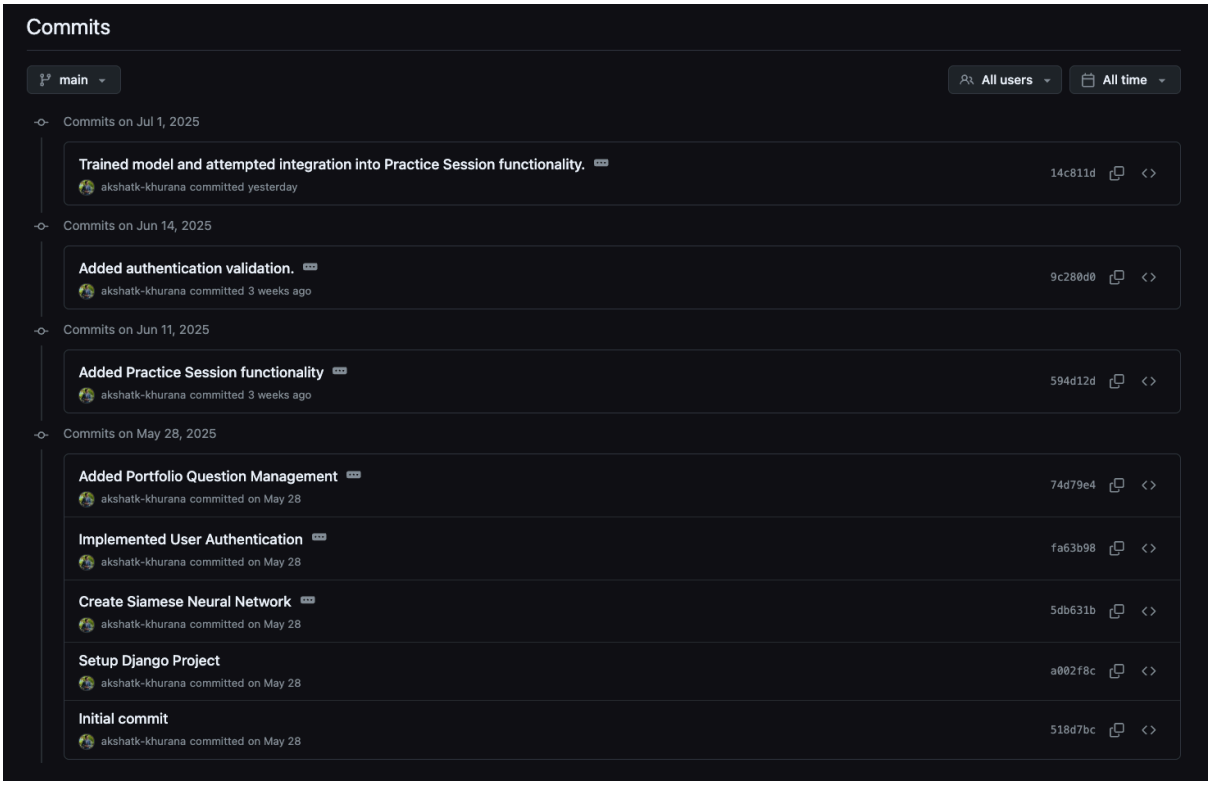

School
127.0.0.1
Open in School ⌘O
My Portfolio
Practice
Logout
Akshat's Speaking Portfolio
Parle moi de toi.
Je m'appelle Akshat
Edit
Delete
Tu as quel âge?
J'ai seize ans.
Edit
Delete





4.4 Version Control Summary

My project's progress has been documented in the Github repository linked above.


Commits
main
All users
All time
Commits on Jul 1, 2025
Trained model and attempted integration into Practice Session functionality.
akshatk-khurana committed yesterday
14c811d
Commits on Jun 14, 2025
Added authentication validation.
akshatk-khurana committed 3 weeks ago
9c280d0
Commits on Jun 11, 2025
Added Practice Session functionality
akshatk-khurana committed 3 weeks ago
594d12d
Commits on May 28, 2025
Added Portfolio Question Management
akshatk-khurana committed on May 28
74d79e4
Implemented User Authentication
akshatk-khurana committed on May 28
fa63b98
Create Siamese Neural Network
akshatk-khurana committed on May 28
5db631b
Setup Django Project
akshatk-khurana committed on May 28
a002f8c
Initial commit
akshatk-khurana committed on May 28
518d7bc

5. Testing and Evaluation

5.1 Testing Methods Used

In the Agile process which was used to develop this French Pronunciation Examiner application, each sprint ends with integration testing and alpha testing to detect any flaws present in the features added in that sprint.

