

ZSL Parliament Interpreter: Design and Implementation of a Speech-to-Sign Language Interpretation System for the National Assembly of Zambia.

By

Martha Nyambe (2022020855)

Danny Lungu (2021461602)

Chilandu Phiri (2022042085)

Supervisors:

Dr. Lighton Phiri¹

¹Mr. Alinani Sichone²

Mr. Paul Chola²

Mr. David Mulenga²

Dr. Mwaula Solopi²

A report submitted to the Department of Computing and Informatics, The University of Zambia,
in partial fulfilment of the requirements of the degree of Bachelor of Science in Computer Science

THE UNIVERSITY OF ZAMBIA

LUSAKA

©2026.

¹*Department of Computing and Robotics, University of Zambia, Lusaka, Zambia.*

²*IT department, National Assembly of Zambia, Lusaka, Zambia.*

Declaration

I declare that this final year project report is my own work and has been prepared in accordance with the academic requirements of the University of Zambia. All sources of information, ideas, data, software, images, and other material that are not my own have been appropriately acknowledged and referenced.

Where this project was undertaken as part of a group, I have clearly identified my individual contributions and have not presented the work of other group members as my own.

Student Name:

Signature:

Computer Number:

Date:

Approval

This final year project report has been submitted for examination with the approval of the project supervisor.

Supervisor Name:

Signature:

Date:

Departmental/Examiner Approval (if applicable):

Abstract

Access to parliamentary debate requires communication channels that accommodate different hearing and language needs. This project presents a browser-based speech-to-sign interpretation prototype for the National Assembly of Zambia. The system captures spoken English, produces a streaming transcript, generates an intermediate gloss representation and plays available recorded movements through a three-dimensional avatar.

The system combines Flask, Flask-SocketIO, SQLAlchemy, NVIDIA NeMo streaming automatic speech recognition, a dictionary-based gloss engine and Three.js/VRM avatar rendering. The application contains 1916 word mappings, 25 phrase mappings and 1941 recorded motion sequences across 28 categories, including alphabet animations for fingerspelling, alphabet word from a to a and a number category. Authentication, email verification, recording permissions, profile management and administrative controls support operation within one application.

Verification established selected software behaviours through component and browser checks, JavaScript regression scripts, and server-side checks of speech processing and asset delivery. Additional formative evaluations examined speech transcription and avatar playback. These results support component execution and integration, but do not establish reliable Zambian Sign Language (ZSL) translation or parliamentary readiness. The project is an engineering prototype; language-specific validation and comprehension studies with intended users remain priorities.

Acknowledgement

We would like to express our sincere gratitude to everyone who contributed to the successful development and completion of this project.

First and foremost, we thank Almighty God for granting us the strength, wisdom, patience, and perseverance throughout the development of this project.

We extend our heartfelt appreciation to our project supervisor, **Dr. Lighton Phiri**, for his invaluable guidance, constructive feedback, encouragement, and continuous support throughout the research and development process. His expertise and recommendations played an important role in shaping the direction and quality of this project. We are equally grateful to the **Department of Computing and Informatics at the University of Zambia** for providing us with the academic foundation, knowledge, and technical skills necessary to undertake this project.

We further acknowledge the contributions of the **National Assembly team**, whose insights, professional experience, and understanding of parliamentary operations provided valuable context for the development of the proposed Speech-to-Sign Language Interpretation System. Their involvement helped us better understand the institutional environment in which the system is intended to operate, as well as the practical considerations associated with introducing technological solutions into parliamentary communication processes. We deeply appreciate the time, attention, and support extended to us throughout our engagement with the institution.

We are also grateful to the **Zambian Sign Language interpreter at Kitwe Teaching College** for sharing valuable knowledge about Zambian Sign Language, particularly its grammatical structure, linguistic characteristics, and differences from other sign languages. These insights contributed to our understanding of the linguistic considerations and challenges involved in developing a sign language interpretation system suitable for the Zambian context.

We also acknowledge the researchers, developers, and open-source communities whose contributions in artificial intelligence, automatic speech recognition, sign language processing, and three-dimensional avatar animation provided important technological foundations for this project.

Finally, we extend our deepest appreciation to our families, friends, and fellow students for their unwavering encouragement, understanding, and support throughout our academic journey.

To everyone who contributed directly or indirectly to the successful completion of this project, we are sincerely grateful. Your support and contributions have been invaluable.

Table of contents

Declaration	2
Approval	4
Abstract	5
Acknowledgement	6
Table of contents	7
List of figures	9
List of tables	9
List of abbreviations	10
Chapter 1: Introduction	11
1.1 Background	11
1.2 Problem statement	12
1.3 Aim and objectives	12
1.4 Scope and significance	12
Chapter 2: Literature Review	13
2.1 Recognition, translation and production	13
2.2 Streaming speech recognition	13
2.4 Avatar-based production	13
2.8 Research gap	17
Chapter 3: Methodology	19
3.1. Research approach	19
3.2. Research design	20
3.2.1. Understanding Zambian Sign Language	20
3.2.2. Speech-to-Sign ZSL Pipeline	21
3.2.3. Evaluation	23
3.3. Ethical Consideration	23
Chapter 4: System Design and Implementation	24
4.1 Use cases and architecture	24
4.2 Technology stack	25
4.3 Class structure and persistent data	25
4.4 Speech acquisition and streaming ASR	26
4.5 Gloss generation and avatar rendering	27
4.6 Deployment	28
Chapter 5: Testing and Results	33
5.1 Verification strategy	33
5.2 Resource and implementation results	33
5.3 Speech recognition and signing observations	34
5.4 Measures needed for empirical evaluation	34

5.5 Threats to validity	35
Chapter 6: Discussion	44
6.1 Interpretation of the system outcome	44
6.2 Acquisition continuity and processing capacity	44
6.3 Vocabulary and linguistic fidelity	44
6.4 Security and operation	44
6.5 Limitations	44
Chapter 7: Conclusion and Recommendations	46
7.1 Conclusion	46
7.2 Recommendations	46
7.3 Future research	47
References	48
Appendix A: Operating workflows	50
Appendix B: Configuration and reproduction	50
Appendix C: Evaluation record and boundaries	50

List of figures

Figure 1.1 : Traditional way of sign language interpretation

Figure 4.4: UML sequence diagram: speech-to-sign.

Figure 4.6: ML deployment diagram.

Figure 4.7: Login and authentication diagram

Figure 4.8: Speech to sign interface

Figure 5.1: Individual participant ratings.

Figure 5.2: Reported transcription outcomes for three tasks.

Figure 5.3 Normalized WER for four trials.

Figure 5.4 Two distinct measured timing intervals.

List of tables

Table 3.2: Functional and quality requirements.

Table 4.3: Technology stack.

Table 4.4: Persistent user data.

Table 4.5: Principal interfaces

Table 4.6: Verification evidence.

Table 5.1 Verification evidence

Table 5.2 Resource inventory

Table 5.3 Evaluation measures

Table 5.4: Interpretation of responses from participants

Table 5.5: Example words that were given to the participants and their responses

Table 5.6: relationship between findings and objectives

Table 5.7: Comparison with existing work

Table 7.1: Recommended research and engineering work

List of abbreviations

Abbreviation	Meaning
API	Application programming interface
ASL	American Sign Language
ASR	Automatic speech recognition
CPU / GPU	Central / graphics processing unit
CUDA	NVIDIA parallel computing platform
EBS / EC2	Elastic Block Store / Elastic Compute Cloud
FPS	Frames per second
JSON	JavaScript Object Notation
PCM	Pulse-code modulation
RBAC	Role-based access control
RES	Representational state transfer
SMTP	Simple Mail Transfer Protocol
TTS	Text-to-speech
UML	Unified Modeling Language
VRM	Humanoid avatar format based on glTF
WER	Word error rate
ZSL	Zambian Sign Language

Chapter 1: Introduction

1.1 Background

Parliamentary debates contain information about legislation, public expenditure and national policy. Spoken broadcasts can exclude people who cannot access the audio. Written captions provide one form of access, while sign-language interpretation offers a visual language channel. These channels serve different needs: a transcript represents spoken wording, whereas sign languages have their own grammar, spatial structure, hand configurations and non-manual features.

The project investigates a software-supported speech-to-sign interpretation workflow in a Zambian parliamentary setting. English speech can enter through a browser microphone, a server-connected microphone or parliamentary radio. The application presents a transcript and an intermediate gloss, then renders available motion through an avatar.

The title reflects the intended local application. The motion resources require linguistic assessment before they can support claims of reliable ZSL interpretation. This distinction determines the scope of the conclusions throughout the report.



Figure 1.1 showing the traditional sign language interpretation

1.2 Problem statement

An accessible speech-to-sign interpretation prototype must connect audio acquisition, speech recognition, gloss generation and avatar presentation without requiring users to restart capture for every result. Audio needs ordered transport and separate client state. Recognition output needs a usable display and a policy for uncertain or unsupported words. The visual output must preserve enough hand and body detail to support comprehension.

Constructing this pipeline is distinct from proving language accuracy. The engineering problem addressed here is the integration of streaming speech, gloss mapping and avatar playback in a browser application. The research problem is whether the generated signs convey meaning accurately in the intended setting; that requires representative recordings and evaluation by competent users.

1.3 Aim and objectives

The aim is to design and implement a speech-to-sign interpretation prototype that converts spoken English into transcript, gloss and avatar-based signing, while providing a foundation for ZSL-specific research.

1.3.1 Specific Objectives

1. To consult relevant sources and establish system requirements.
2. To design and implement a complete speech to Zambian sign language
3. To test and evaluate the functionality of the speech-to-sign system.

1.4 Scope and significance

The implemented scope includes speech acquisition, transcription, gloss generation, animated sign output, account management, model assets and demonstration hosting. It excludes a validated ZSL sentence translator, official chamber installation and a completed large-scale community comprehension study. The system provides an engineering foundation for investigating these questions and makes available motion vocabulary visible to users.

The project does not position automation as a substitute for qualified human interpretation. Its practical significance is the ability to experiment with multiple acquisition channels, observe intermediate results, compare model behaviour and identify the data and linguistic requirements for an appropriate local solution.

Chapter 2: Literature Review

2.1 Recognition, translation and production

Sign recognition assigns labels to visual movement. Translation maps meaning between languages, and sign production generates visual signing. These tasks require different evidence. A classifier that outputs a word label has not necessarily translated a sentence, and a player that renders a clip has not necessarily generated grammatically correct signing.

Camgoz et al. investigate joint continuous recognition and translation using a transformer architecture with gloss supervision [2]. Their work supports treating recognition and translation as connected but separately evaluated tasks. The present application uses explicit component boundaries rather than claiming an end-to-end translation model. Sign Language Transformers

2.2 Streaming speech recognition

A streaming recognizer maintains state across incoming audio windows instead of processing each window as an unrelated recording. The project uses NVIDIA NeMo's cache-aware FastConformer checkpoint, whose model documentation describes streaming English recognition and multiple latency configurations [11]. The deployment uses a CUDA-enabled model runtime. The application's observed execution does not establish the same accuracy or latency as a published model benchmark. NVIDIA model documentation

2.4 Avatar-based production

Saunders et al. study learned sign production from language representations [13]. Their work illustrates that coherent visual production is a temporal modelling problem. This project instead uses recorded landmark clips, making asset coverage, retargeting and transitions central implementation concerns. Progressive Transformers

A recorded motion library supports transparent lookup and reuse. Its limitation is that independent clips do not automatically express sentence grammar, spatial reference or the full range of facial and body signals. Fingerspelling offers a fallback for unfamiliar words and names, but comprehension depends on language conventions, motion quality and playback speed.

