COMP4801 Final Year Project

Final Report

Learning Platform for Minority and/or Endangered Languages

April 2025

Member: Chen Jun (junchen4@connect.hku.hk)

Supervisor: Choi Yi King (ykchoi@cs.hku.hk)

Abstract

Many spoken languages across the world are at risk of decline. At the same time, Hongkongers are generally not knowledgeable about such conditions, and language-learning platforms generally only offer courses on major languages such as Chinese, French or Spanish. The platforms that do offer such courses created by the community are now rolling back their community support, which thus calls for a platform with strong community support, dedicated to learning lesser-known and/or endangered languages.

The project aims to provide a platform for those in Hong Kong or elsewhere who are interested in learning various endangered or minority languages around the world, to raise awareness regarding said languages in wider society, and give a voice to language enthusiasts who may have encountered issues promoting their own languages on other platforms. This was realized using a web application coded using the MERN Stack, with several auxiliary libraries imported for further support, be it for the backend or frontend. The four key features were all successfully implemented, albeit with several changes to the original plans. Due to technological limitations from MongoDB, images can only be stored locally on the device, rather than using an online storage. In addition, the limited time also does not permit a mobile version of the platform to be developed and launched simultaneously.

User testing revealed very positive feedback to the project, with its simplistic design and easy navigation being appreciated. Looking to the future, there is still some time before the platform can be launched. Several features may be added before the platform is launched.

Acknowledgement

The author would like to thank Dr. Choi Yi King for providing early comments on the scope of the project, and gratitude is expressed also to Wiktionary webmasters for making Wiktionary content publicly accessible for further use in different projects.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Background

Around the world, more than 7000 languages are spoken [1] by numerous groups. However, many of those languages are slowly being spoken less. Ethnologue estimates that 44% of languages - more than 3000 in real terms - are endangered today [2] .

In Hong Kong, societal and familial influence is perceived as more important to learning languages rather than self-motivation. In addition, students learning foreign languages are largely motivated to do so because they hope to achieve greater career prospects from it, or to emigrate to a country where that language is spoken. Less thought is given to self-betterment or improving one's knowledge of geopolitics and global affairs through linguistics [3] . In addition, Hongkongers generally do not speak many languages beyond the East Asian sphere [4] , showing their relative lack of linguistic knowledge in other regions of the world.

On the other hand, most language-learning platforms target people who are seeking to learn major languages, as platforms such as Babbel only offer courses in national languages [9]. These platforms therefore do not sufficiently highlight the situation faced by many endangered languages across the world. Furthermore, as other language-learning platforms such as Duolingo and Memrise grow, they have slowly strayed away from the principle of community contribution, thus leaving language enthusiasts with fewer places to contribute their local knowledge.

1.2. Scope of this report

This report will describe the features of the minority-language-learning platform, as well as the technology and procedures required to create the platform. It will also delve into the challenges and limitations of this project, that is, what the platform will not be able to achieve. To conclude, a recommendation will be made on the future steps to be taken.

1.3. Objectives & aims

The objectives of this project are:

- To use MongoDB to create a database of vocabulary and word attributes
- To create a database for storing users and their respective attributes and classes
- To create a visually appealing user interface using React
- To develop a minority-language-learning web application using the MERN Stack

On a more abstract level, the project seeks to aid those looking to broaden their horizons through linguistics. Furthermore, the platform encourages community contributions through creating lessons for various minority languages, and aspires to eventually establish a community of minority-language learners and enthusiasts in Hong Kong and beyond.

1.4. Deliverables

This project will deliver a minority-language-learning platform in the form of a web application, which will include features such as word-matching, an experience points (XP) system, and a community platform to submit languages.

1.5. Outline

The remainder of this report proceeds as follows. Firstly, a literature review will be provided in Section 2, further examining the points made in the background section. Next, the core features of the platform, the technology that had been planned to be used in the project, as well as how data may be collected and processed for use within the platform will be presented in the methodology in Section 3. Then, the final implementation will be presented in Section 4, including screens from the application itself, design principles underpinning the resultant aesthetics of the platform, as well as details on how each core feature was implemented. The exact procedures and results of user testing will also be discussed. To conclude, Section 5 will sum up the entirety of the project and look to the future and mention points for potential future work on the platform.

2. Literature Review

This section will review some existing literature relevant to the project. As mentioned in the background in Section 1.1, more than 40% of languages worldwide are at risk of decline or extinction.

However, this is not something that particularly concerns people in Hong Kong, as a 2005 study on tertiary students in Hong Kong finds that of the students learning French, German and Japanese, two main motivational dimensions for learning those languages were integrative (i.e. aiming to be like a native speaker or integrate into their culture) and instrumental (i.e. gaining success in work). For Japanese learners, the affective dimension was ranked above the integrative and instrumental dimensions, however the latter two also had a higher mean score than those two dimensions for French and German. Among the motivational dimensions included, two other ones were "macro-context", referring to the language attitudes of the wider community (within Hong Kong) and ethnolinguistic identity, as well as "micro-context", that is the language attitudes of peers and parents. It was found that with learners of French, German and Japanese, micro-context ranked above macro-context. In all three languages, the lowest-ranked dimension was "linguistic self-confidence" [3]. In addition, another study was conducted from 2014 to 2015 on the use of spoken languages among Hongkongers. Out of over 2000 respondents, while over 60% reported usage of English, Putonghua and Cantonese, less than 7% reported speaking other Chinese dialects, 5.2% spoke Japanese, 1.8% spoke French, 1.2% spoke Indonesian, and less than 1 percent reported speaking any other languages [4]. This shows that people in Hong Kong in general are neither fluent in many languages nor very interested to learn new ones for non-career purposes.

This could theoretically be remedied, as there are numerous language-learning applications on the market today, either in the form of mobile applications, web applications, or both. For instance, Duolingo is a language-learning application that was first publicly released in June 2012. [5] The platform is identified by a unique light green and white colour scheme, with a mascot owl named "Duo". Since its launch in 2012, it has quickly become by far the most popular language-learning application [6]. Notably, the platform has garnered much attention

due to its features seeking to “gamify” language-learning - with experience points, leagues, and lives which are lost when an incorrect answer is inputted, Duolingo seeks to portray learning languages as an online game. The languages featured in Duolingo are rather extensive, including many major national languages, as well as some lesser-spoken languages like Yiddish and Navajo, and even fictional languages like High Valyrian and Klingon. However, in general, Duolingo covers widely-spoken languages, mostly national languages such as French or Hindi. In 2013, the “Duolingo Incubator” was launched, where communities of speakers of a given language could come together to work on a new language course to then be pushed to the main application. However, in March 2021, this feature was removed, citing alignment with CEFR standards and the issue of making money off of volunteer-produced courses [7].

Another popular language-learning platform is Rosetta Stone. It is comparatively much older than Duolingo, having been established back in 1992 [8] , and is a helpful application for people seeking to learn major languages while getting an insight into the grammar, largely not provided in Duolingo. Languages offered include only national languages, such as Spanish or German. Babbel is a language-learning platform that was launched in 2007 in Germany. Many forms of online language education are provided, from podcasts to online small-group learning sessions [9]. Like Rosetta Stone, Babbel only offers courses in national languages. An internal group of 200 experts are responsible for creating the courses, and no volunteer-led material is used. In addition to Duolingo, one other platform that provides volunteer-made courses is Memrise, which even in 2012 had already accrued materials from over 100 languages [10]. Languages featured on the platform range from the Lingala language in sub-Saharan Africa, to indigenous languages from Taiwan, the USA, and Sweden. Memrise however also announced the closing of their community forums in late 2023 [11], and that community-created courses would move to an entirely separate website [12] .

All of the above reveal three major issues in Hong Kong and language learning: firstly, people in Hong Kong are not interested in learning languages in general; secondly, widely available language-learning platforms largely only offer courses in languages that are already widely-spoken, and thirdly, even platforms which offer courses in minority languages have limited

community involvement in their courses. All of this is occurring while more and more languages worldwide are in decline or even at risk of extinction, thus raising the need for a dedicated language-learning platform for minority languages.

3. Methodology

In the following section, the methodology with which this project was realized will be explained, with the initial planned implementation being compared to the final implementation. In Section 3.1, the core features of the platform will be given. Section 3.2 describes the technology stack used for the platform, as well as a detailed breakdown of the original planned database design. Section 3.3 will discuss the initial planned means of data collection for the platform, ahead of presenting the actual implementation in Section 4.

3.1. Platform features

Four major features are included in the platform, the first of which is word-matching. This is the primary means of language education on the platform, although short sentences are planned for the platform in the future. For the first few times a word appears, an accompanying image may be provided to hint at the English definition of the word, but afterwards a user must remember the meaning of said word. Where applicable, the word-matching may also come with relevant cultural facts.