The testing strategy utilised for this application includes the following four testing methods:

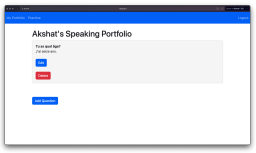
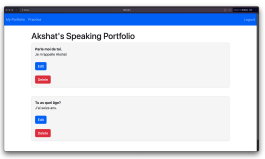
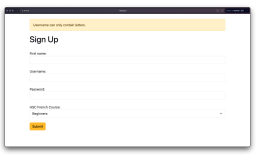
- ➢ Alpha testing: Testing conducted in-house and early-on in the development process by the developers to identify potential bugs in the system before it is sent to be Beta tested externally.
- ➢ Black box testing: A form of testing that involves inputting various values into the system and verifying if they meet the expected outputs, without using the codebase.
- ➢ Integration testing: Testing conducted between separate modules in an application to warrant expected communication and results.
- ➢ Acceptance testing: The final set of testing conducted on the system to verify whether it meets the pre-defined user requirements.

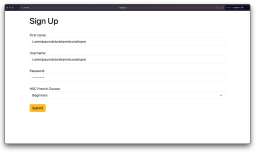
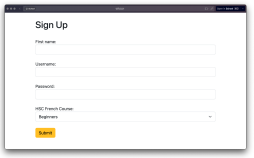
The above testing methods were chosen so that features from the two Agile sprints defined in the Gantt chart could be developed, integrated and have their functionality verified accordingly. Integration testing is to be conducted once each of the sprints is done, to ensure that the pronunciation scoring module (the trained Pytorch model) correctly communicates with the rest of the application and more specifically, the Practice Session functionality. Alpha testing is to be conducted around the midway of the application's development to ensure that all other features of the application, excluding the Practice Session functionality function as expected.

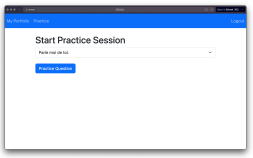
Black box testing will take place after the first sprint to verify whether the authentication system can handle unexpected and invalid attempts from users. Acceptance testing is to be conducted right at the end of the development process; to find and mitigate any errors in the system that have been overlooked during Alpha testing and also ensure that the potential users of the application are satisfied with its functionalities in regards to the requirements presented to them at the start of the SDLC.

5.2 Test Cases and Results

Test Type	No.	Description	Expected Result	Actual Result	P/F
Alpha	01	Create a	The requested	The user is directed	P

		question once logged in.	question is added to the user's portfolio and they are directed to their portfolio page.	to the portfolio page and the newly created question is visible to them.	
	02	Deleting a question.	Question is removed from the users portfolio, and everything else remains unaffected.	The deleted question is not visible to the user.  	P
	03	Selecting a question to practice.	Users are presented with a dropdown of all their questions to choose from.	-	P
	04	Load practice session page for chosen session.	User can see the question and appropriate response.	-	P
Black Box	05	Sign up with numbers in username	Present user with error message and prompt to enter number-less username.	The user is shown a warning box and prompted to try again. 	P
	06	Sign up with numbers in first name.	Present user with error message and prompt to enter number-less first name.	The user is signed up and directed to the login page.	F

	07	Sign up with alphanumeric password	Sign up user and direct to login page.	The user is signed up and directed to the login page	P
	08	Sign up with any field exceeding character limit (as defined in data dictionary):	Prevent users from typing more than the required length.	Data fields restrict any input once the limit is reached. 	P
Integration	09	Score a pronunciation audio clip of a user.	Backend returns a score of either ‘Good, Bad or OK’ to frontend in JSON.	A 500 Internal Server error occurs due to matrix multiplication errors when running the model on provided user audio.	F
Acceptance					
Functional	10	Create an account with a username and password and log in.	If user details meet validation criteria, create an account and direct to login.	- 	P
	11	Add new questions and edit or delete existing questions.	Update the question and show the changes (edited or added) questions	-	P
	12	Begin practice sessions with a chosen question and get real-time feedback.	Allow users to choose a question, and speak while updating colours of words based on pronunciation.	Allow users to choose a question and speak, unable to give real-time feedback.	F

