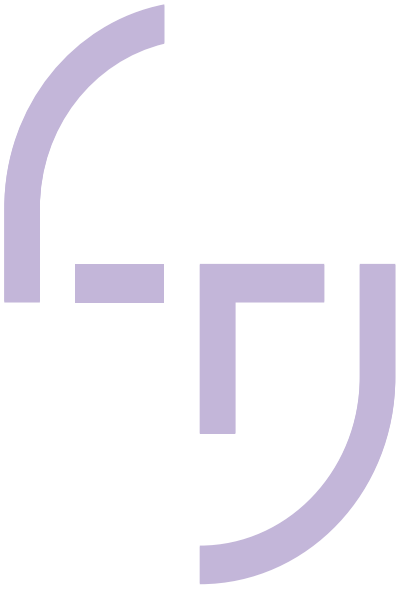
Tampere University of Applied Sciences



Online Language Self-Study Platform

Toan Tan

BACHELOR'S THESIS
May 2023

Degree Programme in Software Engineering

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Bachelor's Degree Programme in Software Engineering

TAN, TOAN
Online Language Self-Study Platform

Bachelor's thesis 54 pages
May 2023

With the globalization and development of information technology, knowing and being fluent in other languages is one of the essential requirements, which will open up many opportunities and advantages in studying and working. Therefore, the need of learning foreign languages has become more and more popular, in which self-studying and practicing are the important factors leading to success in learning any foreign language.

This thesis aimed to build an online platform that makes language self-studying easier and more convenient. The platform is not intended to replace traditional language-learning methods; rather, it is designed as a supplement to make language self-studying less monotonous and encourage regular practice.

The thesis researched the effectiveness of different language-learning methods from scientific articles and published documents. The implementation of the thesis applied full-stack technologies and cloud services from AWS to build an online language self-study platform. The product of this thesis is a platform that has basic features such as learning languages by watching videos and listening to music. The platform also enables practicing vocabulary with flashcards and gamification modes. In addition, it includes built-in tools to support translation and pronunciation of words and sentences.

Key words: language learning, full-stack, serverless architecture, AWS, ReactJS

TABLE OF CONTENTS

ABBREVIATIONS

API	Application Programming Interface
AWS	Amazon Web Services
CDK	Cloud Development Kit
CI/CD	Continuous Integration and Continuous Delivery
CSS	Cascading Style Sheets
CSV	Comma Separated Values
DOM	Document Object Model
ERD	Entity Relationship Diagram
GCP	Google Cloud Platform
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
JSX / TSX	JavaScript / TypeScript syntax extension (used in React)
JWT	JSON Web Token
Kedu	Name of the Online language self-study platform
MUI	Material UI
n.d.	no date
ORM	Object Relational Mapping
RDS	Relational Database Service
S3	Simple Storage Service
SNS	Simple Notification Service
SQS	Simple Queue Service
UI	User Interface
UX	User Experience

1 INTRODUCTION

1.1 Idea and purpose of the online language self-study platform

With the globalization and the rapid development of information technology, the ability to use foreign languages has become a necessity for gaining access to information, facilitating learning, and achieving success in the workplace and in daily life.

In the past, when language courses and resources were limited, it was much more difficult to acquire foreign languages. Today, thanks to the development of information technology, people have more opportunities to access language-learning materials. Utilizing available tools, modern technologies, and applications to enhance language learning is progressively becoming more widespread.

Numerous studies have been conducted to investigate and evaluate the effectiveness of different strategies and methods for learning foreign languages. Recent research by Lin Phoebe (2019) about the study habits of students at Hong Kong University revealed that 75% out of 300 English learners responded that they watched videos, television programs, and movies in English to improve their foreign language skills. Moreover, 10% of respondents said they spent at least 5 hours per week on these activities. This result indicated that teenagers tended to use technology applications in their language learning process, particularly for watching videos and listening to music in the target language that they were studying.

Additionally, learning vocabulary is also a crucial factor in learning and mastering foreign languages. Using flashcards is one of the most effective methods to improve vocabulary acquisition. Komachali and Khodareza (2012) conducted research showing that the use of word cards or flashcards, regardless of paper or digital format, can significantly improve the vocabulary acquisition rate of learners when compared to traditional vocabulary learning strategies. In another study (Spiri 2008), Japanese university students were compared on their ability to learn vocabulary using digital flashcards and physical word lists. The findings of the study showed that students who utilized digital flashcards outperformed those who relied on physical word lists.

Based on the results from the studies above, it can be seen that watching videos and listening to music in the target learning language is one of the most effective methods to enhance foreign language proficiency. It helps learners develop their listening and comprehension skills as well as expose them to real-life language usage. Furthermore, using flashcards has also been proven that it is an interactive and efficient way to memorize and review vocabulary.

The objective of this thesis was to apply full-stack technologies and cloud services from AWS to build a platform that combines these excellent language learning methods in one place. This combination will make learning new foreign languages less monotonous, more enjoyable, and more personalized to the learner's preferences.

1.2 Project information and goals

The online language self-study platform is named Kedu. Kedu is intended to be available on multiple platforms, such as a website, an iOS application, or an Android application, to make it easy for users to access and use. This thesis will focus on the process of researching and building Kedu as a web platform.

The primary goal of the Kedu in this thesis is to create a website that facilitates self-study, practice, and improvement of language skills easily and conveniently. In addition, the user interface (UI) and user experience (UX) are also important targets to be achieved. Finally, features have to work precisely and swiftly, the entire website has to be easy to deploy and manage on the cloud platform with reasonable operating costs.

2 BACKGROUND

This chapter will go through the theoretical background, advantages, and disadvantages of several key technologies that will be applied to the web version of Kedu. Details on how to apply and operate will be discussed and presented in the following chapter.

2.1 ReactJS

ReactJS is one of the most popular JavaScript libraries used to develop user interfaces (UI) for modern web applications. It was developed in 2011 by Jordan Walke and continues to be maintained by Meta to this day. After many upgrades and improvements in many aspects (syntax, performance, etc.), ReactJS today has become one of the most loved and widely used libraries, with more than 40% of users employing it in learning as well as professional jobs, according to a survey (Stackoverflow 2022). Some of the main characteristics and benefits of ReactJS (SolGuruz 2022) can be mentioned as follows:

- Easy to learn and use: by using JSX (which has a syntax similar to HTML) and not too many complex concepts, developers can easily learn and use ReactJS.
- Virtual DOM: ReactJS uses Virtual Document Object Model (Virtual DOM) to enhance performance compared to traditional DOM manipulation.
- Component-based architecture: allows breaking down elements and parts of a website into multiple components (files) to help projects in reducing complexity, making them simple to maintain, and increasing reusability.
- Open source: allows everyone in the world to access, view, and contribute to the source code, thereby improving library stability, quality, and transparency, as well as reducing security risks.

2.2 Tailwind CSS

Tailwind CSS is a widely used utility-first CSS framework developed by Adam Wathan, Jonathan Reinink, Steve Schoger, and David Hemphill in 2017 with the purpose of helping the development of a website's user interface (UI) quickly and conveniently.

Unlike other CSS frameworks such as MUI and Ant Design, Tailwind CSS does not include pre-built components (button, form, input, grid, etc.). Instead, its

concept and operation are quite similar to the Bootstrap CSS framework by providing built-in classes for users to style nearly all element properties such as font color, background color, margin, padding, flex, grid, and border, among others.

style.css U

src > components > TestComponent > style.css > .element

```css
.element {
  display: flex;
  justify-content: space-between;
  align-items: center;
  gap: 0.25rem;
}
```

index.tsx U

src > components > TestComponent > index.tsx > ...

```tsx
import './style.css'

export function SampleComponent() {
  return (
    <div>
      <div className='element'>
        Hello world
      </div>
    </div>
  )
}
```

FIGURE 1: Example of styling HTML elements in a traditional way

index.tsx U

src > components > TestComponent > index.tsx > ...

```tsx
import './style.css'

export function SampleComponent() {
  return (
    <div>
      <div className='flex justify-between items-center gap-4'>
        Hello world
      </div>
    </div>
  )
}
```

FIGURE 2: Example of styling HTML elements by using Tailwind CSS's classes

When using TailwindCSS, users only need to use built-in classes directly in HTML, JSX, or TSX files (Figure 2) to style the website's elements without having to create and name classes, then define those styles' content in separate CSS files like traditional ways (Figure 1). This makes development faster and reduces the risk of class name duplication as the project grows (class names are accidentally reused and redefined across multiple locations).

2.3 JSON Web Token

JSON Web Token (JWT) is an open standard RFC 7519 (M. Jones - Microsoft, N. Sakimura - NRI, J. Bradley Ping - Identity 2015) for transmitting information between parties securely. Nowadays, it is commonly used for authorization and authentication between client and server.

JWT has three parts separated by dots, including the header, payload, and signature, which can be seen in Figure 3 below. Normally, the header contains information about the algorithm used to sign the token and its type. The payload consists of the claims or data to be transmitted, such as user IDs, names, email addresses, roles, permissions, etc. The signature is generated by encrypting the header and payload with a secret key, ensuring that the data is not modified during transmission.

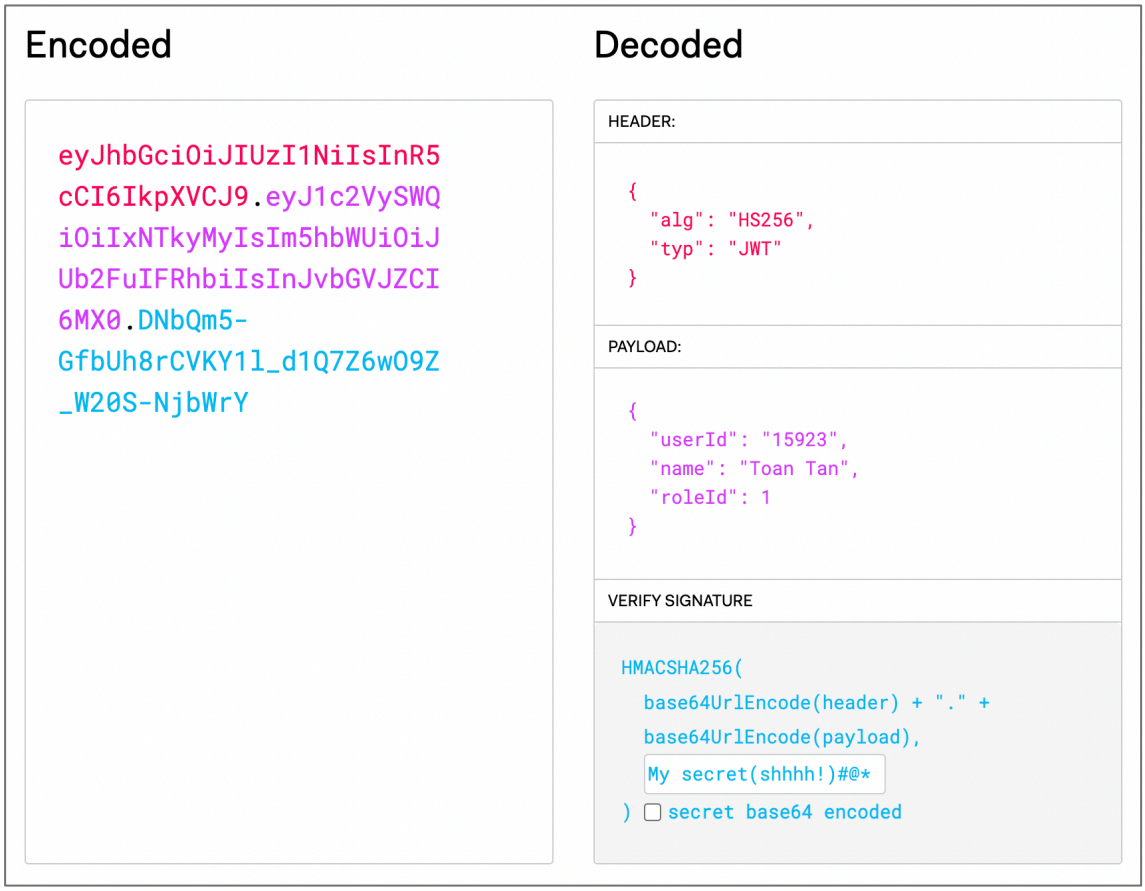


FIGURE 3: Example of encoded and decoded JWT

In practice, JWT is often used in stateless authentication systems, where a user logs in once and receives a JWT token, which can then be attached to subsequent requests (usually in the header of the request) to verify their identity and access without requiring continuous re-authentication. This can improve performance and reduce the load on the authentication server.

2.4 Serverless

2.4.1 Serverless architecture

Serverless architecture is a modern approach for developing and operating cloud computing systems in which the cloud provider administers the allocation and provisioning of required services and resources. This allows users to focus on developing their projects, code, and business logic while reducing the burden of infrastructure administration and complicated configurations.

It is also essential to note that the concept of Serverless or Hostless does not imply that everything runs without a server. The code of users will still be executed on a specific server, but it is fully managed, operated, and provided randomly by cloud providers (Hexaware Technologies Limited 2020).

2.4.2 Advantages

Serverless architecture has many notable characteristics and advantages that can be mentioned:

- Hostless: users do not need to build and manage their own infrastructure. Instead, cloud providers will provide all of them, enabling users to concentrate completely on developing their code and logic.
- High availability: serverless applications are normally deployed on multiple cloud providers' servers, allowing access requests to be handled in a distributed and redundant manner to ensure high availability and reliability.
- Scalability: serverless services can automatically scale up or down in response to changes in access and workload to optimize performance and costs.
- Cost: the majority of services offered under the serverless model permit billing based on the number of requests and used times, which increases convenience and flexibility and lets users pay only for resources they actually use.

2.4.3 Disadvantages

Serverless architecture also comes with some disadvantages that need to be considered when applied to projects:

- Complexity: building, developing, and debugging serverless applications are typically more difficult than traditional architects.
- Customization: users are often required to follow the rules and limitations of services set by cloud providers, making customization almost impossible because they do not have access to the infrastructure.
- Security: because everything is managed and operated by cloud providers, in certain circumstances, businesses or organizations cannot implement a serverless architecture for their projects due to data and security regulations and compliance.

2.4.4 Serverless service providers

There are many cloud providers that offered Serverless services, some of the famous services can be mentioned:

- AWS: AWS Lambda, Amazon S3, Amazon SQS/SNS, Amazon API Gateway, and Amazon DynamoDB.
- Microsoft Azure: Azure API Management, Azure Functions, Azure DevOps, and Azure Cosmos DB.
- GCP: Cloud Run, Cloud Functions, Pub/Sub, Eventarc, Cloud Datastore, and Cloud Storage.

2.4.5 Serverless Framework

Serverless Framework is an open-source JS framework that helps developers in building, managing, and deploying serverless applications to popular cloud platforms such as AWS, Azure, and Google Cloud Platform, among others (Serverless n.d.).

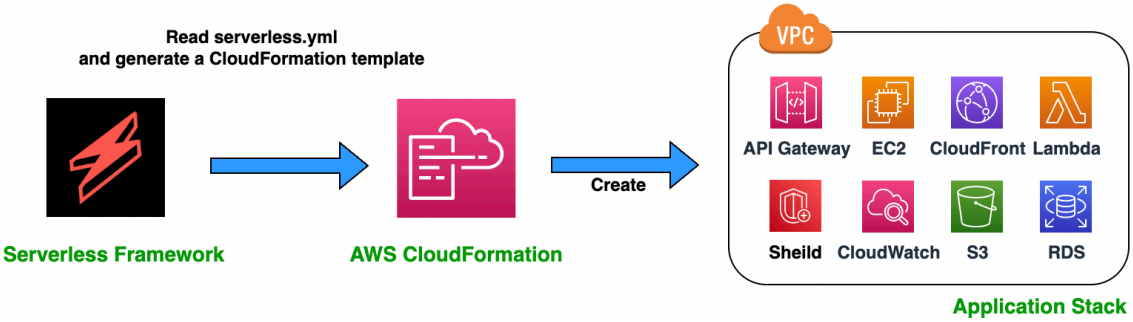


FIGURE 4: How the Serverless Framework manages and creates the necessary resources and services in AWS

Serverless Framework allows users to declare and setup all necessary resources in the serverless.yml file - a form of infrastructure as code (IaC), and deploy to the cloud environment easily in minutes. In Figure 4, for example, an application using the Serverless Framework is deployed to AWS by creating a template from the description in the serverless.yml file and sending that template to the AWS CloudFormation service to create and manage resources.

2.5 Amazon Web Service (AWS)

This section will introduce AWS and some of the main services applied in building the Kedu web version.

2.5.1 Introduction to AWS

AWS is one of the biggest cloud computing platforms nowadays, was founded in 2006 by Amazon and originally provided IT infrastructure in the form of web services to personal users and businesses (Amazon n.d.). AWS has substantially expanded its fields and products to date, offering more than 200 services for building and operating IT infrastructure and applications, including computing, networking, storage, database, analytics, machine learning, and so on.

There are many benefits that can be seen in Table 1 when using clouds in general and AWS in particular. First of all, users will benefit in terms of cost-effectiveness, by reducing the need for upfront capital investments to build the infrastructure

(on-premise), the cost of operation, maintenance, management, and so on. When using AWS, they will only pay for the resources and the duration of its use. Secondly, agility, all the resources and services that users need will be available and ready to run in a short period of time. The other benefit is mobility, allowing users to use and control all services over the network. Users can be in one place, just need an internet connection to be able to work, this is very convenient and efficient, especially during and after the pandemic. In addition, elasticity is a big advantage when using the cloud, allowing users' resources and services to easily and quickly scale up and scale down automatically. This helps their work go smoothly while optimizing costs. Last but not least, backup and restore are other strengths of using resources in AWS. Users' data and applications will be stored and operated in multiple physical servers located in various parts (availability zones and regions) of the world, which will help reduce the risk of natural disasters, network outages, power outages, etc, resulting in lost or unavailable data and applications.

TABLE 1: Comparations between cloud services and traditional computing

Cloud services	**Traditional computing**
No up-front cost	Too much up-front cost
Pay-as-you-go	Pay all time even when not in use
Workload: Elasticity	Workload: Fixed
Locale: Mobility	Local: Fixed
Good security	Hard security
Disaster recovery: Good	Single point of failure (SOF)
Well Backup & Restore	Limited Backup & Restore
More options	Fixed options

2.5.2 AWS Cloud Formation

AWS CloudFormation is a powerful infrastructure as code (IaC) service that enables users to create and organize a collection (Stack) of AWS resources in a predictable and manageable manner. It provides a method to codify the infrastructure in a template format (using JSON or YAML syntax) that can be version controlled and replicated easily. This IaC approach helps users in saving time, increasing consistency, and reducing errors in resource creation and management.

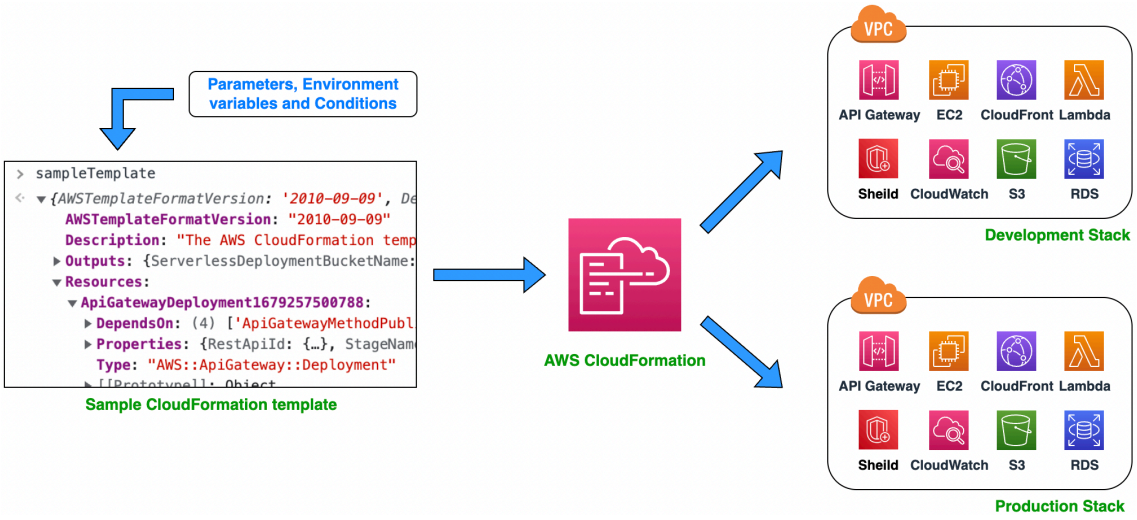


FIGURE 5: How AWS CloudFormation creates and manages AWS resources

In the example depicted in Figure 5, AWS CloudFormation uses a template, which describes resources, to create and manage other AWS services in a stack. Stack is a concept in which resources are centrally managed in a place called stack, allowing for the creation of entire environments with a single click. In addition, when no longer needed, users can simply delete stacks in AWS CloudFormation, and all resources in those stacks will be automatically removed.

2.5.3 AWS API Gateway

AWS API Gateway is a serverless and completely managed service that helps users in developing Application Programming Interfaces (APIs). It provides access points for applications and can communicate with other services on AWS or external services. AWS API Gateway supports a variety of data formats, such as JSON, XML, etc., and allows developers defining security rules and access control to protect their APIs.

HTTP and RestAPI are common use cases in which API Gateway acts as a router or controller, receives requests from clients based on the path, method, headers, body payload, etc., and directs those requests to the corresponding AWS Lambda function for processing (Figure 6).