2.5 South African Sign Language to speech Translation system

Recent work in South Africa has begun to address the lack of assistive technology for South African Sign Language (SASL). Hansrajh developed a real-time SASL-to-speech translation system as a final-year Computer Engineering design project at the University of KwaZulu-Natal [16]. The system uses computer vision, pairing a custom-built AI model with an established machine learning framework to track body landmarks and identify signs as they are made. It was trained on a custom SASL dataset and can recognize individual letters and words, which are then converted to speech by a text-to-speech model. The project is notable because it was reportedly among the first assistive technology projects in the university's Electrical, Electronic and Computer Engineering discipline to target SASL specifically, whereas many comparable systems internationally focus on American Sign Language. This gap matters because SASL was recently

recognized as South Africa's 12th official language. The aim is to let deaf and hard-of-hearing people communicate in settings such as banks and government offices without the other party needing to know SASL. Planned future work includes improving word recognition and producing more fluent, cohesive speech.

2.6 Automatic Speech Recognition for African Low-Resource Contexts

The speech-to-sign direction of the prototype depends on accurate recognition of English speech, specifically parliamentary English as spoken in Zambia. While English is not a low-resource language globally, the specific acoustic and lexical characteristics of Zambian parliamentary speech present challenges that general-purpose ASR models may not handle well.

A 2025 systematic review of ASR for African low-resource languages identified several recurring barriers: data scarcity, linguistic complexity, limited computational resources, acoustic variability, and ethical concerns regarding bias and privacy . Although this review focuses on indigenous African languages rather than English, its findings are relevant in two respects. First, the same infrastructural and resource constraints that affect ASR for Bemba or Nyanja also affect the development and deployment of any speech technology in Zambian institutional contexts. Second, the review emphasises that community-driven data collection and domain-specific adaptation are essential for acceptable performance in real-world applications . [15].

Research on ASR bias has documented systematic performance disparities across speaker demographics, including accent, dialect, and race . In the Zambian context, parliamentary speech may exhibit phonological features influenced by Bemba, Nyanja, Tonga, and other local languages, as well as code-switching and domain-specific terminology related to legislation and public finance. A general-purpose English ASR model, such as the NVIDIA NeMo FastConformer checkpoint used in the prototype, has not been evaluated on this domain. The report's original testing acknowledged this limitation: "parliamentary WER not measured" [from original report]. The expanded literature review reinforces that this is not merely a missing metric but a substantive research question requiring domain-specific evaluation data.

The decision to use a streaming ASR model rather than a batch model is well-supported by the literature. Streaming recognition maintains state across audio windows, enabling incremental output and reducing latency—a requirement for any interpretation system intended for live use. However, the accuracy-latency trade-off in streaming configurations must be evaluated empirically for the target domain.

2.7 Automatic Speech Recognition for African Low-Resource Contexts

The speech-to-sign direction of the prototype depends on accurate recognition of English speech, specifically parliamentary English as spoken in Zambia. While English is not a low-resource language globally, the specific acoustic and lexical characteristics of Zambian parliamentary speech present challenges that general-purpose ASR models may not handle well.

A 2025 systematic review of ASR for African low-resource languages identified several recurring barriers: data scarcity, linguistic complexity, limited computational resources, acoustic variability, and ethical concerns regarding bias and privacy . Although this review focuses on indigenous African languages rather than English, its findings are relevant in two respects. First, the same infrastructural and resource constraints that affect ASR for Bemba or Nyanja also affect the development and deployment of any speech technology in Zambian institutional contexts. Second, the review emphasises that community-driven data collection and domain-specific adaptation are essential for acceptable performance in real-world applications . [15].

Research on ASR bias has documented systematic performance disparities across speaker demographics, including accent, dialect, and race . In the Zambian context, parliamentary speech may exhibit phonological features influenced by Bemba, Nyanja, Tonga, and other local languages, as well as code-switching and domain-specific terminology related to legislation and public finance. A general-purpose English ASR model, such as the NVIDIA NeMo FastConformer checkpoint used in the prototype, has not been evaluated on this domain. The report's original testing acknowledged this limitation: "parliamentary WER not measured" [from original report]. The expanded literature review reinforces that this is not merely a missing metric but a substantive research question requiring domain-specific evaluation data.

The decision to use a streaming ASR model rather than a batch model is well-supported by the literature. Streaming recognition maintains state across audio windows, enabling incremental output and reducing latency a requirement for any interpretation system intended for live use. However, the accuracy-latency trade-off in streaming configurations must be evaluated empirically for the target domain.

4.4 Automatic Speech Recognition for African Low-Resource Contexts

The speech-to-sign direction of the prototype depends on accurate recognition of English speech, specifically parliamentary English as spoken in Zambia. While English is not a low-resource language globally, the specific acoustic and lexical characteristics of Zambian parliamentary speech present challenges that general-purpose ASR models may not handle well.

A 2025 systematic review of ASR for African low-resource languages identified several recurring barriers: data scarcity, linguistic complexity, limited computational resources, acoustic variability, and ethical concerns regarding bias and privacy. Although this review focuses on indigenous African languages rather than English, its findings are relevant in two respects. First, the same infrastructural and resource constraints that affect ASR for Bemba or Nyanja also affect the development and deployment of any speech technology in Zambian institutional contexts. Second, the review emphasizes that community-driven data collection and domain-specific adaptation are essential for acceptable performance in real-world applications. [15].

Research on ASR bias has documented systematic performance disparities across speaker demographics, including accent, dialect, and race. In the Zambian context, parliamentary speech may exhibit phonological features influenced by Bemba, Nyanja, Tonga, and other local languages, as well as code-switching and domain-specific terminology related to legislation and public finance. A general-purpose English ASR model, such as the NVIDIA NeMo FastConformer checkpoint used in the prototype, has not been evaluated on this domain. The report's original testing acknowledged this limitation: "parliamentary WER not measured" [from original report]. The expanded literature review reinforces that this is not merely a missing metric but a substantive research question requiring domain-specific evaluation data.

The decision to use a streaming ASR model rather than a batch model is well-supported by the literature. Streaming recognition maintains state across audio windows, enabling incremental output and reducing latency—a requirement for any interpretation system intended for live use. However, the accuracy-latency trade-off in streaming configurations must be evaluated empirically for the target domain. Computational Work on African Sign Languages

Computational research on African sign languages is sparse relative to work on ASL, British Sign Language, German Sign Language, and Chinese Sign Language. A 2026 survey of sign language recognition and translation research noted that the field has expanded rapidly since 2016, with hundreds of new studies annually, but that this growth has been overwhelmingly concentrated on a small set of high-resource sign languages . African sign languages remain significantly underrepresented.

The most directly relevant recent work is the AfriSign project, which introduced the first parallel dataset for end-to-end machine translation involving African sign languages . The AfriSign dataset comprises videos in six African sign languages—Ghanaian Sign Language, Nigerian Sign Language, Kenyan Sign Language, Zambian Sign Language, Zimbabwean Sign Language, and South African Sign Language—paired with English translations. The source material consists of Bible verses extracted from the Jehovah's Witnesses sign language website, yielding approximately 20,000 unique spoken words and 152 hours of video . The inclusion of ZSL in this dataset represents a significant resource contribution, although the domain (religious text) differs substantially from parliamentary discourse.

AfriSign experiments trained bilingual Transformer systems as baselines and investigated cross-lingual transfer learning and multilingual training. The multilingual model achieved an average improvement of +0.5 BLEU over the mean of individual bilingual models, a modest but meaningful gain given the small size of each language pair's data . Qualitative analysis revealed that models struggled with complex noun phrases and culturally specific expressions, producing translations that were semantically approximate but not equivalent to the target . These findings underscore a fundamental challenge: sign language translation requires more than lexical substitution, and the limited data available for African sign languages constrains the achievable quality of data-driven approaches.

Beyond AfriSign, computational work specifically targeting ZSL is virtually absent from the published literature. The present project therefore fills a gap not by providing a trained ZSL translation model—which would require data and resources beyond the scope of an undergraduate project—but by building an engineering platform that makes the absence of ZSL-specific models visible and provides infrastructure for future data collection and model development.

2.8 Research gap

The project addresses integration and acquisition continuity within a usable browser application. The unresolved gap is language-specific evidence: representative parliamentary audio, signer-diverse local recordings, reviewed motion resources, and comprehension assessment. Published results on external datasets cannot be transferred directly to this implementation or to ZSL.

Gap 1: Absence of ZSL-specific computational resources. The only publicly available dataset containing ZSL is AfriSign, which is domain-limited (religious text) and not designed for parliamentary or formal discourse . No trained ZSL recognition or translation model exists in the published literature. This project does not fill this gap directly but builds infrastructure that could support future data collection and model development.

Gap 2: Lack of evaluation of ASR on Zambian parliamentary English. General-purpose ASR models have not been benchmarked on the acoustic and lexical characteristics of Zambian parliamentary speech. The prototype implements streaming ASR but does not evaluate its accuracy on this domain. The literature on African ASR emphasises the necessity of domain-specific evaluation and community-informed data practices .

Gap 3: Immature avatar production for low-resource sign languages. Clip-based avatar production is feasible with limited data, but linguistic adequacy grammar, spatial reference, non-manual features remains unaddressed. The literature identifies cultural adaptability and community involvement as critical for acceptable avatar systems .

The present project contributes a platform that makes these gaps concrete and addressable. By implementing audio acquisition, streaming speech recognition, gloss mapping and avatar rendering within an authenticated application, the prototype demonstrates that the speech-to-sign integration problem is tractable. Empirical questions concerning transcription accuracy, latency, comprehension and linguistic adequacy remain open and are priorities for future research.

2.9 Summary

This literature review has situated the ZSL Parliament Interpreter within three bodies of work: ZSL linguistics and sociolinguistics, computational research on African sign languages, and the broader technical literature on ASR and avatar production. The review confirms that the project addresses a genuine gap in the absence of integrated interpretation infrastructure for ZSL in formal institutional settings while also confirming that the project does not resolve the deeper empirical questions of recognition accuracy, translation adequacy, or user comprehension. These questions require representative data, linguistic expertise, and community participation that are beyond the scope of the present engineering prototype but that the prototype is designed to support.

Chapter 3: Methodology

3.1. Research approach

System development followed an iterative approach informed by the Scrum framework [14].

