Archisha Bhattacharya
Brooklyn Coulson
Jasmeet Singh

Faculty of Engineering and Applied Science, University of Regina

ENSE 412: Machine Learning

Dr. Timothy Maciag

April 6, 2024

Table of Contents

Project Overview

Problem and Target Audience

Many individuals experiencing hearing loss and those who are non-vocal rely on American Sign Language (ASL) as their primary language. However, there is frequently a significant communication gap between ASL speakers and English speakers. This gap can lead to misunderstandings, limited opportunities, and exclusion for ASL speakers in various settings, such as education, employment, healthcare, and social interactions.

Proposed Solution

Our proposed project will enhance accessibility for non-vocal individuals and those with hearing loss by providing a reliable tool for communicating with English-speaking individuals, thereby reducing communication barriers and fostering inclusion. This application can facilitate communication between ASL and English speakers within organizational settings.

As such, our team created VisuSpeak, an innovative ASL-to-English bidirectional translation application, for our final year capstone project. This application aims to bridge the communication gap experienced by ASL speakers when conversing with English speakers. VisuSpeak enables seamless real-time translation of ASL letters, words, and phrases to the English equivalent, allowing people with special accessibility needs to communicate effectively with others. This brings new opportunities for ASL speakers to gain access to essential services and information in a more natural and humane way.

Project Expectations

Potential Stakeholders

For this project, it was essential to approach several different stakeholders from various industries. Since VisuSpeak is an ASL-English bidirectional translation application, it could be used in several customer service-facing industries. With this in mind, we wanted to engage in three primary industries: customer-facing business organizations, accessibility offices that provide customer support, and an organization that works closely with the deaf and hard-of-hearing community. By networking and connecting with industry professionals, we set out to secure a consultation from industry experts from SaskPower, the U of R Accessibility Centre, and Saskatchewan Deaf and Hard of Hearing Services (SDHHS).

Envisioned Features

In order to bridge the communication gap between ASL speakers and English speakers, our team brainstormed some of the features VisuSpeak must have regarding chat functionality, user support, and teaching ASL to English speakers. One of the main features of chat functionality is converting ASL to English using trained artificial intelligence (AI). The app must be able to accept input in ASL hand gestures using video capture and then predict the ASL

conversions in real time; then, the app would output the predicted words as English text that the user could then send to an Admin. We also envisioned the app to be able to take English text and audio and convert this to ASL hand gestures as well.

We wanted to ensure that both ASL and English-speaking users would be supported throughout their chat experience, fostering a more inclusive and natural communication. To accomplish this, we envisioned ASL users being able to specify their accessibility needs and having the option to change specific conversation settings to create a more tailored experience. We also envisioned that admin users could access user transcripts to support and assist ASL speakers more quickly and efficiently.

Lastly, we wanted VisuSpeak to spark an interest in the deaf and hard-of-hearing community and allow English speakers to learn more about the language. We envisioned the application as a learning component where English speakers may learn how to sign common ASL words and phrases. This learning component would include an ASL dictionary, allowing users to learn how to sign common words and phrases in ASL. We also envisioned an ASL testing feature allowing English speakers to test the new ASL words and phrases they have just learned.

Initial UI/UX Prototypes

When we began the design process, we wanted to keep up with the current UI/UX design trends and keep the interface clean and minimal. We wanted to create a sense of trust and a welcoming look for the user, so we tried to incorporate styles like rounded corners/edges, minimal colours, and well-structured containers that were not cluttered so it was easy to find relevant content. By keeping things simple, we will foster a more pleasant user experience, encouraging them to become returning customers.

We also wanted to ensure that usability was built into our design; this includes efficiency, learnability, and memorability. For efficiency, we ensured that navigation to the other pages would be featured in both the header and footer. For learnability, we included help buttons to help guide the user when chatting about utilizing the ASL signing component. Lastly, for memorability, we wanted as few steps as possible when using our chat to ensure that users can easily remember how to use our application when returning to the application in the future.

