Nazarbayev University

Computer Science Department
Final Report – Spring 2023-2024

Title of the project:	Qazaq Sign Language Dictionary
Team Members:	Batyrbek Orazumbekov Damir Daukarayev Mukhtar Sarsenbay Yelnur Umbetay
Project Advisor/Co-Advisors	Anara Sandygulova Antonio Cerone

Executive Summary (10%)

Our project aims to deliver a comprehensive digital platform focused on learning and using sign language to revolutionize access for the Deaf community. This platform ensures inclusiveness and improved communication by integrating multiple resources to address different learning styles and needs. The main feature of our platform is the Sign Language Dictionary, which includes several categories. These categories cover various aspects of daily life such as greetings, emotions, nutrition, activities and more. Each category is a collection of carefully selected sign language videos, each containing a particular hand or word associated with that category. These videos are designed to be informative, engaging, and accessible, allowing users to learn sign language and learn at their own pace. The Sign Language Dictionary also has a search function that allows users to search for specific words or specific gestures they want to see or find quickly. This feature enhances the usability and efficiency of the platform by allowing users to easily navigate through a large database of sign language content. In addition, our platform includes user-generated content through licensed downloads. This feature allows qualified professionals, such as certified sign language instructors to upload videos to the platform. These videos go through a process of verification and quality, enhance the content of the platform, and enhance the user's learning experience. In addition to the dictionary and custom content, our platform offers a reverse dictionary feature. Our platform allows you to record hand gestures and use them to search for words in a large database of words. We achieved this using a model through which we conducted video clips and output them for all possible predictions and saved them in the database