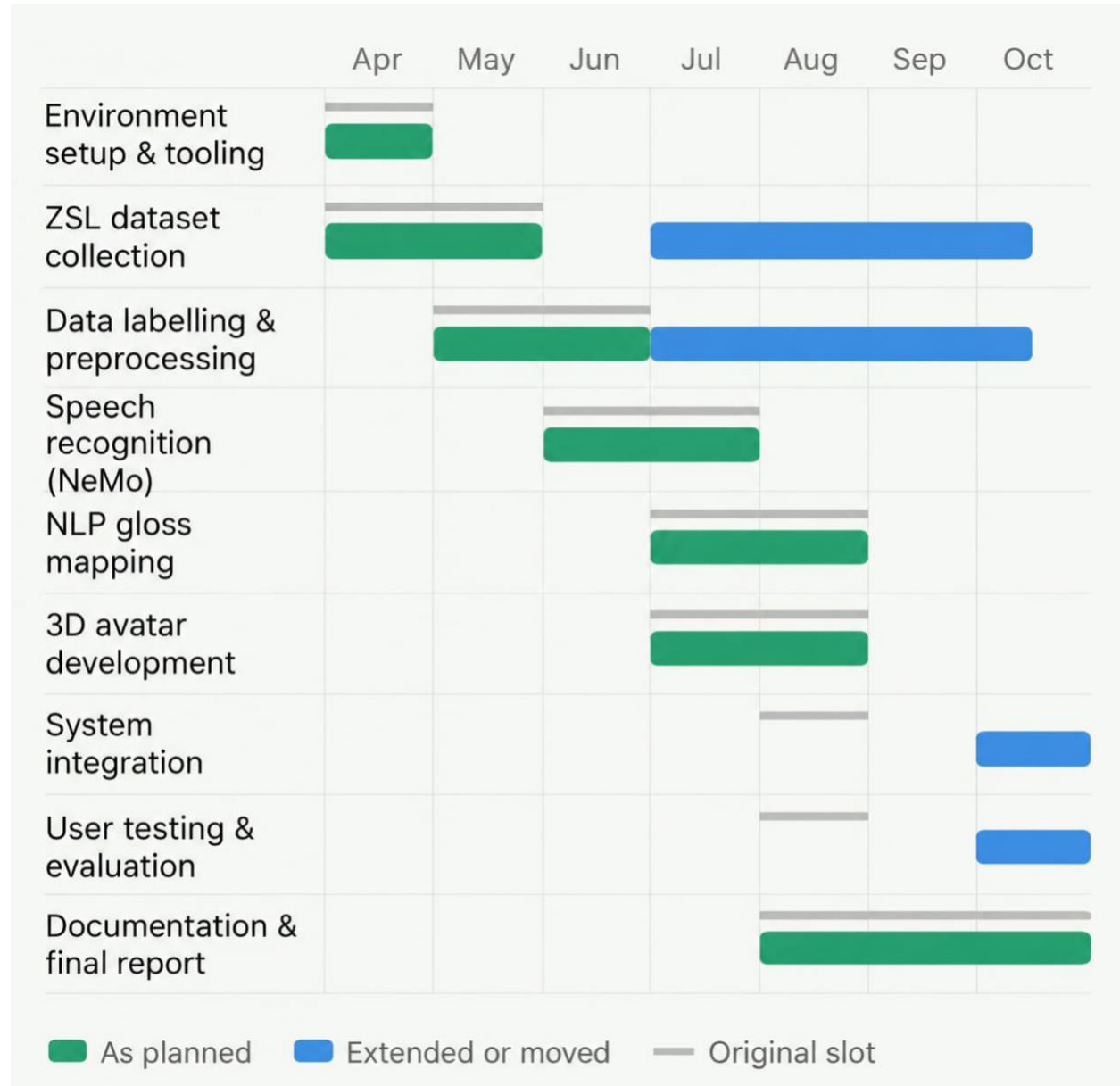


Figure 3.1 SCRUM Sprints.

Development was organised into two-week sprints, with each sprint focusing on specific tasks, such as user interface design, speech recognition, natural language processing for gloss mapping, 3D avatar integration, and visual sign recognition. At the beginning of each sprint, tasks were identified and prioritised according to the project requirements. During the sprint, the selected

tasks were implemented and tested, while the end of the sprint was used to review progress, identify challenges, and determine areas requiring improvement.

The iterative nature of Agile allowed technical challenges to be addressed as they arose and enabled the system components to be refined progressively. Once the individual components had been developed and tested, they were integrated into a single web-based application. Overall, the combination of Agile methodology and the Scrum framework provided a flexible and structured approach to managing development, monitoring progress, and working towards the project objectives.

The study adopted a mixed-methods, design-and-implementation approach to address the project objectives. Qualitative evidence included review of ZSL characteristics, expert consultation and participant feedback. Quantitative evidence included software checks, transcription errors and observed timing measurements.

The product backlog was organized around audio capture, streaming transcription, gloss generation, avatar playback, shared web functions and testing. Work was divided into short development iterations in which features were planned, implemented, integrated and checked before subsequent refinement.

The engineering strand used requirements and UML diagrams to guide modular development of Flask services, streaming speech recognition, gloss mapping and avatar motion playback. The research strand examined linguistic requirements and assessed available software and evaluation evidence. Verification included automated checks and observations of the deployed application.

The mixed-methods approach was appropriate because successful software execution alone cannot demonstrate accurate or comprehensible Zambian Sign Language (ZSL). The prototype uses dictionary-mediated gloss output and a recorded motion library, both of which require language-specific validation. Findings are therefore interpreted as evidence of prototype functionality rather than proof of ZSL translation accuracy.

3.2. Research design

The research design was aligned to the objectives and organized into three activities: understanding the target sign language, constructing the speech-to-sign processing pipeline, and evaluating the implemented prototype. Requirements and design decisions were traced from spoken audio acquisition through transcription and gloss generation to avatar playback.

The design combined qualitative information from ZSL consultation and user responses with quantitative measurements and repeatable software checks. Pretrained speech recognition and existing motion assets were reused rather than described as newly trained ZSL resources. The research design separates implementation success, speech recognition accuracy and linguistic comprehension.

3.2.1. Understanding Zambian Sign Language

Understanding ZSL began with reviewing how sign languages express meaning through hand configuration, movement, location, facial expression, spatial reference and language-specific

grammar. The literature on sign recognition and sign production informed the distinction between identifying isolated labels and translating or generating a complete signed utterance [2], [12], [13].

An expert consultation was conducted with a ZSL interpreter at Kitwe College of Education. It focused on ZSL grammatical structure, differences from ASL and other sign languages, and the use and limitations of written gloss representations. The consultation highlighted why ASL vocabulary or grammatical conventions cannot automatically be transferred to ZSL. These findings guided the description of the prototype’s linguistic boundaries and identified the need for fluent ZSL review of avatar signs and gloss conventions.

The project did not collect a representative parliamentary ZSL corpus or train a ZSL-specific translation model. Available animation sequences were treated as technical assets requiring language-specific validation, rather than proof of ZSL equivalence.

Requirements were elicited through National Assembly of Zambia (NAZ) stakeholder engagement. These activities informed the system’s intended parliamentary use, accessibility needs, and ZSL grammar and vocabulary requirements. The meetings highlighted the importance of simplified glosses, Zambian-specific signs, and stakeholder involvement in real-world testing. The system requirements were obtained from the product owner and documented according to the system’s functional and quality requirements.

These requirements include a system where models are swappable. The team was required to build an architecture that allows different ASR (Automatic Speech Recognition) and gloss conversion models that can be easily plugged in and out. This allows for testing of different models and future upgrades without rewriting the entire system.

The team was asked to add a feature that allows audio input from multiple channels and lets users allow or select the channels.

3.2.2. Speech-to-Sign ZSL Pipeline

Table 3.1 Functional and quality requirements

ID	Requirement	Implementation response
FR1	Capture supported audio sources	Browser, server microphone and radio adapters
FR2	Display incremental speech output	Socket.IO transcript and gloss events

ID	Requirement	Implementation response
FR3	Render available signs	Phrase lookup, word lookup and alphabet fallback
FR4	Restrict acquisition controls	Interpreter and administrator recording permissions
FR5	Support account administration	Role management, password workflows and account controls
QR1	Preserve acquisition order where required	Browser audio queue and per-client ASR state
QR2	Handle interruption	Audio capture cleanup and stream finalization
QR3	Support different screen sizes	Responsive avatar layouts and controls
QR4	Permit reproducible verification	Tests, configuration and deployment utilities

To address the first objective, the system receives English audio from supported microphone or radio sources and carries incremental audio to NVIDIA NeMo streaming recognition for transcription [11]. To address the second objective, the resulting English text is normalized and mapped to an intermediate gloss, then to available recorded avatar motions or fingerspelling clips. This sequence produces visual output but does not establish grammatical ZSL translation.

For the third objective, audio capture, transcript display, gloss generation and avatar playback were integrated into a single authenticated Flask web application. HTTP routes deliver pages and avatar resources, while Socket.IO carries incremental speech events. The functional and quality requirements in Table 3.2 guided implementation checks.

The principal resources were an English speech-recognition checkpoint, a dictionary with word and phrase mappings, and recorded avatar-motion clips. These serve different purposes and cannot be combined into a single ZSL training dataset. No locally collected parliamentary ZSL training corpus is claimed.

Audio windows are queued to preserve processing order. Unsupported gloss words may use available alphabet animations. The sign animation queue preserves available output clips, although

sustained speech may increase playback delay. These behaviours provide a functional speech-to-sign prototype but do not solve ZSL sentence grammar or comprehension.

3.2.3. Evaluation

The third objective was addressed through component, integration and preliminary user-oriented evaluation. Functional verification checked authentication, recording permissions, stream handling, transcript and gloss events, asset delivery and avatar playback. Automated Python and JavaScript tests examined repeatable software behaviours, while recorded trials assessed speech transcription and presentation timing.

Quantitative evaluation considered normalized word error rate on speech trials, estimated delay between spoken words and displayed text, and observed avatar playback intervals. Qualitative evaluation considered the two respondents' task feedback, perceived comprehensibility of rendered movements and interface observations. The small samples, dependence on a limited number of speakers or respondents and absence of blinded linguistic scoring restrict generalization beyond the recorded tests.

Evaluation of the speech-to-sign pipeline distinguished accurate transcription, functioning gloss and avatar playback, and actual comprehension of signs. Recognition errors can propagate from speech to gloss, while motion quality, fingerspelling speed and linguistic grammar affect the usefulness of avatar output. Existing results are reported with their measurement limits rather than as proof of reliable ZSL interpretation.

3.3. Ethical Consideration

Microphone access is controlled by browser permission requests and application roles. Evaluation recordings and participant feedback require informed consent, limited access, secure handling and an appropriate retention policy. A temporary audio-processing workflow does not replace assessment of application logs and hosting infrastructure.

The study must avoid presenting recorded avatar clips as validated ZSL translation or overstating the comprehensibility of generated signs. Community participation, accessible consent, respectful treatment of Deaf and signing users, review of source-asset permissions and qualified linguistic evaluation are necessary before institutional use.