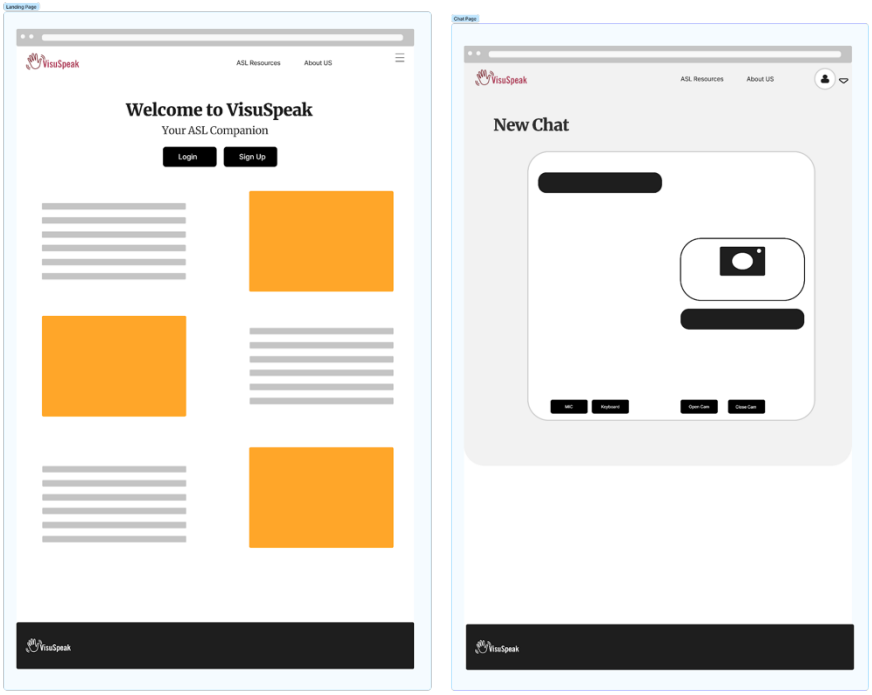

Figure 1: Initial high-fidelity protypes developed

Projected Technical Requirements

To support our envisioned application, we decided upon a list of technical requirements that would be needed in order to create an enjoyable chatting experience between ASL speakers and English speakers. In terms of hardware requirements, both users would need access to a desktop computer or laptop that must have browser support; if the user wishes to utilize the ASL conversion capabilities of the application, then they must also have access to a webcam that can capture their hand gestures.

Regarding software requirements, we felt that the application's design and user interface must be accessible and inclusive to those with visual and hearing impairments. VisuSpeak must keep user credentials, conversations, and other data safe and secure. Lastly, the application must use an AI model that is both efficient and accurate. We aim to train a model that will recognize hand gestures as ASL words or phrases and provide accurate predictions of the English equivalent within 15-30 seconds of receiving the input.

Target Minimum Viable Products

After reviewing our project requirements and the features we wished to include, we divided VisuSpeak's deliverables into five Minimum Viable Products (MVP). For the first MVP, we heavily focused on creating the AI model that would produce the ASL-English translation predictions, creating the bare skeletons of the website pages, and implementing the signup/login functions for user accounts. During the second MVP, we planned to have the application accept spoken audio as input for creating messages. The third MVP would focus on the conversation transcript functionality for admin users, while the fourth would be tailored to

creating the learning component of the website; this would include the ASL dictionary and the ASL accuracy testing. The final MVP would explore other languages the AI model could translate ASL into, including French and various Indigenous languages.

Technology Stack Utilized

Front-end Development

React hosted on Nginx: The choice of React for the front-end is strategic for developing a dynamic and responsive user interface. React's component-based architecture allows for efficient updates and rendering, making the application fast and interactive. Hosting the application on Nginx ensures reliable service delivery and efficient static content serving, which is crucial for a high-performance web application.

Back-end Development

Firebase: Utilizing Firebase for the back-end infrastructure offers several advantages, including scalability, real-time database updates, and authentication services. This choice supports the real-time updating of messages within the VisuSpeak chat feature, enabling instantaneous communication between users. Firebase's scalability and ease of use make it an ideal solution for a project aiming to reach a wide audience with varying accessibility needs.

AI Model Development

Python and MediaPipe for Gesture Recognition: Python is a versatile programming language with extensive support for machine learning and AI development. MediaPipe, an open-source framework from Google, is employed for hand gesture recognition, which is central to translating ASL to English. This combination allows for the development of a robust model capable of recognizing complex hand gestures with high accuracy.

TensorFlow for Model Optimization: TensorFlow is used to optimize the AI model by converting it into a lighter version deployable in web environments. This step ensures that the model runs efficiently in real-time applications, minimizing latency and enhancing the user experience.