					
	13	Hover over the question or a particular word in the practice session and hear correct pronunciation.	When the user clicks on the question chosen, they can hear it being pronounced and hover over other words to hear pronunciations.	-	P
Non-Functional	14	Performance	Respond within 2 seconds for all operations, with minimal lag.	Responds within 1 second for all portfolio activities.	P
	15	Security	Confirmation prompts for critical actions with user passwords hashed for storage.	Passwords hashed using Django's auth system.	P
	16	Reliability	No data loss within portfolio and handling of edge case deletion of questions.	-	P
	17	Usability	Accessible and intuitive interface compatible with browsers.	-	P

5.3 Evaluation Against Requirements

As shown by the Acceptance testing portion of the testing table above, my solution met all functional and non-functional requirements that were defined during the design phase, leaving only one failed requirement (functional) that being of the real-time feedback within

Practice session. In terms of the security of the application and its data, this aspect was kept in mind from the beginning of the development process and thanks to Django's auth system, all user passwords are hashed with SHA256. In addition to this, validation functions have been implemented in the web application to ensure that sensitive user information meets specific criteria and that all users have a strong password when signing up. Finally, to prevent any forms of cross site scripting attacks, Django's CSRF tokens have been enabled in each of the application's pages, with the CSRF_exempt decorator applied only to endpoints that don't involve the usage of any user data. As a result of all these secure coding practices, all students who access the application can stay assured that their portfolios will be modified only by them and that no one else will be able to access their account.

5.4 Improvements and Future Work

As mentioned, I would simply spend more time debugging the Practice Session functionality to ensure that the trained Siamese neural network integrates with the web application. Another key improvement I would make is that of actually training the neural network more thoroughly. As there were time constraints and a lot of time had already been spent in receiving pronunciation audio files, I was only able to end up with 650 processed and usable samples for training and testing. Hence, if more time was to be spent on this solution, I would survey more people for their pronunciations and try to get as close to 1000 samples once all the received audio files are separated by word.

6. Feedback and Reflection

6.1 Summary of Client Feedback

Clients were interviewed during a French continuers period and without explicitly mentioning their names, the following feedback was received:

Student X: *The interface is user friendly and intuitive. A randomised way of practicing the entire portfolio would be good. Every requirement except for pronunciation scoring has been executed well.*

Student Y: *Except for the practice sessions, the application functions well. I think allowing longer passwords with symbols would be my only feedback.*

6.2 Personal Reflection

While making the Django web application was previously explored territory, a lot was learned during this project. Specifically, training the Siamese neural network was a new experience, as although I have attempted to train models before, the architecture of the model was different, and as audio files were involved, significantly dissimilar to preprocessing was required. I learned a large amount on how to manipulate audio files (trimming, splitting, etc) and how to use Mel spectrograms to turn them into tensors that could be fed into the model. Lastly, while I was unable to complete the real-time pronunciation feedback feature, I did learn about how to record audio and send it over an endpoint in Javascript.

7. Appendices

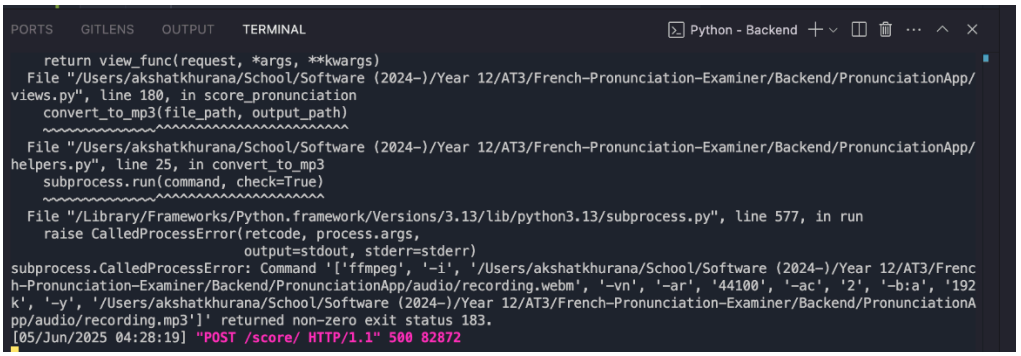
7.1 Full Gantt Chart (based on actual implementation)