Chapter 4: System Design and Implementation

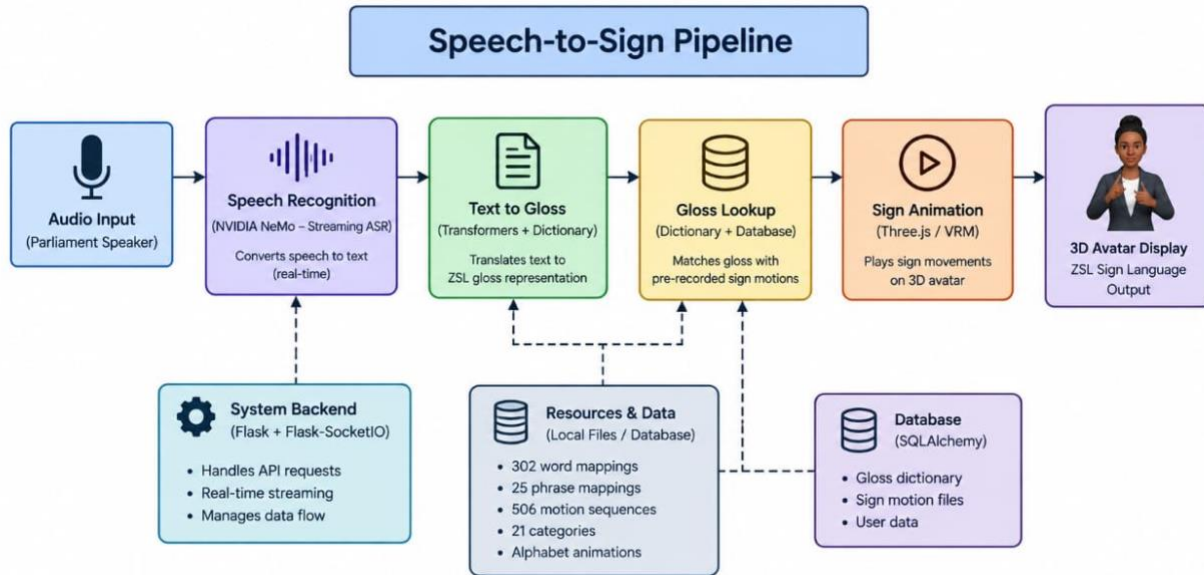


Figure 4.0 processing pipeline.

4.1 Use cases and architecture

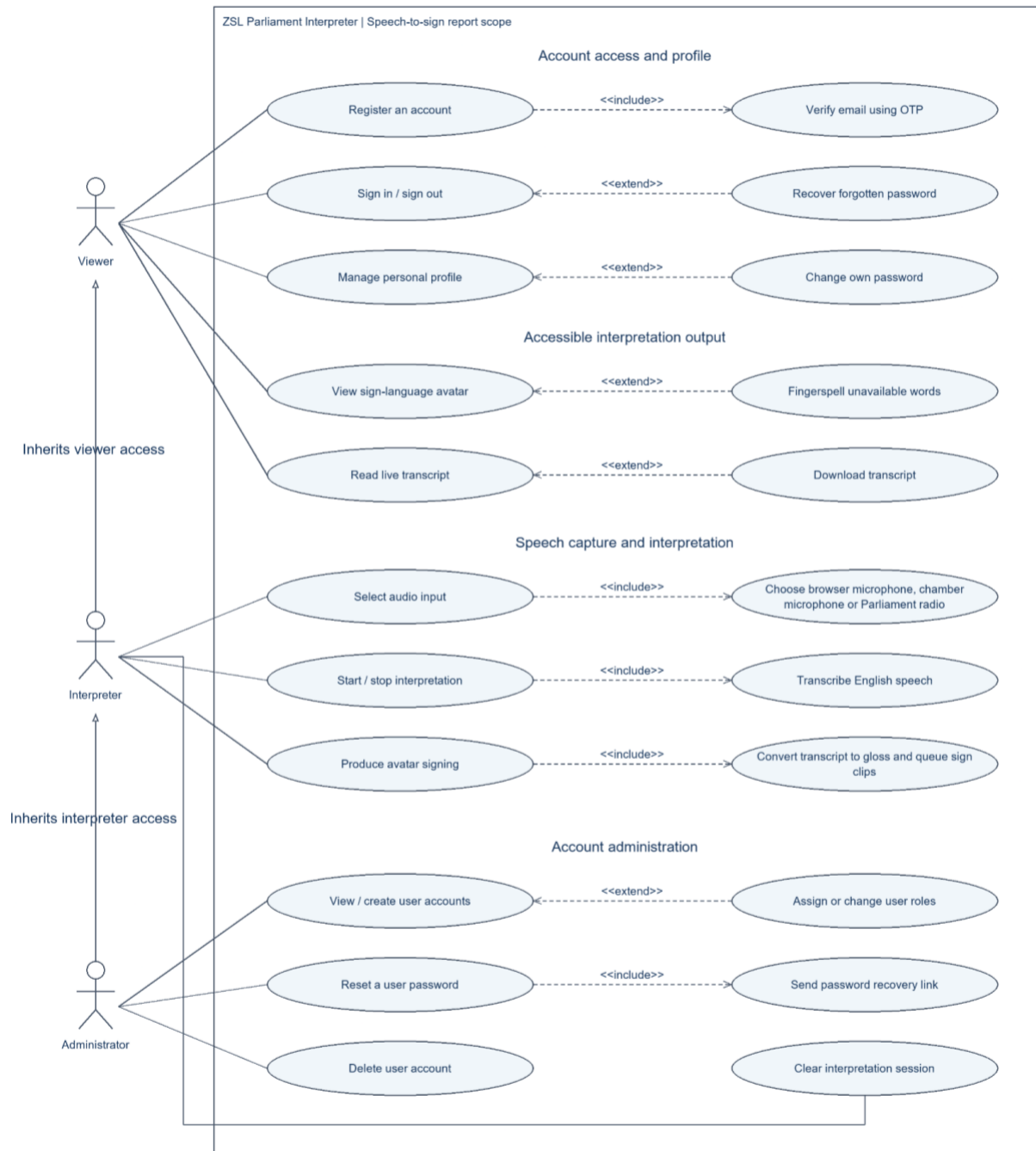


Figure 4.1 use case diagram for different users. The use cases are role-based: the admin can add or remove users, the interpreter can record audio for the system to translate, and the viewer can only view the system interface.

The application presents the speech-to-sign route within one authenticated service. Viewers can open the page; interpreters operate audio recording controls; administrators also manage accounts

and roles. Viewing a page does not imply that a shared chamber broadcast is distributed to all viewer sessions.

The architecture separates browser audio capture and avatar rendering from backend speech inference, gloss mapping and account services. This supports incremental transcription and animation playback with explicit queue policies.

4.2 Technology stack

Table 4.3 Technology stack

Layer	Technology	Purpose
Application	Python, Flask and Flask-SocketIO	Pages, REST routes and incremental events
Accounts	Flask-Login, SQLAlchemy and SQLite	Authentication, user records and roles
Speech inference	NVIDIA NeMo, PyTorch and CUDA	Cache-aware English ASR
Gloss	JSON dictionary; optional Hugging Face pipeline	Text-to-gloss mapping
Browser acquisition	Web Audio API	PCM audio windows
Avatar	Three.js, VRM and recorded landmarks	Skeleton, hand and expression animation
Radio	ffmpeg and streaming audio adapters	Broadcast decoding and audio input
Hosting	AWS EC2, EBS and Cloudflare Tunnel	GPU demonstration environment and HTTPS

4.3 Class structure and persistent data

The speech-processing backend contains a StreamingParakeetASR wrapper around the configured NeMo FastConformer checkpoint. `_StreamState` holds client-specific buffers and caches, while `TranscriptionResult` carries text, completion and error information. The user model supplies authentication and recording permissions. These components support the speech-to-sign pipeline.

Table 4.4 Persistent user data

Field group	Fields	Purpose
Identity	id, email, name, avatar_url	Account identification and profile
Authentication	password_hash, is_active	Password checking and account status
Authorization	Role	Viewer, interpreter or administrator access
Login metadata	created_at, last_login, login_count	Account and login history

Passwords are stored as hashes through Werkzeug helpers. The model implements `can(action)` for recording and administrative permissions. Login metadata should not be described as a complete event audit trail. Session state, recognition buffers and playback queues are separate from the persistent user table.

4.4 Speech acquisition and streaming ASR

The browser captures microphone samples through Web Audio and encodes them as mono PCM WAV windows. The application requests a 16 kHz audio context and uses a 500 ms transport window. A client energy check suppresses some inactive windows, and the server validates speech activity before inference. These gates are heuristics; they are not a measured voice-activity classifier.

Waiting audio windows are queued in capture order. A completion event releases the next window. Stopping capture forward the remaining samples and requests finalization after pending audio drains. During a disconnect, pending client audio is cleared, so continuity is not guaranteed across network interruption.

The backend resamples when necessary and maintains a streaming buffer and decoder caches for each Socket.IO client. Partial recognition can produce incremental gloss output, while finalization supplies the remaining utterance. The implementation requires CUDA-enabled PyTorch for live speech inference. The environment variable `PARAKEET_MODEL` identifies the selected checkpoint despite the wrapper's naming.

Server-connected audio adapters provide microphone and radio input. Radio decoding requires `ffmpeg`. A source is associated with a client session, and source selection restores server-side capture after reconnection. A single shared parliamentary interpretation channel with consistent multi-viewer delivery requires separate design and verification.

4.5 Gloss generation and avatar rendering

The baseline gloss engine normalizes input, expands selected contractions and applies phrase and word mappings. Entries may suppress selected grammatical words. Unknown words remain available for visual fallback. An optional transformer interface can be configured, but dictionary mode is the established baseline. Uppercase tokens are an intermediate representation, not proof of sign-language grammar.

The avatar manifest maps normalized names to recorded JSON clips. Phrase matching takes precedence over shorter matches. Unmatched words are split into alphanumeric characters; available alphabet and number clips are queued in order. Repeated letters are retained. Punctuation does not produce a motion clip, and unsupported character sets need a separate fallback policy.

The player fetches upcoming clips and caches bounded clip promises. It retargets pose and hand landmarks to a VRM skeleton, applies motion smoothing and retains cached recordings without repeated in-place alteration. Playback uses 54 frames per second for recordings described as 30 FPS. Animation completion advances the queue, avoiding reliance on estimated duration. Partial/final gloss overlap is compared by tokens so the final transcript contributes its novel suffix.

The sign queue preserves waiting clips rather than truncating the utterance. This improves completeness within an active session but permits growing delay when input exceeds playback capacity. Comprehension at the configured speed and the grammatical quality of joined clips remain evaluation questions.

Table 4.5: Principal interfaces

Interface	Purpose
/	Speech-to-sign page
/health	Application and speech/gloss readiness
/api/config	Runtime settings and user-facing configuration
/api/avatar/manifest	Available clips and avatar asset endpoints
/api/avatar/clip/<path>	Restricted landmark delivery, with compression support
audio_chunk, transcription_done	Ordered live audio processing
transcript_partial, transcript, gloss	Incremental interpretation output
end_audio_stream, set_source	Stream finalization and source selection

Avatar resources are served from a configured asset directory. The clip manifest is authoritative for allowed landmark paths, and requests outside the indexed resources are rejected.

4.6 Deployment

The verified demonstration environment is an On-Demand AWS g4dn.xlarge in us-east-1, with four vCPUs, 16 GiB system memory and one NVIDIA T4 GPU. The root disk is a 50 GiB EBS volume. A separate local NVMe instance-store device is present and is not the application root disk. GPU-enabled PyTorch and OpenCV are installed in the application environment.