This leads into the discussion of our second feature, which is an experience points (XP) and level-up system. For each lesson that a user does, they will receive 20 XP for completing that lesson. If a culturally significant word appears in said lesson, the user will receive 30 XP.

Once a user accrues a certain amount of XP, they will be able to level up, which may enable them to a selection of rewards. One such reward would be the third major feature - a personalized dictionary. This dictionary would store the words that have been learned by the user thus far, and would aid the user in memorizing the meaning of words. For the aforementioned culturally significant words, a description is provided for said word, in addition to its definition.

The final major feature is a community portal for speakers of lesser-known languages to submit their own language to the platform, as well as for users to see what languages other users have submitted to the platform. Thorough inspection will take place before a language is released publicly onto the platform to prevent vandalism or otherwise malicious behaviour. In addition, as words are sourced through Wiktionary, the language must have at least 200 entries on Wiktionary to qualify for approval and use on the platform. This minimizes the chance of under-attested languages being added to the platform when the documentation is not yet complete.

3.2. Technology

The MERN stack [13] was chosen to implement the platform. That is, MongoDB was used to create and host the database; Express was used for dynamically rendering the HTML page by passing arguments to templates; React was used to design the front-end architecture, as it provides flexibility and numerous modules for designing an aesthetically pleasing website; and Node.js was used to provide an interface between front-end and back-end architecture, and allowed both sides to be coded with JavaScript. The combination of the four technologies in the MERN stack allow for creating visually appealing web applications with interactive features, and is widely used across the web development field for a variety of projects.

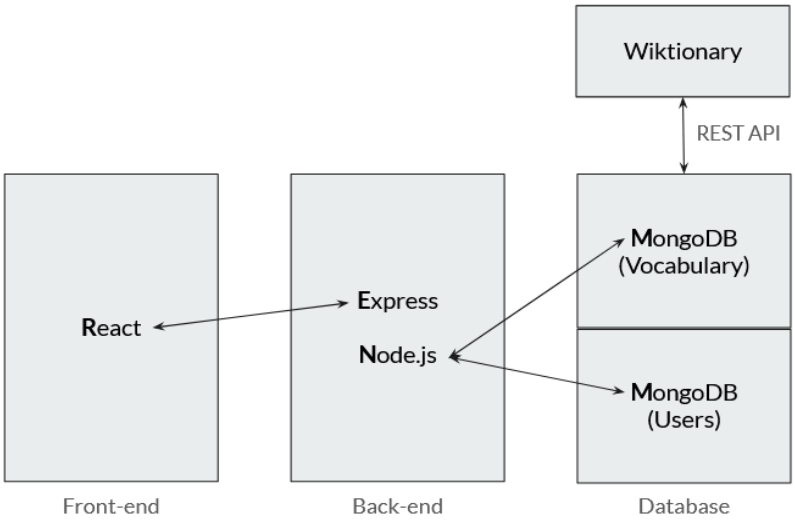

Fig. 3.1. The macro structure of the language-learning platform, using the MERN stack for the core infrastructure and a REST API to retrieve data from Wiktionary.

Furthermore, three different classes of objects were envisioned for the project, presented below in an SQL database diagram:

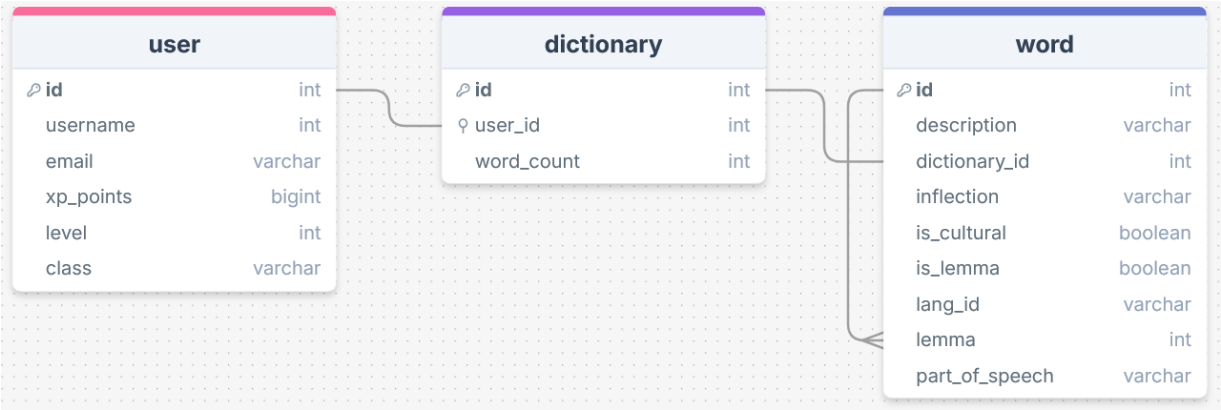

Fig. 3.2. The initial planned database diagram, displaying objects “user”, “dictionary” and “word”, as well as their respective parameters connected in SQL format.

In the “user” object, *xp_points* represents the number of experience points currently possessed by a user. *class* represents whether the user is a regular user or an admin. In the “dictionary” object, *user_id* refers to the ID of the user who owns that particular dictionary, such that the dictionary can be matched accordingly when the user enters the application. Other parameters are likely self-explanatory.

In the “word” object, *dictionary_id* provides the dictionary which holds this word. *inflection* refers to the form of the word. For instance, in English, the inflection of “ate” is simple past tense. *is_cultural* is a Boolean used to denote whether a word is considered culturally significant in that language, such as the name of a specific food, dance, or festival. This is used in the application to display such cultural words differently to generic words that do not necessarily carry cultural significance, say “car” or “water” in English. A lemma in linguistics is defined as a “word considered in its citation form” [14], or in other words, the base form that is used when considering a word with inflections or conjugations. For example, “cat” and “be” are lemmas in English, while “cats” and “am” are non-lemmas. The *is_lemma* parameter therefore indicates whether a word is in its base form, while the *lemma* parameter links back to the lemma entry if the current word is not a lemma. *lang_id* provides the two or three-letter Wiktionary ID of the language the given word is in. For instance, English is “en”, and French is “fr”. Finally

part_of_speech gives the part of speech of a word, that is, whether the word is a noun, adjective, verb, adverb, or something else.

3.3. Data collection

For this project, the bulk of the vocabulary and other data comes from Wiktionary. It is one of the largest databases containing words from numerous languages and dialects. While countless resources exist which document various languages, many have not been digitized properly, and are thus challenging to import into a database for the immediate purposes of the project. For instance, Latin encoding may be applied to a dictionary which contains both terms in the Latin script and terms in the Cyrillic script, which would make it impossible to search for terms written in the Cyrillic script.

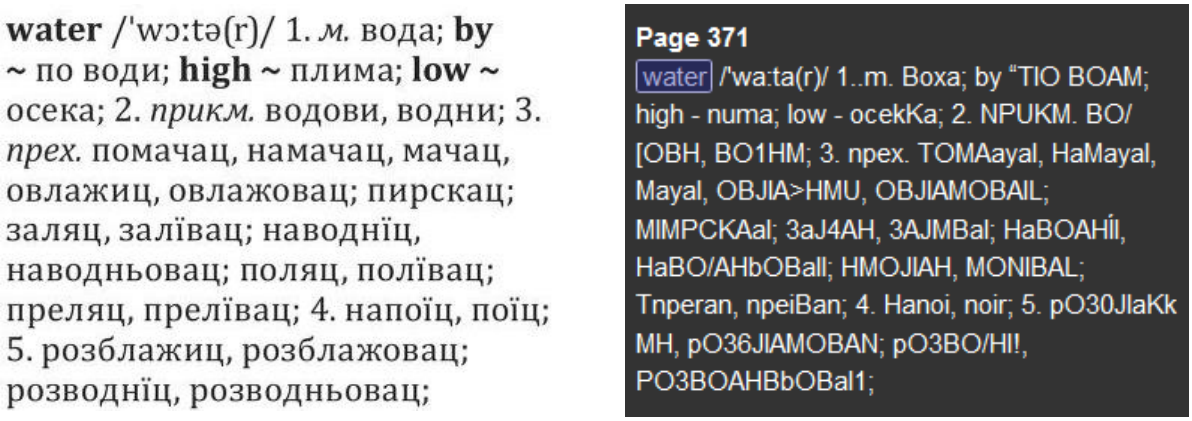

Fig. 3.3. An entry in an English-Pannonian Rusyn dictionary [14], with text in both Latin and Cyrillic scripts. This is contrasted with a search result on archive.org for the same entry in the dictionary, showing the Latin-only encoding used for both Latin- and Cyrillic-script text.

As time is limited for the project, Wiktionary is thus the most readily accessible and uniformly encoded database for a variety of language resources. It was initially planned that the Wiktionary API [16] and Wikimedia REST API [17] would be used to obtain words from Wiktionary for use in the platform. However, this was not possible, and it will be further expanded on in Section 4.1.3.