TensorFlow's optimization capabilities were instrumental in refining the model for deployment, ensuring that it could deliver real-time performance without compromising accuracy or responsiveness.

Custom Dataset and Model Training: The project team created a custom dataset with assistance from ASL experts, allowing for the training of a model specifically tailored to recognize ASL letters and words accurately. This approach addresses the limitations of pre-existing datasets by incorporating real-world variations and nuances of ASL, ensuring a higher level of precision in gesture recognition.

Achieving Desired Accuracy

The development process culminated in achieving a model with 78% accuracy, a significant improvement over initial trials. This level of accuracy represents a promising step toward breaking down communication barriers for the ASL community, offering a new level of interaction and accessibility.

Project Results

Stakeholder Engagement

Our team utilized our network connections throughout the past eight months and secured industry experts from every stakeholder we intended to consult. For expertise regarding business organizations, we consulted with Rob Anderson, P.Eng. and Manager of Innovative Solutions at SaskPower. We virtually met with Rob several times throughout the project for both consultation and user testing, and he gave invaluable advice from an organizational point of view. Some of the advice he gave related to checking AAA standards for our user interface, potential features that would enhance the application's accessibility, and suggestions on how to improve the admin side of our application.

For expertise regarding user accessibility, we connected with Jasmine Owens, a student accessibility officer from the U of R Accessibility Centre. We met with Jasmine a few times for consultation and user testing, and she provided us with advice focussing on accessibility and inclusivity. Jasmine's advice highlighted the accessibility standards associated with colour schemes, font size, and website navigation.

Lastly, we connected with Patti Spicer, who gave us expert advice and feedback from an ASL-speaking point of view. Not only does Patti Spicer work as a vocational expert with Saskatchewan Deaf and Hard of Hearing Services, but she also speaks ASL as her first language. As such, her consultation was pivotal in our project and significantly improved our application in more ways than one. Some of the ways she helped us include thoroughly testing the application, improving the accuracy of our model by expanding our dataset, allowing us to record tutorial videos for the ASL Dictionary, and providing cultural insight regarding the ASL community.

Actual Features Implemented

From our initial list of potential VisuSpeak features, not all of them were included in the final product. After consulting with our stakeholders, mentors, and facilitators throughout the semester, we took the collective feedback. We ensured the application was heavily focused on chat functionality and ASL translation capabilities so ASL speakers and English speakers could converse seamlessly and humanely. By shifting more focus, effort, and time on this central component, we avoided featuritis while shifting our application from being perceived as confusing to having a healthy level of complexity.

As a result, we were able to implement core features like the conversion of ASL gestures into text to facilitate communication with English speakers. We also fostered accessible and inclusive chatting experiences by allowing ASL users to alter chat settings to suit their unique needs. ASL users can tailor their communication experience by having the ability to create a message using text, audio, or signing. ASL users can also alter the hand gesture acceptance speed to 5, 10, or 15 seconds when using the signing option.

From an admin point of view, we also implemented some chat features to enhance their user experience. One of these features includes the ability for an admin to pull up a history of conversations from past interactions. They can see this transcript history and easily filter through these conversations to find a specific conversation, archive essential conversations, or delete any they no longer need to have on record. Another admin chat feature we implemented is the ability to copy messages from a conversation so they can be efficiently pasted into their organization's internal corporate ticketing systems. Lastly, admins also have the option to change their passwords and the language of preference (English or French) associated with their accounts.

For our final set of features, we added components and pages focusing more on supporting VisuSpeak's English and ASL-speaking users rather than trying to teach English speakers ASL. We still implemented the resource aspect of our application; however, we pivoted to allow English speakers to see what letters, words, and phrases our AI model can predict and how the user can sign them. We also implemented a Support page that helps users understand how to use various features of VisuSpeak. By shifting this component to a secondary support rather than a primary learning feature, we make VisuSpeak's purpose more apparent: it is intended to be an ASL-English bidirectional chat application.

Final UI/UX Developed

Over the past eight months, we dedicated ourselves to creating an inclusive user experience for VisuSpeak, specially tailored to meet the needs of ASL users, our primary target audience. We explored multiple design ideas for creating an application that serves our ASL users while benefiting organizations. After all, our vision was to offer support in day-to-day customer service settings. Throughout this iterative process, we collaborated closely with industry experts, rigorously evaluating our UI's functionality and core design features. Their invaluable input ensured that VisuSpeak met the highest level of accessibility standards. The attached screenshot illustrates the significant change from our initial prototypes.