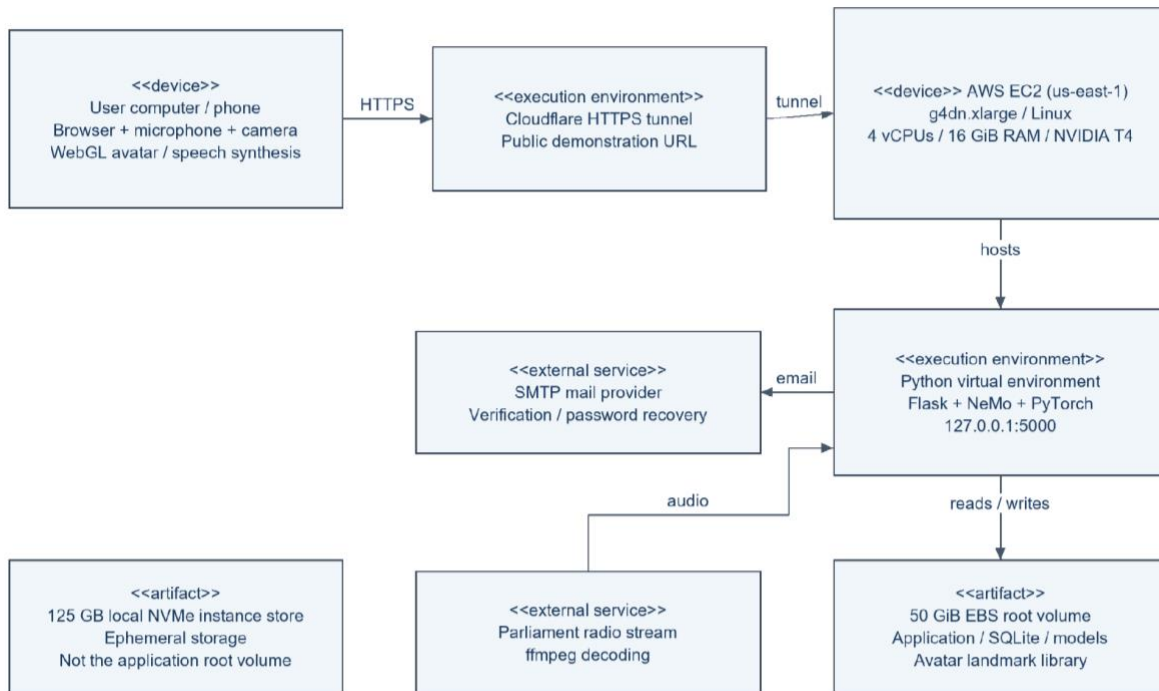


Figure 4.6 UML deployment diagram. The browser accesses the loopback-bound application through HTTPS tunneling. External mail and radio services remain separate dependencies.

The application listens at 127.0.0.1:5000, with public HTTPS provided by Cloudflare Quick Tunnel. The tunnel URL is temporary rather than a permanent institutional address. Deployment tooling stages a bundle, checks Python syntax and recognition execution, backs up replaced files, starts the application and checks readiness. Formal production operation requires supervised services, a stable endpoint, recovery testing and an appropriate serving configuration.

4.7 User interface design

The user interface of the ZSL Parliament Interpreter is designed to provide a browser-based environment for speech-to-sign interpretation. It connects the user to audio capture and processing while presenting transcripts, intermediate gloss and avatar animation in an organized format.

The principal interpretation interface is the speech-to-sign page. Additional interfaces support authentication, account management and administration.

The interface provides audio source selection and recording controls alongside transcript, gloss and animated signing displays, keeping user actions and feedback clear.

The interface communicates with the backend through HTTP endpoints and Socket.IO events. HTTP requests support page and avatar-resource retrieval, while Socket.IO supports incremental speech transcription and gloss updates during audio streaming.

The major user interface components are described below.

4.7.1 Login and Authentication Interface

The login interface provides access to the application for registered users. It collects the user's authentication details and submits them for verification by the backend.

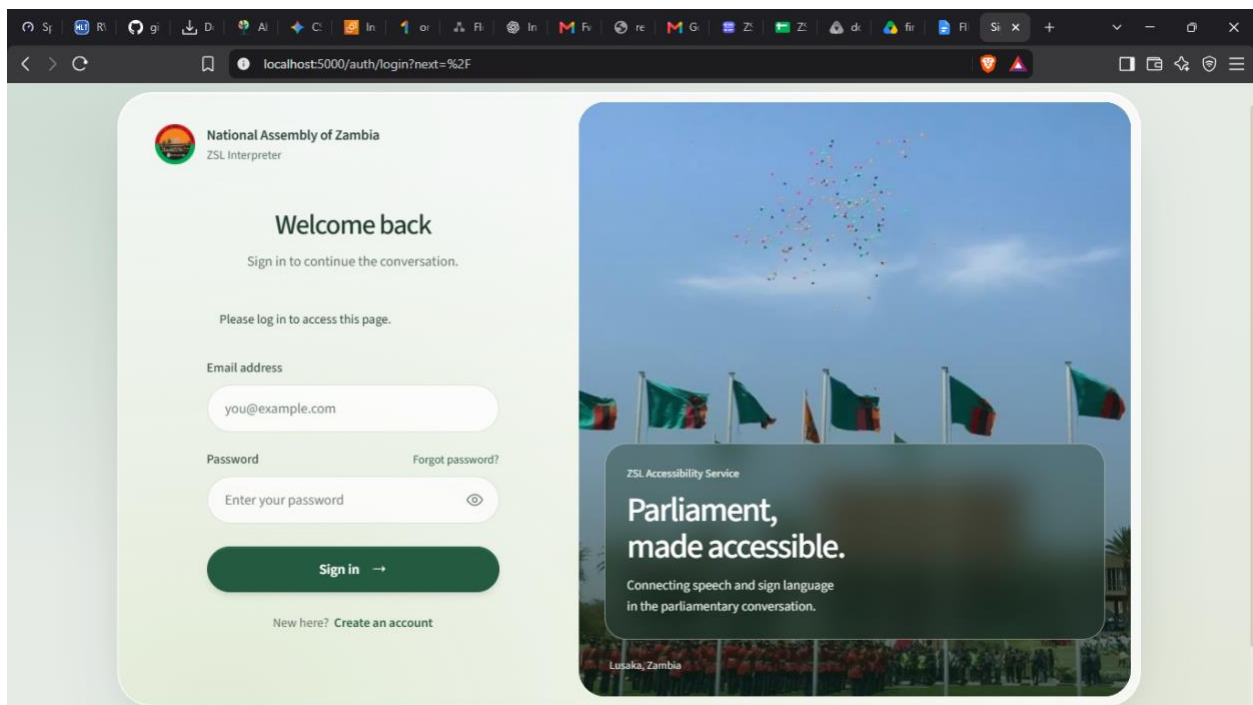


Figure 4.7 Login and authentication figure - User login page.

The authentication system uses Flask-Login and persistent user records managed through SQLAlchemy and SQLite. Passwords are stored as hashes rather than plain text.

The application supports different access roles, including viewers, interpreters, and administrators. Viewers can access the interpretation pages, interpreters can operate authorized recording controls, and administrators can manage user accounts and roles.

The application also supports registration verification and password recovery workflows, which depend on the appropriate email configuration.

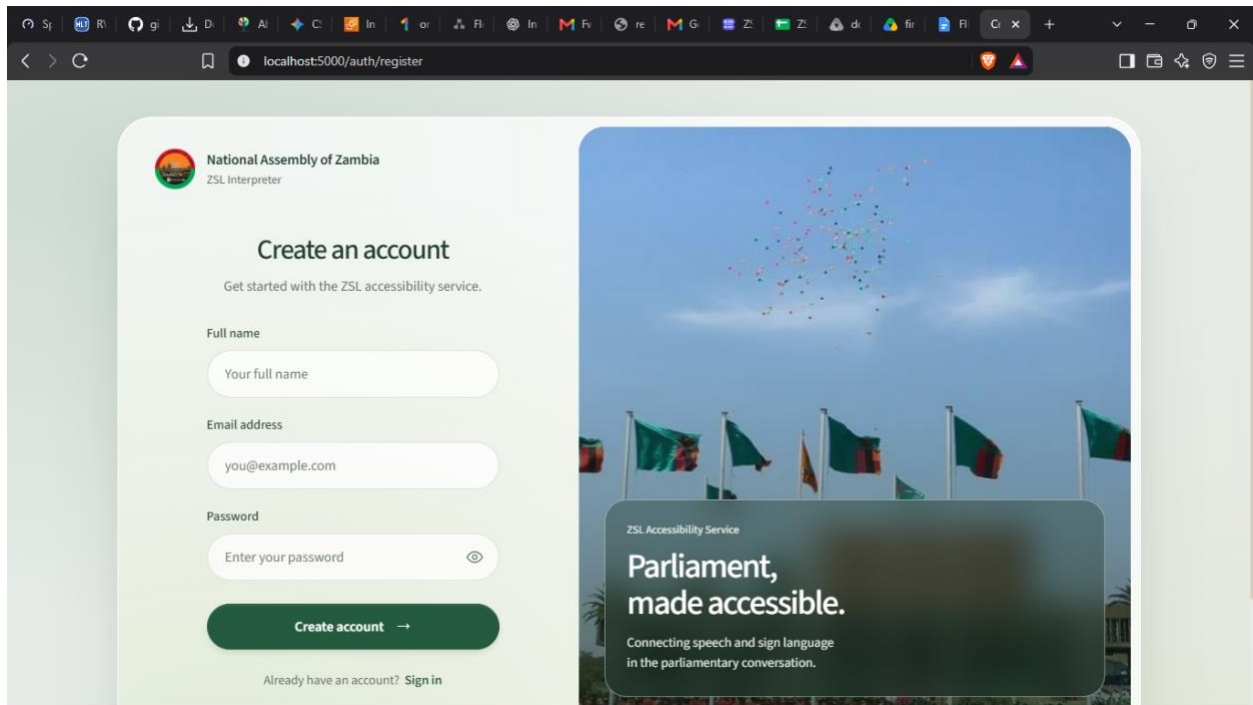


Figure 4.8 Login and authentication figure - User account creation page

4.7.2 Speech-to-Sign Interface

The speech-to-sign interface allows an authorized interpreter to capture spoken English and display the corresponding visual output.

The interface supports audio acquisition through the browser microphone and provides options for supported server-side audio sources. During operation, the application displays the recognized transcript, the intermediate gloss representation, and the corresponding avatar animation.

The principal interface components include:

- Audio source selection: Allows the user to select an available audio input source.
- Recording controls: Allow authorized users to start and stop browser audio capture.
- Transcript display: Presents recognized English speech as text.
- Gloss display: Shows the intermediate representation generated from the transcript.
- Three-dimensional avatar display: Presents available recorded sign movements through a VRM avatar.
- Processing status: Communicates relevant recording and processing states.
- Transcript export: Provides a way to export available session text.

The interface receives incremental transcript and gloss updates through Socket.IO events. This allows the application to display intermediate results while speech processing continues.

The avatar retrieves available motion sequences from the configured library. When a word or phrase has no direct animation mapping, the application can use available alphabet or number animations to represent supported characters.

The displayed gloss and avatar output are intermediate representations. Their presence does not establish that the resulting animation accurately expresses ZSL grammar or meaning.