4. Findings & Discussion

In this section, the findings of the project will be presented and discussed. Firstly, the final implementation will be revealed in Section 4.1, detailing the exact processes that created the application. This is split into further sections discussing the technology used, screens of the design as well as design principles, and details on data processing. Section 4.2 lays out the procedure for user testing, and discusses its results. Section 4.3 takes a look at difficulties encountered throughout the project, and finally Section 4.4 will discuss the limitations of the project, or the reasons for which features cannot be implemented during the course of the provided project period.

4.1. Final implementation

4.1.1. Overall platform technology

As mentioned in Section 3.2, the MERN stack was chosen for this project, creating a full stack application. More specifically, the front-end was designed using a base of React and Vite infrastructure. The JSON Web Token library [18] was also imported for user verification, as well as dynamic access and modification of session storage. Bcrypt [19] was imported to encrypt users' passwords, such that they cannot be easily read by humans from within the database. In addition, to streamline frontend design, shadcn [20] was used for reusable, flexible and configurable components, while Tailwind [21] was used to enable inline CSS specifications, with minimal use of a separate CSS file.

4.1.2. Platform design

4.1.2.1. Walkthrough

When the user first enters the platform, they are greeted with the login screen, from which they can either login if they have an account, or create an account if they do not.

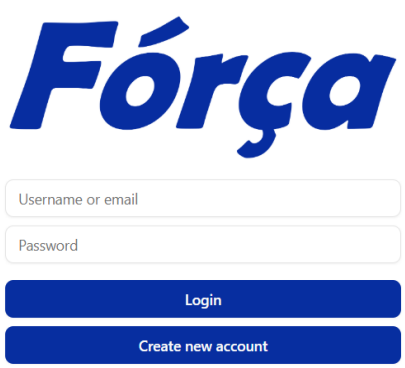

Fig. 4.1. A screenshot of the login page.

Fig. 4.2. A screenshot of the account creation page.

Once the user creates an account and logs in, they are met with the home page, from which they can select a language whose lessons they would like to try. On the top left of the page, to the right of the logo, an XP bar can be seen, displaying the user's current level, as well as XP within the current level and the amount of XP required to level up again.

Fig. 4.3. A screenshot of the home page, displaying a welcome message and some of the languages available for learning.

Fig. 4.4. A screenshot of the language page for Macanese, showing some of the lessons accessible to the user.

In each lesson, there will be a set of questions, each of which has 4 possible answers. Depending on the word, there may also be an accompanying graphic or image to describe the concept. Correct answers on culturally significant words are marked in purple.

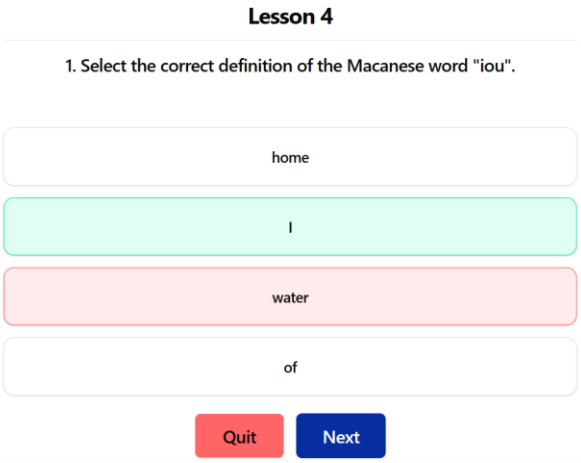

Fig. 4.5. A screenshot of a question with an incorrect answer.

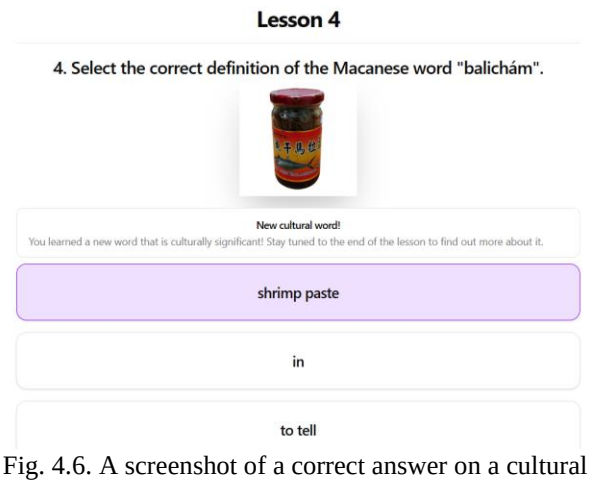

Fig. 4.6. A screenshot of a correct answer on a cultural word.

Once a user reaches level 5, they may access their personal dictionary, where words are first sorted by alphabetical order of the language, then by alphabetical order within the language. Culturally significant words are once again highlighted in purple.

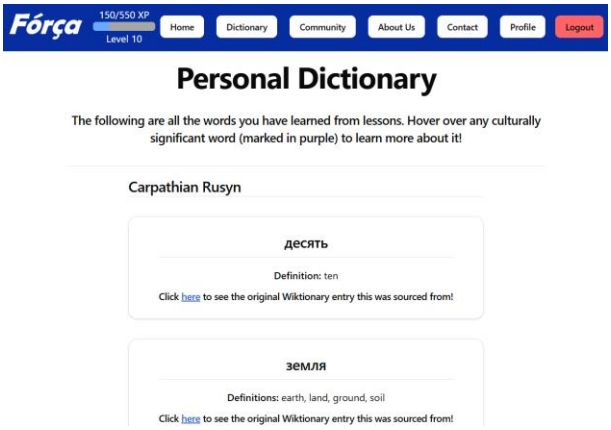

Fig. 4.7. A screenshot of the top section of the personal dictionary.

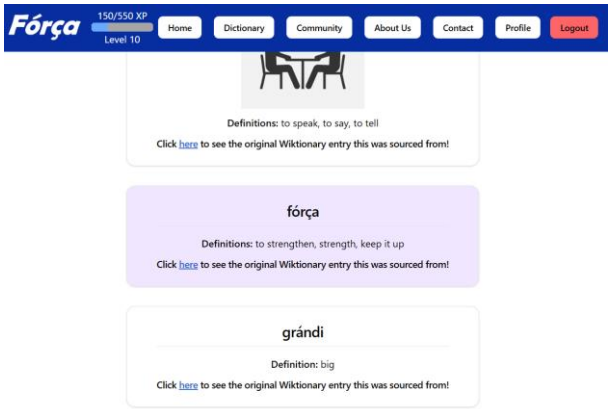

Fig. 4.8. A screenshot of a cultural word highlighted in purple, displayed in the dictionary.

Users who speak or are very passionate about a certain minority language may access the community portal, where they can check other users' submitted languages, or make their own submissions.

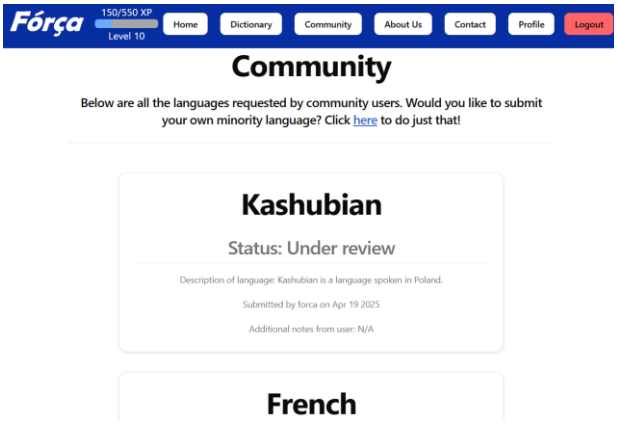

Fig. 4.9. A screenshot of the community portal, displaying Kashubian and French as requested languages.

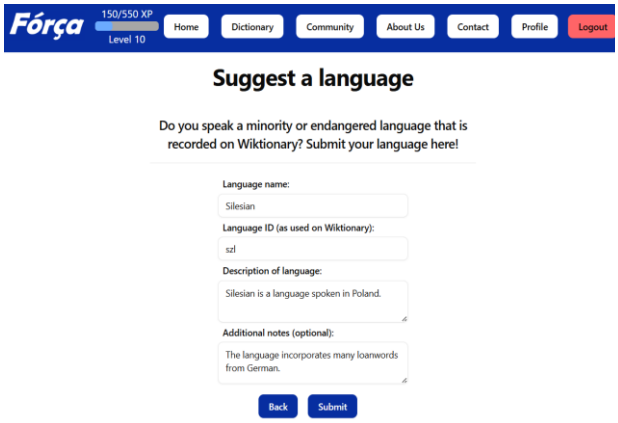

Fig. 4.10. A screenshot of the language suggestion page, with the user intending to suggest that Silesian be added to the platform.