Stage	5th - 11th May	12th - 18th May	19th - 25th May	26th - 1st June	2nd - 8th June	9th - 15th June	16th - 22nd June	23th - 29th June	30th June - 4th July
	W2	W3	W4	W5	W6	W7	W8	W9	W10
Planning									
Design									
Development Sprint 1									
1.1 User Authentication									
1.2 Managing Portfolio Questions									
1.3 Practice Sessions									
1.4 Alpha & Black Box Testing									
Development Sprint 2									
2.1 Preparing Received Data									
2.2 Training Pronunciation Model									
2.3 Quantitative Model Testing									
Acceptance Testing									
Evaluation									

7.2 Errors Encountered

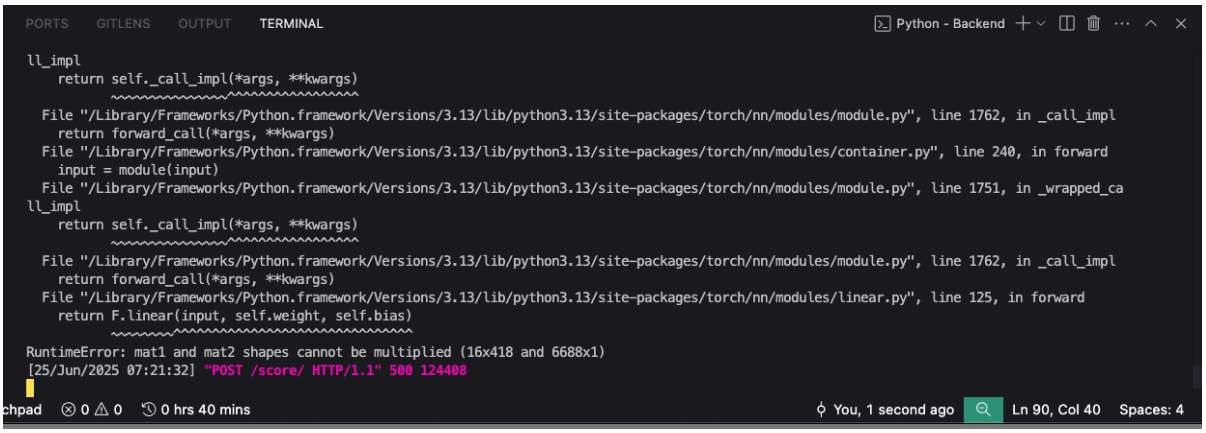
While the first sprint had minor errors occur, the second sprint - attempting to train and integrate the model into the mainstream application - did have some major complications.

When trying to convert the audio files received from the browser, the following ffmpeg error arose, causing the backend to throw an internal server when called by the Javascript of the frontend. The error was resolved after debugging with different scenarios and changing some settings in the command to be run by Python's subprocess module.



The next major issue occurred while training the model on the processed audio data and not seeing any significant changes in its loss. While this wasn't a typical error, it was an indication that the model was not training as desired. After several iterations of the training loop with altered learning rates and epochs, a sufficiently low loss of 0.1474 was reached.

The final issue occurred when trying to integrate the trained Siamese neural network with the remaining Django application. The frontend was successfully able to record and send pronunciation audio to the backend, when the backend ran the trained model on the given audio files (once processed the same way to produce tensors as in training), the following matrix multiplication error occurred:



While multiple attempts were made to resolve this shape mismatch, by altering the model architecture and ensuring that the preprocessing of received audio files was identical to that done in the training, due to a lack of time and expertise with similar models, this error remained and development had to be halted.