In shaping VisuSpeak's final look, we incorporated industry-standard practices, such as Level AAA of WCAG standards. Prioritizing simplicity, we adopted a monochromatic colour palette to enhance clarity and ease of use. Additionally, we utilized colour-blind checkers and maintained consistency in fonts to optimize readability for all users.

Our dedication to prioritizing accessibility and inclusivity demonstrated our commitment to empowering everyone in the digital ecosystem. By embracing diverse perspectives from our industry experts and following industry best practices, we have created an application that not only meets the accessibility standards but also ensures a seamless experience for all users.

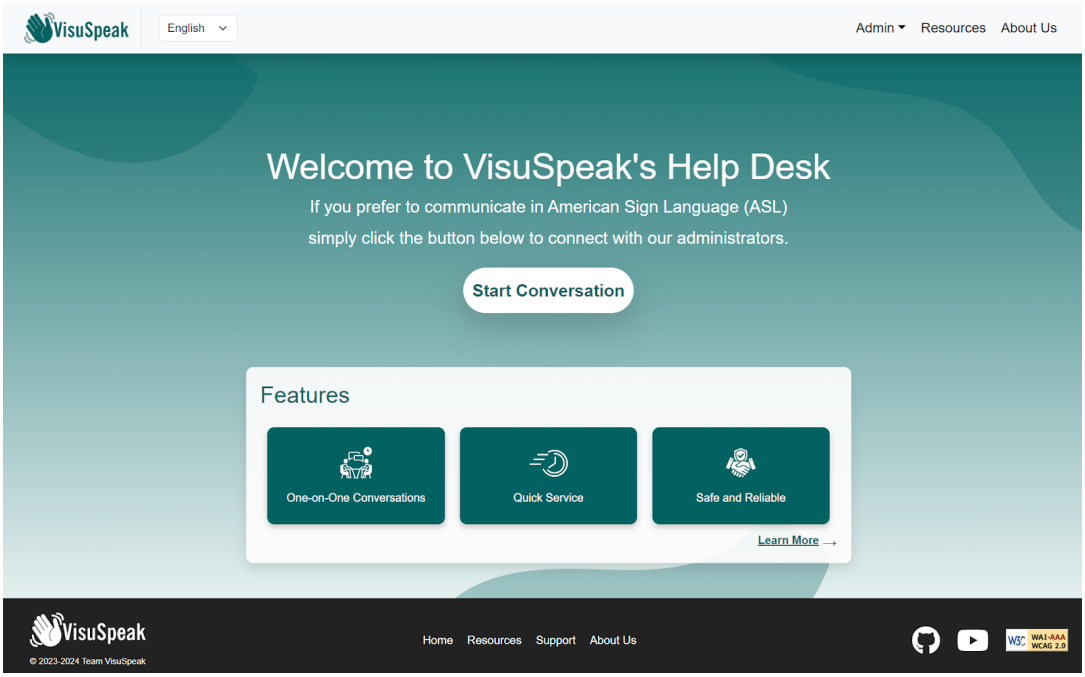

Figure 2: Final VisuSpeak Landing Page

Achieved Minimum Viable Products

Projects tend to change, mainly when stakeholders, mentors, and other interested parties review, test, and give feedback on your progress throughout the project's life; VisuSpeak was no exception. Our MVP Story Map has significantly changed over these past eight months as our application shifted to focus more heavily on the ASL-English bidirectional chat features. As such, we rearranged deliverables, added new features, and altered or removed old ones. Once this restructuring was completed, we had a total of six MVPs, and by the end of this project, we had successfully implemented five of them.

The first MVP largely remained the same; we trained our MediaPipe AI model to accurately recognize the hand gestures representing all 26 letters of the English Alphabet and 20 common English words and phrases. Our AI model could recognize these gestures and put the predicted English word into a chat message so the user could construct sentences to send to the other user. We also implemented the signup and login functionality so that a user can quickly create an account and securely login with their credentials.
For the second and third MVPs, we decided to swap them as we felt that the transcript history should take precedence over improving upon the ASL speaker's chatting experience. Once we implemented the ability for a chat conversation to be automatically saved within our database, we added filter controls to make it easier for the admin to find, open, archive, and delete chat

histories as needed. Once this was completed, we moved on to the third MVP, which allowed the users to use speech to create a message, meaning they could now either speak, type, or use hand gestures to generate chat messages. We also added features and indicators to make the signing process easier for ASL speakers. We implemented a countdown to indicate when a hand gesture would be accepted for translation, a go button in case the ASL speaker does not wish to wait for the countdown to reach zero, and three different times the countdown clock will start counting down from.