Otherwise, the user may choose to browse other pages on the platform, such as the About Us page or the profile page. The Contact page is relatively simplistic and only includes a line with an email address for contact, thus a screenshot will not be provided here.

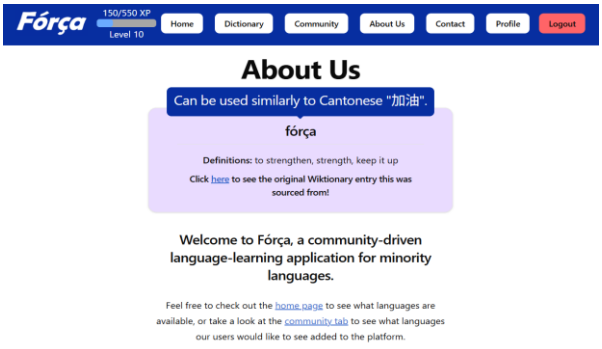

Fig. 4.11. A screenshot of the About Us page on the platform, displaying the word "fórça", the namesake of the platform, as well as a brief introduction to the premise of the language-learning platform.

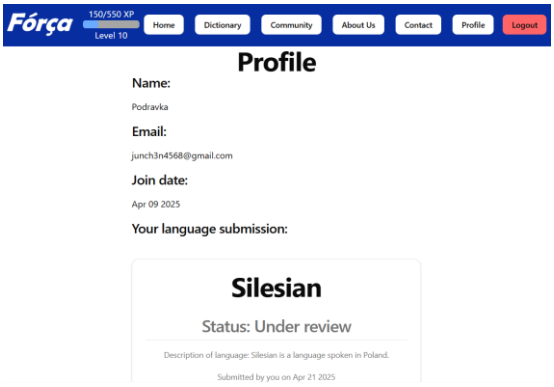

Fig. 4.12. A screenshot of the profile page, displaying the username, email, join date, and language submissions of the current logged-in user.

4.1.2.2. Theming

The theming has deliberately been made relatively simplistic in order to minimize clutter and enforce uniformity across the platform.

Firstly, the platform has been named “Fórça”, pronounced “FOR-suh”. This is a Macanese word which derives from Portuguese, with the word sharing a common origin with “force” in English. The Macanese word can have multiple meanings; it can be used as a verb, to mean “to strengthen”; it can be a noun meaning “strength”; or the word can be used as an interjection of encouragement, similar to “keep it up” in English, or 「加油」 in Cantonese [22]. The idea is that this language-learning platform can be used to “strengthen” endangered languages, and to say “keep it up” to minority-language speakers in their efforts to preserve their language, which can be done through this platform by suggesting their languages on the community portal as well as having their languages be learnt by other users on the platform. The logo was created by choosing a supported font and writing the word “Fórça” on a blue background in Microsoft PowerPoint, and cropping out the resultant graphic as an image and importing it into the application. The opposite was done for the logo on the login and account creation page, which has blue on white instead of white on blue.

This brings us to the colour scheme of the application, which is white and a darker blue. As mentioned in a previous report, this platform aims to minimize comparisons with Duolingo, which also utilizes word-matching as a primary means of learning languages. Therefore, as Duolingo primarily uses a colour scheme of white and light green, a darker blue was selected for Fórça instead. The sharper contrast also brings more visual attention to the page.

4.1.2.3. Design principles

For the design of the platform, Nielsen’s usability principles [19] were consulted. These 10 usability principles were first created in 1990 by Jakob Nielsen as a set of guidelines for digital interface design, further refined in 1994 and re-clarified again in 2020. The usability heuristics are as follows:

1. Visibility of System Status
2. Match Between the System and the Real World
3. User Control and Freedom
4. Consistency and Standards
5. Error Prevention
6. Recognition Rather than Recall
7. Flexibility and Efficiency of Use
8. Aesthetic and Minimalist Design
9. Help Users Recognize, Diagnose, and Recover from Errors
10. Help and Documentation

These heuristics help improve usability of the page, reduce user confusion and increase average user retention, that is, their time spent staying on the site. And for the design of this project, while all heuristics have been taken into account, at this stage some of them are more applicable than others. For instance, Help and Documentation stipulates that help content and documentation should be concise and easy to find for more complex pages or tasks, however the language-learning platform is quite straightforward in its design and does not require documentation at a user level. Other heuristics may only partially apply to the platform, in part due to its simplistic design. Therefore, heuristics 1, 3, 4, 5 and 8 are the main focus, which will be further discussed individually below.

Visibility of system status stipulates that the design should keep users aware of the current system status, i.e. where the user is in the site. And as seen from the screens in section 4.1.2.1, all pages, minus the home page, have large text on the top reminding the user what the current page is. This ensures that users are aware of their current location within the platform, even if the current page is not listed in the top navigation bar, such as lessons or the language submission component of the community portal.

As for user control and freedom, Nielsen refers to the need for there to be an “emergency exit” for users to exit out of unwanted actions. In the project, Fórça allows users to exit from lessons at

any point. They may either click the “Quit” button below the options to return to the language page, or they may click onto any page in the navigation bar, both of which will automatically remove any progress made on the lesson so far. Other pages also have similar “Back” buttons to allow users to return to a previous page, such as going from the language submission page back to the community portal.

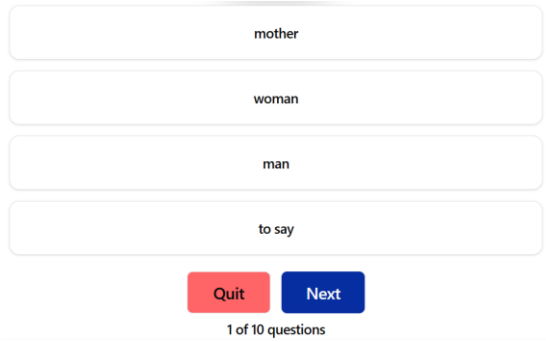

Fig. 4.13. A screenshot of the lower part of a lesson question, clearly displaying a “Quit” button to allow for exiting out of a lesson.

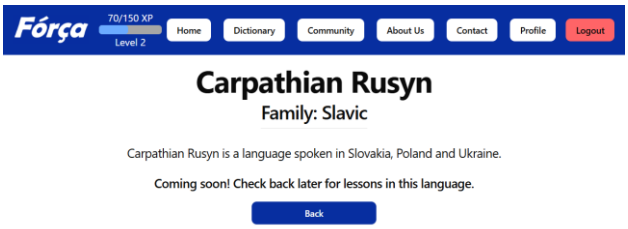

Fig. 4.14. A screenshot of the Carpathian Rusyn language page, where the “Back” button is clearly visible and will bring users back to the home page when clicked.

Consistency and standards can be divided into two sections: external consistency and internal consistency. For external consistency, industry standards are employed in both the design of common objects and in the naming of functions within the code. A `UseState` variable which accesses the user for example would be named `[user, setUser]`, while a function dealing with the value change in input boxes would be named `handleChange` in the application. Meanwhile, internal consistency is maintained by using the same typeface and colour scheme across the platform, as well as the navigation bar being visible from all pages of the app once logged in. In the code, different files fulfilling similar functions are also named similarly. For instance, the backend file handling requests involving the user object is named `userRoutes.js`, while the backend file handling requests involving the dictionary object (shortened to “dict”) is named `dictRoutes.js`.

For error prevention, any inputs made on the platform are checked for invalid inputs. For instance, all fields must be filled for a user account to be created. Furthermore, the email field is subject to a regex check, such that the inputted address contains three groups of text separated by an “@” sign and a dot, following the standard format of an email address. The username and

email are also checked for duplicates within the database, so users cannot create an account with the username and/or email address of another account. For the language submission page, it is required for users to input the language name, language ID and a short description of the language. Character limits are also put on specific fields. The language ID for example uses the Wiktionary ID, which does not exceed three digits, and is thusly reflected on the language submission page. In both cases, this ensures that all inputs made are valid and will not cause display issues later on. A user cannot, for instance, input a username of 100 characters in length, or an email address with no "@" sign.

The aesthetic and minimalist design, although first conceived by Nielsen in 1990 ahead of the age of cluttered HTML sites in the 2000s, was written ahead of its time, as simplistic website and app designs are once again in vogue. In line with this trend in interface design, the platform design strives to not display any more information than is necessary, with any extra elements being minimal and conducive towards the simplistic design, such as short descriptions for the function of a page.

4.1.3. Platform content

It was decided that Macanese would be the first language added to the platform. Macanese is a creole language formed from the combination of Portuguese, Malay, India and Cantonese influences, and is spoken by a small Macanese minority in the city of Macau today, as well as a Macanese diaspora abroad in Portuguese-speaking countries such as Brazil or Portugal [20]. Occasionally, plays are still performed in the Macanese language, such as "Vêm di volta pa jantâ", which was performed in November of 2024 [21]. However, it is estimated that fewer than 20 native speakers of Macanese remain today, most of them being elderly people.

