AI-YM: AI-based Solutions for Kazakh-Russian Sign Language

Aruzhan Kaltay, Bekzat Ongdassynov, Dilnaz Sembekova, Yerzhan Yerbatyr, Karina Iskakova
Computer Science Department
School of Engineering and Digital Sciences, Nazarbayev University
Astana, Kazakhstan

I. Executive Summary

AI-YM - is the interactive web application powered by the artificial intelligence designed to teach hearing individuals Kazakh and Russian Sign Language (K-RSL). This platform addresses the lack of accessible educational tools for learning K-RSL by employing AI to recognize signs performed by users and provide feedback, thereby enhancing the learning experience. The system addresses a significant gap in accessible K-RSL resources, especially for non-deaf users aiming to build inclusive communication skills.

The methodology involved designing and integrating AI gesture recognition and developing a full-stack solution. The project's final result is a fully functional web application that combines AI-based gesture recognition, user role management, interactive lesson creation, and a gamified user experience. Our AI model recognizes K-RSL signs from video input, compares to the original sign trajectory, and provides feedback to learners. Additionally, the learners can track their progress by earning XP points, use the streak system, and enroll in the desired courses. Finally, teachers can create their own learning content, and contribute their materials to the AI-YM platform.

This project demonstrates the end-to-end design, implementation and evaluation of our solution to a real-world social challenge. The project employs machine learning, human-computer interaction, and web technologies to develop a responsive, inclusive, and adaptive educational tool.

II. Introduction

A. Problem statement

Effective communication between hearing and non-hearing individuals is a growing social priority, especially in multilingual and multicultural environments like Kazakhstan. However, no structured platforms exist to teach K-RSL, particularly for hearing individuals who want to bridge the communication gap with the deaf community, and this topic remains under-resourced.

Traditional sign language learning platforms are static and limited in scope. Most rely solely on pre-loaded content designed by a small team of educators or developers. As a result, learners are limited by fixed curricula and subject matter experts, including members of the deaf community, who are often excluded from contributing their valuable insights.

B. Motivation and Significance

The primary motivation behind this project is to create an accessible and scalable platform beyond traditional sign language education. The platform is driven by the idea that allowing people to become educators will make a richer, more diverse set of learning materials. It supports the idea of active participation of all type of users at any stage of the platform and especially those from the deaf or educational communities, which enables them to share their unique perspectives and experiences.

The significance of this project lies in its potential to change the way sign language is taught and learned. It creates a cooperative learning environment where the curriculum is directly influenced by the community. By doing so we are solving the initial problem of the static platforms, and guarantee that the curriculum is always current, culturally appropriate, and responsive to the needs of both teachers and students.

C. Overview of the Proposed Solution

The proposed solution is AI-YM, a web application platform that designed to teach Kazakh-Russian Sign Language. This platform allow learners to practice signs, get feedback, and track their progress. Also, we implemented a content-creator studio, where experienced learners can become a teacher, and create their own courses. Overall, the platform aims to offer an inclusive, scalable solution for learning sign language.

D. Agenda of the report

The report will first provide background information from the extensive literature review, following this introduction. It will then transition to the project approach, where we will describe the algorithms for the AI model, the software selection, the overall idea creation, and show the use case diagram and the database structure. After that, we will dive into the project execution, focusing on how the work was structured and the job done in this and past semesters. The next section of the report will focus on user testing, the feedback from that evaluation, and our responses to it. Lastly, we will summarize key findings, suggest improvement areas in the conclusion section, and list all references in the references section.

III. Background and Related Work

The World Health Organization estimates that more than 446 million people worldwide suffer hearing impairments. Sign language is the main form of communication for these people. However, a significant obstacle to inclusive, real-time communication is that most hearing people are not proficient in sign language [1], [2]. To fill this gap, the field of sign language processing was created.

Assistive services like Siri, Alexa and Alisa mainly use voice and text input and are examples of current solutions. These tools are helpful in some situations, but are unsuitable for sign language users because they do not support visual-gestural language input. Furthermore, essential factors like the necessity of varied, real-world datasets and the linguistic subtleties of various sign languages are frequently overlooked by researchers [1].