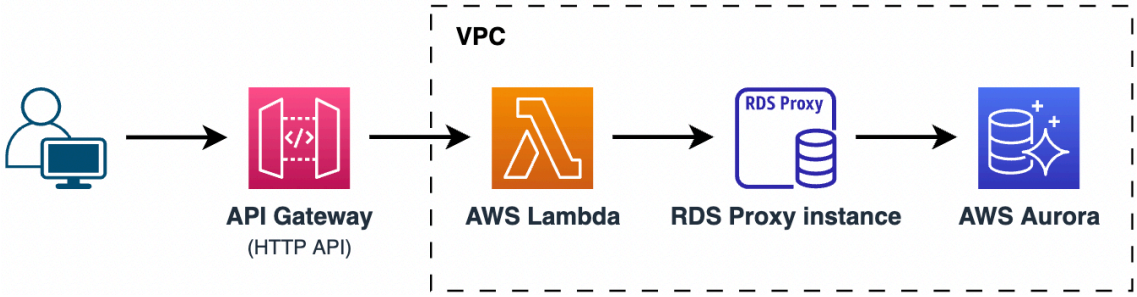


FIGURE 6: Sample flow diagram of an HTTP API built on Serverless architecture

Moreover, AWS API Gateway can be combined with AWS CloudFront to optimize performance, AWS WAF (Web Application Firewall), and AWS Shield to enhance security and protect APIs from cyber attacks (Figure 7).

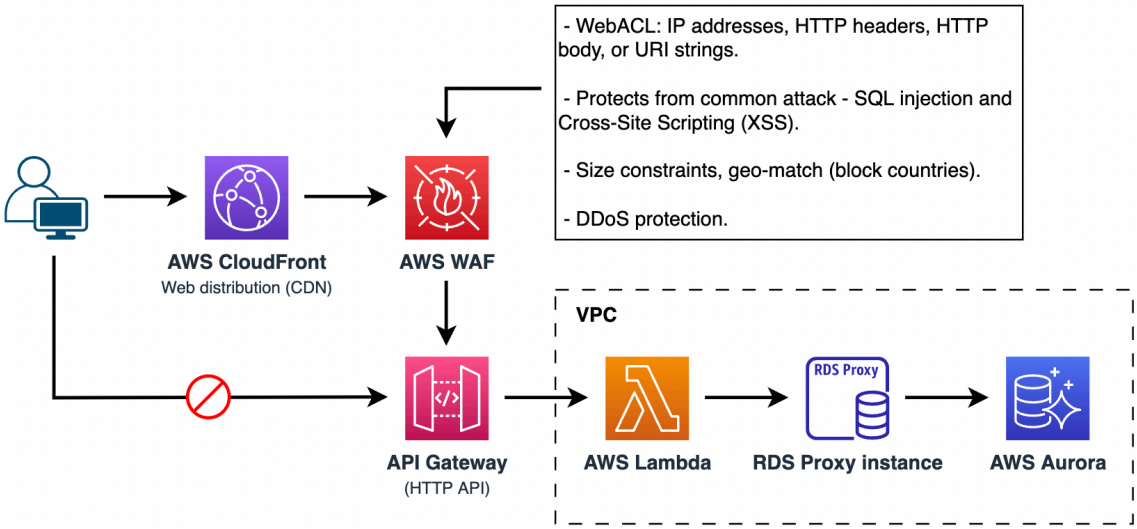


FIGURE 7: Using AWS API Gateway combined with AWS CloudFront and AWS WAF to improve performance and security

2.5.4 AWS Lambda

Lambda is a serverless cloud computing service offered and managed by AWS. It allows developers to run their code without having to manage servers or virtual machines, allowing them to focus on application development without concern for infrastructure management and complex configurations.

Lambda was chosen to execute the code in the backend of Kedu because of some key characteristics and advantages, such as:

- Multiple languages support: AWS Lambda supports many popular programming languages and runtimes, including Node.js, Python, .NET, Java, Go and Ruby.
- Scalability: is the ability to flexibly scale according to user needs. AWS Lambda can automatically scale up or down based on workload (by default, the total concurrency limit is 1000 across all functions in a specific region). In addition, users can easily modify the computing capacity and memory of their functions (from 128 MB to 10240 MB) to optimize performance and cost.
- Integrate easily with other AWS services: Lambda can be combined with other AWS services to create complex and useful applications. For example, it can work with Amazon S3 to process files, use with Amazon DynamoDB for database processing, combine with AWS API Gateway to build RestAPIs, and so on.
- Flexible pricing: Lambda offers a pay-per-run pricing model. Users only pay for the duration and memory used by their functions, eliminating the unexpected cost by unused resources.

2.5.5 Amazon Translate

Amazon Translate is an AWS fully managed translation service that enables users to translate their text or content between languages. By using intelligent automatic translation technologies such as machine learning and neural machine translation, Amazon Translate can provide fast, automatic, and high-quality translations for documents up to millions of pages.

Some of the main features and benefits of Amazon Translate can be mentioned:

- High-quality translation: by using neural machine translation technologies, Amazon Translate can provide high-quality translations.
- Multiple language support: Amazon Translate supports a wide variety of popular languages, including English, Chinese, Arabic, French, Spanish, German, Italian, Japanese, Portuguese, Russian, and a number of others.
- Scalability: Amazon Translate facilitates the ability for translation from a few rows to millions of pages with flexible scalability.
- Rapid response and easy integration: Amazon Translate provides quick translations and can be integrated with other AWS services to help users to create multilingual applications.
- Flexible pricing: Amazon Translate pricing model is based on the number of translated characters and used languages.

2.5.6 Amazon Polly

Amazon Polly is a text-to-speech service that allows users to generate natural-sounding speech from text in multiple languages and a diversity of accents. For instance, newspapers can integrate with AWS Polly to provide a feature enabling users to listen to content from articles instead of having to read them.

Some of the key characteristics and benefits of Amazon Polly can be mentioned:

- Multiple language support: Amazon Polly supports multiple common languages, such as English, French, Chinese, Arabic, German, Spanish, and many more.
- Multiple voice integration: offers users a variety of voices of both male and female genders in many accents and tones from different regions and countries.
- Multiple audio file formats support: Amazon Polly allows users to create audio files in many different formats, including MP3, OGG, PCM, and so on.
- Voice stress and pitch customization: Amazon Polly allows users to customize the intonation and accent of sentences or words to create more natural speeches.
- Voice speed adjustment: allows users to adjust the speed of generated voices from slow to fast to accommodate a variety of usage scenarios.
- Flexible pricing: Amazon Polly offers a usage-based pricing model where users only pay for the number of requests and characters converted to speech.

3 IMPLEMENTATION

This chapter focuses on describing the main components of the Kedu web version, including their design, development, operation, and interrelationships.

3.1 System architecture

3.1.1 Overview diagram and introduction

Kedu web version is a full stack project, the main components and how they interact with each other are depicted in Figure 8 below. In this system, the client side is a single-page application (SPA) that communicates with server side and other assets via HTTP REST API requests. The server side is built based on the serverless architecture, connects to the relational database and interacts with other AWS services to provide necessary data and features for the client side.

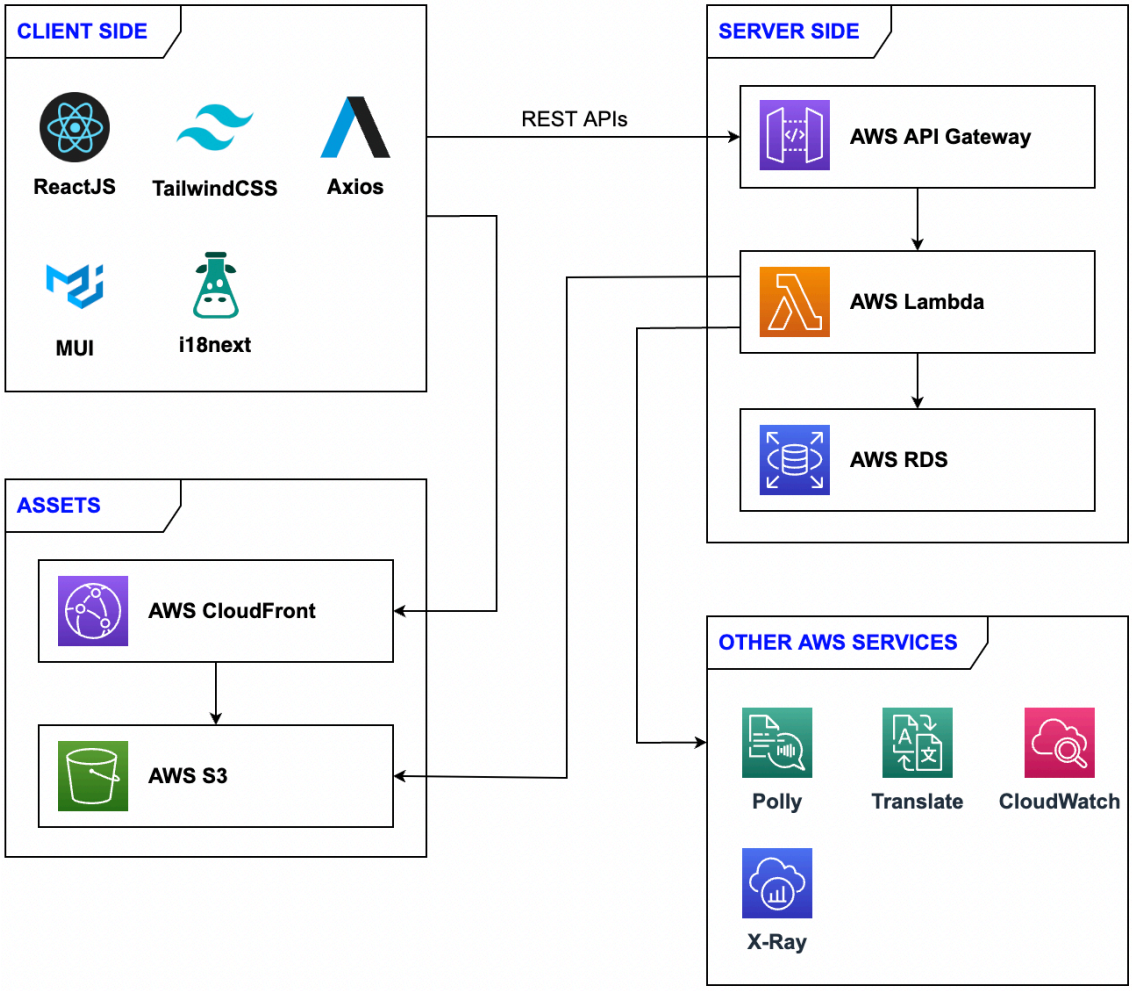


FIGURE 8: Overview diagram of the Kedu web version

3.1.2 Interpretation of main components

The main components, their purposes and roles in the Kedu web version are described in Table 2 below.

TABLE 2: Description of main components in the Kedu web version

	Components	Description
CLIENT	User Interface	A single-page application built using ReactJS, TailwindCSS, MUI, etc.
	HTTPClient	The website uses Axios and TanStack Query v4 (react-query previously) to call Restful APIs.
SERVER	AWS API Gateway	Acts as a router or controller for the API server.
	AWS Lambda	Serverless functions that contain system logic and handle requests sent from API Gateway.
	AWS RDS	A relational database (MySQL), normally used by AWS Lambda to perform querying, adding, editing, and deleting necessary information.
ASSETS	AWS S3	Used to save images, files, audio, videos, and other assets.
	AWS CloudFront	Act as a content delivery network (CDN) to cache, offload, and speed up the loading of resources from S3 buckets.
OTHER AWS SERVICES	AWS Polly	Used for text-to-speech conversion. In this project, it will help in playing audio of words and sentences from videos or flashcards on some screens.
	AWS Translate	Assists in the translation of words and sentences (from videos, flashcards, etc.) into other languages.
	AWS CloudWatch	A service supports storing and viewing logs from other services such as Lambda, S3, RDS, API Gateway, and more.
	AWS X-Ray	Used to monitor and analyze the running application in order to identify bugs and performance issues.

3.2 Client side

3.2.1 Main technologies

The client side is a single-page application developed using the following technologies:

- Main UI library: ReactJS
- Programming language: Typescript
- UI components and styling: MUI and TailwindCSS
- Other libraries: React router dom, React hook form, Yup, Dayjs, React icons, Lodash, React toastify, and more.
- AWS Services: AWS Amplify, Amazon S3

3.2.2 Development and management tools

Build tool

The front end of the Kedu web version utilizes ViteJS to develop, manage, and build a ReactJS project instead of using webpack, as specified in the official ReactJS project creation statement (npx create-react-app). ViteJS has many nice advantages and benefits, and speed is one of the vital factors in considering using it in this project. For example, in practical tests, ViteJS needs less than 200 milliseconds on average to make a website ready for showing and developing locally (Figure 9), and about 10 seconds to build and optimize a complete product ready for deployment in production (Figure 10).

```
~/Desktop/Kun/TAMK/Thesis/Kedu-Frontend alpha +1 ?1
❯ yarn dev
yarn run v1.22.19
$ vite --host

  VITE v4.1.4  ready in 129 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: http://192.168.3.27:5173/
  ➜  press h to show help
```

FIGURE 9: The speed of building and running the website on localhost

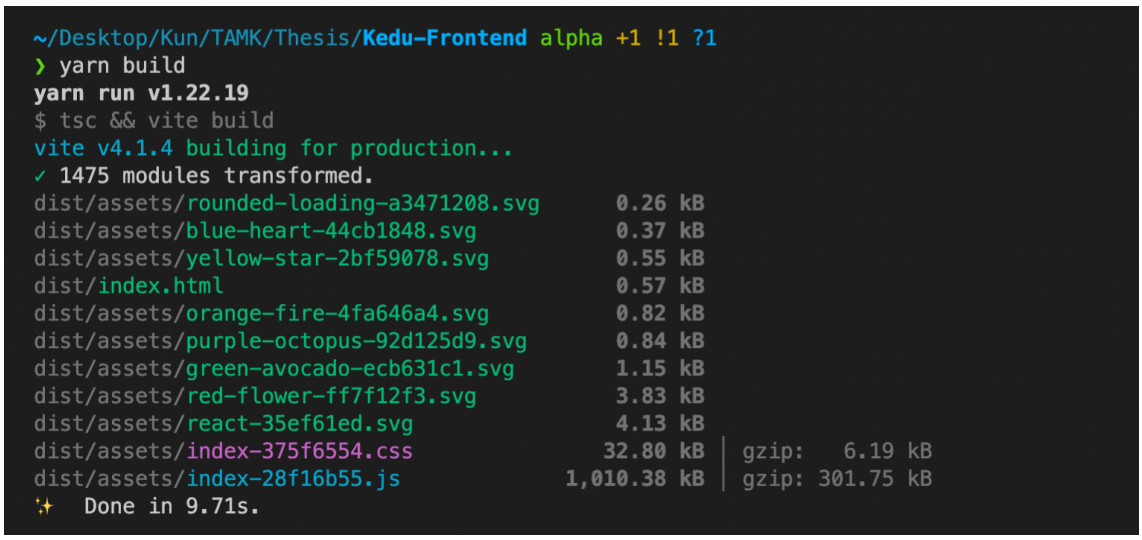


FIGURE 10: The speed of optimizing and building the complete product for deployment when using ViteJS

Formatter and analysis tools

To improve the code quality, Prettier – an automatic code formatting and analysis tool, is applied in the project (Figure 11), which helps in unifying the common coding style of the whole project and produces beautiful, straightforward, and easier-to-read code.

FIGURE 11: How Prettier's rules, code conventions, and format are defined

In addition, by utilizing ESLint – a tool for analyzing, locating, and highlighting problems and errors in Javascript and Typescript code based on standard coding style guides, the code of the project will always comply with certain standards and patterns, thereby enhancing code quality and reducing potential risks.

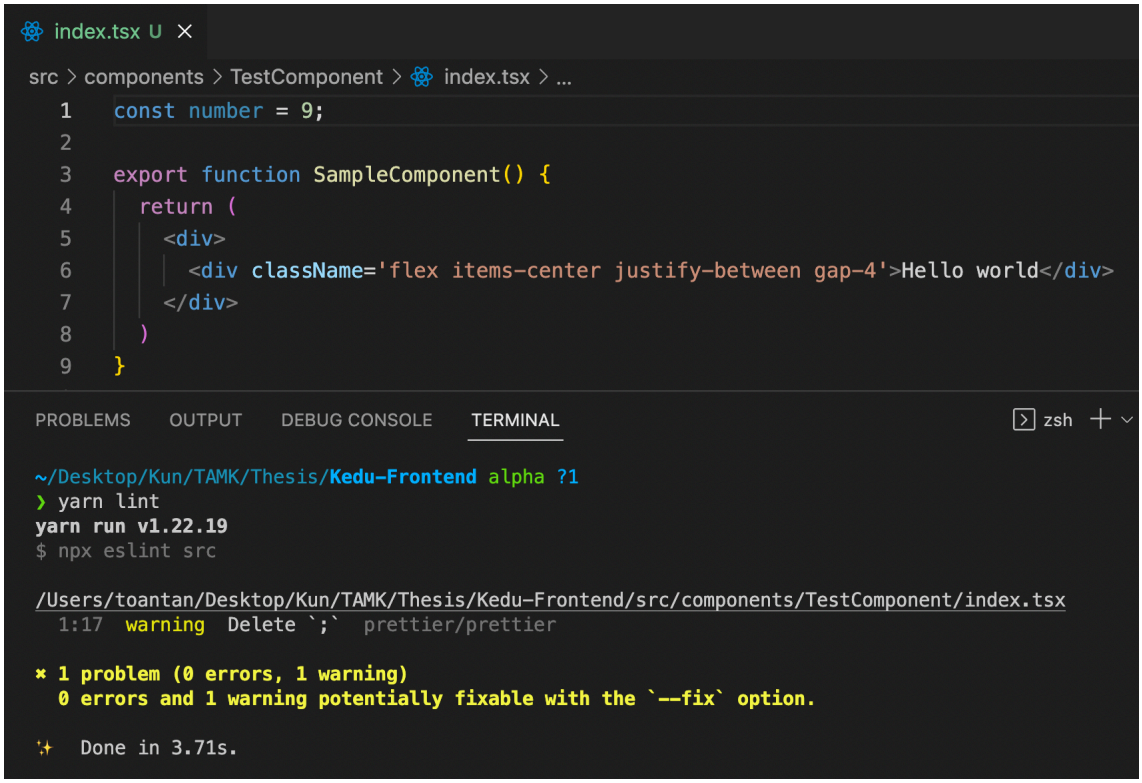


FIGURE 12: Example of how ESLint inspects and throws warnings and errors in the source code

In the example above, the Prettier format rule in the 4th line of Figure 11 declares that semicolons are not needed at the end of lines. However, in the first line of Figure 12, there is a redundant semicolon placed at the end of the statement, causing ESLint to issue a warning.

Source code hosting

The project uses GitHub for source code management and version control.

3.2.3 Deployment

AWS Amplify is applied for Continuous Integration and Continuous Delivery (CI/CD) implementation. By creating a project in Amplify, authorizing and connecting to the repository on GitHub, and then declaring the build specification in the amplify.yml file (Figure 13), Amplify will listen, build, and deploy code whenever a new commit is pushed to the branch of a particular GitHub repository.

App build specification Download Edit

To modify your app's build spec choose 'Edit'. Alternatively, you may download the YML file and deploy in the root of your repository to as amplify.yml to override this setting.

File: amplify.yml

```yaml
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - yarn install
    build:
      commands:
        - yarn build
  artifacts:
    # IMPORTANT - Please verify your build output directory
    baseDirectory: dist
    files:
      - '**/*'
  cache:
    paths:
      - node_modules/**/*

```

FIGURE 13: Build specification written on amplify.yml file

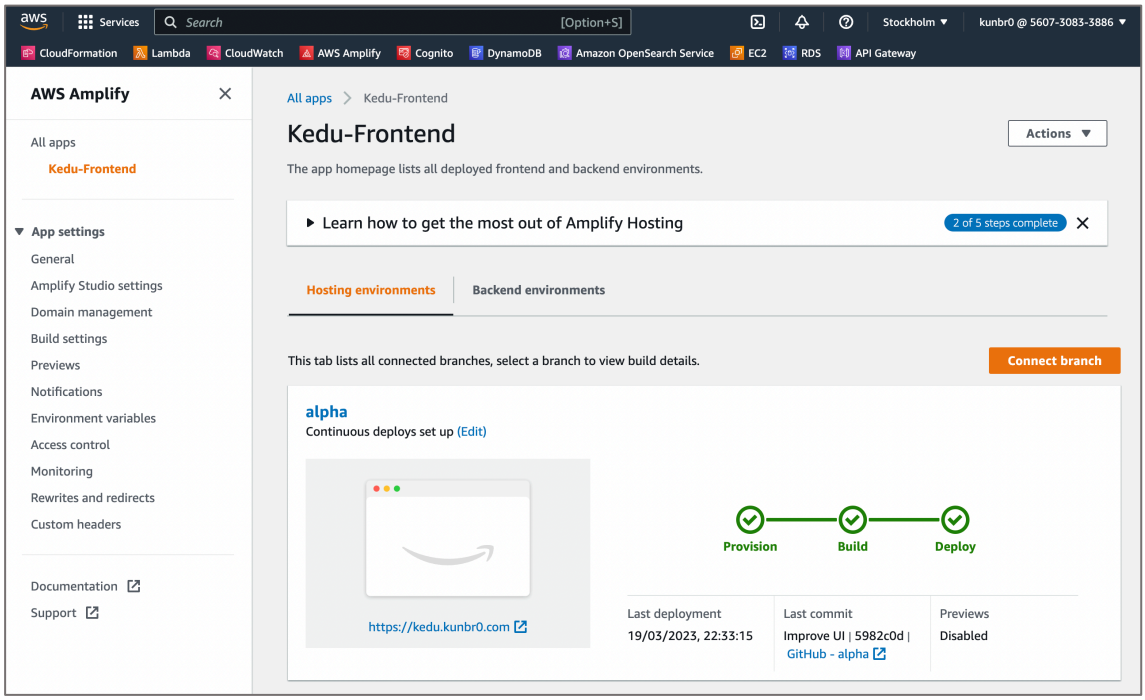


FIGURE 14: Using AWS Amplify to implement CI/CD and deploy a ReactJS project from source code hosted on GitHub

In the example illustrated in Figure 14, AWS Amplify will listen to the “alpha” branch of the repository, and it will provision the necessary resources to build and deploy the website whenever any changes are made to this branch (for example, when commits are pushed or pull requests are merged to this branch).