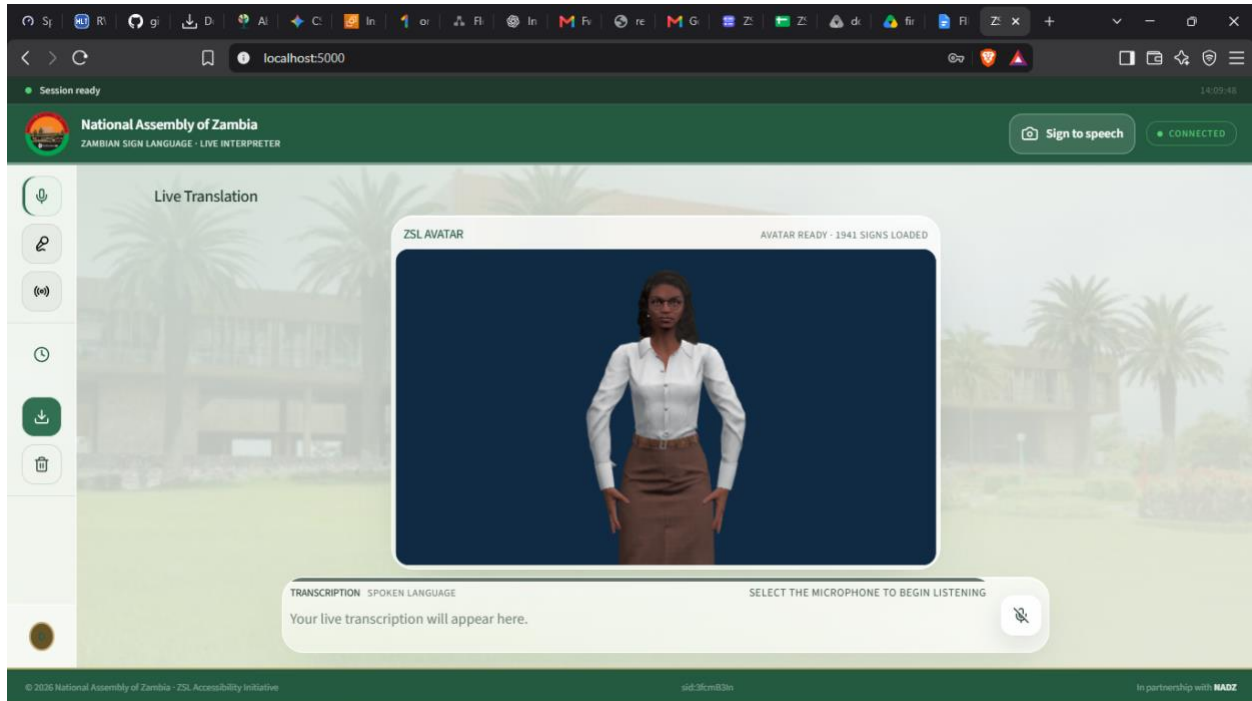


Figure 4.8 Showing the speech to sign window

4.7.3 Administrative Interface

The administrative functionality supports the management of application accounts and access permissions.

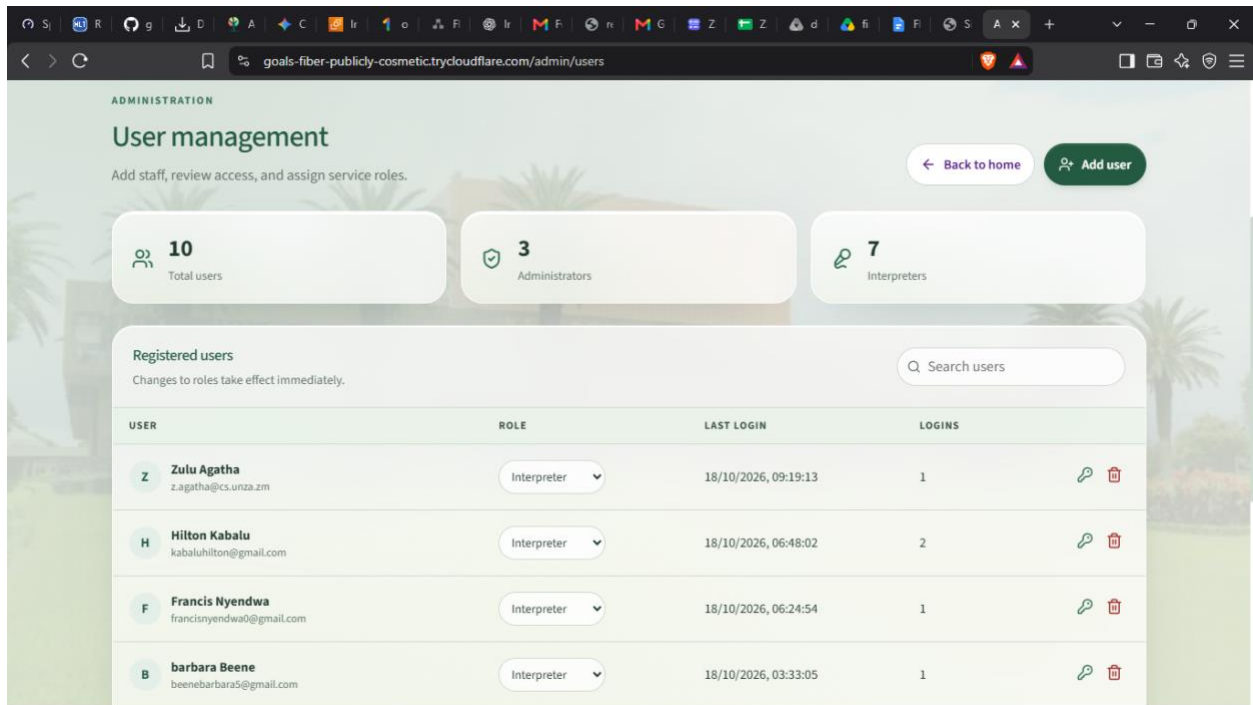


Figure 4.9 Admin Dashboard showing users registered for the system.

Administrators can manage users and assign roles according to the application's authorization rules. The role system separates ordinary viewing privileges from recording and administrative operations.

These controls help restrict access to microphone functions and support more controlled operation of the prototype.

Chapter 5: Testing and Results

5.1 Verification strategy

Testing separates software contracts from language accuracy. Automated tests verify speech capture, event ordering, text processing, avatar motion playback, authentication and resource delivery without treating successful execution as evidence of correct ZSL interpretation.

The relevant verification results comprise component and browser checks of audio handling, transcription, gloss output, avatar playback and authentication. Passing tests establish selected software behaviours, not speech-recognition accuracy or ZSL comprehension.

Table 5.1 *Verification evidence*

Verification group	Observed result	What it establishes
avatar-player.test.cjs	Passed	Startup buffering, ordered signs, alphabet fallback, final suffix and caching
avatar-finger-curly.test.cjs	Passed	Selected bent/straight finger-chain geometry
audio-queue.test.cjs	Passed	Audio ordering, stop-tail forwarding and finalization
Public HTTPS request	HTTP 200 at verification	Login-page reachability, not continuous availability

5.2 Resource and implementation results

Table 5.2 *Resource inventory*

Resource	Verified count	Interpretation
Gloss word entries	1916	Mapping entries, not all distinct animated words
Gloss phrase entries	25	Text phrase mappings
Landmark sequences	1941	Recorded clips, including variants and duplicate vocabulary

Resource	Verified count	Interpretation
Landmark categories	28	Parent category directories
Alphabet clips	26	Recorded A–Z animations

The delivered application connects audio acquisition, transcript and gloss generation, and avatar playback within an authenticated account system. Speech output reaches a sequential avatar queue, and unknown word tokens can reach alphabet playback.

These are integration results. Correct transcript and gloss event handling or successful animation loading do not demonstrate linguistic correctness. Similarly, correct finger-chain geometry does not establish the semantic intelligibility of a complete signed utterance.

5.3 Speech recognition and signing observations

The observations concern transcription quality, timing, avatar motion and intelligibility. Speech recognition errors can propagate into the gloss and visual output, while gaps in recorded signs can require slower fingerspelling. The participant-response and recorded-speech evidence are evaluated in the subsequent sections.

Word-level mappings and alphabet fallback improve coverage but do not guarantee sentence-level ZSL accuracy. Assessment should distinguish the accuracy of the transcript from the linguistic quality and comprehensibility of the avatar output.

5.4 Measures needed for empirical evaluation

Table 5.3 Evaluation measures

Measure	Procedure	
Speech Recognition Accuracy	Compare the generated transcript with the reference transcript and calculate the Word Error Rate (WER).	
Output coverage	Count direct signs, fingerspelled words and unsupported characters	
Latency and backlog	Timestamp capture, inference, output and queue waiting	
Signing comprehension	Qualified users assess semantic understanding	

Measure	Procedure	
Usability	Task completion, observation and accessible questionnaires	

WER can be calculated as $(S + D + I) / N$, where substitutions, deletions and insertions are compared with the number of reference words. Speech-to-sign evaluation should also report output coverage, speech-to-text delay, animation queue waiting and participant comprehension separately.

5.5 Threats to validity

Synthetic or controlled input may not represent actual parliamentary speech, varied Zambian accents, complex names or sustained live debate. Dictionary coverage and animation playback on a test device cannot establish comprehension across ZSL users. Generalization therefore requires representative audio and independent sign-language review.

The report makes no claim of statistically significant accuracy, a participant satisfaction score or a verified ZSL corpus. These evidence gaps limit deployment conclusions even when the application executes successfully.

chapter 5 Evidence and participant-response evaluation

This analysis contains two consenting respondents, P01 and P02. P01 described their ZSL experience as intermediate and reported use several times a week; P02 described advanced experience and daily use. Both tested on Android phones with Chrome and stable internet. The results therefore describe a small formative evaluation under similar device and connection conditions, not a representative study of all intended users. Participant codes do not indicate the number recruited or imply that eight people responded.

Table 5.4: Interpretation of responses from participants

Measure	Finding from the supplied responses	Interpretation
Login	Both logged in without help	Encouraging evidence of basic access on the tested devices
Finding the browser microphone	Rating 5 and 4; mean 4.5	Generally favorable rating, although P02 also selected

		microphone access as an issue
Menu labels and icons	Both said all were understandable	Navigation was understandable to these respondents
Display	P01 reported no problem; P02 found the avatar/signing area difficult to see	Visibility remains an accessibility concern despite successful navigation
Task transcription	Five reports of “Correct”; one of “Partially Correct” across six participant–task reports	Self-reported correctness, not an independently scored recognition accuracy rate
Avatar response	Signing appeared in all six tasks reports	Output was triggered; this does not establish that the intended meaning was correctly signed
Overall ease of use	Rating 5 and 4; mean 4.5	Favorable initial usability feedback
Potential usefulness after problems are addressed	Both rated 5: mean 5.0	Conditional perceived usefulness, not a rating of present reliability

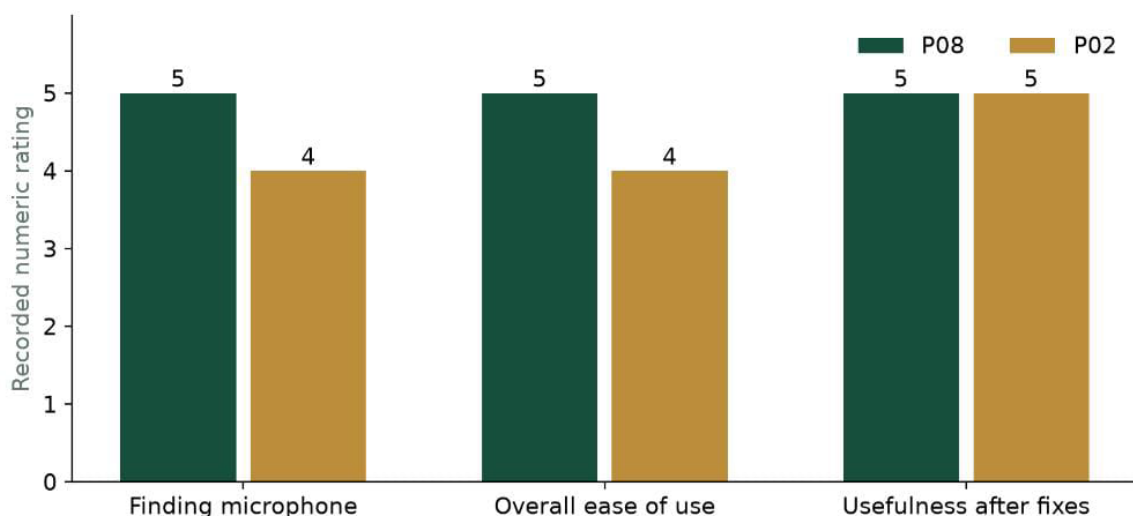


Figure 5.1: Individual participant ratings. *The export does not supply scale endpoints or anchors; usefulness is conditional on addressing the identified problems.*

The exported questions do not include the numeric scale endpoints or anchors. The means above use the recorded numeric values; they are not a validated usability score, SUS result, or percentage satisfaction measure. There are two respondents, not six independent participants: each respondent completed three tasks.