One of the noteworthy contributions to Kazakh-Russian Sign Language is the FluentSigners-50 dataset, which comprises more than 43,000 annotated samples of daily K-RSL usage by local Deaf people [3]. This dataset has dramatically improved the performance of machine learning models for translation and recognition tasks. However, few sign language datasets are available, particularly for underrepresented languages like K-RSL.

Researchers identify the need for an interdisciplinary approach in sign language technology—the convergence of artificial intelligence, linguistics, engagement with Deaf communities, and human-computer interaction [1]. Educational devices have also been a highly sought-after approach to achieve this method. For example, the "Shuwa" project provided an AI-powered learning platform to support a range of sign and spoken languages, including English, Traditional Chinese, Japanese, and Hong Kong and Japanese Sign Languages. The site offers a chance to rehearse signs using a webcam and receive immediate machine learning-driven feedback. Notably, it relies on low-cost hardware such as ordinary webcams instead of expensive 3D sensors that tend to be a barrier to mass adoption [4].

Among these are quiz-based learning programs for Indian Sign Language, avatar-based 3D systems, and AR-capable smart glasses. These programs have shown technology's capability to significantly assist the teaching and learning of sign languages [5]. However, some of these programs still rely on pre-curated content and are afflicted by the lack of scalable, diverse datasets.

IV. Project Approach

The solution is designed as a web-based platform for teaching Kazakh-Russian Sign Language (K-RSL) using AI-driven gesture recognition. The platform allows users to practice sign language through feedback from the AI system, which processes user gestures and provides accuracy scores for each landmark.

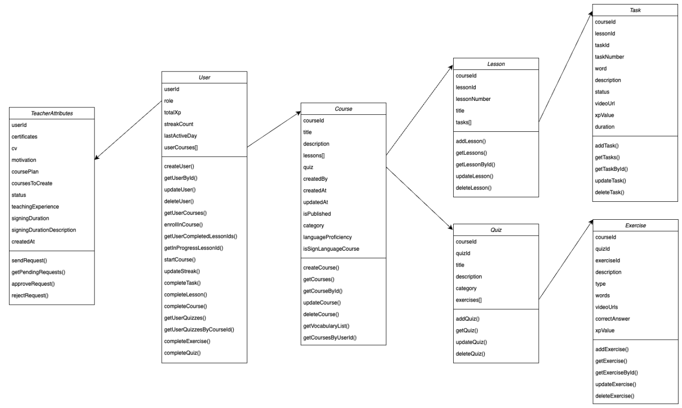

Fig. 1. Class diagram.

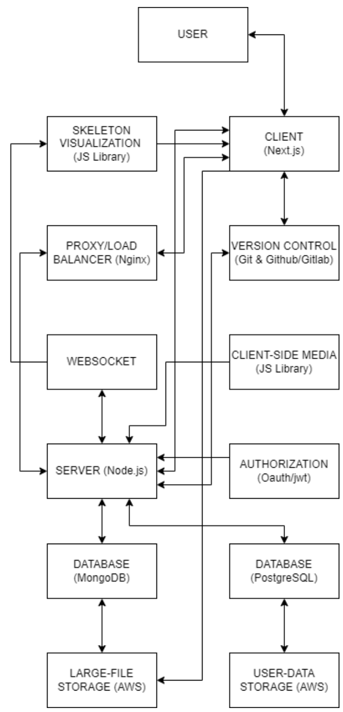

Fig. 2. Architectural model of the AI-YM.

A. Software and Hardware Architecture

The platform has a modular design with a front-end and back-end linked by RESTful APIs. The frontend is developed using **Next.js** to ensure a responsive and user-friendly interface. The backend, where the main logic and data processing occur, is built with **Node.js** and **Express**, which handle user management, course progression, and interactions with the AI components. The database is managed using **MongoDB** for its flexibility in storing user data, lessons, and progress tracking. As shown in Figure 1, the class diagram illustrates the database structure.