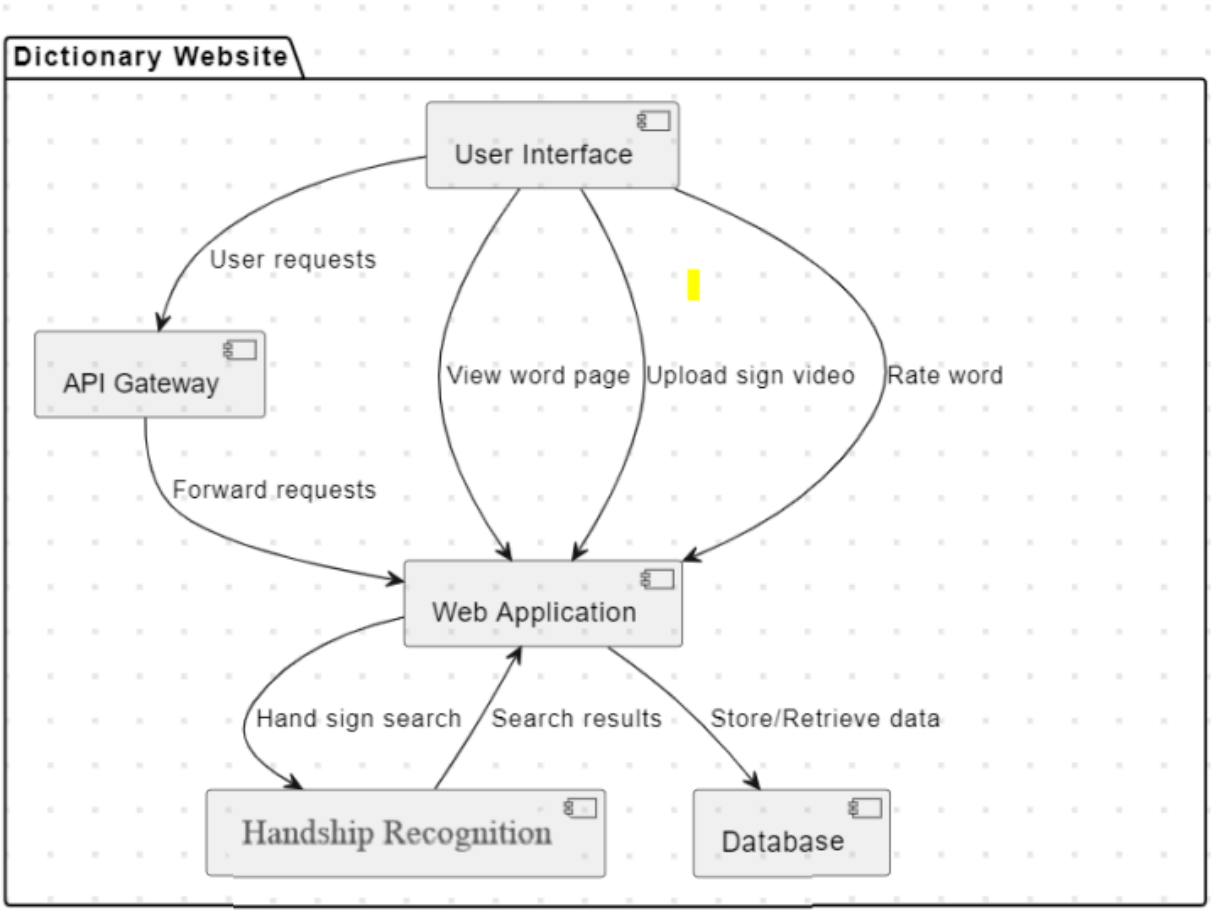
Nazarbayev University

Introduction (10%)
In a world where communication is vital, it's crucial for everyone to have equal chances, no matter their abilities. Hearing impairment is a challenge hindering free communication. Addressing the needs of people with hearing difficulties is a significant concern. Sadly, there's currently no digital resource specifically aiding in learning sign language, a gap also felt in our country. The absence of a necessary tool for socially vulnerable citizens motivates our project. We aspire to join a community working towards supporting the deaf community in Kazakhstan, aiming to fill this gap and contribute positively to the lives of those facing hearing challenges.
Background and Related Work (15%)
In the field of digital resources for learning sign language, the need for inclusive platforms focused specifically on the Kazakh language is becoming increasingly obvious. People need not only easy-to-use, but also affordable tools for learning Kazakh sign language. Let's look at why we need a new website for this purpose. Sign language learning has changed with digital platforms. But students who want to learn Kazakh sign language are missing something. Although there are some common platforms, none of them actually take into account the unique way Kazakh sign language works. This made us think about creating a website specifically designed for people who want to learn sign language in the Kazakh cultural context. Other sites dedicated to sign language usually focus on large, widely spoken languages. They often forget about languages such as Kazakh, which are not so common all over the world. Common platforms may not be good enough for learning Kazakh sign language. To correct this, we need to understand how the Kazakh sign language differs, both linguistically and culturally. As a result, we found that there is a need for a special website for learning Kazakh sign language.

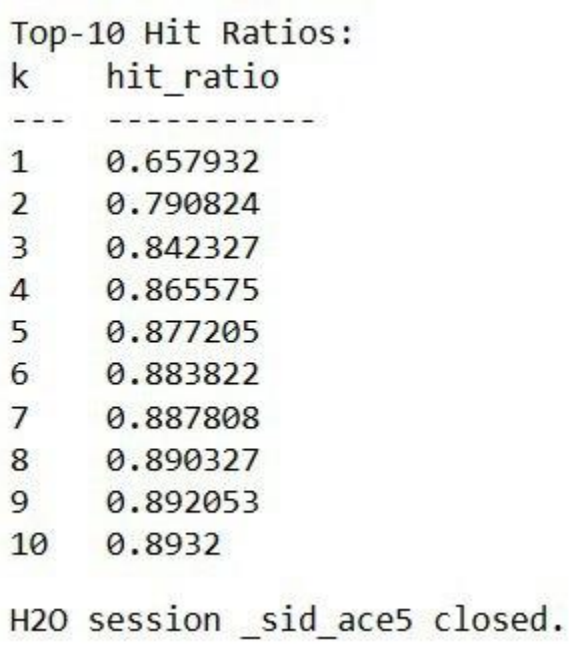
Other websites don't really pay attention to the unique parts of Kazakh sign language. With the right approach, our website can become a fascinating and easy way for people to learn Kazakh sign language, glorifying the beauty of its culture and language.