3.3 Server side

3.3.1 Main technologies

The server side is an API server developed on the serverless architecture with the following main technologies:

- Framework: Serverless
- Programming language: Typescript
- Object-relational mapping (ORM): Typerom
- Database: MySQL
- AWS Services: AWS CloudFormation, Amazon API Gateway, AWS Lambda, AWS Aurora, Amazon CloudWatch, etc.

3.3.2 Entity Relationship Diagram (ERD)

The database used in the system is AWS Aurora MySQL - a relational database. Figure 15 below presents the structure of the tables and their relationships to other tables.

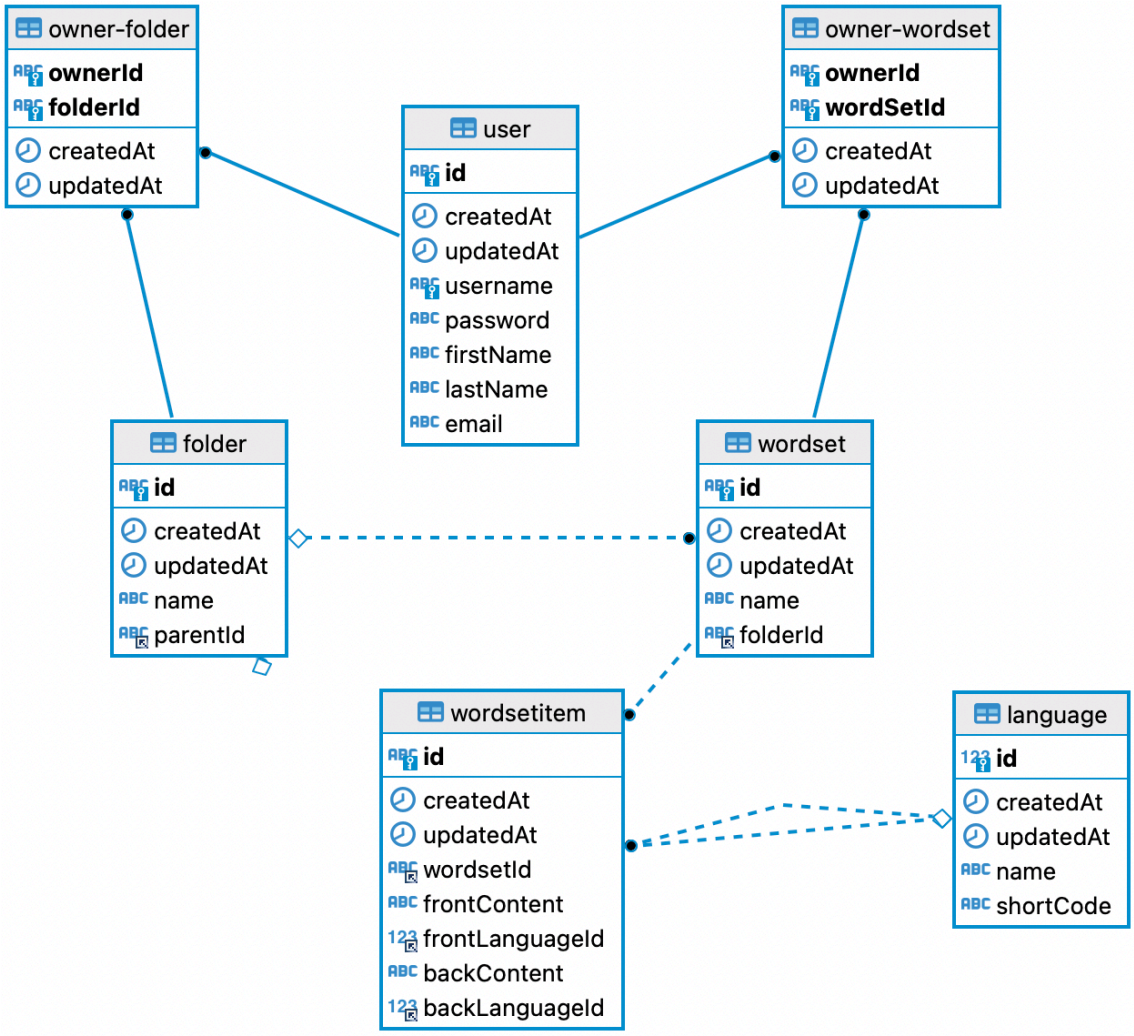


FIGURE 15: Entity Relationship Diagram (ERD) of the Kedu's database

3.3.3 Database entity's relationships and description

Some of the main relationships between the tables in Kedu's database (Figure 15) can be mentioned:

- One user can have many folders, and ownership is represented through the owner-folder table.
- One user can have many word sets, and ownership is represented through the owner-wordset table.
- One folder can have many other word sets or folders within it.
- One word set can store many word set items.
- Content on one side of a word set item can be written in one language.

3.3.4 Authentication module

This module is responsible for allowing users to register, log in, and control user access to private resources.

After registering an account (Figure 16) and logging in with the newly created account and password (Figure 17), the user on the client side will receive a JWT token.

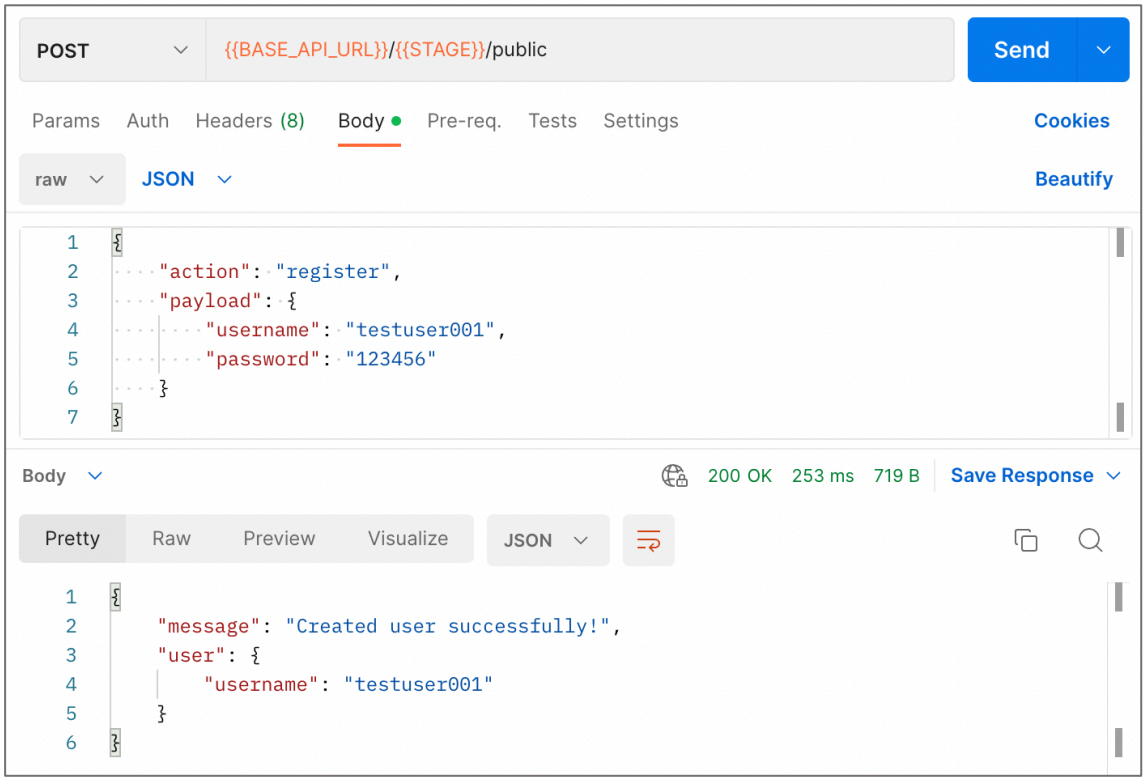


FIGURE 16: HTTP request of the Register action

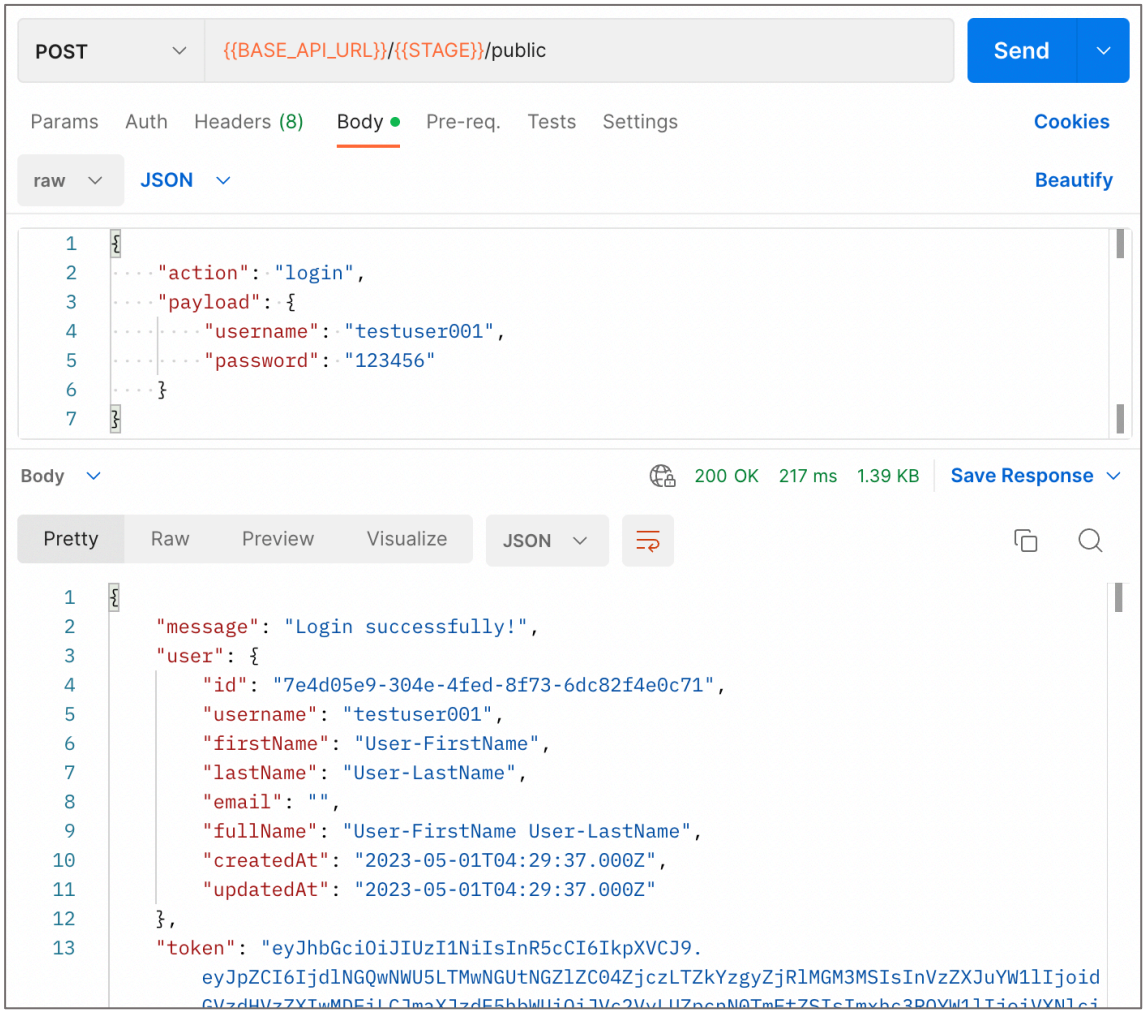


FIGURE 17: HTTP request of the Login action

To be able to access private resources such as folder data, word set data, profiles, etc., users must attach the JWT token received after login (Figure 17) to the header of those requests (Figure 19) to verify permission access. If the token is not provided or is invalid, the request will be terminated and returned with the HTTP status 403 Unauthorized (Figure 18).

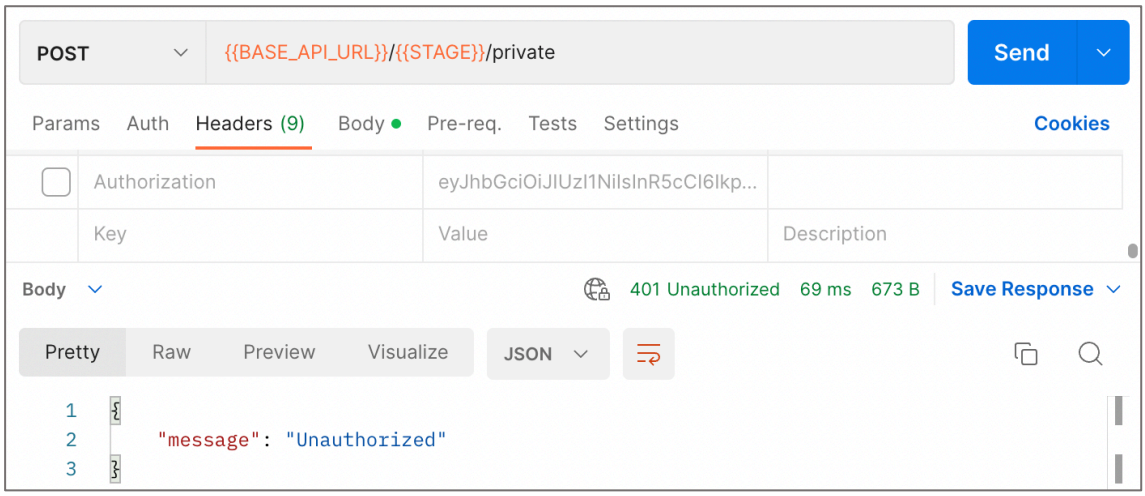


FIGURE 18: Accessing a private route without a valid authorization token (JWT)

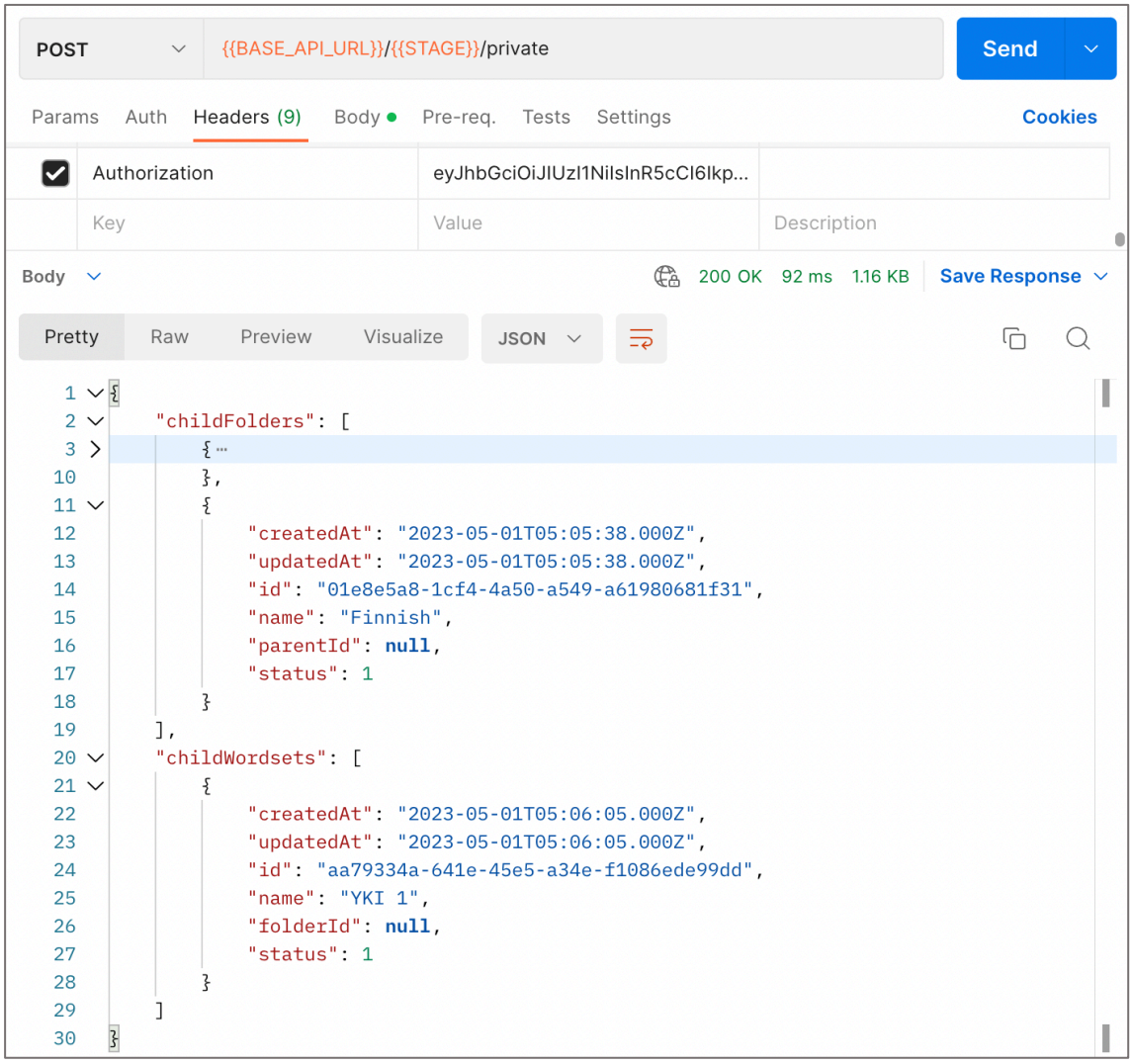


FIGURE 19: Accessing a private route with a valid authorization token (JWT)

Kedu's backend uses Lambda authorizers to perform authentication at the API Gateway before allowing access to private resources (Figure 20).

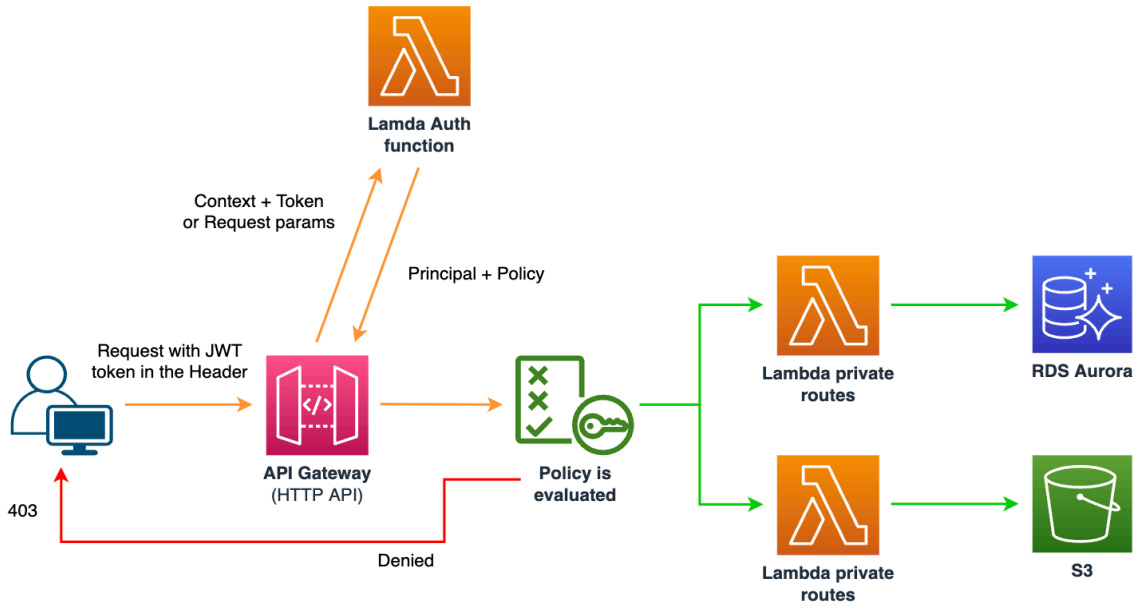


FIGURE 20: Authentication at the API Gateway with Lambda authorizers

3.3.5 Text-to-speech module

The Text-to-speech module is built for the purpose of generating speech audio files from texts, which will be applied to playing audio about the content of the flashcards as well as texts or sentences on the website. The diagram of this module can be seen in Figure 21. In which code is written in the AWS Lambda function that takes a text in string format as input, then calls the Amazon Polly service via CDK provided by AWS to generate and return an mp3 audio file in base64 format.