Table 5.5: Example words that were given to the participants and their responses

Task	P02 transcription report	Avatar response
A: Hello	Correct	Both: signing appeared :under 3 seconds
B: Thank you	Correct	Both: signing appeared: under three seconds
C: How's the baby	Correct	Both: signing appeared: under 3 seconds

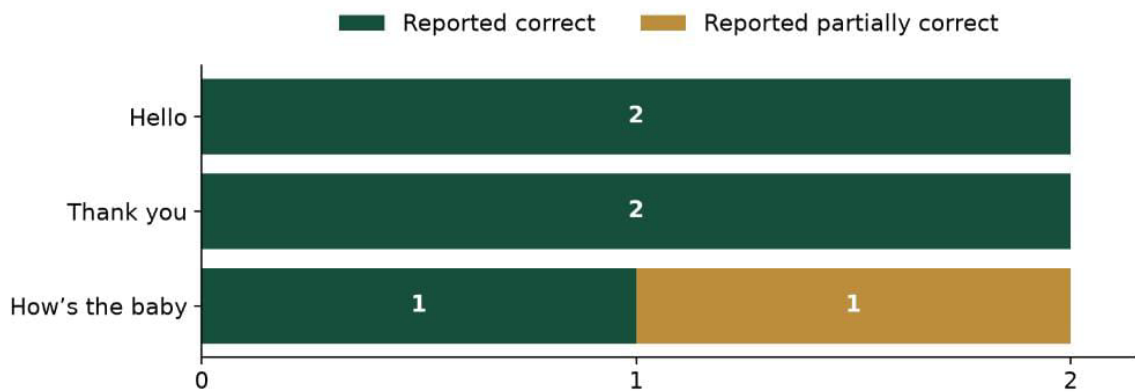


Figure 5.2: Reported transcription outcomes for three tasks. *Six reports come from two respondents, not six independent participants; this is self-report rather than independently scored accuracy.*

P08 described recognizable Hello and Thank you movements and reported that repetition corrected an initial transcription problem on Task C. However, P08 also noted that some signs were understandable because the intended word was already known. P02's free-text interpretations for Hello and Thank you referred to children and teachers rather than clearly reproducing the assigned phrase. These responses may reflect additional testing or task misunderstanding; the export does not resolve that ambiguity. Consequently, the responses cannot support a reliable percentage for sign comprehension or validated ZSL translation accuracy.

1. What do the results mean?

The combined evidence supports technical feasibility and promising initial usability, while identifying limitations in transcription, sign production and playback throughput.

The software verification recorded 80 passing Python test cases and seven passing JavaScript test files. These checks establish selected account, processing, recording-integrity and playback behaviours. They do not establish linguistic correctness. In four recorded speech trials, normalized word error rate was 11.54%, comprising six-word edits across 52 speaker-confirmed reference words. Individual trials ranged from 0% to 25% WER, indicating that performance varied by utterance. The observed errors included incorrect wording, a truncated proper name and duplicated words.

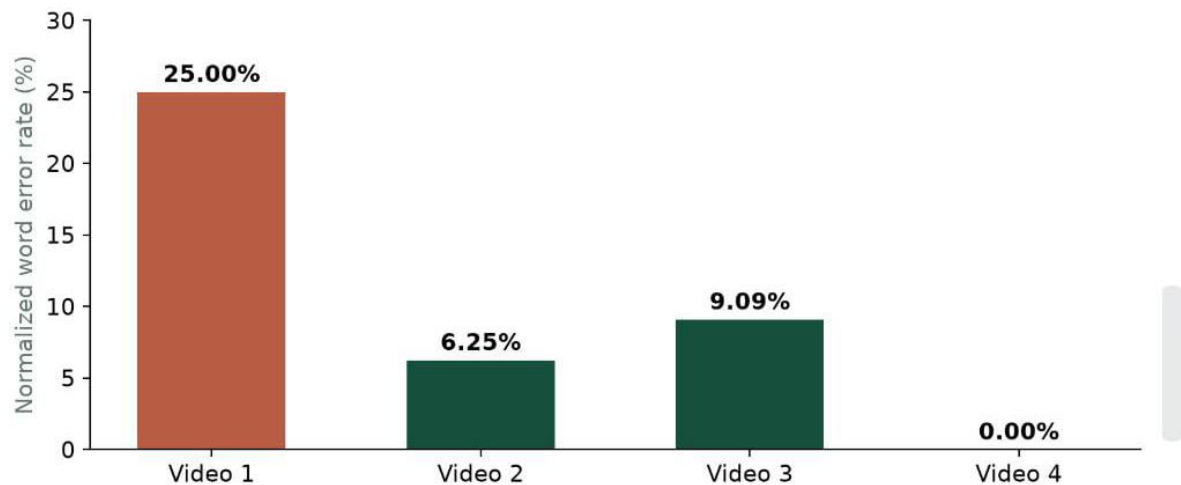


Figure 5.3 Normalized WER for four trials. Pooled WER is 11.54%: six edits across 52 speaker-confirmed reference words. Case, punctuation and numeric wording were normalized as documented in the video evaluation.

The estimated median spoken-word-end-to-text delay was 2.18 seconds across 40 matched displayed words; the 95th percentile was 2.92 seconds. This suggests that text was reasonably prompt in these recordings, but it is not evidence that every spoken word appeared correctly within three seconds. Incorrect or missing words were excluded from those matched-word timing observations. Speech boundaries were automatically aligned and screen transitions sampled every 0.25 seconds.

Word-level medians use 6, 15, 10 and 9 matched words in videos 1–4. Speech boundaries are automatically aligned; screen observations are sampled every 0.25 seconds.

Participants' reports of final text appearing under three seconds after stopping recording are broadly compatible with a responsive text interface. They do not directly confirm the measured 2.18-second median: stop-to-final-text delay and word-end-to-first-visible-text delay have different start and end points.

Avatar output was slower when words were fingerspelled. Across 46 observed word-label playback intervals, the median saved-sign interval was 2.00 seconds, compared with 9.25 seconds for a fingerspelled word. In one example, Nigeria's playback label began 36.41 seconds after its estimated spoken-word end because earlier words occupied the queue. These intervals include loading, transitions, letters and pauses; they are not expert-verified physical sign durations.

The participant comments help explain why output speed alone is insufficient. Both raised transcription concerns and hand/finger/movement concerns. Both asked, in different wording, for common words with established signs to be signed rather than spelled. P08 specifically reported an incorrect N handshape and difficulty triggering the same word or sentence repeatedly. P02 reported confusing movement and an avatar area that was difficult to see. These are reported observations requiring reproduction and linguistic review, not independently confirmed defects in every case.

2. How do the findings address the original problem?

The original problem was the limited accessibility of audio-led parliamentary communication for audiences who need visual language support. The prototype addresses part of that problem by combining visible transcripts and animated output in an accessible browser workflow. Both respondents could access the system without login assistance, and they regarded its potential usefulness favourably after the identified problems are addressed. P08's positive interpretation of Hello and Thank you provides a small amount of encouraging qualitative evidence for familiar signs.

However, providing visible output does not automatically provide reliable access to meaning. Incorrect transcripts can feed incorrect words into gloss and animation. Unclear handshapes and movement can reduce intelligibility even when the transcript is correct. A long fingerspelling queue can cause the avatar to fall behind the speaker, which matters for following a live debate.

The defensible conclusion is that the project demonstrates a potential supporting accessibility channel, rather than a proven replacement for qualified interpretation or an established parliamentary service. The questionnaire tasks were short everyday phrases, not a representative parliamentary comprehension test, and there was no controlled comparison with an existing accessibility service.

3. How do the findings relate to the objectives?

Passing all 26 alphabet recording-integrity tests does not contradict the reported N concern. Those tests check nonempty recordings, expected landmark counts, finite coordinates and usable tracking. They do not check whether the displayed handshape is the correct ZSL letter. Similarly, a passing repeated-playback regression does not prove every real session can replay identical input successfully. The questionnaire contains no build identifier or detailed reproduction trace, so the precise scenario needs further investigation.

P02's observation that "is" was not signed also needs linguistic interpretation. The gloss engine deliberately omits some English function words. Word-for-word correspondence with English is

not a sufficient test of ZSL grammar; a fluent reviewer must judge whether the intended meaning is preserved.

4. How do the findings compare with existing work?

The comparison is about approach and evaluation scope, not a numerical claim that this prototype outperforms published systems.

Table 5.7: *Comparison with existing work*

Existing work	Relationship to this project	Appropriate comparison
Saunders et al., Progressive Transformers	Joint continuous sign recognition and translation, evaluated on PHOENIX14T	Recognition and translation require task-specific evidence. The project's English audio-transcription WER is not comparable with signed-video recognition WER or translation BLEU scores. Sign Language Transformers
Camgoz et al., Sign Language Transformers	Learned generation of continuous 3D signing poses, including a gloss-mediated configuration	This project uses recorded clips and retargeting. Its participant feedback and queue measurements identify handshape, continuity and throughput concerns for that approach; they do not show comparative superiority or inferiority. Progressive Transformers

The main contribution demonstrated here is the integration of accounts, speech capture, transcripts, a motion library and avatar playback within a parliamentary-oriented browser application. It is not evidence of a fully validated ZSL translation service.

5. What is the most important finding?

A functioning and initially usable application does not yet guarantee understandable, timely signing. The central remaining challenge is the fidelity and throughput of the visual-language output: handshape and movement quality, appropriate use of established signs, and queue delay during fingerspelling.

This finding combines measured evidence with user experience. Software checks passed and respondents could operate the application, but the recordings showed fingerspelling intervals substantially longer than saved-sign intervals, while the respondents requested clearer movements and more appropriate signs. The reported N handshape is a concrete example of why valid landmark data is not the same as linguistically correct signing. Its priority is review and verification, not an assumption that the entire alphabet has failed.