Project Approach (20%)

This architectural model illustrates various components including User Registration, Video Streaming, Content Upload, and more, each responsible for specific functions. These components work together to create a seamless user experience, allowing users to register, watch videos, upload content. Also, a dedicated engine can determine the gesture(sign analyze).



For the reverse dictionary, we had to create a model using the dataset provided to us by our advisers. Initially, we tried to use different automated machine learning libraries such as h2o. However, in the end, we ended up choosing sci-kit learn's Random Forest algorithm, due to the fact that it showed better accuracy results and was relatively fast for the real time hand gesture predictions. The results of the testing part showed accuracy of about 78%. Unfortunately, in the real time case, the model works not work as good as we planned. In general, the dataset includes 42 features indicating different parts of the hand. Regarding the structure of the dataset, it consists of 83 gesture classes and over 4 million samples. The main problem is that some classes include over 500.000 samples while some didn't even reach 100 samples showing unbalancedness of the dataset. That's the reason for the unsatisfactory performance of the model. For real time hand gesture capturing we used Google's MediaPipe library which allows us to get 42 key features from each frame. In the website, user's camera captures each frame and then the model predicts the hand gesture based on the key features from the mediapipe get 42 key features from each frame. In the website, user's camera captures each frame and then the model predicts the hand gesture based on the key features from the mediapipe.



```
Top-10 Hit Ratios:
k    hit_ratio
---  -----------
1    0.657932
2    0.790824
3    0.842327
4    0.865575
5    0.877205
6    0.883822
7    0.887808
8    0.890327
9    0.892053
10   0.8932

H2O session _sid_ace5 closed.
```

img1(h2o testing performance)

Here from the image 1 we can see that the accuracy is equal to the 65.7 percent due to the fact that it is random forest algorithm it chooses from the first attempt.

```python
#h2o

import pandas as pd
import pickle
from glob import glob
from sklearn.model_selection import train_test_split
from h2o.automl import H2OAutoML
import h2o

# Start an H2O cluster with increased memory
h2o.init(max_mem_size="16G")  # Example: set maximum memory to 16 GB

csv_dir = "./"
find_pattern = "*.parquet"
files = glob(f"{csv_dir}{find_pattern}")
print(files)

df_all = pd.DataFrame()

for file in files:
    df_c = pd.read_parquet(file)
    df_all = pd.concat([df_all, df_c])

X_all = df_all.loc[:, 'x1':'y21']
y_all = df_all['class']

pkl_filename = "6mil_18h_300prtl_120gb_nolda_nopreproc_h2o_automl.pkl"

X_train, X_test, y_train, y_test = train_test_split(X_all, y_all, random_state=1)

# Convert pandas DataFrame to H2O Frame
train = pd.concat([X_train, y_train], axis=1)
test = pd.concat([X_test, y_test], axis=1)
h2o_train = h2o.H2OFrame(train)
h2o_test = h2o.H2OFrame(test)

# Identify predictors and response
x = h2o_train.columns[:-1]
y = 'class'

# Run AutoML
aml = H2OAutoML(max_runtime_secs=3600)  # 1 hour
aml.train(x=x, y=y, training_frame=h2o_train)

# Print leaderboard
lb = aml.leaderboard
print(lb)

# Get model performance
perf = aml.leader.model_performance(h2o_test)
print(perf)

# Save the model
h2o.save_model(aml.leader, path=pkl_filename, force=True)
```