The AI module uses computer vision and machine learning models, specifically MediaPipe and OpenCV, to recognize and assess user gestures. The system processes webcam input,

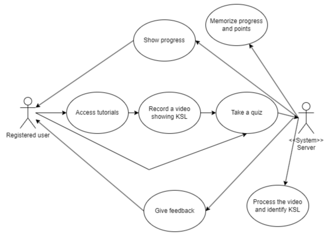

Fig. 3. Use-case diagram for the learner.

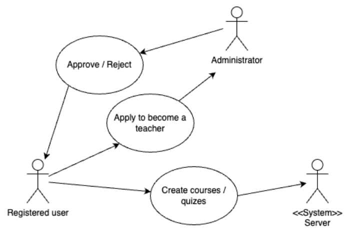

Fig. 4. Use-case diagram for the teacher.

compares it to an original gesture video, and provides feedback based on gesture accuracy. Using Python was an architectural challenge since our backend framework is built on Javascript. To tackle that, we had to create a separate FastAPI service that handles exclusively the use of the model. The model consists of 2 main parts - landmarks extraction from a video, and comparison of 2 videos (reference and user uploaded). Since reference videos in our database are inconsistent in timing of signs and starting positions, we had to implement an algorithm called dynamic time warping. It finds the best matching of frames between 2 videos, which accounts for issues in timing and velocity (temporal alignment). The comparison is preceded by 3 step normalization - general (centering and scaling), based on hip position, based on hand orientation. Gestures are then formed and compared, accounting for distances and angles. At the end of the pipeline, the landmarks are applied on top of the user uploaded video to serve as a feedback. The parts where landmarks differ greatly are highlighted in a spectrum of green-red.

B. User Workflows

Registered users automatically start as learners, and their workflow begins when they select a desired course. The course consists of several lessons with several sub-tasks, or in other words, sign videos. The user watches the videos and performs sign language gestures for a specific task. The AI model evaluates the accuracy of these gestures and provides feedback. Users can review their performance, track their learning progress, and move on to subsequent tasks. The use-case diagram for the learner is illustrated in Figure 3.

A learner can apply to become a teacher by sending their CV, motivation, teaching and signing experience, and the approximate plan of their coursework structure. After the system administrators approve them, teachers can create new lessons, quizzes, and interactive content using an intuitive course creation tool integrated into the platform. The use-case diagram for the learner is shown in Figure 4.

C. Roles and Features

Students, Teachers, and Admins are the three main user roles. Students access classes, finish assignments and monitor their progress. Teachers create new and manage their lessons and quizes. Admins are in charge of managing content and user accounts. Some key features include a content-creator studio for teachers, progress tracking via an XP and streak system, and feedback on gesture performance.

D. Third-Party Components and Integration

We utilized a number of third-party tools in order to improve the platform's functionality and do not create difficult algorithms and services from the scratch. MediaPipe and OpenCV libraries of Python were used for our AI model, MongoDB was used for the storing of user and course information. Due to the lack of enough space to store large data, such as videos and documents, we implemented the upload feature through the UploadThing service. Overall, all third-party components were seamlessly integrated and tested by our team.

E. Team Functioning and Development Process

The project was developed using Agile methodology and team members were assigned tasks according to their areas of interest. The team divided into three main sub groups: AI model development, frontend, and backend development. Every two week we held sprint meetings, to make sure that our team maintained alignment with project goals and timelines. Additionally, the status meetings with project advisors were held in order to keep them updated and gathering their feedback. From the software, last semester we used Notion for project management, but this semester we moved to google documents and github dashboards. Github still remained for the version control.

V. Project Execution

A. Design Evolution and Key Changes

We created the user interface design of the platform using Figma in the first semester. Initially, the major components were quizzes and courses, each on a separate page without any backend connection. After discussing the idea with our instructors and receiving feedback from them, we arrived at the conclusion that courses will be made up of lessons and each lesson will have a title deriving from the course's core topic. Initially, we referred to the learning items as "tasks." The "Sozder" dataset containing word pairs and videos was used as the source for such tasks and was provided by the Nazarbayev University Human-Robot Interaction (NU HRI) Lab. To start with, we just implemented a "next" button and a dropdown menu for navigating between tasks and assignments, later recognizing it to be restrictive and confusing via user tests. Furthermore, we implemented an LLM-based system that generated personalized quizzes based on the user's performance. Implementing it, we discovered the logic of tracking and completing tasks, lessons, and courses to be more complex than expected, as we were also enhancing the backend, and it brought with it new issues.