This language has been chosen to add to the platform because taking into account that the primary demographic for this platform is Hongkongers, Macau has a similar cultural background to Hong Kong, and Macanese itself borrows quite significantly from Cantonese which is widely spoken in both Macau and Hong Kong. In addition, the simplistic grammar of Macanese allows for a simpler tentative implementation of word-matching and display in dictionaries.

4.1.4. Data collection & processing

As mentioned in Section 3.3, it had been planned to use the Wiktionary REST API to obtain words from Wiktionary. However, its functionality only permitted accessing one word at a time. Furthermore, each word required the addition of different properties and parameters, and this would not be possible based on an on-request retrieval system. It was therefore decided to use a service to download Wiktionary words, and manually import them into MongoDB. This was done using a Wiktionary JSON parser [22], created by user TallonKH. Wiktionary XML dumps are publicly accessible, and a Python-based JSON parser was used to select words from desired languages and output a JSON file with a predetermined format.

Several issues were encountered in this process. Firstly, the JSON parser used was 7 years old as listed on GitHub. Possibly as a result of this, the outputted JSON file was incomplete, and had missed almost 400 of the originally desired 700 words. Secondly, the output format was not as desired. Words were grouped by each language having its own JSON object, rather than each word having a parameter indicating the language. While words would be grouped by language in the dictionary, this was not desirable elsewhere on the platform. The JSON file therefore had to be manually edited using a text editor to rectify this issue.

The first issue, on the other hand, did not turn out to be very detrimental, as the majority of the words did not end up being used in lessons. In the final phase of data processing, all words would have specified parameters added to them, and the JSON file containing all words would be imported into the MongoDB server. Each word was first evaluated for cultural significance, with culturally significant words being assigned “true” in the *is_cultural* parameter. More parameters had been planned for the word object, such as *lemma* and *part_of_speech* as mentioned in Section 3.2, however, to simplify the display and prevent confusion with different parts of speech, this was not done. In addition, in Macanese grammar, words are not inflected or conjugated, therefore all Macanese words are “lemmas” and do not require linking to a lemma entry.

Afterwards, the words were curated for use in Macanese lessons. This means selecting words which have a common theme, and/or are suitable for a certain phase of learning a language, and including them in specified lessons. For the ten lessons implemented, 58 words were curated, mostly basic words such as the words for “father”, “mother”, “house”, “flower” and “happy”. Most such words were selected according to the Swadesh list, which is a list of 200 stable concepts, originally used to date the divergence of a language from its ancestor [23], but also happens to be mostly basic human concepts.

Each lesson will have progressively more words available. For Macanese in particular, the first lesson only has five words, while the final lesson has 58 possible words and definitions at its disposal. Therefore, many words still remain un-curated in the database, and if important basic words were missing from the JSON parser, they were added in manually. Other missing words from Wiktionary will be added in at a later date for use in more advanced lessons.

Several of the words that were curated were also assigned with an image. In line with gamification, most images were colourful vector graphics, often stock images, found on various online graphic databases. These images were then cropped into a square shape, reduced to 250 * 250 pixels, and inserted into the “images” folder in the project directory, which would then be called when the word was selected either in a lesson or in the dictionary. Images were only provided for more concrete concepts, such as “talk” or “boy”; concepts such as “you”, “yes” or “hello” are more difficult to express in abstract graphics without turning to text or being confused for another concept – for example, if a thumbs up icon was used for “yes”, then when learning the word for “yes” in a minority language, the graphic could be misinterpreted as “good”; or a handwaving graphic could be reasonably interpreted as *both* “hello” and “goodbye” – therefore those words were not assigned with any images.

More objects were also introduced for implementing different features. In addition to the initial *user*, *word* and *dict*, the objects *submission*, *language* and *lesson* were also added. Therefore, the collections in the final database are as follows. Note that despite the SQL-style presentation, the

database was implemented in MongoDB, and the attributes are only “connected” with one another using front-end probes.

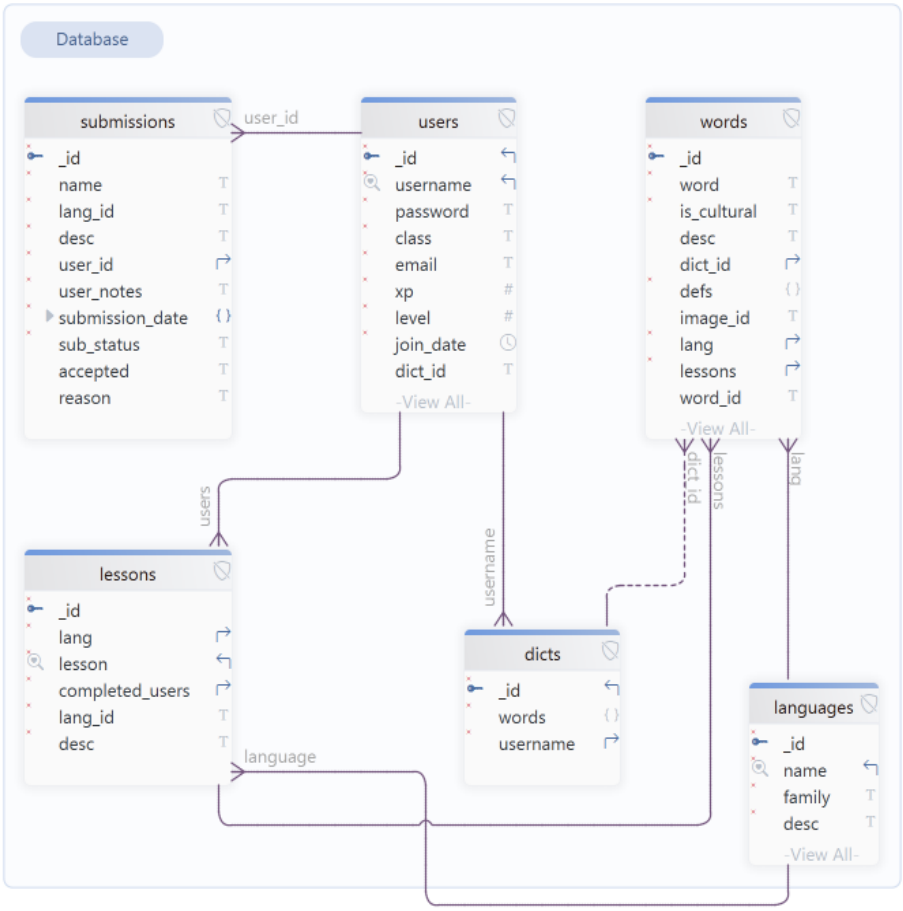

Fig. 4.15. The final implemented database, presented in a SQL-style diagram, showing the collections “submissions”, “users”, “words”, “lessons”, “dicts” and “languages”.

Note that in the “words” collection, the parameter *image_id* does not point to the actual ID of an image object, but rather simply lists the full name of an image used for the word. The Macanese word for “man”, for instance, has “man.png” as its *image_id*. Instead of hosting the images online, the current implementation simply stores the images within the “assets” directory in the project folder, and loads them from there when requested. The reasoning for this will be discussed in Section 4.4. In addition, *defs* is short for “definitions”, and is an array of strings which describe the different definitions of a word. *lessons* is an array of integers, and the numbers represent the lesson which a word is in. For example, if a word can be found in lessons 8, 9 and 10, then the word’s *lessons* parameter will return [8, 9, 10].

For the “languages” collection, it is used for display on the home page. *family* refers to the classification of the language, such as Slavic, Sinitic, Germanic, and so on.

In the “lessons” collection, *lesson* is an integer which refers to the number of lesson that this Lesson object is. `{'lesson': 1}` means that this is the first lesson in the language for instance. *completed_users* is an array of user IDs, and helps to keep track of whether a user can access a lesson, as seen below.

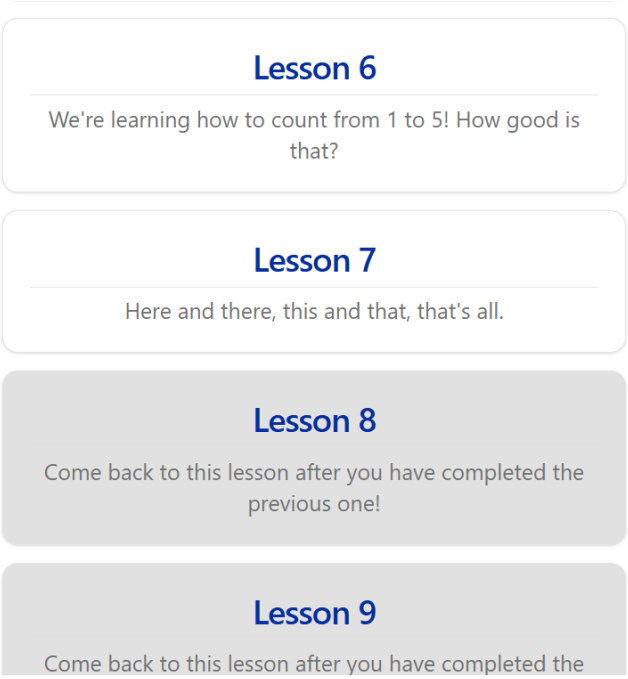

Fig. 4.16. A screenshot displaying two unlocked lessons and two locked lessons. Here, the user has completed lesson 6, which unlocked lesson 7. However, lesson 8 remains locked, as lesson 7 has not been completed yet.