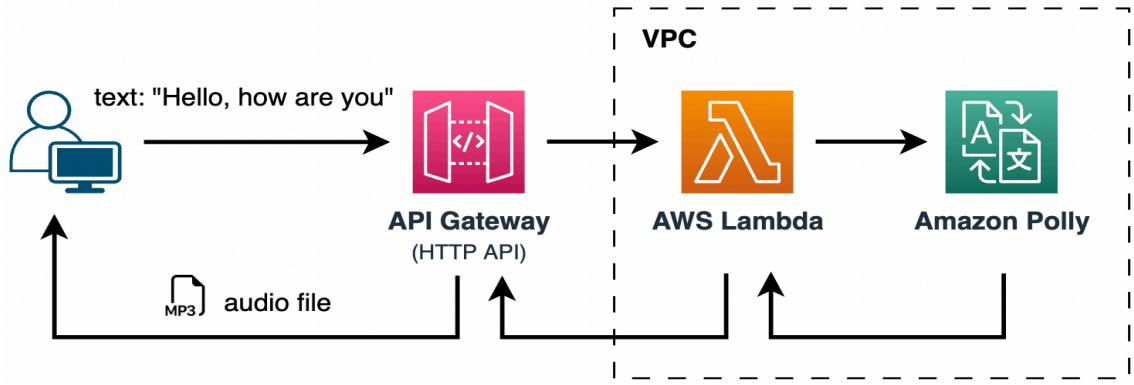


FIGURE 21: Diagram of the Text-to-speech module

In addition, the Text-to-speech module also allows customizing the voices of many people from many different countries and adjusting the speed of the audio file. Figure 22 below is an example of creating a speech from a text with the custom voice of Emma in British English (en-GB).

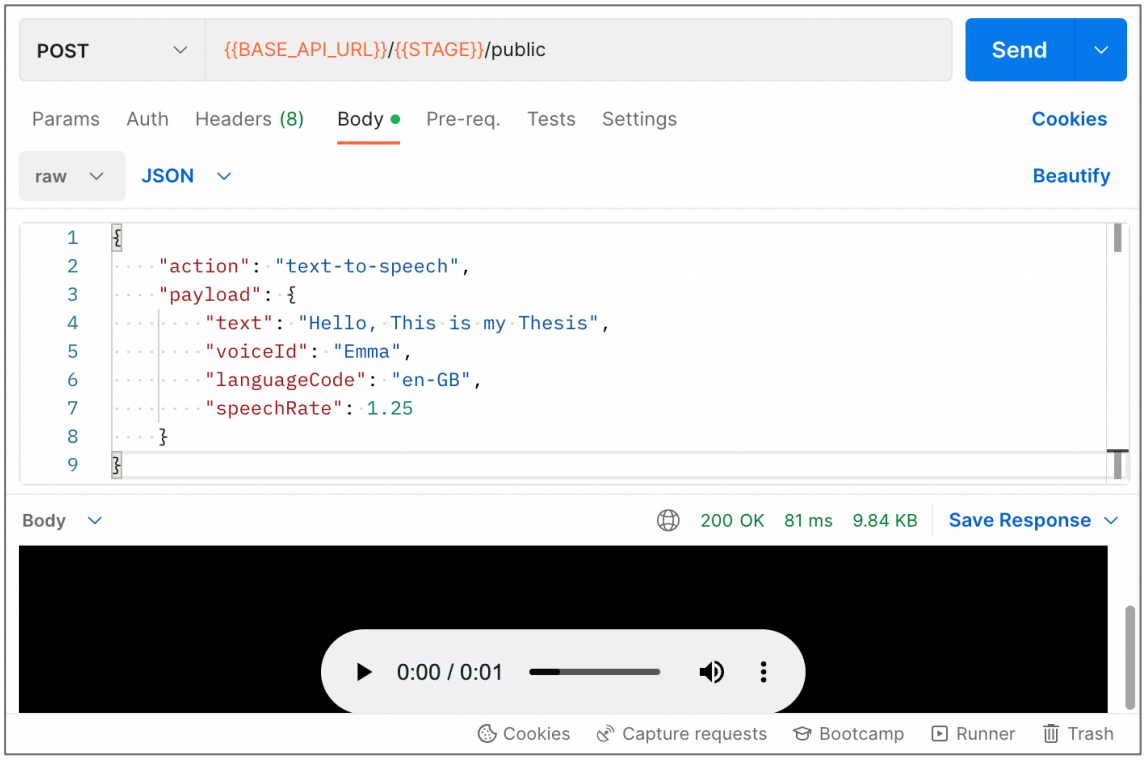


FIGURE 22: HTTP request of the Text-to-speech module

3.3.6 Translation module

The Translation module helps translate text into many different languages. It is used to translate paragraphs when watching videos or the content of flashcards in Kedu. The diagram of this module can be seen in Figure 23. In which the logic is written in the AWS Lambda function that takes a text in string format, the source and target language codes in the ISO 639-1 two-digit string format (Codes for the Representation of Names of Languages n.d.) as input, then calls the Amazon Translate service via CDK provided by AWS to translate and return the text translated in the desired language.

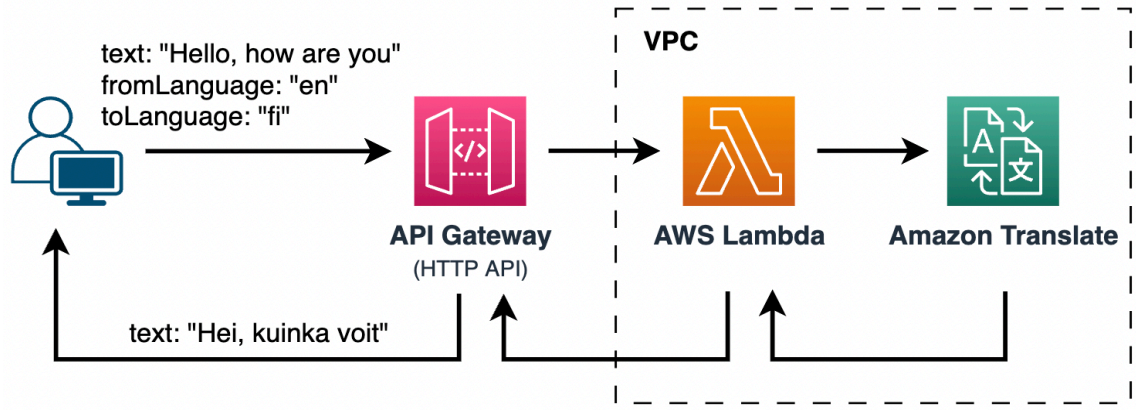


FIGURE 23: Diagram of the Translation module

Figure 24 below is an example of using the Translation module to translate a text from English to Finnish.

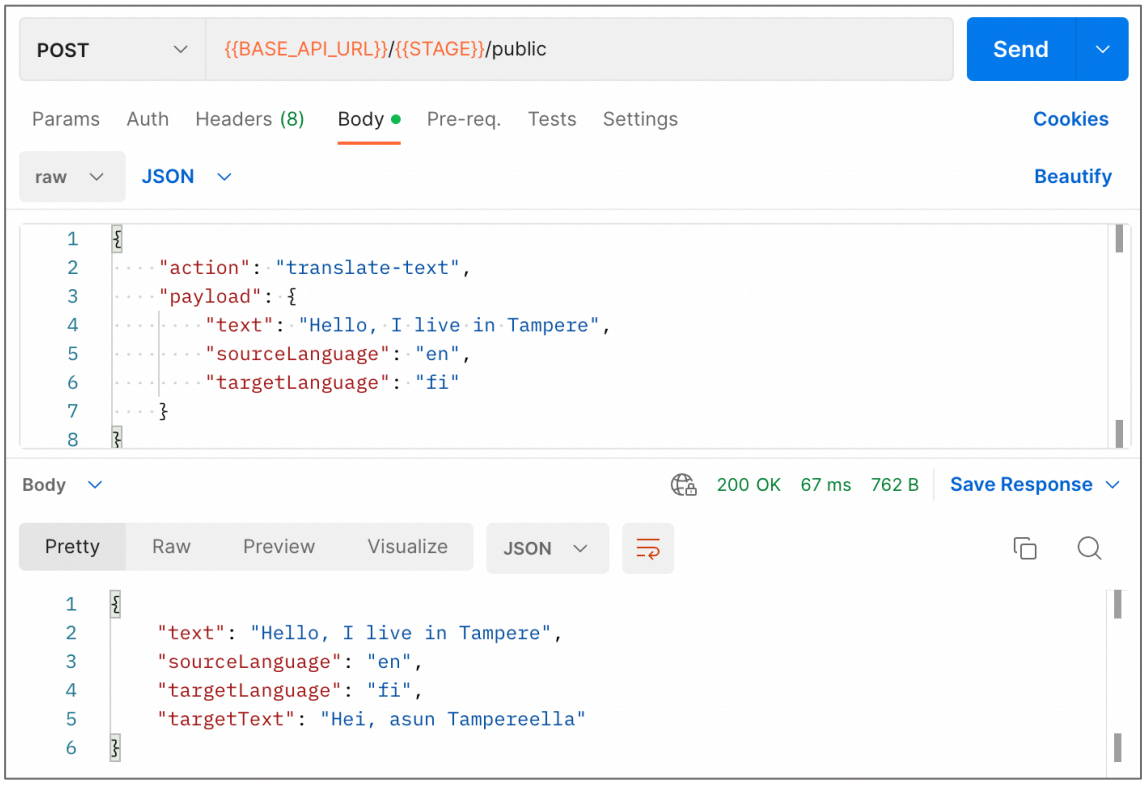


FIGURE 24: HTTP request of the Translation module

3.3.7 YouTube module

The YouTube module is responsible for handling tasks and logic related to YouTube, such as searching videos, getting search completion suggestions, getting video detail, getting video captions, and so on. Unlike most of the other modules on Kedu's server side, which are written in NodeJS, this module is written in Python programming language to take advantage of existing famous open-source libraries such as “yt-dlp” and “youtube-search-python” which help to search and retrieve information about videos, channels, playlists, and more from Youtube. The diagram of the YouTube module is illustrated in Figure 25 below.

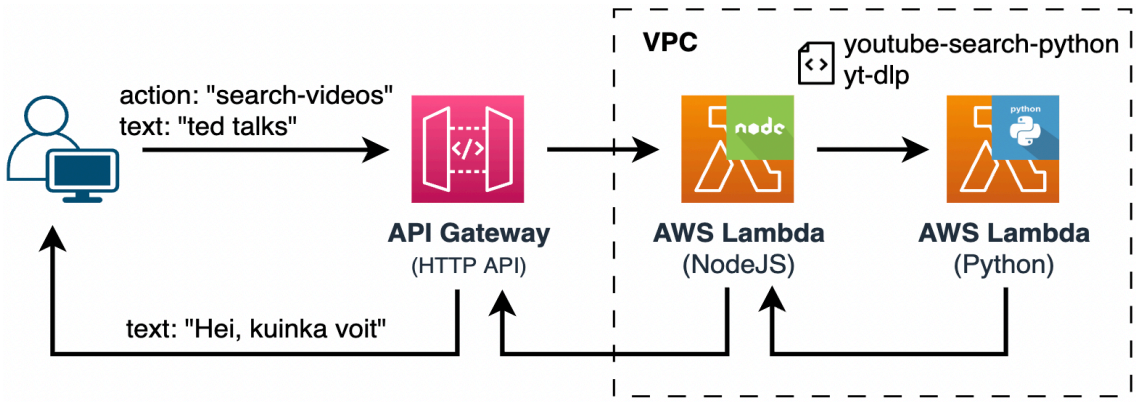


FIGURE 25: Diagram of the Youtube module

Some main actions of the Youtube module are applied in Kedu, such as:

Action: Get search completion suggestions

This action is used to suggest complete search keywords to users when they enter a few characters in the search bar (Figure 26).

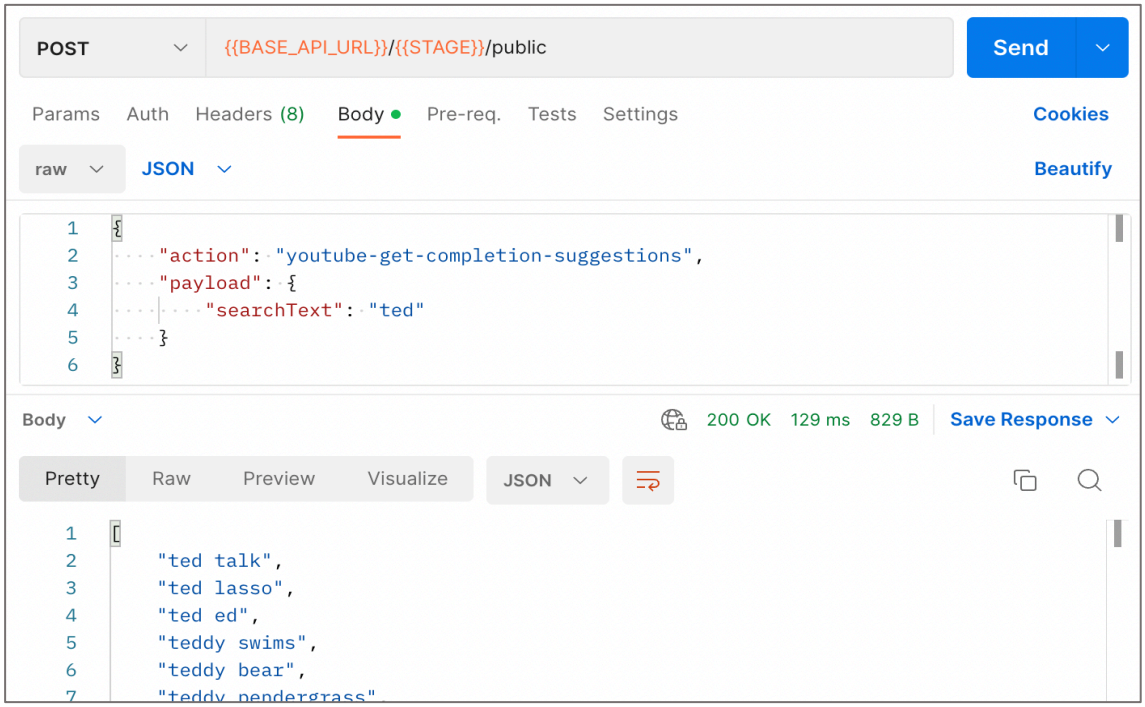


FIGURE 26: HTTP request of the Get search completion suggestions action

Action: Search videos

This action is used in the Videos Discovering page of Kedu (which will be illustrated and discussed in detail in the next section), allowing users to search for videos from YouTube based on the keywords they enter (Figure 27).

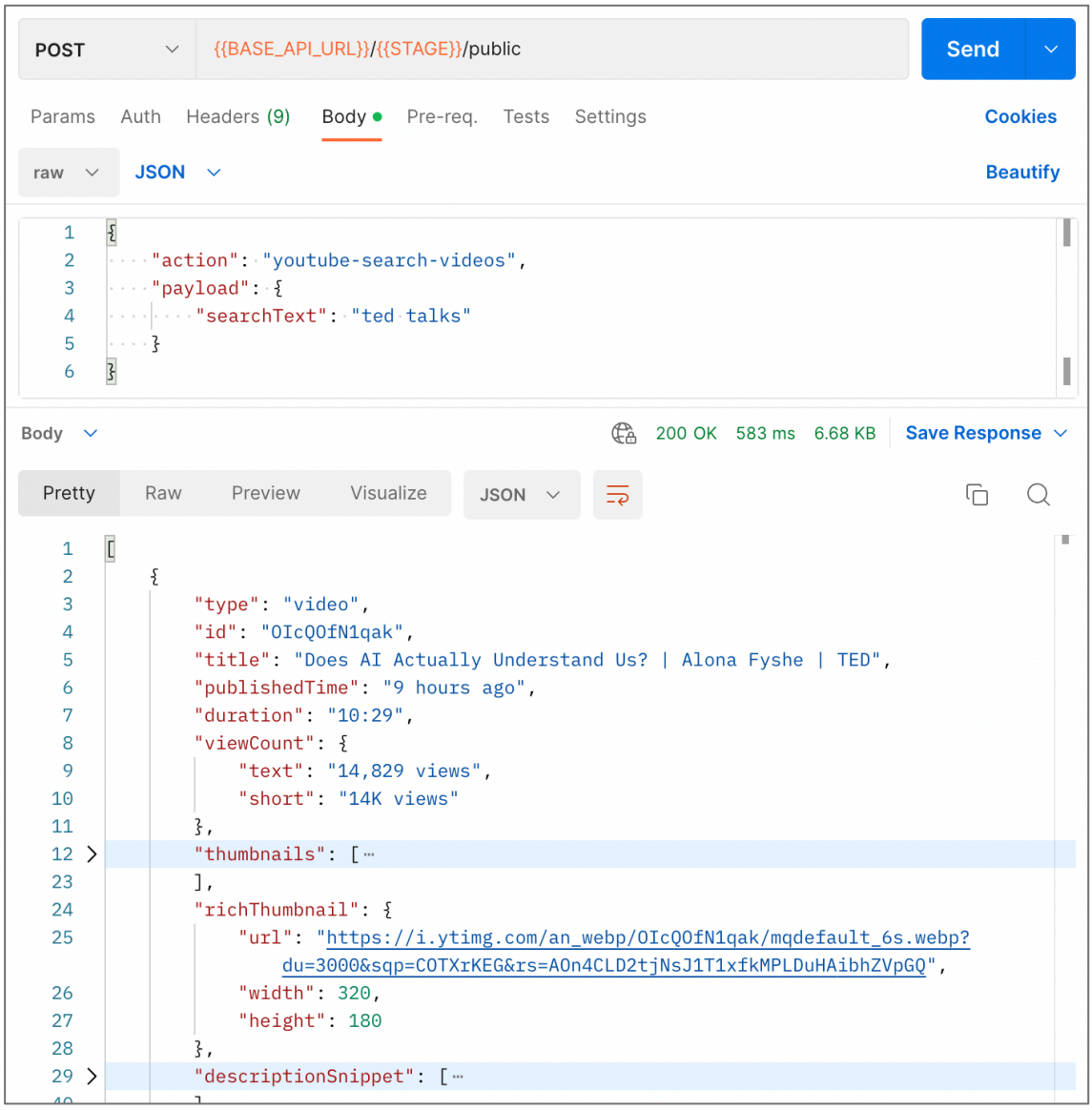


FIGURE 27: HTTP request of the Search videos action

Action: Get video detail

This action helps to get detailed information about a particular video, such as title, duration, view count, thumbnail, description, channel, keywords, publication date, translation language, and so on (Figure 28).

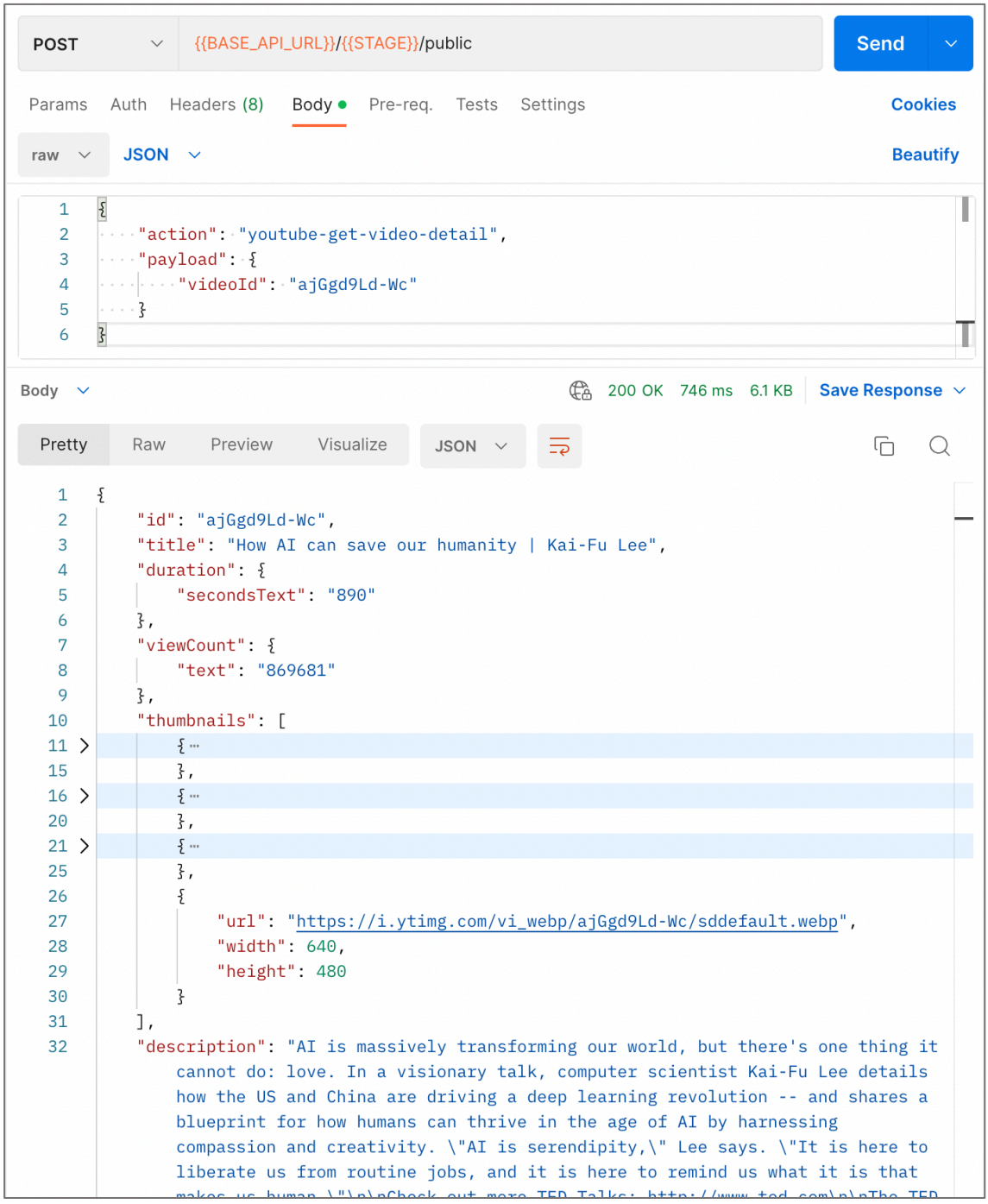


FIGURE 28: HTTP request of the Get video detail action

Action: Get video caption

This action is used in the Video Watching page (which will be illustrated and discussed in detail in the next section) to get captions in different languages of a video. Figure 29 below is an example of getting the English caption (through languageParams) from a video (through videoId).