Our project underwent major structural and functional changes in the second semester. To better link each quiz to the related course, we rearchitected the system and the database. This required revising all API routes, rewriting the database schema, and including the quiz interface as part of the course view rather than putting it on a separate page. We also redesigned the task user interface entirely. We streamlined the general design to improve usability and aesthetics, provided a persistent navigation sidebar, and made navigation buttons more visually accentuated based on user feedback. To better match the educational intent of the items, we also renamed ”tasks” to ”activities.” A few weeks out from the final deadline, the interface for the content creator was one of the major last-minute features. We needed to implement a new feature for the addition, allowing users to create and manage their courses. We also implemented a course discovery webpage to allow students to decide on the courses in which to enroll. Upon analyzing the usage and expense of the use of LLMs to create personalized quizzes, we determined that the approach was not scalable and not viable. Rather, we decided to have subject experts create quizzes manually to control for quality and congruence with the course material.

B. Teamwork and Collaboration

The team was divided across the project into three specialisations: backend development, frontend development and design and artificial intelligence model development. Each team member was delegated the major responsibility over one of these specialisations with the understanding that all the team members would work and support one another across various areas to ensure the success of the project.

Issues were tackled as soon as they arose throughout the project. When issues arose — was it technical issues with the performance of the AI models or difficulty merging external pieces — the team collectively brainstormed solutions. Great communication ensured the timely recognition of the issues, and debugging moved into the limelight. Bi-weekly meetings were planned for our advisers to report on the outcomes of the sprints, identify any challenges, and set the task priorities for the upcoming sprint. With each sprint focusing on specific deliverables, the Agile methodology facilitated continuous improvement.

The team successfully executed the project through the cooperative style of operation and the clear identification of leadership roles. This enabled it to tackle issues as they arose and continuously improve the product using ongoing feedback and testing.

VI. Project Evaluation

We assessed the solution of our project in a number of ways to check if the platform achieved the objectives outlined in the introduction and to what extent it solved the issue of teaching Kazakh-Russian Sign Language.

A. Lighthouse Performance Analysis

Analysis of the platform’s web pages via Lighthouse tool showed generally excellent performance in various areas, with some areas identified for improvement. As demonstrated in Table I, the *Content Creator Dashboard* appeared to have the best score in performance (95) and best practices (100), meaning it is optimally designed and modern development standards. Similarly, pages such as *Create New Course, My Courses, New Courses, and Take Quiz* all achieved consistently high best practices scores of 100 and high-level performance above 70, indicating that these pages were efficiently designed and have a seamless user experience. However, the *Lesson* page showed the lowest performance score (62), possibly due to the use of more elaborate or heavier resources like embedded media or animations that might delay load and interactivity. For accessibility, scores varied from 69 to 79, with the *Take Quiz* page scoring the lowest. This indicates that some interactive elements may lack proper labeling or semantic structure, potentially limiting usability for users with disabilities. SEO scores were consistent at 91 on all the pages, indicating strict compliance with search engine optimization practices. Generally, results illustrate a high-performing and accessible platform with specific areas of improvement targeted in some of the pages in order to have a uniformly high-quality user experience.

TABLE I
Lighthouse Scores Summary per Page

Page	Performance	Accessibility	Best Practices	SEO
Content Creator Dashboard	95	77	100	91
Create New Course	83	78	100	91
Exercise	76	78	78	91
Lesson	62	70	74	91
My Courses	73	73	100	91
New Courses	73	79	100	91
Review	78	72	78	91
Take Quiz	84	69	100	91

B. Live User Feedback Collection

Four students, two teachers, two Deaf community members, one native sign language user, and one sign language interpreter participated in two rounds of user testing to ensure our platform satisfied its intended users’ needs. We aimed at evaluating the usability, accessibility, and clarity of the content for the platform’s hearing and non-hearing users. Participants in the in-person test were asked to perform specific exercises such as accessing quizzes, completing exercises and navigating the lesson plans. We used direct feedback, observation, and short questionnaires to obtain data.