In the “dicts” collection, *words* is an array of all the IDs of words rather than the words themselves. This makes it easier to look up and filter words for display.

Finally, in the “submissions” collection, *sub_status* is a parameter which determines whether the language has been approved for adding to the platform, rejected, or still under review. By default, when a new submission is sent, this parameter is set to “under review”.

4.1.5. Word-matching

The primary mode of learning minority languages on the platform is word-matching. This is implemented in two ways:

1. The word in the minority language is provided, and the user is prompted with four English words (or phrases), with one of them being the correct definition of the word. This mode has a 70% chance of being used for a given question. This increased percentage was chosen, as this encourages users to actively focus on the minority language word provided, rather than be distracted by three other such words.
2. An English word or phrase is provided. The four options are different words in the minority language, and the user must choose the correct one. This mode has a 30% chance of being used for a given question.

The lessons are entirely randomly generated, meaning that each lesson may have entirely different groups of words depending on the amount of curated vocabulary. Firstly, the word being tested is selected randomly. Then, the question mode is selected randomly, with a 70/30 ratio as mentioned above. The correct answer is added to the array of answers, and then three more incorrect answers are randomly selected to complete the array of four. This answer array is then shuffled, so that the correct answer is not always the first one in the list. In terms of numbers of questions, lesson 1 only consists of five questions as there are only five curated words for that lesson, but all lessons from lesson 2 and onward have 10 questions per lesson, regardless of the number of total curated words. However, each question can have three wrong answers chosen from any of the curated words for that lesson. Under the current curation scheme, all words from previous lessons are retained for lessons of a higher number, that is, a word from lesson 1 could appear in lesson 10.

Culturally significant words begin appearing in the Macanese lessons starting from lesson 4. They are no more likely than any other word to appear in the lesson; however, if the cultural word question is answered correctly, the user will receive 30 XP at the end of the lesson instead

of the usual 20, provided that the user gets an overall score of 50% or higher in that lesson. If that cultural word is new to the user, then the word is revisited at the end of the lesson.

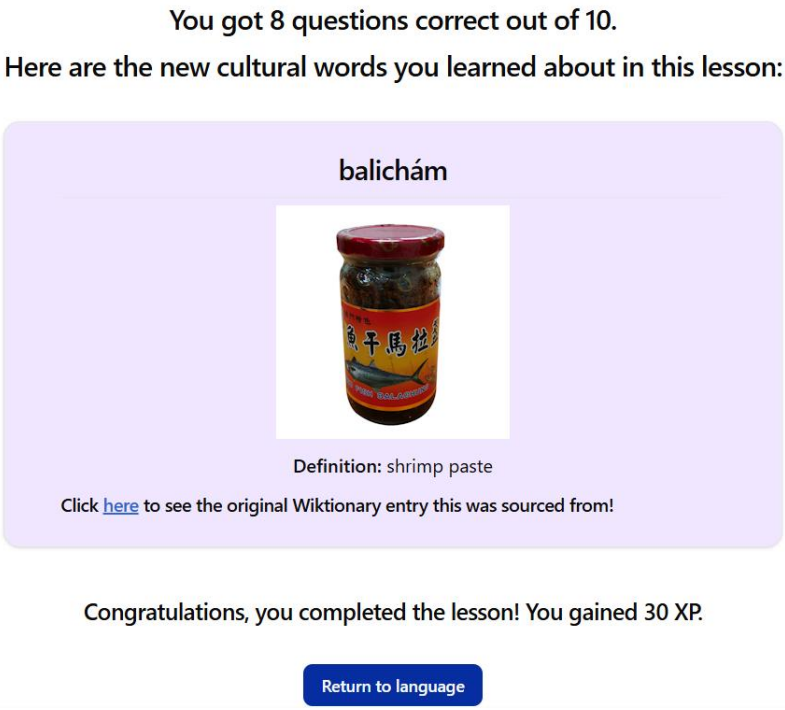

Fig. 4.17. A screenshot of a user obtaining 30 XP and a new cultural word at the end of a lesson.

4.1.6. Experience points and level-up system

In line with gamification principles, a level-up system was implemented, with the user gaining experience points every time a lesson is successfully finished. This is defined as correctly answering 50% or more of the questions in that lesson. For each level, the user will have to earn 50 more XP than in the previous level. Thus the experience points and level grading, up to Level 10, is as follows:

Table 4.1: a table describing the experience points required for a user to reach a certain level.

Level	Experience points	Cumulative XP
0	0	0
1	50	50

2	100	150
3	150	300
4	200	500
5	250	750
6	300	1050
7	350	1400
8	400	1800
9	450	2250
10	500	2750

4.1.7. Personal dictionary

Once the user reaches 750 total experience points, or in other words, the user reaches level 5, they will be able to access their personal dictionary. Every time the user correctly matches a word and its definition, the word will be added to the dictionary, regardless of whether the user fails or passes the lesson. The dictionary is created upon user account creation, and will collect words even when the user is not yet at level 5; however, this dictionary will be obscured from the user's view until the user has reached level 5.

To obtain the display of the dictionary, the system first gets the user's dictionary by filtering through usernames. Then, all words are obtained in a big array using an Axios GET command. Using the *words* array, those words are filtered down into an array of words which are only in the dictionary. Finally, words in the dictionary are first sorted by alphabetical order of the language, then by alphabetical order according to the rules of the language, as seen below in Fig. 4.18.

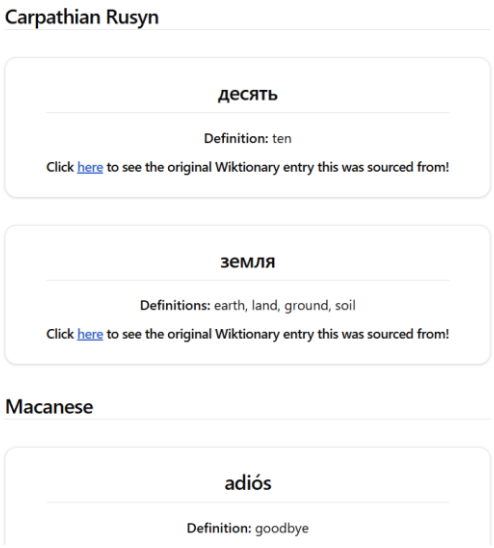

Fig. 4.18. A screenshot of the personal dictionary, where the language “Carpathian Rusyn” is sorted before “Macanese”.

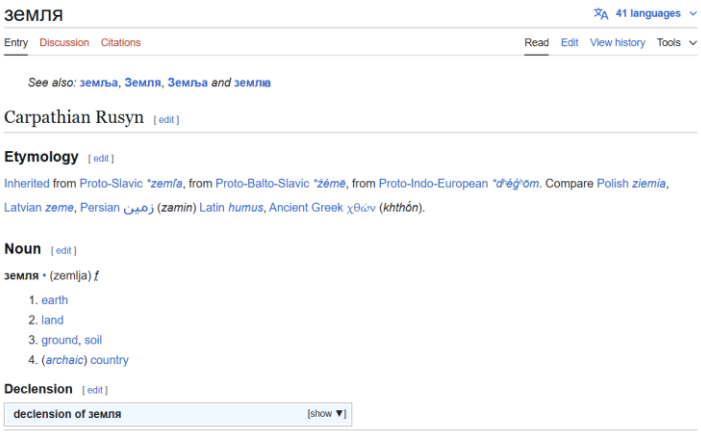

Fig. 4.19. A screenshot of the Wiktionary entry for « земля », which results when a user clicks onto the “here” portion of the card for each word.

And as seen in an earlier screenshot, culturally significant words are also highlighted in purple in the dictionary, not only in the lessons.

4.1.8. Community portal for submitting languages

Speakers of endangered languages or advocates for them may request for languages to be added through the community portal. This feature was inspired by the now-defunct Duolingo Incubator, whose features were mentioned in Section 2. Implementation of the feature was quite straightforward, with users first greeted by a list of current submissions as obtained from the *submissions* collection in the database. Users can then add to that database by clicking a link in the introductory text of the portal, which brings them to a form with parameters for the name and ID of the language, a mandatory description box, and an optional notes section for if the user would like to include additional details regarding the language, such as historical influences, or even suggest their own culturally significant words. Due to the limited personnel, a detailed scheme for evaluating whether a language should be added to the platform has not yet been devised.