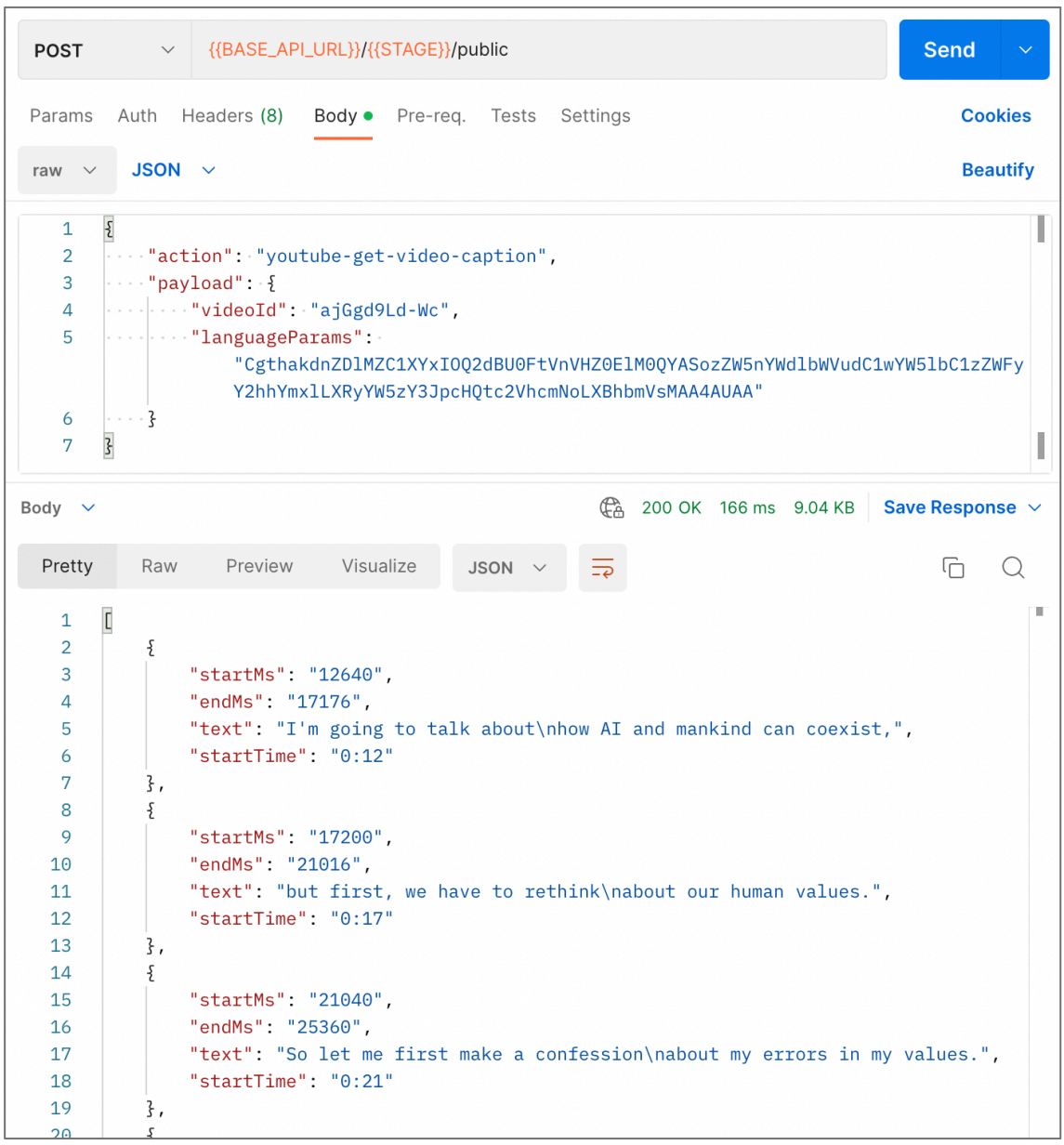


FIGURE 29: HTTP request of the Get video caption action

3.3.8 Deployment

Kedu's backend uses the Serverless framework, deploys and operates on the AWS cloud platform. Before deployment, all required resources, such as AWS Lambda (functions, memory, and runtime configurations), API Gateway, cloud provider, region, environment variables, and so on, are declared in the serverless.ts file (Figure 30).

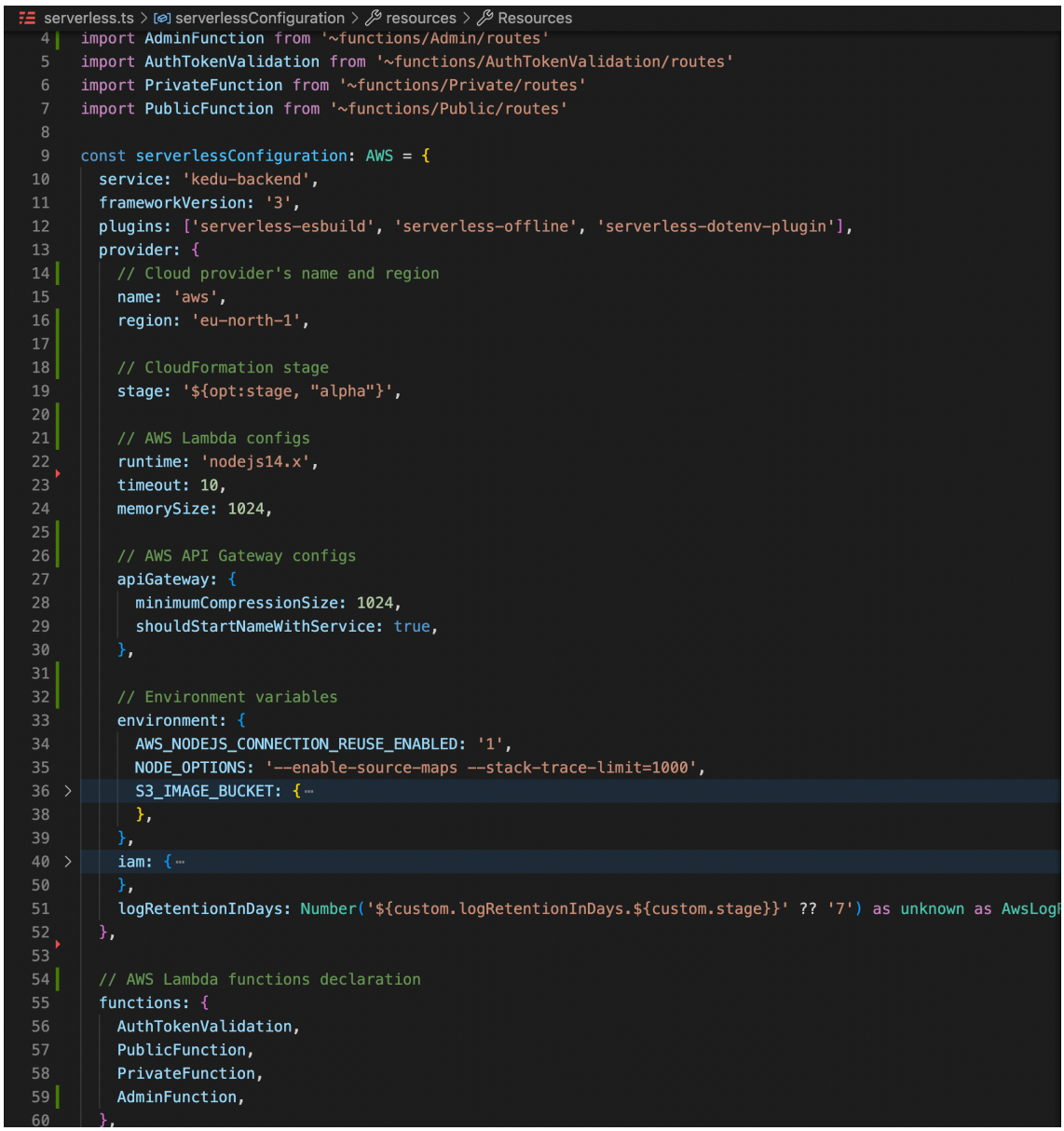


FIGURE 30: How resources and configurations are defined in serverless.ts file

Deployment is done by executing the command “serverless deploy”, with optional parameters like --stage, --aws-profile, and so on (Figure 31). In which:

- stage: can be anything, usually it will be named after the environment and working process of the team. For example, a product can have stages like alpha, beta, and production.
- aws-profile: is the name of the AWS profile and credentials configured on the current machine.

```
~/Desktop/Kun/TAMK/Thesis/Kedu-Backend alpha !4 ?4
❯ serverless deploy --force --aws-profile aws-kunbr0 --stage alpha
Running "serverless" from node_modules
DOTENV: Loading environment variables from .env:
        - MYSQL_DB_HOST
        - MYSQL_DB_NAME
        - MYSQL_DB_USERNAME
        - MYSQL_DB_PASSWORD
        - AUTH_LOGIN_JWT_TOKEN
        - AUTH_LOGIN_JWT_TOKEN_EXPIRES_IN
        - LAMBDA_YOUTUBE_GET_COMPLETION_SUGGESTION
        - LAMBDA_YOUTUBE_SEARCH_VIDEOS
        - LAMBDA_YOUTUBE_GET_VIDEO_DETAIL
        - LAMBDA_YOUTUBE_GET_VIDEO_CAPTION

Deploying kedu-backend to stage alpha (eu-north-1)

✔ Service deployed to stack kedu-backend-alpha (46s)

endpoints:
  POST - https://l9tbbet910.execute-api.eu-north-1.amazonaws.com/alpha/public
  POST - https://l9tbbet910.execute-api.eu-north-1.amazonaws.com/alpha/private
  POST - https://l9tbbet910.execute-api.eu-north-1.amazonaws.com/alpha/admin
functions:
  AuthTokenValidation: kedu-backend-alpha-AuthTokenValidation (315 kB)
  PublicFunction: kedu-backend-alpha-PublicFunction (3.9 MB)
  PrivateFunction: kedu-backend-alpha-PrivateFunction (3.8 MB)
  AdminFunction: kedu-backend-alpha-AdminFunction (3.9 MB)
```

FIGURE 31: Serverless framework's deploy command execution and result

After the deployment is done successfully, the resources can be viewed and managed centrally on a Stack of the AWS CloudFormation service in different convenient modes, such as Designer mode (Figure 32) or List mode (Figure 33).

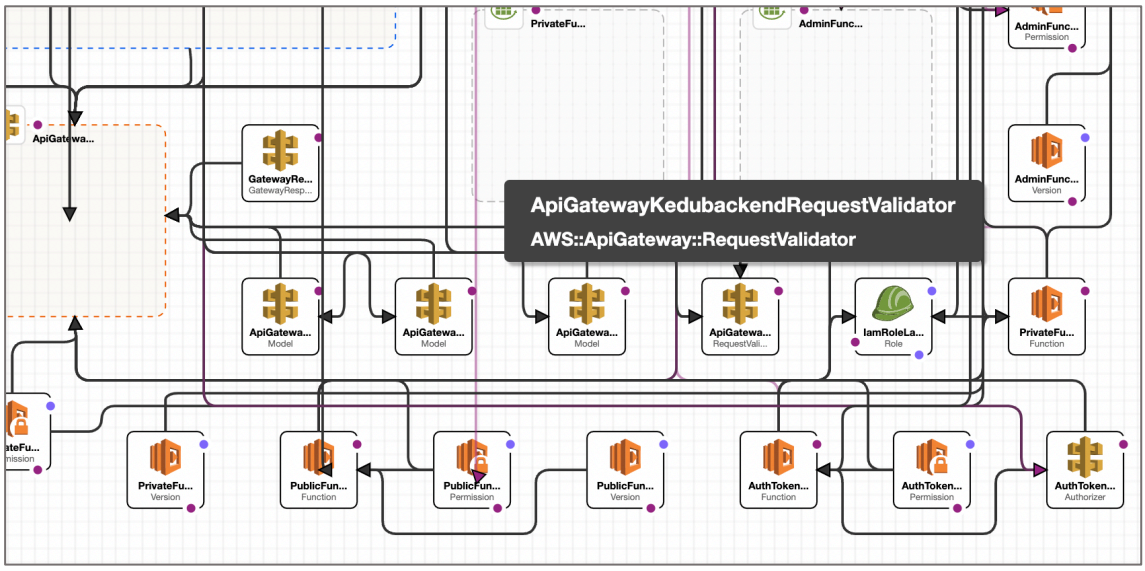


FIGURE 32: View and manage created resources in the Designer mode

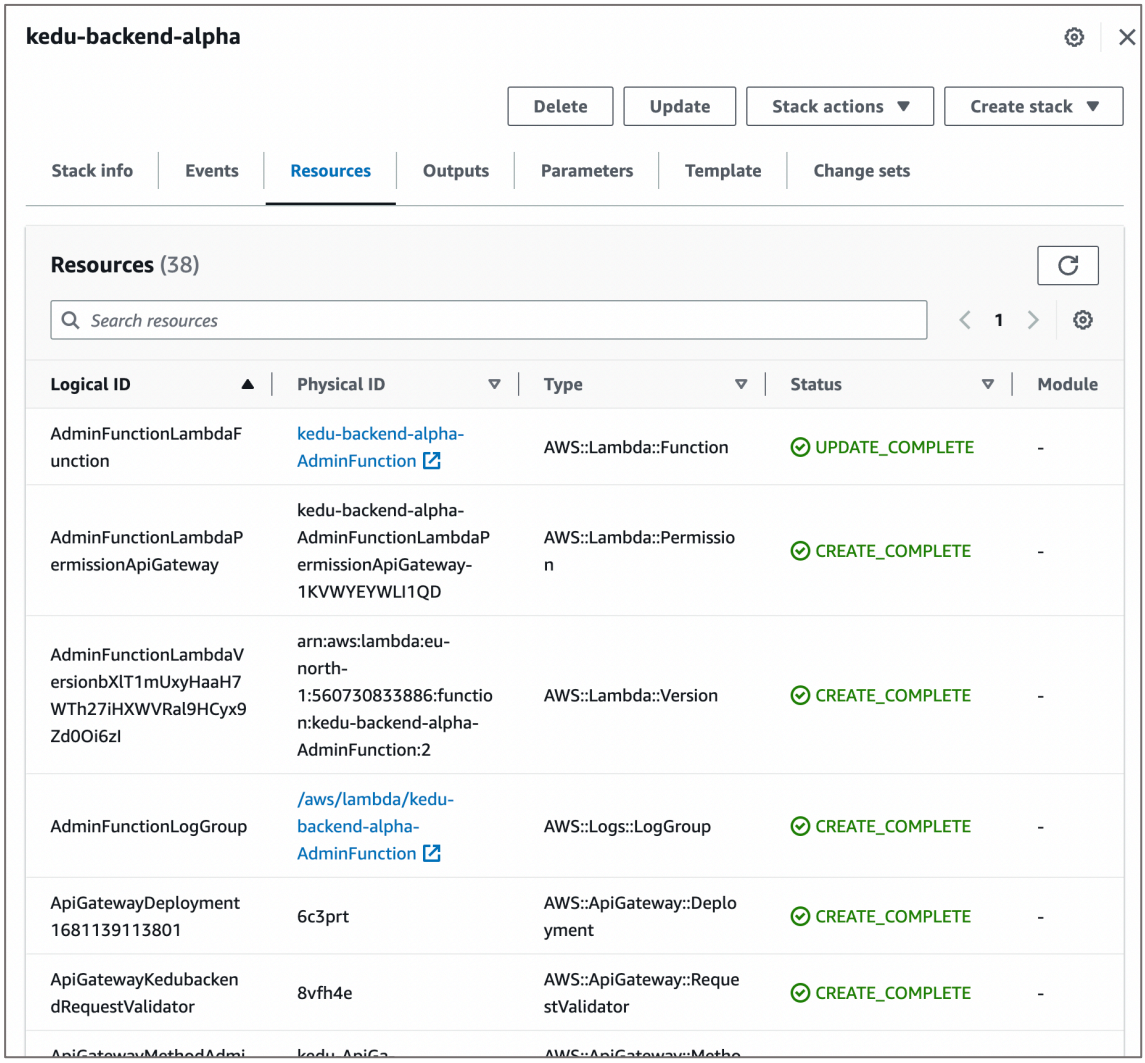


FIGURE 33: View and manage created resources in List mode

4 RESULTS AND DISCUSSIONS

This section presents the outcomes achieved in the development of the Kedu web version through screenshots as well as discussion and explanation.

4.1 Authentication

On the home page, there is a sidebar menu on the left; some options in the menu are disabled because they require authentication (Figure 34).

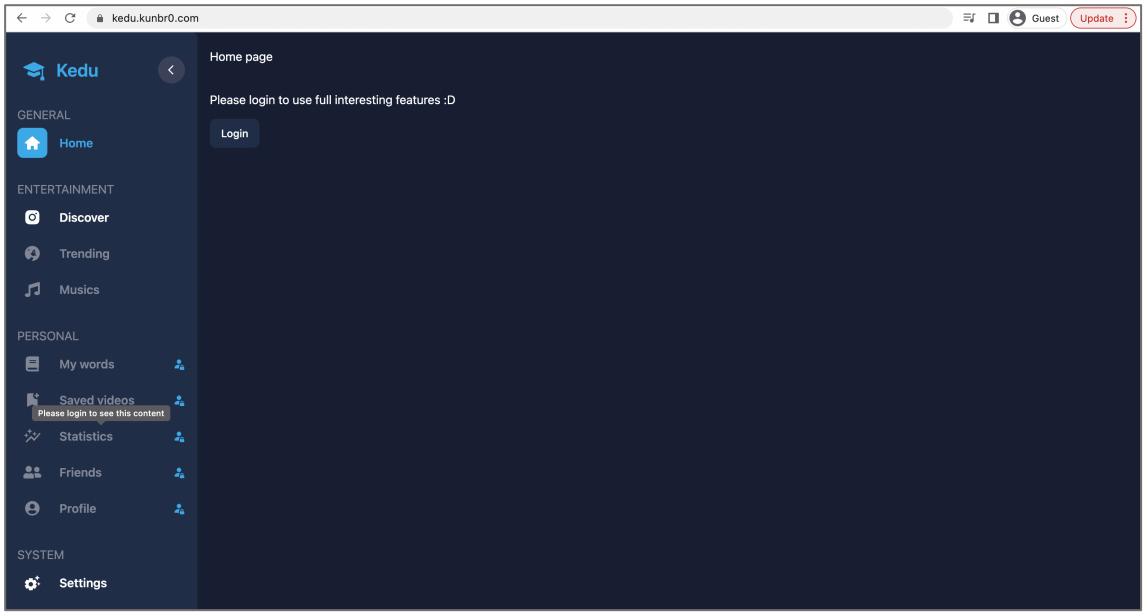


FIGURE 34: Home page before login

Users can log in by clicking on the Login button on the home page or any disabled items in the left sidebar menu to open the Login popup (Figure 35).

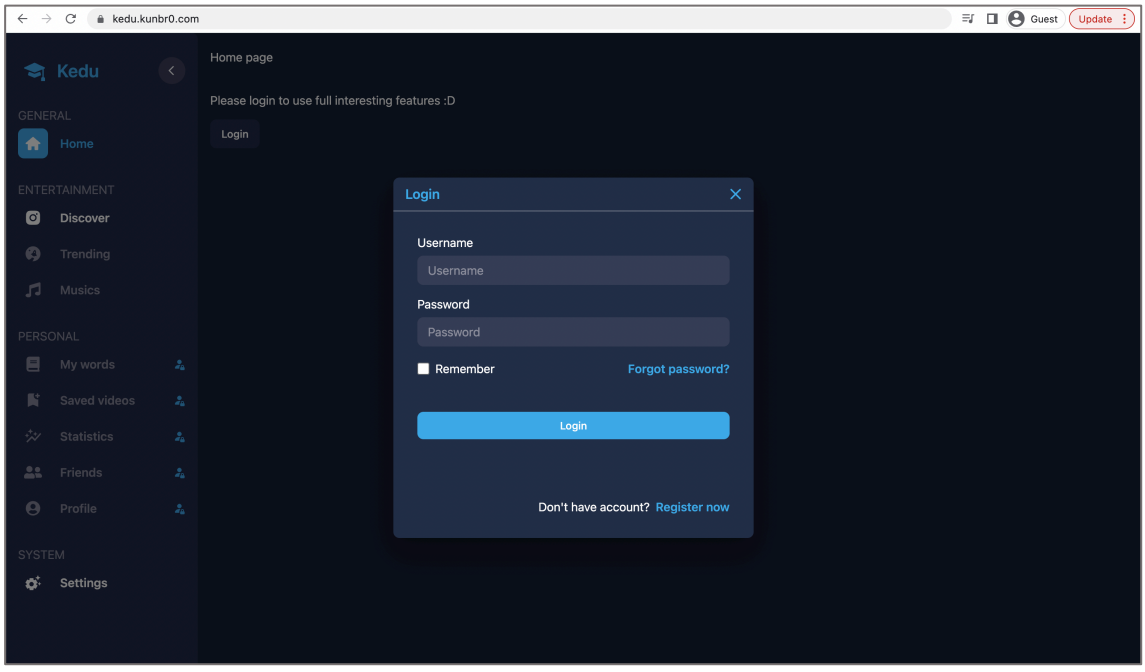


FIGURE 35: Login popup

If the user doesn't have an account, they can create a new one by clicking “Register now” in the Login popup (Figure 36).