For the fourth MVP, we focused on the type of users that will be utilizing VisuSpeak, the setting in which it is used, and how we can differentiate these two use cases. This is when we developed the Admin Account and general ASL User account. We chose to separate these two since there are features that either only admins or ASL users should have access to; for example, the transcript histories of users they have chatted with in the past should only be accessed by admins, and ASL speakers who come into the organization for customer service should have the option to sign as an input type.

The final MVP we completed consisted of features surrounding the ASL dictionary resource available to users, enhancing both the admin and ASL user experience. As mentioned previously, the ASL dictionary was developed to show what English letters, words, and phrases our AI model can recognize and provide English speakers instructions on how to sign or recognize these supported words themselves. To enhance user experience, we implemented the option for an ASL user to override the countdown so the hand gesture can be accepted by pressing a go button. We also added a toggle so the ASL user can switch between signing words or fingerspelling; this was done so that the user is not limited to the 20 words/phrases our AI model recognizes and can continue to chat by spelling out words and phrases that are not supported. On the admin side, we enhanced their user experience by implementing a message notification system and the ability to copy messages to be pasted into internal ticketing systems easily. To help foster more inclusion and a better user experience, we have also implemented a feedback system so that the admin can gain immediate feedback on how the chat experience was for the ASL user once their chat has ended. This immediate feedback lets the organization see where they excel in customer service and understand how to improve this experience.

Project Reflections

Implementation of the MediaPipe AI Model

1. Integration with MediaPipe:

- The team began by integrating MediaPipe with their application, focusing on the hand tracking solutions that MediaPipe offers. This step required setting up the environment to support MediaPipe and ensuring compatibility with the rest of their tech stack, particularly Python and TensorFlow.

2. Custom Dataset Training:

- A custom dataset, enhanced with input from ASL experts, was used to train the model. This involved capturing a broad range of hand gestures and positions to ensure the model could accurately recognize ASL signs under various conditions.

3. Model Optimization:

- The trained model was optimized for web deployment. This likely involved using TensorFlow to convert the model into a lighter version that maintained high accuracy while ensuring fast, real-time performance on the web.

4. Testing and Iteration:

- Extensive testing with real users from the ASL community was conducted to gather feedback on the accuracy and usability of the gesture recognition feature. The team iterated on the model based on this feedback to improve its performance and user experience.

5. Ensuring Accessibility and Performance:

- Accessibility was kept at the forefront of the development process. The team ensured that the gesture recognition was fast and accurate enough not to hinder the user experience, especially for users who relied on this technology for effective communication.

Implementation of Real-time Chat Feature

1. Real-time Database Setup:

- Firebase was utilized as the real-time database solution to manage chat data. Firebase allowed for real-time data synchronization across user devices, making it an ideal choice for building interactive chat features.

2. User Authentication:

- User authentication mechanisms were implemented to enable users to create and sign into their accounts securely. Firebase provided authentication services that were integrated into the application to handle sign-ups, logins, and user session management.

3. Chat Interface and Data Management:

- A user-friendly chat interface was designed that allowed for easy message exchange. The necessary front-end and back-end logic to fetch, display, and store messages in real-time was developed. The team ensured that messages were linked to the correct user accounts and that users could only access their conversations.

4. Testing Across Different Devices:

- The chat functionality was tested across a variety of devices and network conditions to ensure reliability and performance. Special attention was given to the responsiveness of the chat interface and the real-time update mechanism.

5. Security and Privacy:

- Security best practices were implemented to protect user data and messages. This included securing the database, enforcing data validation, and ensuring that user conversations were encrypted and could not be accessed by unauthorized parties.

6. UI/UX Considerations:

- Given the focus on accessibility, the chat interface was made accessible, with options for users to customize text sizes, contrast, and other UI elements to suit their needs.

Additionally, One of the most rewarding aspects of this project was how engaged our stakeholders were and how passionate they were about the ASL community. It was reassuring to know that they genuinely believed in what we were developing and wanted to help us create something they desperately needed within the community. This passion helped to fuel our motivation and keep us determined to develop a higher-quality application.