image 2(h2o creating model)

```
: #Random forest ALL FILES at ONCE

import os
import pandas as pd
import pickle
from glob import glob
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

csv_dir = "./"
find_pattern = "*.parquet"
files = sorted(glob(f"{csv_dir}{find_pattern}"))  # Sort files alphabetically

# Initialize lists to store data from all files
X_all_list = []
y_all_list = []

# Loop through files to collect data
for file in files:
    df_c = pd.read_parquet(file)
    X_all_list.append(df_c.loc[:, 'x1':'y21'])
    y_all_list.append(df_c['class'])

# Concatenate data from all files
X_all = pd.concat(X_all_list)
y_all = pd.concat(y_all_list)

# Split the combined dataset into train and test sets
#X_train, X_test, y_train, y_test = train_test_split(X_all, y_all, random_state=1)
X_train, X_test, y_train, y_test = train_test_split(X_all, y_all, test_size=0.5, random_state=1)

# Initialize and train the Random Forest model
rf_classifier = RandomForestClassifier(n_estimators=40, random_state=42)
rf_classifier.fit(X_train, y_train)

# Save the trained model using pickle
pkl_filename = "random_forest_combined_data.pkl"
with open(pkl_filename, 'wb') as f:
    pickle.dump(rf_classifier, f)

print("Random Forest model trained on all files and saved successfully!")
```

```
Random Forest model trained on all files and saved successfully!
```

image 3(random forest creating mod random forest creating mod)

```
In [6]: # Load the trained model

        pkl_filename = "random_forest_combined_data_20_estimates.pkl"

        with open(pkl_filename, 'rb') as f:
            rf_classifier = pickle.load(f)

        # Make predictions on the test dataset
        y_pred = rf_classifier.predict(X_test)

        # Evaluate the model
        from sklearn.metrics import classification_report, accuracy_score

        print("Classification Report:")
        print(classification_report(y_test, y_pred))

        print("Accuracy Score:", accuracy_score(y_test, y_pred))
```

```
Classification Report:
              precision    recall  f1-score   support

          pa       0.77      0.49      0.60      2221
        pa23       0.76      0.39      0.51       171
          pk       0.85      0.44      0.58        25
       pka23       0.81      0.72      0.76      1709
          pl       0.85      0.76      0.81     33556
         pla       0.79      0.36      0.50       488
       pla23       0.89      0.48      0.62       459
           q       0.76      0.65      0.70     33321
          qa       0.81      0.57      0.67      2842
         qka       0.72      0.56      0.63      4798
         qla       0.80      0.45      0.58       710
           r       0.76      0.22      0.34        87
          rk       1.00      0.03      0.05        37

    accuracy                           0.79   2342938
   macro avg       0.77      0.56      0.63   2342938
weighted avg       0.79      0.79      0.79   2342938

Accuracy Score: 0.7887562539000178
```

image 4 (random forest model evaluation accuracy)

Project Execution (15%)

During the first meeting with the advisors team we discussed the details of the project, visions and possible implementation. Having already formed a solid base in the form of image processing algorithms and training samples - it was crucial to implement the existing work into a feasible product. It was determined that we would be creating an online dictionary. The dictionary would consist of a word catalog divided into categories, the word page would have its description, interpretation in kazakh sign language. On top of that we would implement a reverse dictionary solution, so that a word could be

found via sign demonstration. The initial step was the analysis of existing projects that could be taken as an example, the project that caught our eye was the ASL STEM online American Sign Language dictionary. After the analysis we understood that our product should have a simple and neat design, for better user experience.

During the first semester, we mostly worked on the website itself. The first stage was outlining the structure of the website and its webpages. The initial structure is outlined below:

- The UI/UX prototype for different pages
  - Main page
  - Category page
  - Word page
- The header and the footer
  - The logo that returns to the main page
  - Search bar
- The main page
  - Button for different category choices (traditions, cuisine, locations, etc)
- The category page
  - Includes all the words corresponding to that page
- The word page
  - Video showing the sign
  - Word description
  - Upload button so people can upload their variants of the sign
  - Like and dislike buttons

Initial design sketches for the website are provided below:

Figure 5. Initial design layout

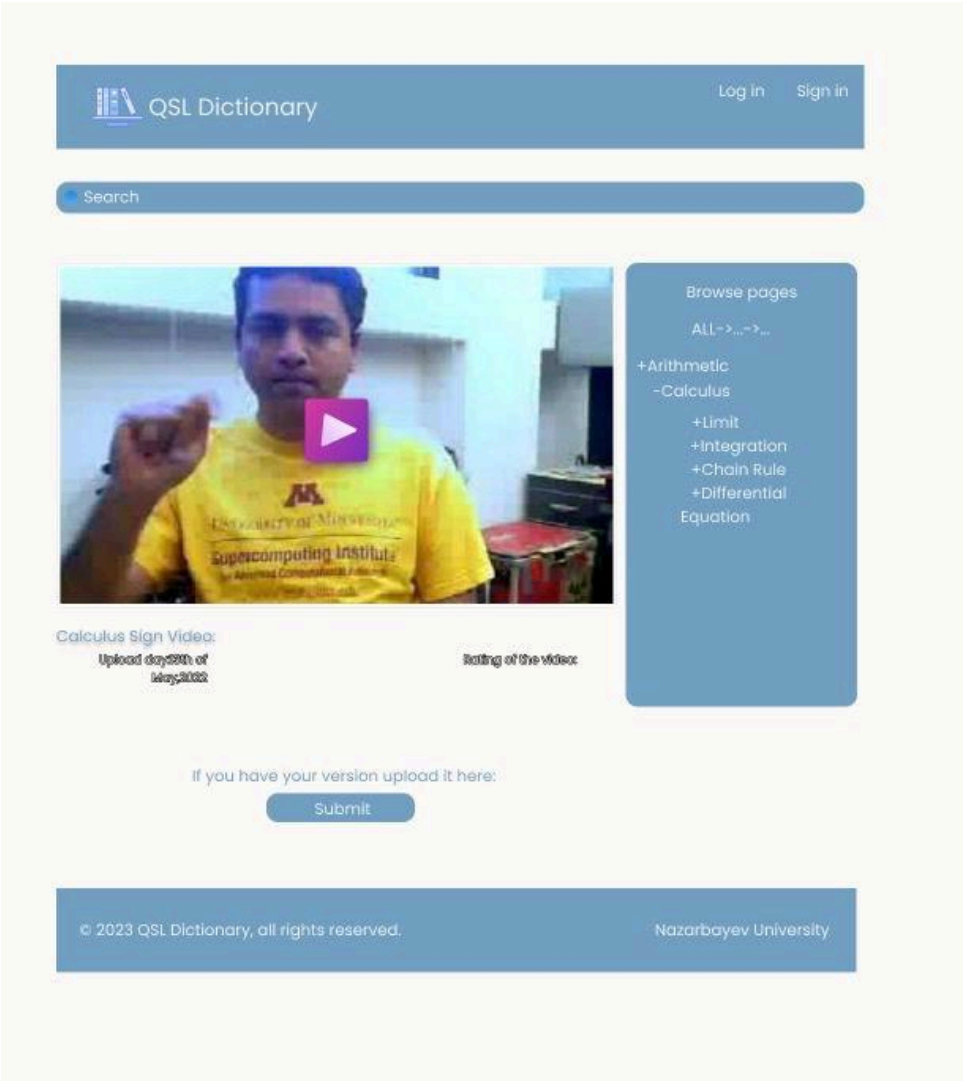


Figure 6. Word page design

The next step was the website development. As we progressed further we adjusted the website structure and design choices. During that process button design, color choice, and container placement were adjusted for better user experience. Whilst working on the website we understood that there should be a restriction on upload feature. What we mean by that is that there should be a limited amount of users that would have a possibility to upload their own interpretations to the words. For that, in the sign up stage, we would separate the admin/super users from regular users, the first category would be

granted the upload possibility. Simultaneously we were filling up the database with the words and their respective interpretation provided by the research team.

During the second semester we were mostly focused on the implementation of the existing ML algorithms for gesture recognition, their improvement, reverse dictionary system , and website outlook adjustments. Random Forest was found to have the greatest accuracy and was the approach taken for further improvement.