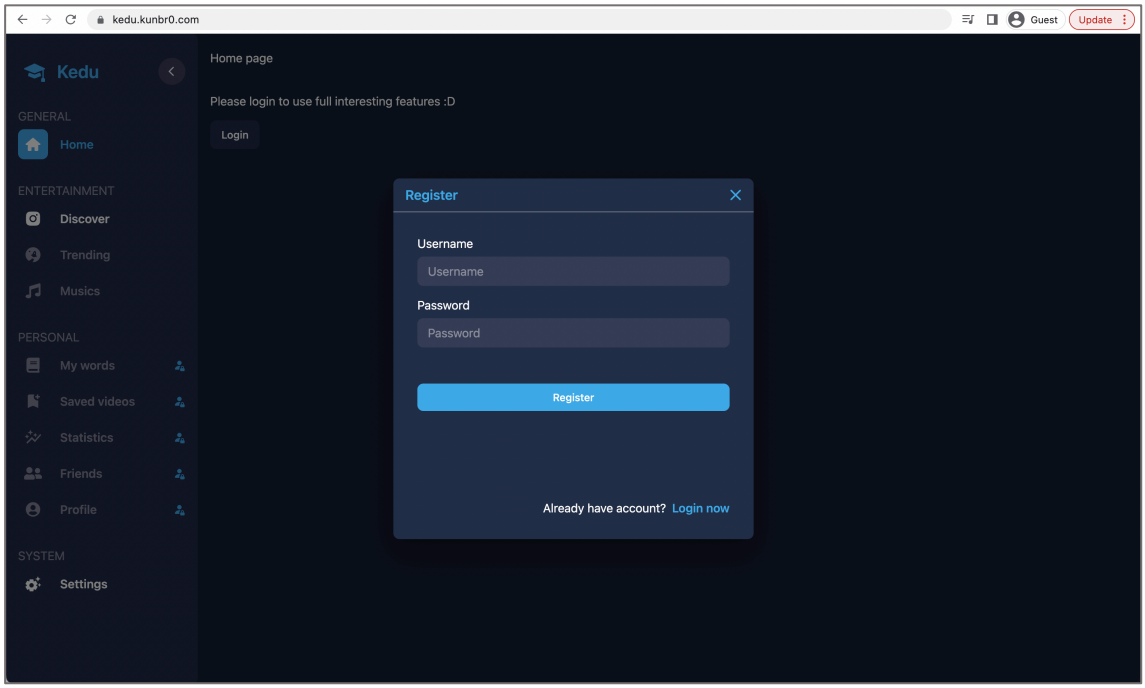


FIGURE 36: Register popup

After successfully creating an account and logging in, they will be returned to the Home page with a successful login message (Figure 37).

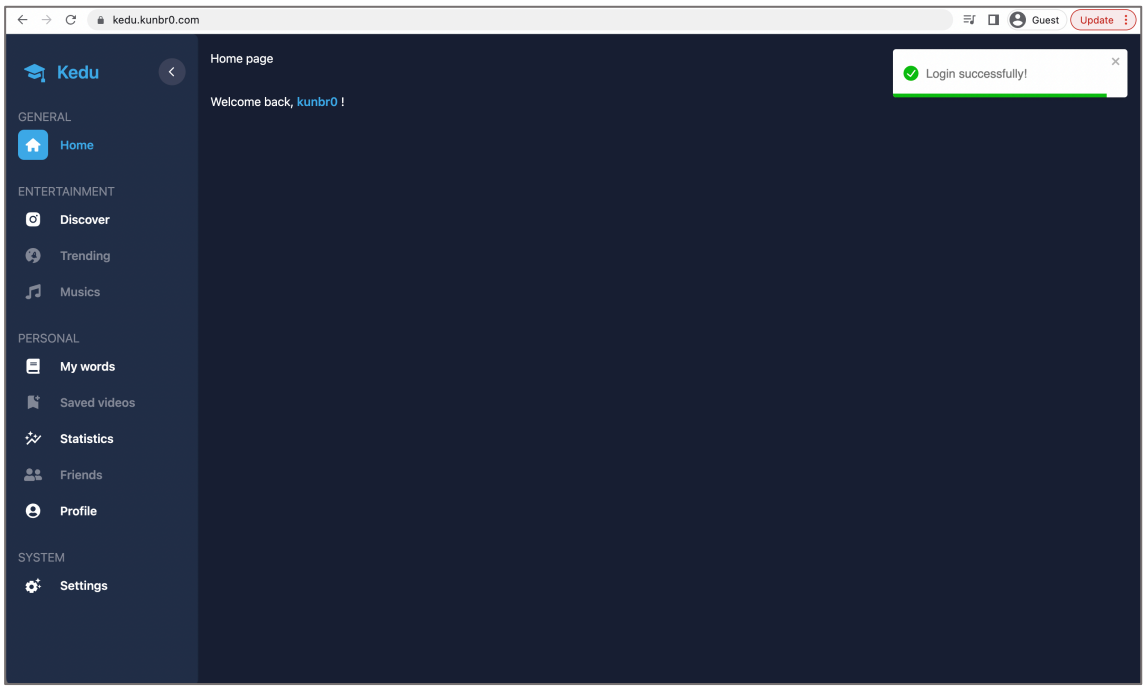


FIGURE 37: Home page after Login

4.2 Managing folders

Users can manage all of their folders and word sets on the My Words page. They can create folders or word sets by clicking the CREATE button in the upper right corner of the screen (Figure 38).

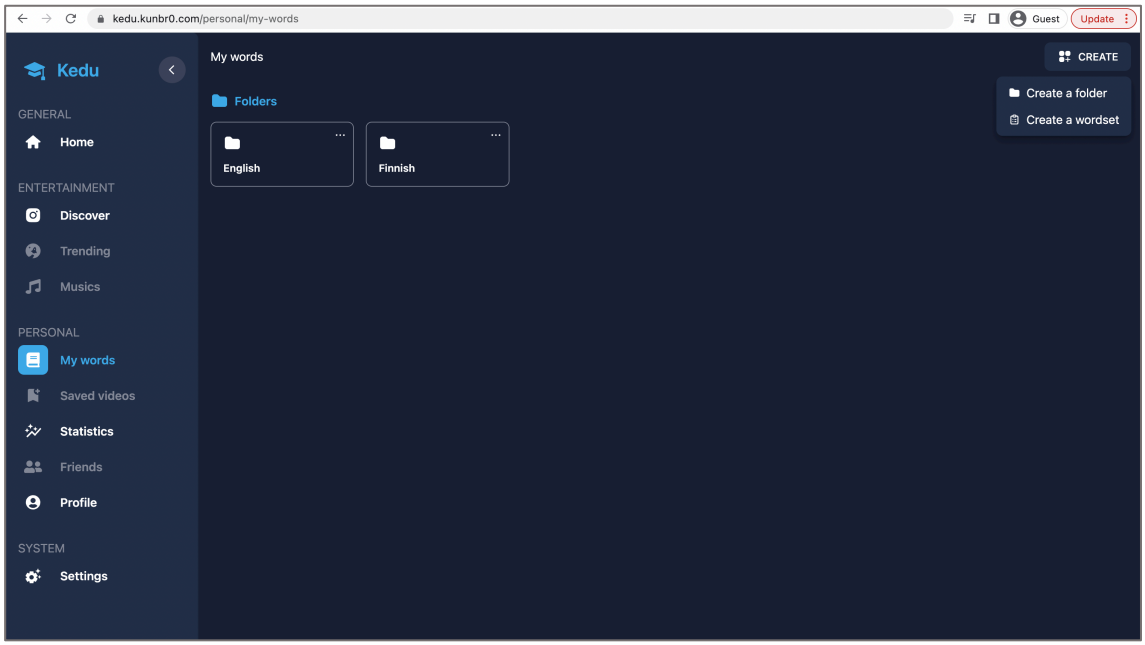


FIGURE 38: My Words page

This page displays folders and word sets and operates like the File Explorer in Windows or MacOS operating systems. One folder can contain many other sub-folders and word sets. For example, in Figure 39, the user goes to the Finnish folder and creates a new sub-folder named “Sub 3”.

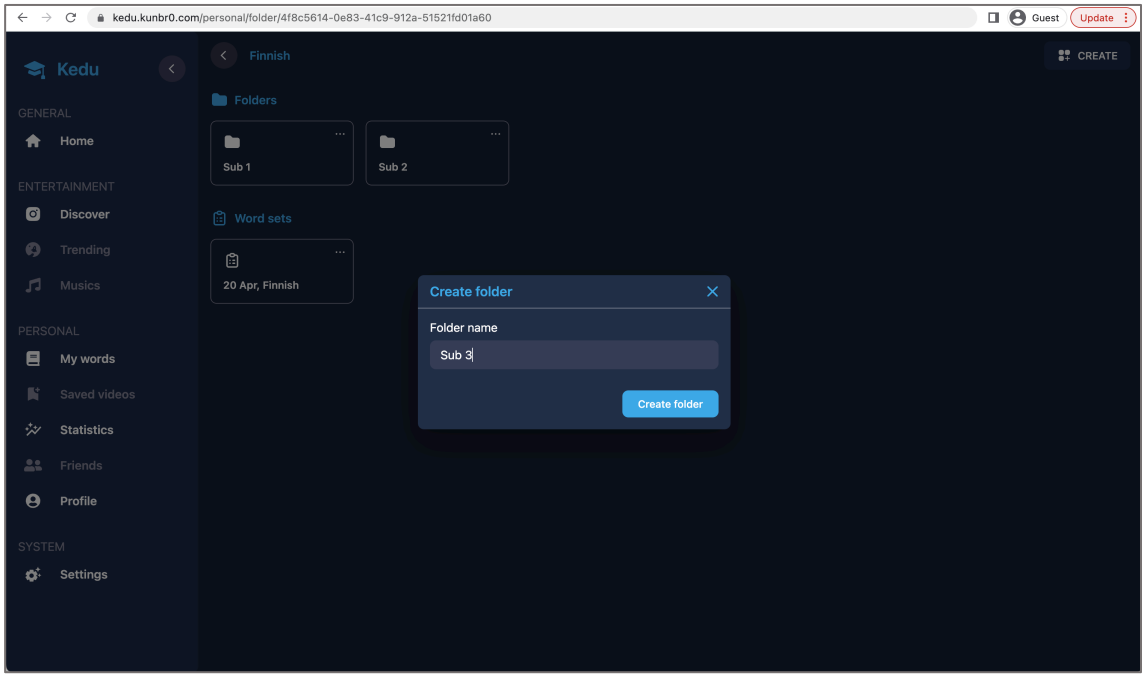


FIGURE 39: Naming and creating a sub-folder inside a folder

The “Sub 3” folder will be displayed in the “Finnish” folder after it is successfully created (Figure 40).

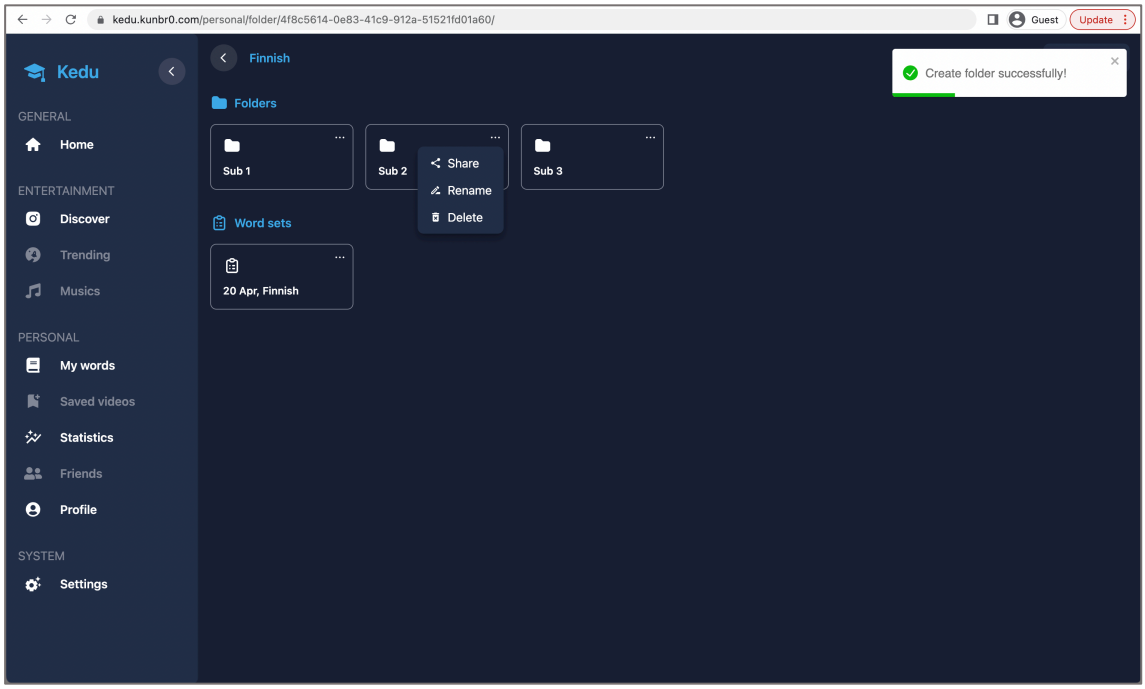


FIGURE 40: Finnish folder after creating a sub-folder

In addition, users can also share, rename, or delete folders or word sets that they have created by clicking on the 3-dot icon in the upper right corner of the folder or word set (Figure 40). For example, if users choose to delete the folder Sub 2, a confirmation popup will show up for them to review and confirm this action (Figure 41).

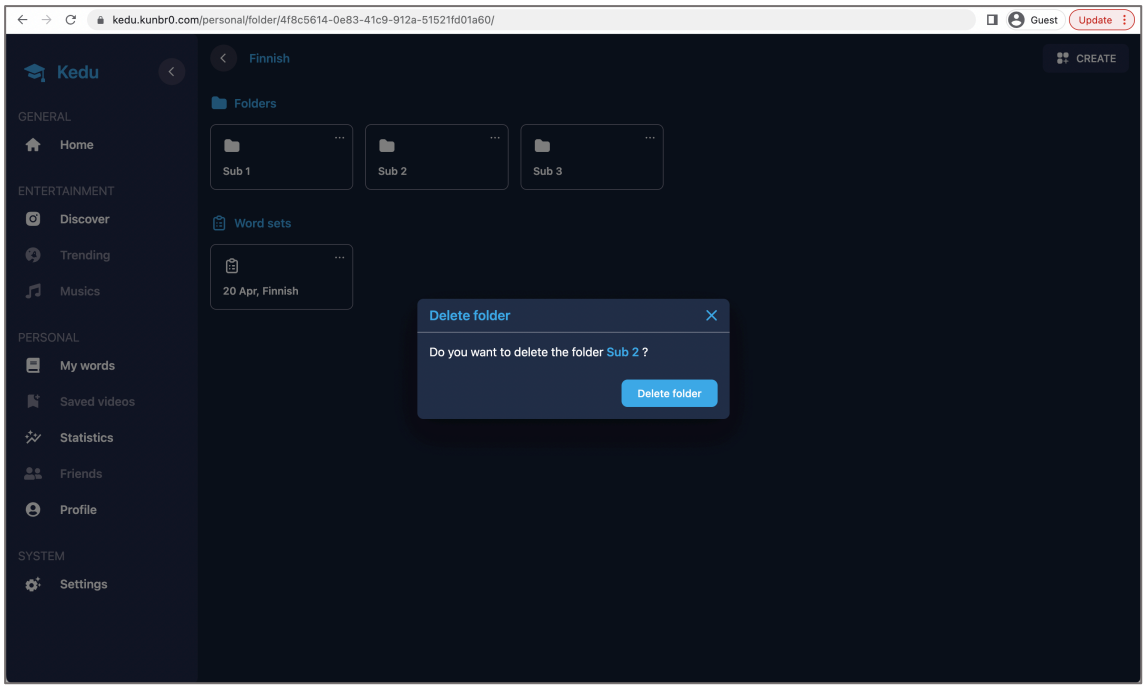


FIGURE 41: Delete folder confirmation popup

The folder “Sub 2” will disappear after users confirm the deletion (Figure 42).

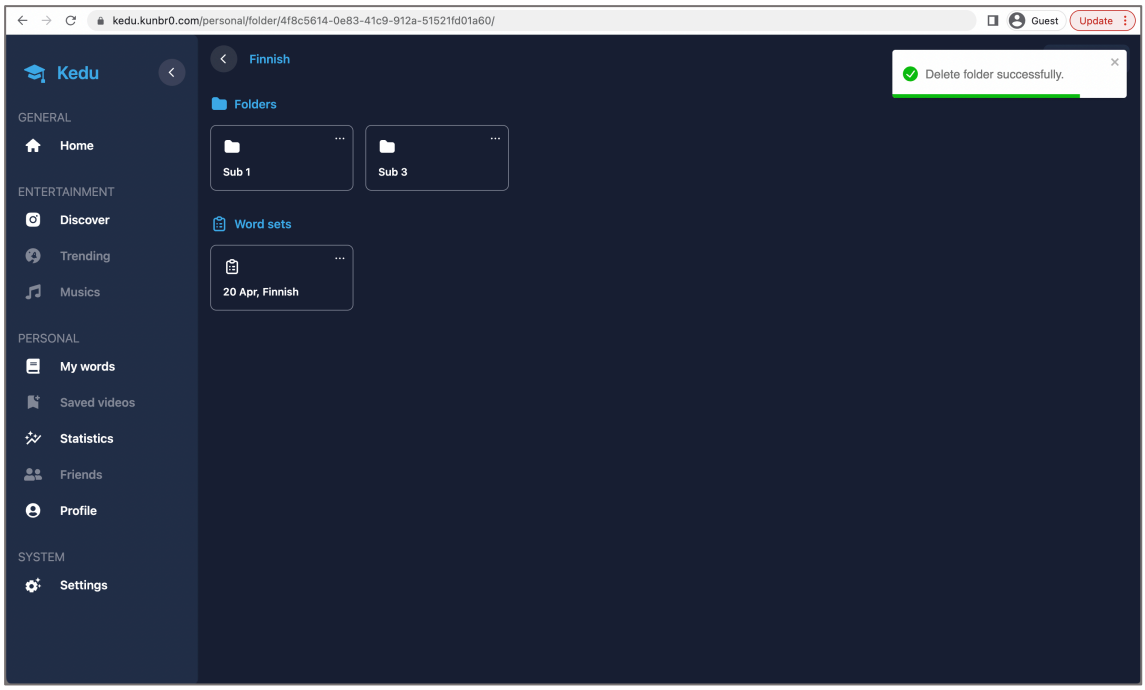


FIGURE 42: Finnish folder after deleting a folder

4.3 Managing and using word sets

A “word set” means a set of words. Users can create a word set in any folder or sub-folder they want. For example, they have a Finnish lesson today, and they want to create a wordset named “23 Apr, Finnish” to store all the vocabulary they have learned in the classroom for reviewing and practicing later (Figure 43).

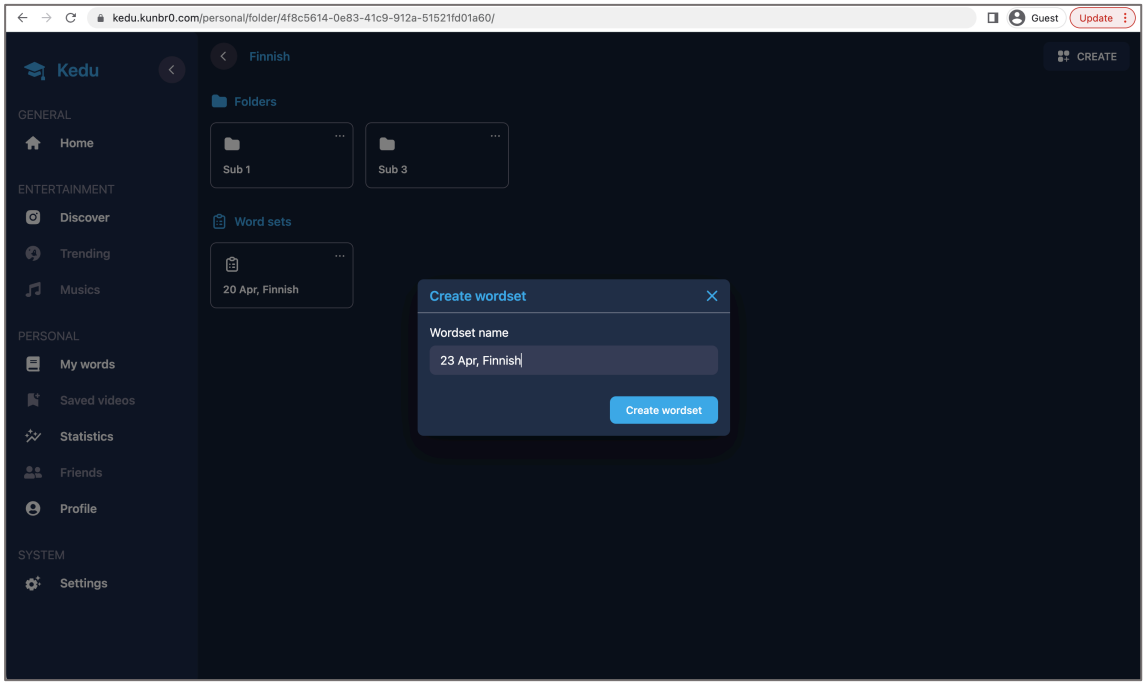


FIGURE 43: Naming and creating a word set

After successfully creating the “23 Apr, Finnish” word set, it will appear in the current “Finnish” folder (Figure 44). Users can access the word set management page by clicking on it.