Challenges

Training the Model with Large Datasets and Handling Network Interruptions

1. Handling Network Interruptions:

- The team faced significant challenges when network interruptions and computing limitations occurred during model training with large datasets. To mitigate this, we implemented checkpointing in their training process. This technique involves saving the model's state at intervals, ensuring that training could resume from the last saved state rather than starting over.

2. Incremental Training Approaches:

- The team adopted incremental training approaches, where the model is trained on smaller subsets of the dataset in sequences. This method not only helps in managing memory and computational resources more effectively but also reduces the setback caused by network interruptions.

Learning to Use Different Technologies Together

1. Embracing Community and Documentation:

- Without prior experience in using the chosen technologies, our team leaned heavily on official documentation, community forums, and tutorials. Sites like Stack Overflow, GitHub discussions, and technology-specific communities would have been invaluable resources for solving problems and learning best practices.

2. Prototyping and Testing:

- We adopted a rapid prototyping approach, creating small, manageable projects to test how different technologies worked together. This hands-on experience allowed for a deeper understanding of integration points and potential issues.

3. Seeking Mentorship and Collaborations:

- Mentorship from industry experts played a crucial role in guiding us throughout the development process.

4. Continuous Learning and Iteration:

- The development process involved continuous learning and iterative improvement. Each challenge encountered would have provided a learning opportunity, allowing the team to refine their approach and improve their understanding of the technologies involved.

5. Using Integration Tools and Frameworks:

- To facilitate the use of different technologies together, the team explored various integration tools and frameworks designed to work with their chosen tech stack. This includes looking into libraries and APIs that offer compatibility layers between different systems.

Understanding our Audience

A significant challenge we faced with this project was tackling such a unique problem without having any ASL knowledge before starting. It became clear early on that due to this lack of knowledge, we would need to seek out many different experts to ensure that not only are we solving the correct problem but that our solution is relevant and valuable to the customers we are targeting. Luckily, we overcame this challenge by connecting with accessibility experts and those fluent in American Sign Language.

Lessons Learned

One lesson we learned was that project work can be difficult, especially when working with programs, languages, and technologies you are not familiar with or have never seen. That said, it is essential to know that even in large projects, you should be bold and ask for help when you are stuck or need help making something work. Being transparent and open about how tasks are going and seeking out assistance when necessary, helps to make the development process run more smoothly, and it also helps to allow others to plan their work more efficiently. We are all engineering students, still learning, and asking for assistance should be considered proactive rather than admitting inadequate skills.

Conclusion

The journey of developing VisuSpeak, our ASL-to-English bidirectional translation application, has been one of innovation, inclusion, and learning. Our primary objective was to bridge the communication divide between ASL and English speakers, a goal that carried the promise of opening new avenues of understanding and interaction. The project is a testament to the dedication and collaborative effort of our team, stakeholders, and the ASL community. Through the iterative design, development, and testing process, VisuSpeak has matured into a tool that not only meets the technical aspirations we set out but also resonates deeply with the values of accessibility and empowerment. Integrating a reliable AI model for gesture recognition and real-time chat functionality has been pivotal in creating an inclusive platform where ASL speakers can communicate their thoughts and needs effectively and with dignity.

We have successfully implemented five of the six envisioned MVPs, and our application now offers a robust user experience tailored to the unique needs of its users. The combination of technical prowess and human-centred design has resulted in an application that's functionally rich and empathetic to its users' experiences.

Moreover, the stakeholder engagement has been enriching and enlightening, bringing to light the intricacies of real-world applications and technology's impact on people's lives. The feedback loop established with experts and end-users has been invaluable, leading to a product that not only meets but anticipates and addresses the needs of the community it serves.

Future Improvements

As we look to the future, VisuSpeak holds the potential for scalability and expansion into new languages and dialects, promising to be a beacon of innovation in the field of communication accessibility tools. It also opens pathways for further research and development in the realm of assistive technologies.

When discussing what MVPs were realized within this project, we mentioned that our team needed to complete MVP6. As such, one future improvement would be adding more languages to increase inclusivity and accessibility. We want to expand the translation capabilities of our AI model and train it to translate ASL into other languages like French and various Indigenous languages. A few other features that were included in this MVP were allowing admins to flag, delete, and archive inappropriate messages, implementing a Frequently Asked Questions section catered to admins, and integrating a ticket managing system directly into VisuSpeak so that admins do not need to rely on other external software.