4.2. User testing

4.2.1. Procedure

After the vast majority of the platform was completed, people were invited in person to take part in user testing. First, they were required to answer a questionnaire about their daily language usage habits, as well as their views towards minority languages and language-learning platforms. Then, the premise and broader aims of the project were introduced to the test participant. User testing was performed afterwards, where with minimal instruction, they were required to perform four tasks, corresponding to the four key features, which are as follows:

Table 4.2: a table describing tasks undertaken by testers during the user testing phase.

Given condition	Task	Success criteria	Notes
The tester has never done a lesson.	Pass lesson 1 in Macanese.	The lesson is passed in fewer than 5 attempts.	Lesson 1 has only five words compared to later lessons.
The tester is currently at level 0.	Reach level 1.	The user sees on the XP bar on the top left of the navigation bar that it displays “Level 1” instead of “Level 0”.	Each lesson gives at least 20 XP. Level 1 is achieved at 50 XP, therefore three lessons are required to reach level 1.
The tester is at level 5 and can access the dictionary. In addition, lesson 4 in Macanese is unlocked.	Find a cultural word in the dictionary and check its description.	The user finds the word “balichám” and learns that it is a condiment often used in Macanese cooking.	The word “balichám” first appears in lesson 4. In addition, all cultural words are marked with purple, both in the lesson and in the dictionary.

The tester has never submitted a language.	Submit a language to the community portal.	The tester can see their own language submitted.	

After the test was complete, participants were requested to fill out a post-test questionnaire, where they would provide their feedback on the design and ease of use of the platform. Finally, they were asked to give one suggestion to improve the platform, as well as to provide any other comments regarding it.

4.2.2. Results

From the pre-test questionnaire, it was found that most of the testers are of ages 25 or under. 56% of the users listed themselves as having been raised in East Asia, whether in Hong Kong or elsewhere. 94% of the participants had an education level of secondary/high school or above. 87.5% of the users spoke at least two languages at a conversationally fluent level. For the minority language speakers, when asked to rate their frequency of using said minority language, a plurality of them responded that they used it every day, with the majority of participants also using it more often than not, giving a rating of 3 or higher in a scale from 0 to 5. In the agree-disagree questions, the vast majority agreed that it was important to conserve endangered languages, and that existing language-learning platforms do not do enough to preserve such languages. However, 25% of participants responded that they “strongly disagree” that they learn languages in order to learn more about different cultures of the world. This matches with the problem outlined in Section 2, where many people learn languages for career prospects rather than to learn about different countries and lifestyles across other continents.

All testers were able to successfully complete the 4 required tasks with little confusion. The task that took the most time was the second task, as they had no knowledge of Macanese and took time to remember the words and their respective definitions.

In the post-test questionnaire, the vast majority of respondents either agreed or strongly agreed that the design of the platform was intuitive and easy to navigate, and that the key features (such

as the dictionary or community portal) were easy to access. When prompted for suggestions, some suggested for the personal dictionary to be unlocked at a lower level than level 5, as well as a map to see where the different languages are actually spoken. In the final comments, several participants commended the platform for filling an important niche, and “solved real-life issues”.

Therefore, to sum up, testers found the design of the platform to be well-made, and thought positively of its unique role amongst existing language-learning platforms. The detailed results, as well as the Google Forms used for the pre- and post-questionnaires, may be accessed in the Appendix.

4.3. Difficulties

The main difficulty relates to displaying languages. There are numerous scripts used to write languages across the world: Latin, Cyrillic, Greek, Arabic, and so on. Some of these scripts have not been encoded into Unicode yet, and so those languages may only be covered in passing due to technological limitations. Even if the scripts themselves are displayable on electronic screens, there may be individual characters which may not be supported. For example, the Cyrillic “tje” letter, used in the Khanty language, cannot be displayed on Wiktionary even on modern computers running Windows 10 at the time of writing this report [24], as shown in Fig. 4.20.

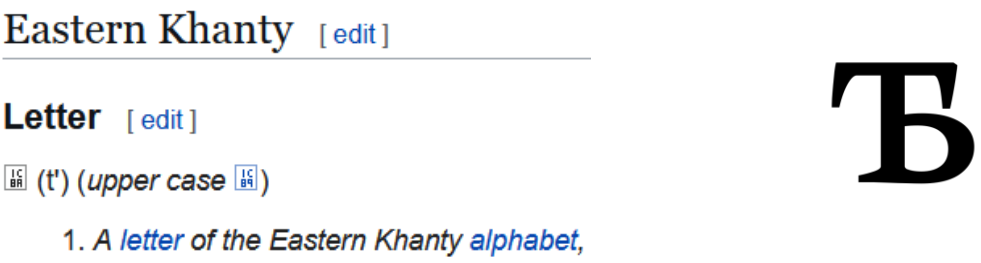

Fig. 4.20. The letter “tje” displayed on Wiktionary. On the right, the actual shape of the letter, as used in the Khanty language [25] .

This issue is only exacerbated with antiquated systems. In addition, some typefaces, created with English in mind, may not support accented letters. Fortunately, the default font from shadcn was able to display both Latin and Cyrillic lettering correctly. For future languages in different scripts, caution will be taken when choosing a suitable typeface. Otherwise, display issues may occur if the wrong typeface is selected.

Even if the correct or a suitable typeface is selected, there may still be issues with sorting words. As mentioned in Section 4.1.7, words are sorted by alphabetical order. However, this can be wrong, based on how different languages sort words. For instance, in English, letters with accents are sorted *after* the letter Z. Consider this sample of Macanese words below.

adiós, ali, dôs, falâ, águ, êle

These words are sorted by the English alphabetical order. However, Macanese uses the Portuguese alphabetical order, which treat accented letters the same as accent-less letters. The correct order of the above words is thus as follows:

adiós, águ, ali, dôs, êle, falâ

This can be done using the `localeCompare` function, as seen here:

```
userWords.sort((a, b) => a.word.localeCompare(b.word, 'pt'))
```

Fig. 4.21. A sample code, using the localeCompare function to correctly sort Macanese words by alphabetical order. "pt" in this case refers to Portuguese, and this uses the Portuguese locale for sorting.

4.4. Limitations

One major limitation is that it is not possible within the allotted project development time period to create a mobile equivalent of the platform. If such an equivalent were to be made, it would have to go through a nearly entirely different design process. Both forms would have to undergo rigorous user testing - especially with both Android and iOS for the mobile version - before being pushed out at the same time for the project, which is impractical based on the limited personnel working on the project.

The second limitation is that sign languages cannot be covered with the platform, as concepts of words and morphemes cannot be effectively expressed in a text-based format focussed on word matching as a primary feature. This is also a limitation encountered by the aforementioned major language learning platforms. There certainly exist endangered or minority sign languages, and there do exist dedicated platforms for learning sign languages such as Sign It ASL [26] or Sign Language 101; however, the mode of presentation is much more visual rather than textual, and is massively different to platforms providing largely text-based lessons on written languages.

Another technological limitation is that MongoDB is a text-only database, and cannot store images. This is the reason that images for individual words are simply stored in the project folders, rather than being stored on the cloud. Online storage of images can be configured using AWS (Amazon Web Services); however, as AWS is also used for deployment of web applications, it may be more preferable to implement online image storage when the platform is ready for full deployment. Note that AWS only changes the method by which images are imported, and not their display on-screen, therefore changes in code would be relatively minor.

5. Conclusion & Future Work

This report presents a learning platform for minority and/or endangered languages, given the current sociopolitical context where a sizeable percentage of languages are at risk of decline or extinction, in addition to the general low level of knowledge or interest regarding global languages across Hong Kong society as a whole. In addition, existing language platforms are overly focussed on teaching national and/or widely-spoken languages, and the platforms that do provide community-run minority language courses have been rolling back their community support as of late. This thus creates an opportunity for a dedicated learning platform for languages that are endangered or going into decline, which among other features, provides for speakers of said languages to submit their own languages and courses onto the platform.

The project aims to provide a platform for those in Hong Kong who are interested in learning various endangered or minority languages around the world, to raise awareness regarding said languages in wider society, and give a voice to language enthusiasts who may have encountered issues promoting their own languages on other platforms. In more concrete terms, the platform, now named “Fórça”, was realized through a web application coded using the MERN Stack, including a database for users and a database for vocabulary created using MongoDB, and an attractive user interface using React, shadcn and Tailwind. The four key features – word-matching, the experience points and level-up system, personal dictionary and community portal – were all successfully implemented, albeit with several changes to the original plan as set out in the Interim Report due to various reforms in the architecture. Nonetheless, user testing revealed

very positive feedback to the project, with its simplistic design and easy navigation being appreciated. Due to technological limitations from MongoDB, images can only be stored locally on the device, rather than using an online storage. In addition, the limited time also does not permit a mobile version of the platform to be developed and launched simultaneously.