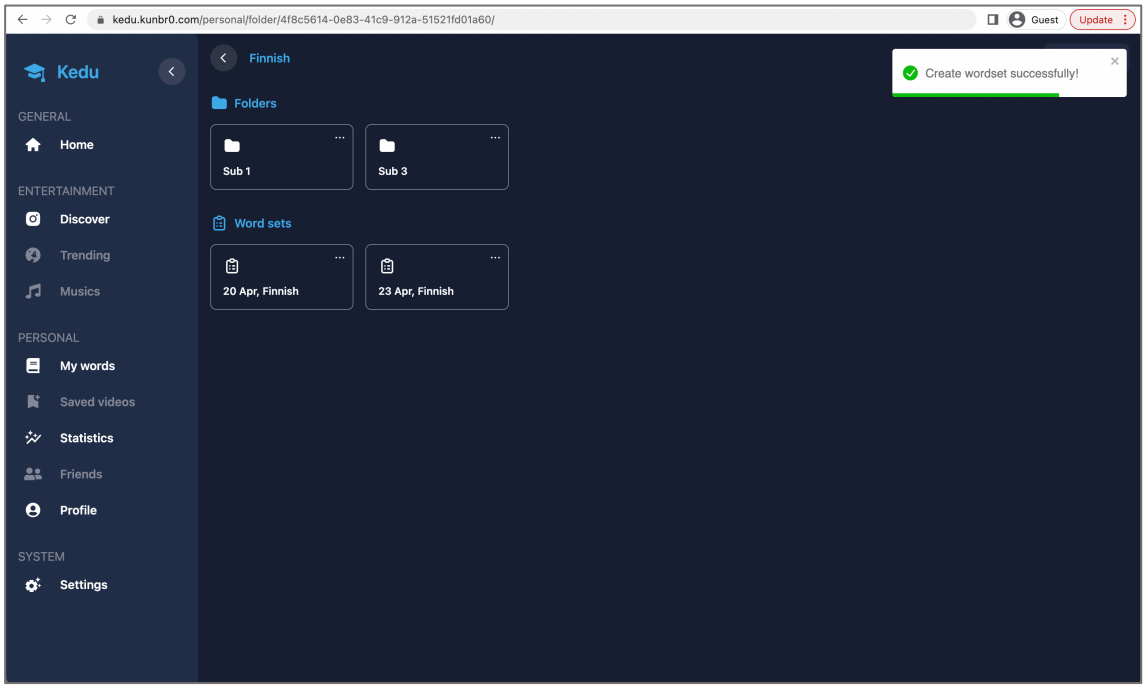


FIGURE 44: Folder after successfully creating a word set

Currently, the word set is empty because it was just created. Users can add items or vocabulary by clicking the EDIT button in the upper right corner of the screen (Figure 45).

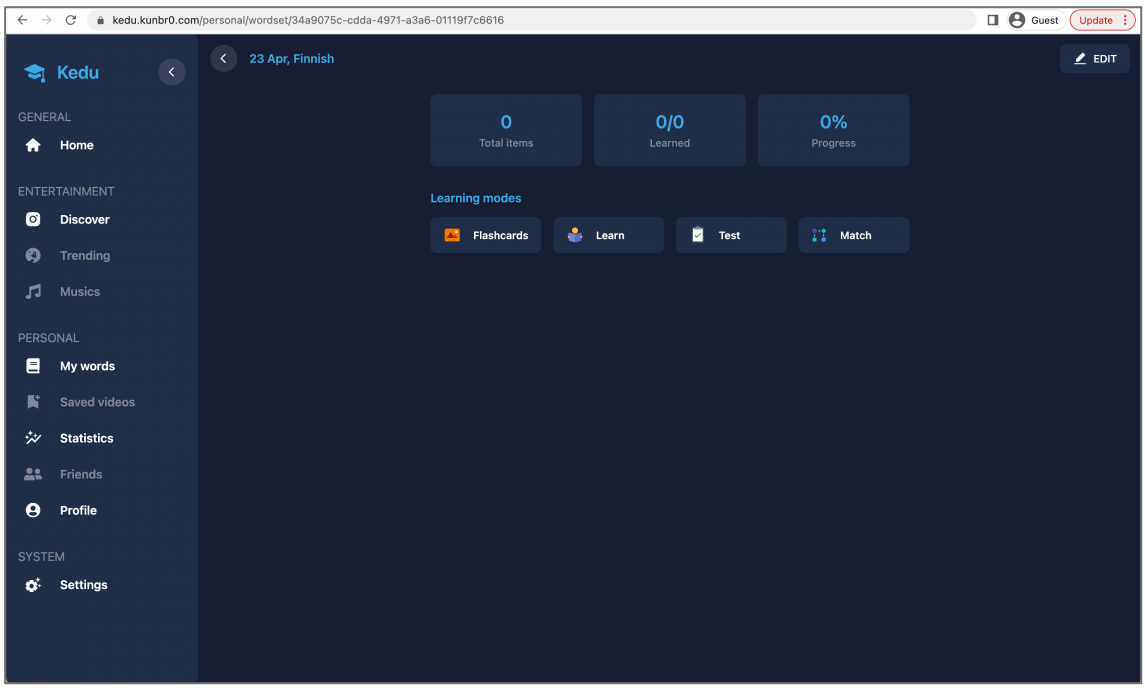


FIGURE 45: Word set management page

On the Edit a word set page, users can add a new word by clicking the Add Item button, selecting the language pair of the word, and filling in the content of the word. Alternatively, they can delete an existing vocabulary by clicking the X button in the upper right corner of each word (Figure 46).

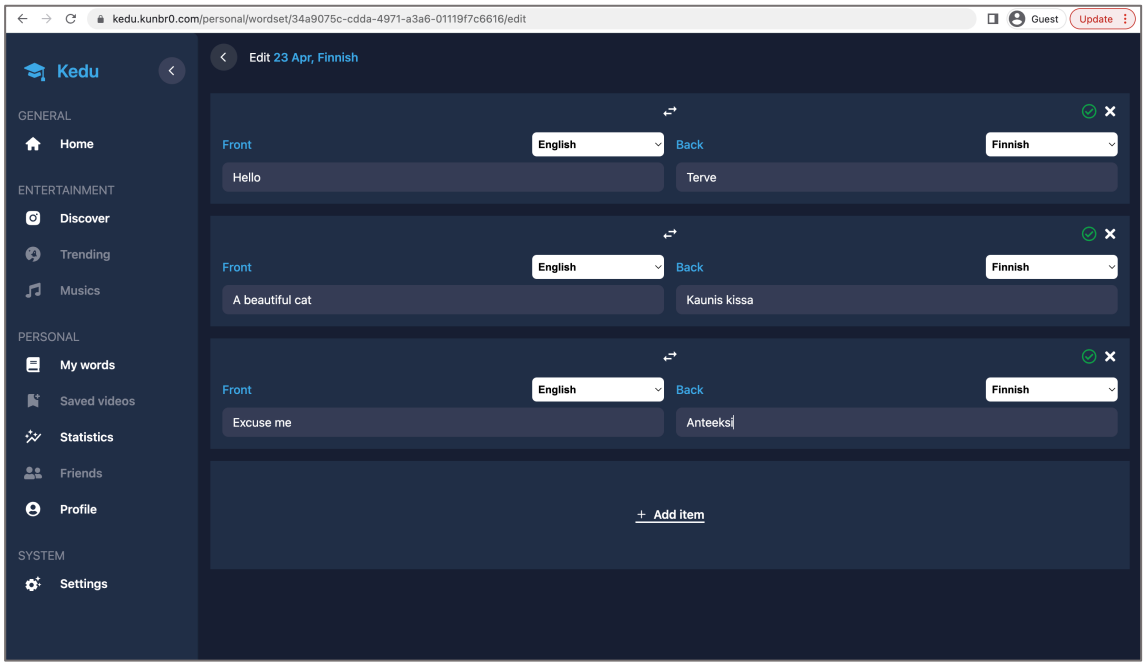


FIGURE 46: Edit a word set page

After adding some items to the word set, the user can click the back icon in the upper left corner to return to the word set management page. It can be seen in Figure 47 that some statistics of the word set have changed compared to Figure 45 after adding some new items to this word set.

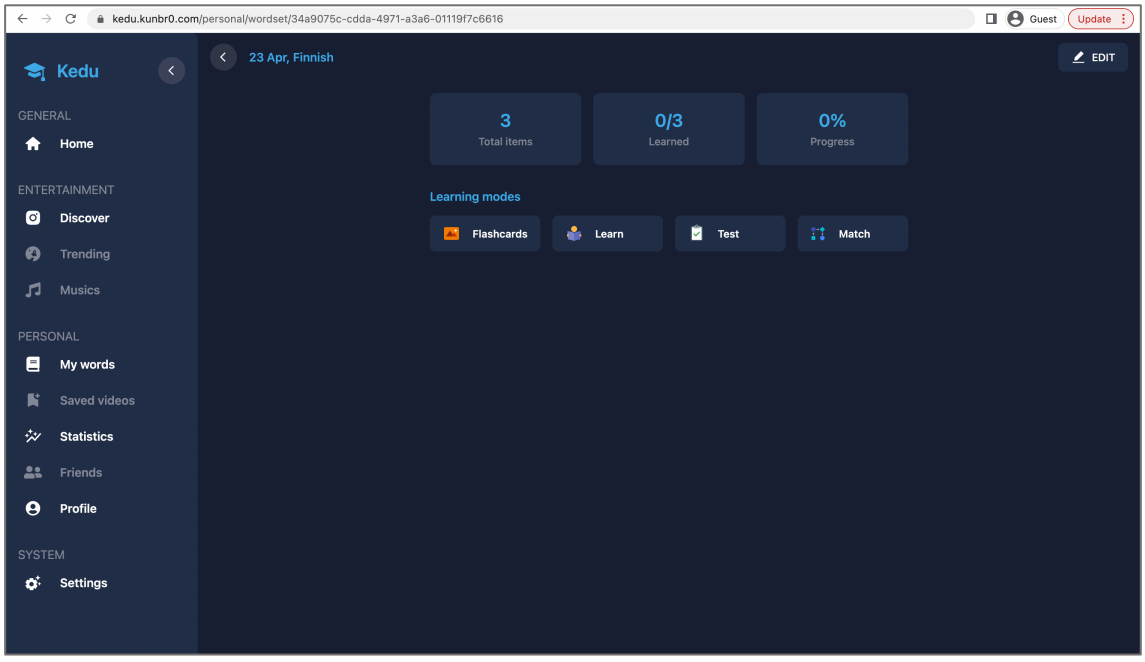


FIGURE 47: Word set management page after adding some items

4.4 Practicing vocabulary by using flashcards

To access this feature, users can click on the Flashcards button on the word set management page. Flashcards mode is a feature that helps users review and practice vocabulary effectively by memorizing words, their meaning, as well as how they are pronounced. A flashcard includes two sides, the front side (Figure 48) and the back side (Figure 49). The audio of the text will be automatically played when switching or flipping flashcards thanks to the Text-to-speech module (Section 3.4.4).

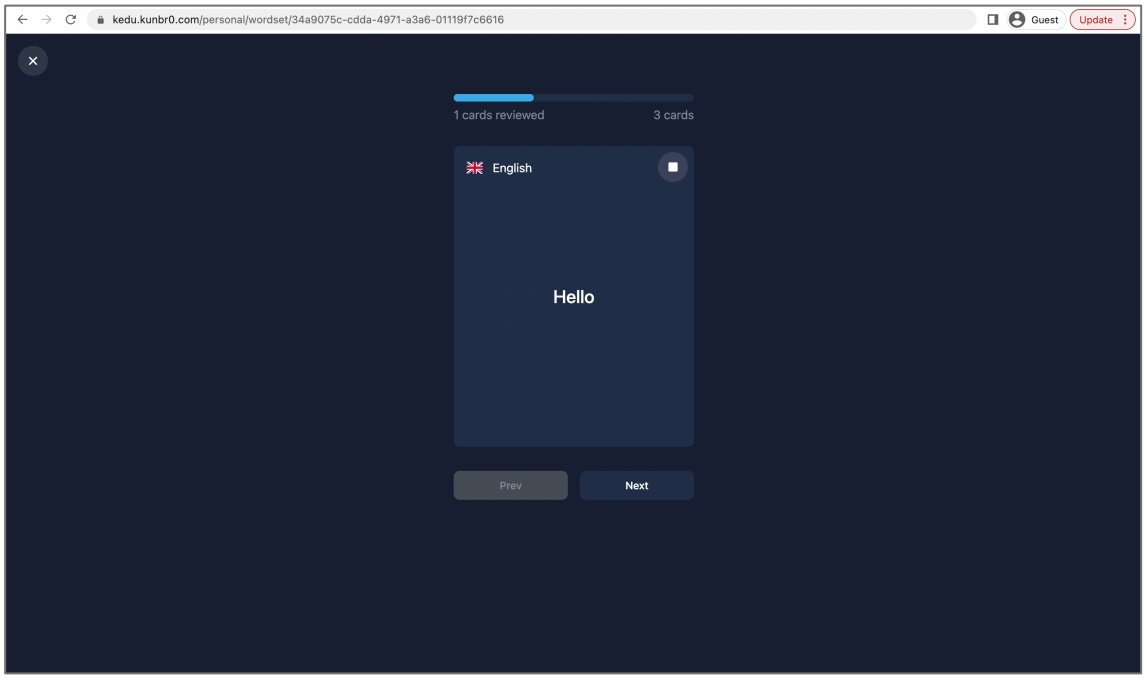


FIGURE 48: Front side of a flashcard

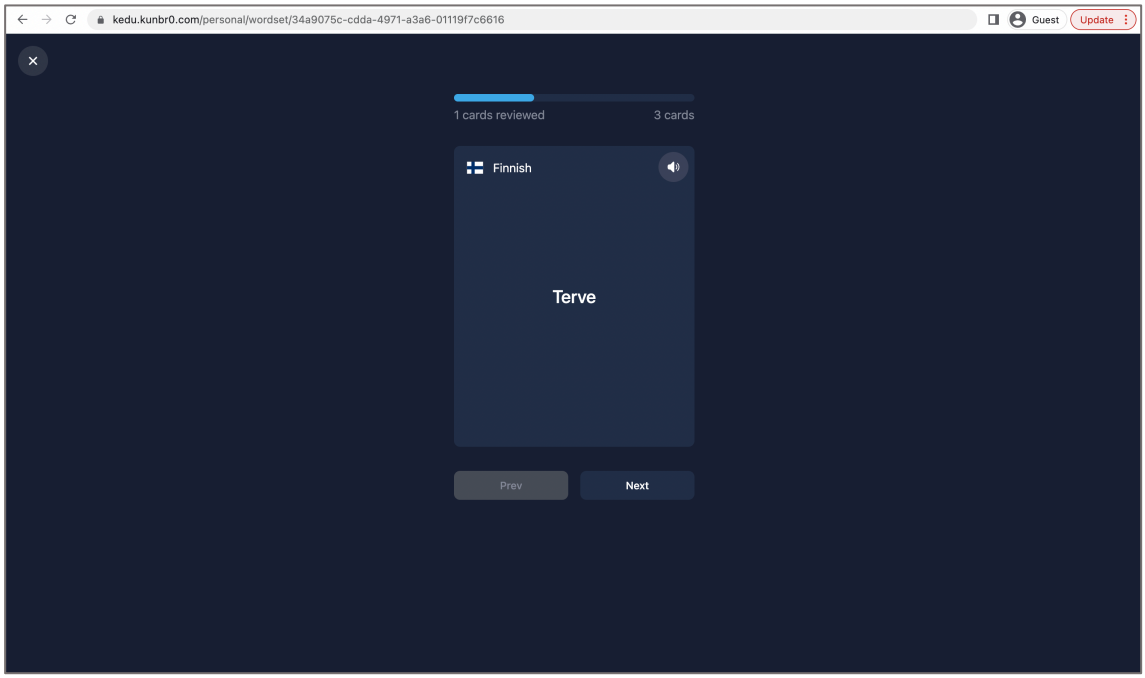


FIGURE 49: Back side of a flashcard

4.5 Learning foreign languages by watching videos

This is a feature that helps users learn foreign languages by watching videos. Users can access it by clicking the Discover item in the left sidebar menu. On this screen, users can search for any video on YouTube according to their interests thanks to the YouTube module (Section 3.3.6). Figure 50 is an example of searching YouTube videos with the keyword "ted".

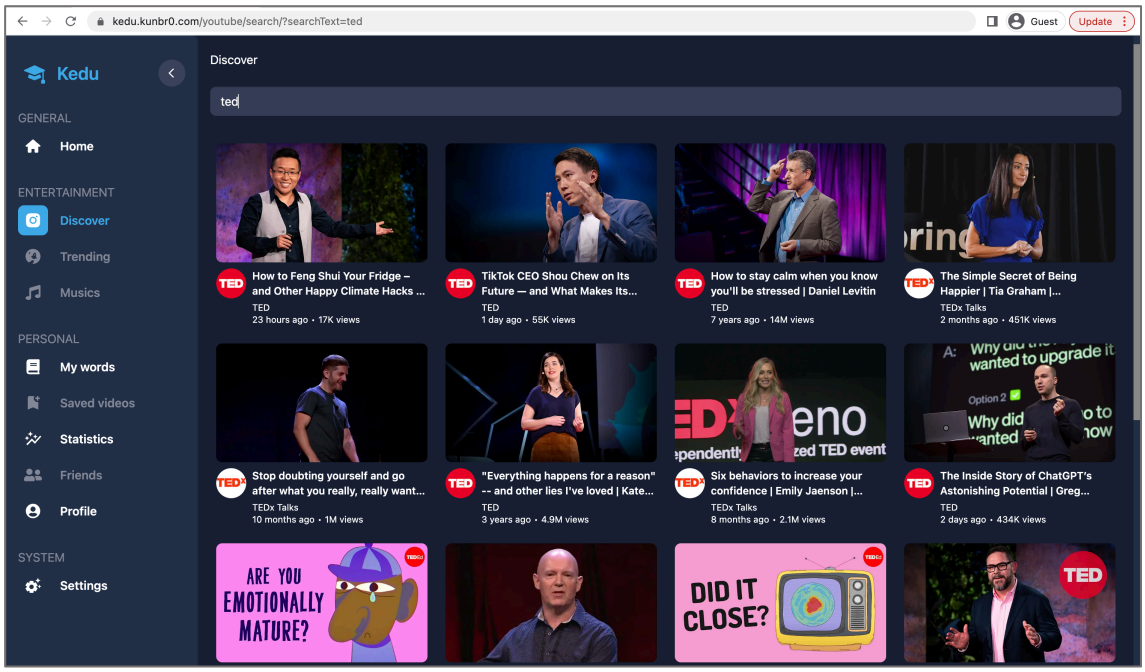


FIGURE 50: Videos Discovering page

After finding and choosing a favorite video, users can watch a video by clicking on it to navigate to the video watching page (Figure 51).

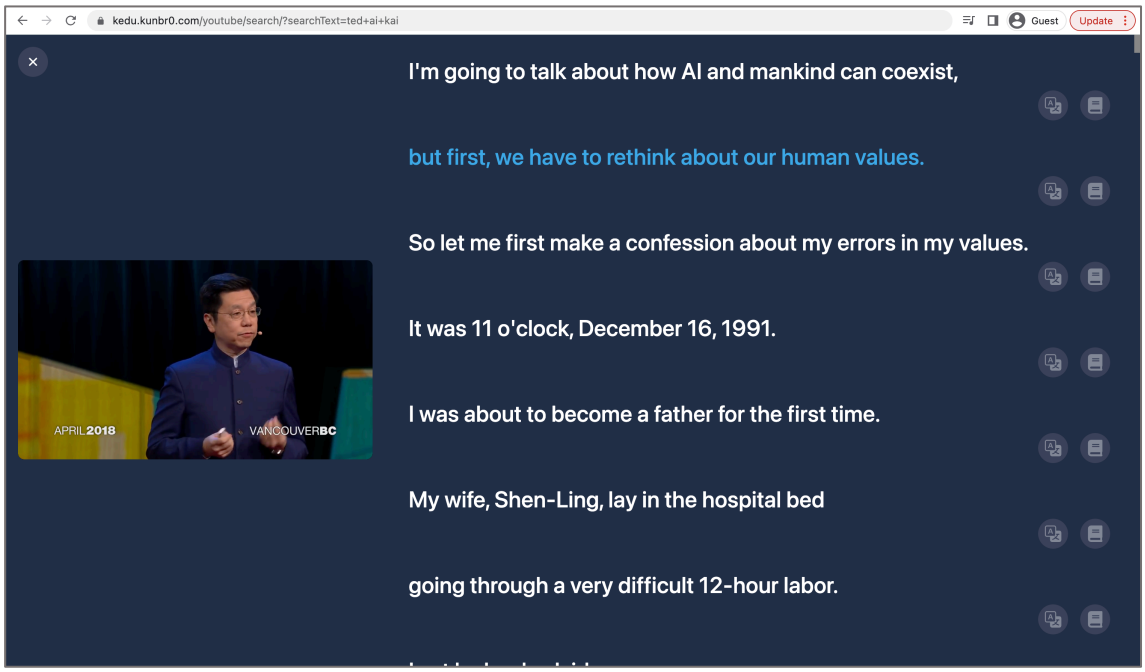


FIGURE 51: Video watching page on Kedu

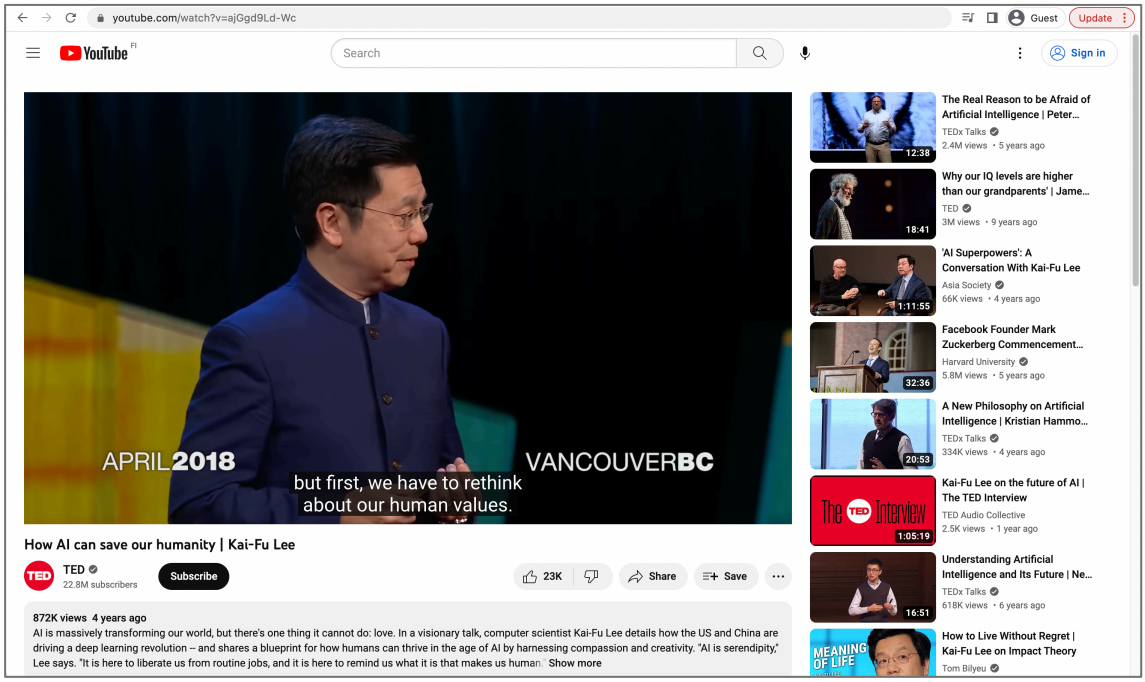


FIGURE 52: Video watching page on Youtube

The difference between the two video watching pages on Kedu (Figure 51) and Youtube (Figure 52) in learning foreign languages will be illustrated in Table 3 below.