Throughout the development of the project we did face numerous difficulties, those include the choice of video uploading server, adding word pictures, providing description for each word, choosing a reliable database, and unbalanced dataset. The issue with an unbalanced dataset lies in the fact that there was a total of 84 classes, where a number of classes had less than 100 samples and a number of classes had more than 100,000 samples. This in turn made the actual performance not as great as expected. At some point. During the early stages of the development we ran into a problem of choosing a reliable server that wouldn't cost a lot for the team and be able to store all of the needed data (video material), so we chose Amazon's S3 cloud storage. Having dozens upon dozens of words present on the website, we had to provide visual material for each word, having no way of creating every single image on our own, we used pixabay's API for free-for-use images to be hosted on our website. Alongside with the visual representation, we needed word description for each single word as well, so in order to resolve the issue we used Wikipedia's API for description generation.

Evaluation (20%)

Whilst surfing the web for any online kazakh sign language learning platforms, we came to the conclusion that there were none. Considering the fact that a respectable research team consisting of certified linguists, experienced developers and Doctorate degree students took part in the development of the project, we can definitely say that the QSL web platform would be able to contribute to the solution of previously mentioned problems. The QSL platform would provide all of the necessary means for people to learn and practice their kazakh sign language knowledge.

In order to test the usability of the product, we went to the center that helps people with disabilities. We had 4 testers that knew the kazakh sign language and had hearing difficulties. Concluding the functionality of the product, everyone was satisfied with the reverse dictionary functionality, search function implementation, and description provided with each word. This especially hints that we were able to reach our main goal. Whilst testers were able to retrieve the words that they were signing, it is clear that the reverse dictionary needs further improvement for better gesture recognition and greater accuracy, which currently is at 78.8% (figure 4). The product needs a lot of design rework for better user experience. Testers, especially those who don't have much of an experience in working with computers, didn't find it as easy to navigate through the website.

Conclusion and Possible Future Work (5%)

In order to advance further with the product, we need to improve the hand gesture recognition and work on the website layout. The testers didn't find it as easy to navigate through the website as it was expected. The search field and language toggling have to stand out more. The best solution for this issue could be hiring a professional, a UX/UI expert in order to improve user experience. On top of that, for better sign learning rate, we have to provide supportive videos that would demonstrate the usa cases of signs in sentences or phrases.

To resolve the issue with gesture recognition, we need to balance out the dataset. As it was mentioned previously, we had classes containing hundreds of thousands of data samples, and we had classes containing less than a hundred data samples. To improve accuracy of recognition, we have to provide more data samples for better learning.

References (5%)

Annotation tool. Annotation Tool. (n.d.). https://slan-tool.com/

A signer-independent benchmark dataset for Sign Language Processing. FluentSigners-50. (n.d.). https://krslproject.github.io/FluentSigners-50/

ASL wiki - https://danibragg.com/papers/ASL_Wiki.pdf

Cambridge, D. B. M. R. (n.d.). *Sign Language Recognition, Generation, and Translation | Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*. ACM Conferences. https://dl.acm.org/doi/10.1145/3308561.3353774

Designing an Online Infrastructure for Collecting AI Data From People With Disabilities - https://www.microsoft.com/en-us/research/uploads/prod/2021/01/Inclusive_AI_Datasets_FINAL.pdf

D1.40 ReportonEurope'sSign languages - european language equality. (n.d.). https://european-language-equality.eu/wp-content/uploads/2023/06/ELE__Deliverable_D1_40__Europe_s_Sign_Languages_.pdf

Evaluation of manual and non-manual components for Sign language recognition. KRSL20. (n.d.). https://krslproject.github.io/krsl20/

Glasser, A., Minakov, F., & Bragg, D. (2022, October 22). ASL Wiki: An Exploratory Interface for Crowdsourcing ASL Translations. *The 24th International ACM SIGACCESS Conference on Computers and Accessibility*. https://doi.org/10.1145/3517428.3544827

Hugging face – the AI community building the future. Hugging Face – The AI community building the future. (n.d.). https://huggingface.co/datasets/masya420/all_dataset

Report on Europe's Sign Languages - https://european-language-equality.eu/wp-content/uploads/2023/06/ELE__Deliverable_D1_40__Europe_s_Sign_Languages_.pdf

Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective - https://dl.acm.org/doi/10.1145/3308561.3353774

Sign2MINT. (n.d.). https://sign2mint.de/