We used the first round to pilot the initial version of the platform and received constructive feedback. As participants found the navigation buttons unclear and the flow of activities unintuitive, they branded the lesson page as not intuitive. For facilitating easier tracking of their progress, users also requested a list of all the words utilized in each of the courses. Substantial modifications were then made based on the feedback, such as redrawing the interface of the lesson, the addition of visually appealing navigation sidebar, the clearer navigation buttons and labels, and the provision of a word list for each specific course.

The revised solution was received positively by users in the second round of testing. They complimented the capability to see all course vocabulary in one place, noted easier navigation and commented that the lesson organization was straightforward. Deaf community participants especially complimented the intuitiveness of the interface and indicated they were more at ease navigating the site on their own. We were able to create a more accessible, user-centered, and inclusive learning space due to the iterative testing process.

VII. Conclusion and Future work

The AI-YM platform successfully fills the need for accessible, scalable, and easily accessible resources for learning sign language, particularly for Kazakh-Russian Sign Language (K-RSL) users. The project has achieved significant success in bridging the communication barrier between hearing and non-hearing individuals by employing AI-driven gesture recognition to deliver feedback to the users regarding signs made. With the addition of user-led course creation, the platform is now a dynamic user-centered learning space.

The primary conclusions of the evaluation showed the high performance of the platform in content delivery, accessibility, and usability. Based on the Lighthouse performance audit, the platform adheres to current development best practices, ensuring the safe and efficient user experience. The platform was redesigned iteratively to provide a more inclusive and user-friendly experience based on the feedback from the deaf and hearing participants of live user testing.

Although the existing version of the platform has been successful in reaching its primary goals, there are a number of areas where it can improve and grow in the future. Most importantly, with the development of the system and addition of increasingly varied data of sign language, it will be critical to enhance the gesture recognition algorithm to handle continuous signing and enhance accuracy. Next-generation research might also include advanced deep learning models to further enhance the recognition of signs and to support a broader vocabulary of signs and phrases.

Adding to the platform's capability to include new course content, focused for Deaf community, like coding, traffic violation laws, or other useful life skills, delivered by deaf instructors and for the deaf, is another promising direction promoted by our advisors. Additional gamification aspects to more strongly motivate users, maybe in the form of challenges or multi-player collaboration, friend streaks and tournaments, can also increase retention and interaction.

Finally, the scalability of the platform will be a key area to improve. When the user base grows, the system needs to be able to handle increasing traffic and user-contributed data without compromising the user experience.

References

[1] D. Bragg, T. Verhoef, C. Vogler, M. Morris, O. Koller, M. Bellard, L. Berke, P. Boudreault, A. Braffort, N. Caselli, M. Huenerfauth, and H. Kacorri, "Sign language recognition, generation, and translation: An interdisciplinary perspective," pp. 16–31, 10 2019.

[2] A. Imashev, N. Oralbayeva, and A. Sandygulova, "An exploratory user study towards developing a unified, comprehensive assessment apparatus for deaf signers, specifically tailored for signing avatars evaluation: challenges, findings, and recommendations," *Multimedia Tools and Applications*, pp. 1–38, 10 2024.

[3] M. Mukushev, A. Ubingazhibov, A. Kydyrbekova, A. Imashev, V. Kimmelman, and A. Sandygulova, "Fluentsigners-50: A signer independent benchmark dataset for sign language processing," *PLOS ONE*, vol. 17, p. e0273649, 09 2022.

[4] The Chinese University of Hong Kong, "Meet signtown handbook: Ai-powered online sign language learning platform," *CUHK Faculty of Arts News*, 11 2024. Available: https://www.arts.cuhk.edu.hk/web/news-and-events/meet-signtown-handbook-ai-powered-online-sign-language-learning-platform.

[5] V. Falvo, L. P. Scatalon, and E. Francine Barbosa, "The role of technology to teaching and learning sign languages: A systematic mapping," in *2020 IEEE Frontiers in Education Conference (FIE)*, pp. 1–9, 2020.