Looking to the future, there is still some time before the platform can be launched. Firstly, sentence-matching can be implemented in addition to word-matching. Second, an in-game chat and friend system can be added, such that different users can friend request one another and discuss different languages. An AI chatbot could possibly be added to facilitate users in searching for content across the site, or to perform searches in the ever-growing personal dictionary. Finally, when nearing a full launch, AWS can be used to store graphics and images online, and then be used for full deployment complete with a unique domain name.

6. References

The following is the list of references.

[1] “How many languages are there in the world?,” Ethnologue, https://www.ethnologue.com/insights/how-many-languages/ (accessed Oct. 14, 2024).

[2] “How many languages are endangered?,” Ethnologue, https://www.ethnologue.com/insights/how-many-languages-endangered/ (accessed Oct. 14, 2024).

[3] G. Humphreys and M. Spratt, “Many languages, many motivations: A study of Hong Kong students’ motivation to learn different target languages,” *System*, vol. 36, no. 2, pp. 322, Jun. 2008. doi:10.1016/j.system.2007.09.010

[4] J. Bacon-Shone, K. Bolton, and K. Luke, “Table 3.1 Knowledge and use of spoken languages by percentage”, in *Language Use, Proficiency and Attitudes in Hong Kong*, Hong Kong: Social Sciences Research Centre, the University of Hong Kong, 2015, pp. 18

[5] S. Loeb, “When duolingo was young: The early years,” VatorNews, https://vator.tv/news/2018-06-22-when-duolingo-was-young-the-early-years (accessed Oct. 15, 2024).

[6] Straits Research, “The best language learning apps of 2024 by downloads.,” Straits Research, https://straitsresearch.com/statistic/the-most-popular-language-learning-apps (accessed Oct. 15, 2024).

[7] Matt, “What happened to the Duolingo Incubator?,” duoplanet, https://duoplanet.com/what-happened-to-the-duolingo-incubator/ (accessed Oct. 15, 2024).

[8] “About Us,” About Rosetta Stone® | Company History and Mission, https://www.rosettastone.com/about/company/ (accessed Oct. 15, 2024).

[9] Babbel GmbH, “It starts here,” Babbel, https://www.babbel.com/about-us (accessed Oct. 15, 2024).

[10] D. Baker, “Crowdsourced Mnemonics: How Ed Cooke teaches languages with memrise,” Wired, https://www.wired.com/story/crowdsourced-mnemonics/ (accessed Oct. 15, 2024).

[11] “Closure of the MEMRISE Community Forum,” Memrise Classic, https://memrise.zendesk.com/hc/en-us/articles/20736101603473-Closure-of-the-Memrise-Community-Forum (accessed Oct. 14, 2024).

[12] Memrise Team, “Major update: A new version of the app is coming!,” Memrise, https://www.memrise.com/blog/major-update-a-new-version-of-the-app-is-coming (accessed Oct. 15, 2024).

[13] “Mern stack explained,” MongoDB, https://www.mongodb.com/resources/languages/mern-stack (accessed Oct. 15, 2024).

[14] “Lemma definition and meaning,” Collins English Dictionary, https://www.collinsdictionary.com/dictionary/english/lemma (accessed Apr. 20, 2025).

[15] Fejsa, M., Šlemender, M., Čel′ovski, S. “water”, *Английско-руски словнїк* [English-Rusyn Dictionary], Novi Sad: Faculty of Philosophy, 2022, pp. 371

[16] “REST API documentation,” Wiktionary, The free dictionary, https://en.wiktionary.org/api/rest_v1/ (accessed Oct. 15, 2024).

[17] “MediaWiki API help,” Wiktionary, The free dictionary, https://en.wiktionary.org/w/api.php (accessed Oct. 15, 2024).

[18] “JSON web token introduction,” JSON Web Tokens - jwt.io, https://jwt.io/introduction (accessed Apr. 20, 2025).

[19] “Bcryptjs,” npm, https://www.npmjs.com/package/bcryptjs (accessed Apr. 20, 2025).

[20] “Introduction,” shadcn/ui, https://ui.shadcn.com/docs (accessed Apr. 20, 2025).

[21] “Get started with Tailwind CSS,” Tailwind CSS, https://tailwindcss.com/docs/installation/using-vite (accessed Apr. 20, 2025).

[22] “Fórça,” Wiktionary, The free dictionary, https://en.wiktionary.org/wiki/f%C3%B3r%C3%A7a (accessed Apr. 20, 2025).

[19] J. Nielsen, “10 usability heuristics for user interface design,” Nielsen Norman Group, https://www.nngroup.com/articles/ten-usability-heuristics/ (accessed Apr. 18, 2025).

[20] S. Reporter, “Macau’s ‘sweet language’ on verge of disappearing,” The Mail & Guardian, https://mg.co.za/article/2011-11-02-macaus-sweet-language-on-verge-of-disappearing/ (accessed Jan. 20, 2025).

[21] TFadmin, “戲劇農莊【黑盒劇場演出計劃2024】 第三回 《回家吃飯 Vêm Di Volta Pa jantâ》,” 戲劇農莊, https://www.tfarmers.org/2024/11/19/%e6%88%b2%e5%8a%87%e8%be%b2%e8%8e%8a%e3%80%90%e9%bb%91%e7%9b%92%e5%8a%87%e5%a0%b4%e6%bc%94%e5%87%ba%e8%

a8%88%e5%8a%832024%e3%80%91-%e7%ac%ac%e4%b8%89%e5%9b%9e-%e3%80%8a%e5%9b%9e%e5%ae%b6%e5%90%83/ (accessed Jan. 20, 2025).

[22] TallonKH, "TallonKH/WiktionaryParse," GitHub, https://github.com/TallonKH/WiktionaryParse (accessed Apr. 18, 2025).

[23] M. Swadesh, "Lexico-statistic dating of prehistoric ethnic contacts. With special reference to North American Indians and Eskimos," *Proceedings of The American Philosophical Society*, vol. 96, no. 4, pp. 456–457, 1952. Accessed: Apr. 20, 2025. [Online]. Available: https://cdstar.eva.mpg.de/bitstreams/EAEA0-BF5B-6FD1-C12C-0/Swadesh1952.pdf

[24] "□ - wiktionary, The free dictionary," Wiktionary, https://en.wiktionary.org/wiki/%E1%B2%8A (accessed Oct. 15, 2024).

[25] Kwamikagami, "File:unicode 0x1C8A.SVG," Wikimedia Commons, https://commons.wikimedia.org/w/index.php?curid=152543159 (accessed Oct. 15, 2024).

[26] J. Duffy, "The best sites for learning American Sign Language in 2025," PCMAG, https://www.pcmag.com/picks/best-sites-for-learning-american-sign-language?test_uuid=01nEKitrPAQmw6jRcSPVArQ (accessed Apr. 20, 2025).

7. Appendix

7.1. Table 4.1

Table 4.3: a table describing the experience points required for a user to reach a certain level.

Level	Experience points	Cumulative XP
0	0	0
1	50	50
2	100	150
3	150	300

4	200	500
5	250	750
6	300	1050
7	350	1400
8	400	1800
9	450	2250
10	500	2750

7.2. Table 4.2

Table 4.4: a table describing tasks undertaken by testers during the user testing phase.

Given condition	Task	Success criteria	Notes
The tester has never done a lesson.	Pass lesson 1 in Macanese.	The lesson is passed in fewer than 5 attempts.	Lesson 1 has only five words compared to later lessons.
The tester is currently at level 0.	Reach level 1.	The user sees on the XP bar on the top left of the navigation bar that it displays “Level 1” instead of “Level 0”.	Each lesson gives at least 20 XP. Level 1 is achieved at 50 XP, therefore three lessons are required to reach level 1.
The tester is at level 5 and can access the dictionary. In addition, lesson 4 in Macanese is unlocked.	Find a cultural word in the dictionary and check its description.	The user finds the word “balichám” and learns that it is a condiment often used in Macanese cooking.	The word “balichám” first appears in lesson 4. In addition, all cultural words are marked with purple, both in the lesson and

			in the dictionary.
The tester has never submitted a language.	Submit a language to the community portal.	The tester can see their own language submitted.	

7.3. User testing data

The Google Forms used for user testing can be accessed here:
https://forms.gle/6P3p7qwEtvtcLjW96
While the responses from the Google Forms can be accessed here:
https://docs.google.com/spreadsheets/d/1kadWunKSOjJhS_taJ-cnUIqstFKgmtfOrq1JJFqLz4Y/edit?usp=sharing

END OF REPORT