Priorities arising from the evidence

1. Review sign meaning and handshapes with fluent ZSL reviewers. Start with N and the evaluated phrases. Use unlabelled or blinded identification trials so knowing the intended word does not inflate comprehension.
2. Review common-word coverage and fingerspelling choices. Map established ZSL signs where appropriate and measure queue growth during longer utterances. Do not speed up or substitute signs without checking comprehension.
3. Reproduce session and transcription concerns. Test repeated identical utterances, proper names, numeric statements and microphone behaviour under recorded conditions.
4. Improve avatar visibility and motion quality. Check the viewport, finger articulation, wrist posture and coordinated arm movement on Android Chrome, then test additional devices and browsers.
5. Broaden the evaluation. Recruit more ZSL users, include representative parliamentary content and varied connections, measure comprehension separately from text accuracy, and evaluate avatar signing across longer utterances.

Presentation-ready discussion

- Meaning: The prototype is functional and received favourable initial usability feedback, but transcription and avatar quality remain uneven.
- Original problem: It provides a potential visual channel for parliamentary information; reliable access to meaning has not yet been established.
- Objectives: Speech acquisition, transcription, gloss mapping and playback were demonstrated; ZSL correctness and sustained playback throughput need further evaluation.
- Existing work: The contribution is speech-to-sign system integration; its speech WER cannot be compared directly with published sign-recognition or translation scores.
- Key finding: Clear, linguistically appropriate signing at a sustainable pace is the main remaining challenge, beyond simply making the avatar respond.

Sources and evidence boundaries

- Participant evidence: supplied pasted Google Sheets export, two responses dated 18 October 2026. No live sheet was accessed.

- Software evidence: Testing results, reporting 80 Python cases and seven JavaScript files passed.
- Recorded-video evidence: Video evaluation and its detailed result sheets, four trials from one speaker.
- Objectives and context: Project report, sections 1.1–1.4 and 2.1–2.5. Its earlier empirical-scope statements are not used to erase the subsequently supplied evaluation evidence.
- Comparison sources: the three primary research papers linked above, checked on 19 October 2026.

No statistical significance, validated comprehension percentage, population-level satisfaction rate, or institutional readiness claim is supported by these data. The participant ratings, six participant–task reports, four recordings and software checks are different units of evidence and are not combined into a single accuracy or pass percentage.

Chapter 6: Discussion

6.1 Interpretation of the system outcome

The system demonstrates a practical composition of streaming speech recognition, gloss generation and avatar-based visual output within an authenticated web application. This provides a basis for investigating speech-to-sign accessibility in the parliamentary context.

English speech produces dictionary-mediated motion with alphabet fallback. The generated output requires validation against ZSL linguistic conventions, especially grammar, spatial reference and non-manual features.

6.2 Acquisition continuity and processing capacity

Continuous audio acquisition, continuous transcription and continuous sign-language interpretation are different properties. The recognizer can maintain audio state, but an ordered avatar playback queue may still accumulate delay during sustained input.

Audio and avatar queues favour ordered completeness during an active connection. This protects the intended sequence of spoken words and visual clips, while potentially increasing delay when speech input exceeds playback capacity.

6.3 Vocabulary and linguistic fidelity

Fingerspelling improves handling of unfamiliar names and unsupported words, but spelling every word does not replace sign-language translation. Alphabet speed, repeated letters, word boundaries and non-English characters need comprehension testing. Recorded clip provenance and language identity must also be reviewed.

The available word and phrase motion vocabulary is finite. A fallback to fingerspelling provides additional coverage but may slow playback and cannot substitute for a verified ZSL sign or sentence grammar.

6.4 Security and operation

The application provides hashed passwords, login sessions, email verification and role checks. These form an access-control foundation, not a complete production security assessment. Institutional hosting requires review of route permissions, cross-origin configuration, request forgery protection, logging, retention and administrative operations.

A temporary tunnel is suitable for demonstration access but is not a stable public service endpoint. Operational readiness also requires service supervision, monitoring, backups, controlled dependency versions and recovery procedures. GPU inference availability is distinct from successful HTTP reachability.

6.5 Limitations

The principal limitations are incomplete evaluation of ZSL comprehension, no validated ZSL sentence translator, dictionary and motion-asset coverage constraints, potential speech-recognition

errors, queue-related delays, and external mail/radio dependencies. Institutional use also requires representative testing and security review.

Chapter 7: Conclusion and Recommendations

7.1 Conclusion

The ZSL Parliament Interpreter provides a browser-based speech-to-sign prototype combining English audio acquisition, streaming transcripts, gloss generation and avatar motion. Account controls and deployment utilities support demonstration and testing. The result establishes an integrated engineering foundation, not validated ZSL sentence translation.

Software verification and preliminary speech/avatar evaluations support selected component execution and usability observations. They do not establish universal accuracy, low latency under all loads or linguistically valid ZSL interpretation. The project therefore contributes a foundation for focused linguistic and accessibility research.

7.2 Recommendations

Table 7.1: Recommended research and engineering work

Priority	Work	Required evidence
1	Improve speech recognition for Zambian accents and parliamentary terminology.	Evaluation transcripts, WER scores and documented recognition errors.
1	Validate the avatar's signs with qualified ZSL interpreters.	Expert assessments of sign accuracy and comprehension.
1	Measure end-to-end delay and overload	Capture-to-output distributions, queue depth and omissions
1	Review motion-language identity and avatar comprehension	Asset provenance and qualified linguistic assessments
1	Assemble representative parliamentary English evaluation audio	Independent transcripts, WER and domain-term errors
2	Collect and annotate consented ZSL data	Documented protocols, signer diversity and community review
2	Adapt models and translation to the intended language	Held-out results and comprehension evidence
2	Define a shared interpretation session for viewers	Multi-client delivery, access-control and recovery tests

Priority	Work	Required evidence
2	Harden institutional hosting	Stable domain, supervised services and security review
3	Conduct accessible usability studies	Task outcomes and documented participant feedback

7.3 Future research

Future work should measure speech-recognition performance using representative, labeled Zambian parliamentary recordings and evaluate avatar output separately with fluent ZSL reviewers. This distinction helps identify whether errors originate in transcription, text-to-gloss mapping or animated sign production.

Speech-to-sign research should address sentence grammar, spatial reference, non-manual expression, appropriate sign vocabulary and motion transitions. Further studies should measure comprehension, sustained playback throughput and accessibility across devices and user groups.

References

- [1] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, “Neural sign language translation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 7784–7793, doi: 10.1109/CVPR.2018.00812.
- [2] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, “Sign language transformers: Joint end-to-end sign language recognition and translation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 10023–10033, doi: 10.1109/CVPR42600.2020.01004.
- [3] A. Radford et al., “Robust speech recognition via large-scale weak supervision,” in Proc. 40th Int. Conf. Mach. Learn. (ICML), vol. 202, 2023, pp. 28492–28518.
- [4] S. Stoll, N. C. Camgoz, S. Hadfield, and R. Bowden, “Text2Sign: Towards sign language production using neural machine translation and generative adversarial networks,” Int. J. Comput. Vis., vol. 128, pp. 891–908, 2020, doi: 10.1007/s11263-019-01281-2.
- [5] L. Naert, C. Larboulette, and S. Gibet, “A survey on the animation of signing avatars: From sign representation to utterance synthesis,” Comput. Graph., vol. 92, pp. 76–98, 2020, doi: 10.1016/j.cag.2020.09.003.
- [6] M. Aziz and A. Othman, “Evolution and trends in sign language avatar systems: Unveiling a 40-year journey via systematic review,” Multimodal Technol. Interact., vol. 7, no. 10, Art. no. 97, 2023, doi: 10.3390/mti7100097.
- [7] T. Ananthanarayana et al., “Deep learning methods for sign language translation,” ACM Trans. Access. Comput., vol. 14, no. 4, Art. no. 22, 2021, doi: 10.1145/3477498.
- [8] “SignMate: Real-time speech to sign language translation,” in Proc. 2nd Int. Conf. Trends Eng. Syst. Technol. (ICTEST), 2025, doi: 10.1109/ICTEST64710.2025.11042509.
- [9] SYSTRAN, “faster-whisper: Faster Whisper transcription with CTranslate2,” GitHub repository. [Online]. Available: <https://github.com/SYSTRAN/faster-whisper>. [Accessed: Oct. 23, 2026].
- [10] International Organization for Standardization and International Electrotechnical Commission, Information Technology—User Interface Component Accessibility—Part 24: Visual Presentation of Audio Information in Sign Languages, ISO/IEC DIS 20071-24.
- [11] NVIDIA, “STT En FastConformer Hybrid Transducer-CTC Large Streaming Multi,” NVIDIA NGC Model Catalog. [Online]. Available: https://catalog.ngc.nvidia.com/orgs/nvidia/nemo/models/stt_en_fastconformer_hybrid_large_streaming_multi. [Accessed: Oct. 23, 2026].

- [12] D. Li, C. Rodriguez Opazo, X. Yu, and H. Li, “Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison,” in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV), 2020, pp. 1448–1458, doi: 10.1109/WACV45572.2020.9093512.
- [13] B. Saunders, N. C. Camgoz, and R. Bowden, “Progressive transformers for end-to-end sign language production,” in Computer Vision – ECCV 2020, 2020, pp. 687–705, doi: 10.1007/978-3-030-58621-8_40.
- [14] K. Schwaber and J. Sutherland, The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game, Nov. 2020. [Online]. Available: <https://scrumguides.org/scrum-guide.html>. [Accessed: Oct. 23, 2026].
- [16] S. Frost, “UKZN graduate develops tech to bridge SA sign language communication gap,” University of KwaZulu-Natal, Sep. 22, 2026. [Online]. Available: <https://caes.ukzn.ac.za/news/ukzn-graduate-develops-tech-to-bridge-sa-sign-language-communication-gap/>. [Accessed: Oct. 23, 2026].
- [15] S. H. Imam et al., “Automatic speech recognition for African low-resource languages: Challenges and future directions,” in Proc. AfricanNLP Workshop, 2025. [Online]. Available: <https://aclanthology.org/2025.africanlp-1.13/>. [Accessed: Oct. 23, 2026].

Appendix A: Operating workflows

For speech-to-sign, an interpreter signs in, chooses an audio source and starts browser acquisition or selects server-side input. Transcript, gloss and avatar output provide complementary feedback. The transcript export records available session text. Stopping the browser microphone drains pending audio and finalizes the stream.

Administrators manage users and roles. Registration verification and password recovery require configured SMTP access. Viewer accounts can open the pages but cannot initiate capture. Account permissions do not by themselves create a shared interpretation broadcast.

Appendix B: Configuration and reproduction

Configuration is read from .env. Use a compatible NeMo, PyTorch and CUDA environment for live speech recognition. The configured speech checkpoint should support the cache-aware FastConformer interface and the speech-to-sign application routes.

Avatar assets require a VRM model, animator module and landmark directory, located through AVATAR_ASSETS_DIR or the project configuration. Audio capture, speech recognition and gloss mapping settings are documented in the repository configuration.

The selected verification commands are documented in the repository README. The remote verification utility uses an isolated in-memory test account to exercise pages and inference without creating a permanent account in the deployed database. A synthetic inference result verifies model execution, not recognition correctness.

Appendix C: Evaluation record and boundaries

Verification evidence for this report focuses on speech capture, text/gloss events, avatar playback, authentication and asset delivery. Relevant JavaScript tests cover avatar playback, finger geometry and audio ordering.

Formal parliamentary ASR benchmarks and large-scale validated ZSL comprehension studies are not yet established. Future evaluations should retain consent information, model identifiers, hardware details, timing traces and linguistically reviewed reference material.