TABLE 3: Comparisons between watching a video on Kedu and Youtube

Kedu	Youtube
Small video player area	Big video player area
Big area for displaying the caption	Small area for displaying the caption, display in single sentences
Easily following the previous and next sentences of the caption => helps to understand the entire meaning of the video.	New sentences come too fast, leading users easily forget previous sentences => Hard to follow and understand the video.
Focused on reading and listening	Focused on watching video
Very friendly for learning languages	Not effective for learning languages

It is clear to see the difference and effectiveness of learning foreign languages between watching videos on Kedu and Youtube in listening and reading, which are two extremely important skills for any foreign language learner.

Furthermore, users can easily translate any sentence from the caption that they do not understand while watching a video by clicking on the translate icon to the right of each sentence (Figure 53). This makes the learning process more convenient and saves time compared to using another website or application (Google Translate) for translation purposes.

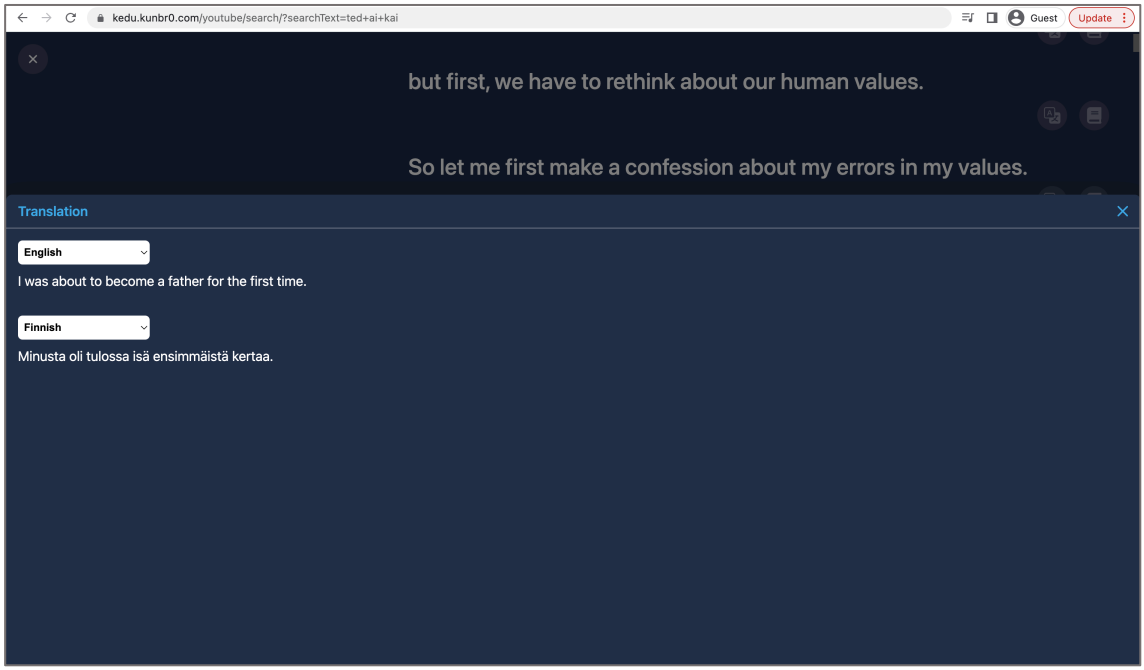


FIGURE 53: Translate a sentence while watching a video

Last but not least, when seeing a favorite sentence while watching a video, users can easily save it to existing word sets for reviewing and practicing later by clicking on the book icon to the right of each sentence (Figures 54, 55, and 56).

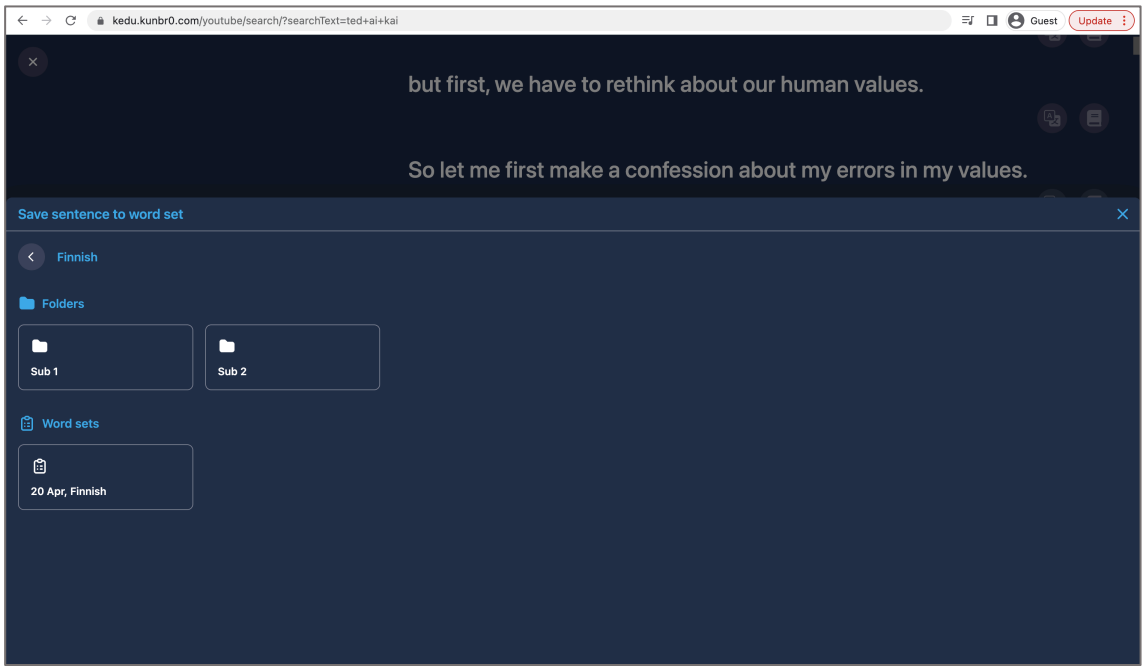


FIGURE 54: Selecting a word set for saving a sentence

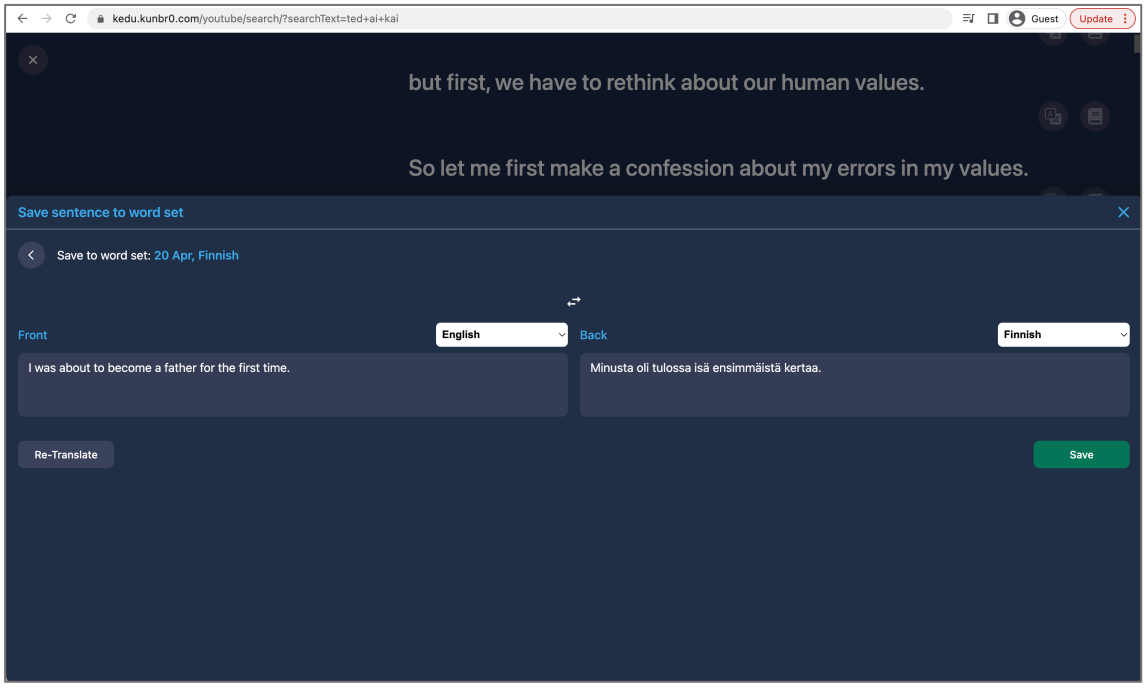


FIGURE 55: Reviewing the sentences before saving to the word set

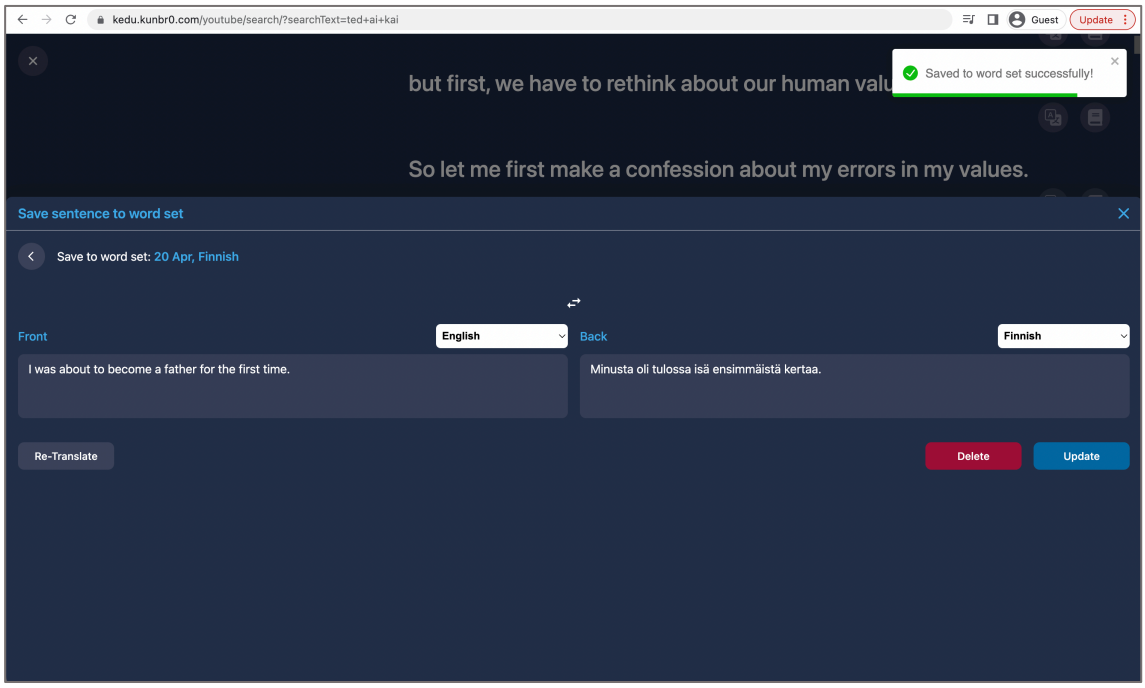


FIGURE 56: Saved the sentence to the word set successfully

In addition, users can also update or delete a saved item after saving it to a word set by clicking on the “Delete” or “Update” button (Figure 56).

4.6 Personal information and statistics

In Figure 57, users can view and update their account information or change their password on the profile page.

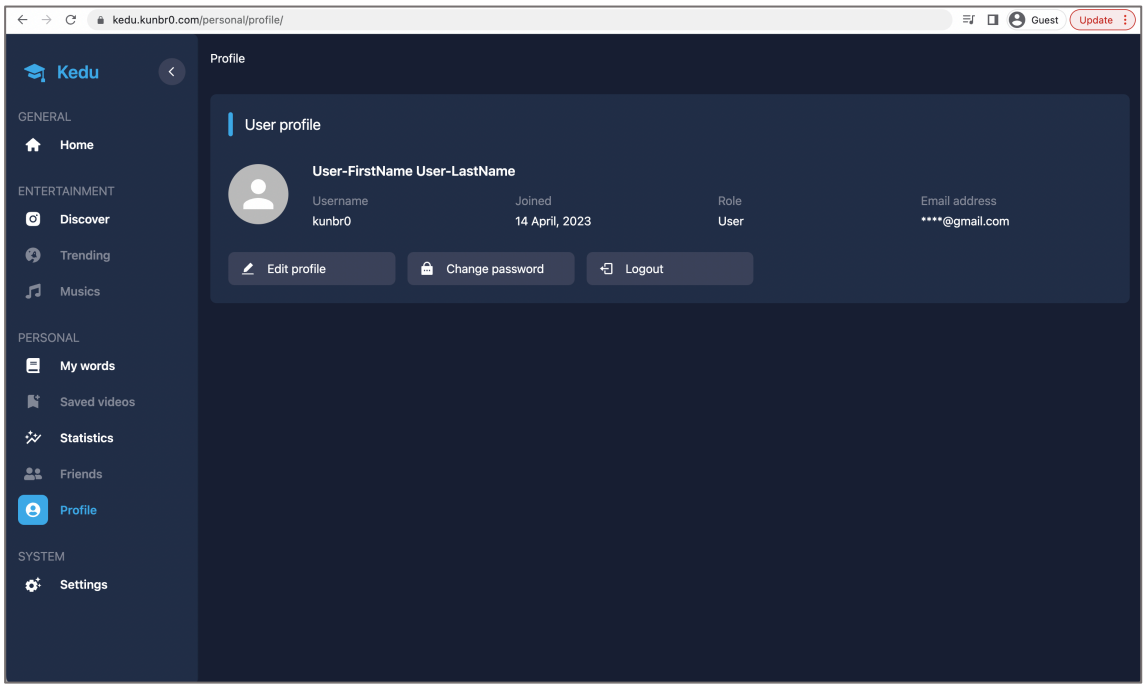


FIGURE 57: Profile page

In addition, users can also view summaries and statistics about their learning process on the statistics page (Figure 58).

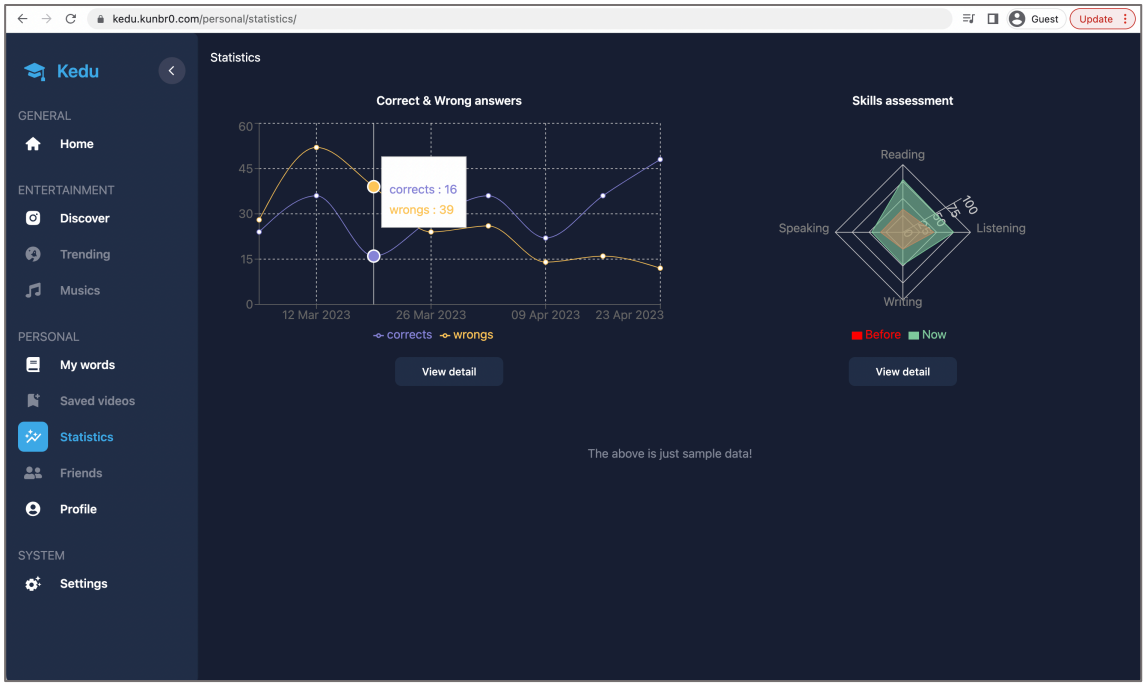


FIGURE 58: Statistics page

4.7 Customization

This is a feature that allows users to freely customize the system according to their preferences to make learning more interesting. By clicking on the Settings item in the left sidebar menu, users can access the Theme settings popup, where they can choose their own favorite background color and main color tone of the website (Figure 59).

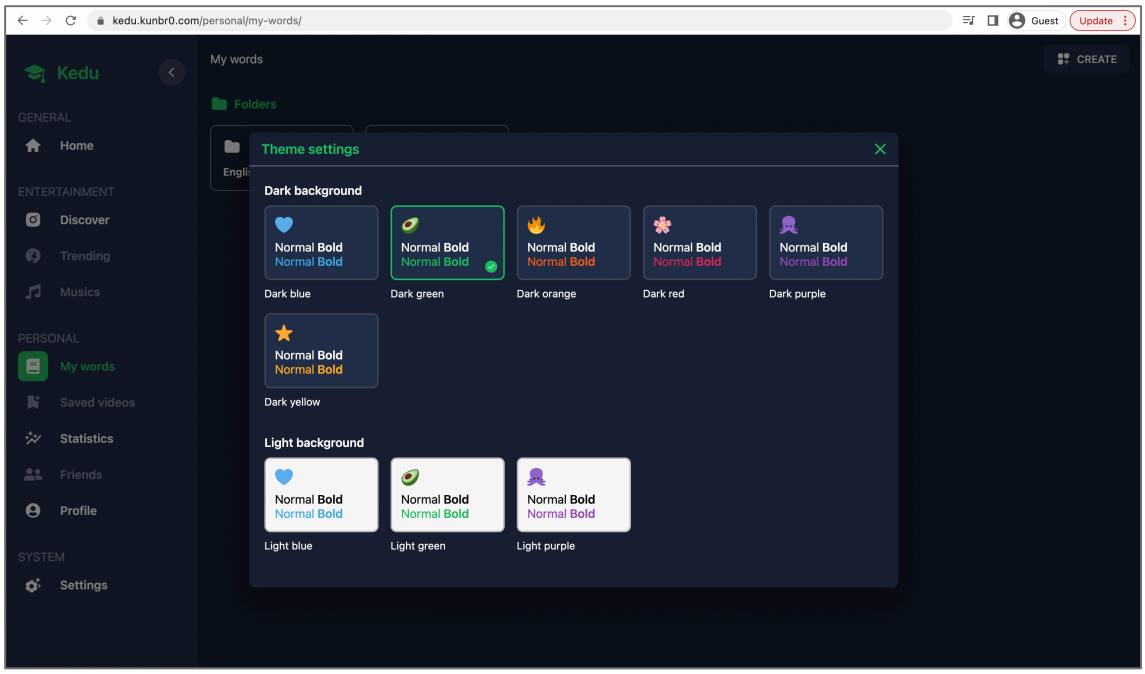


FIGURE 59: Theme settings popup

Changes to the theme background and font color are applied immediately to all screens and components of the website (Figure 60).

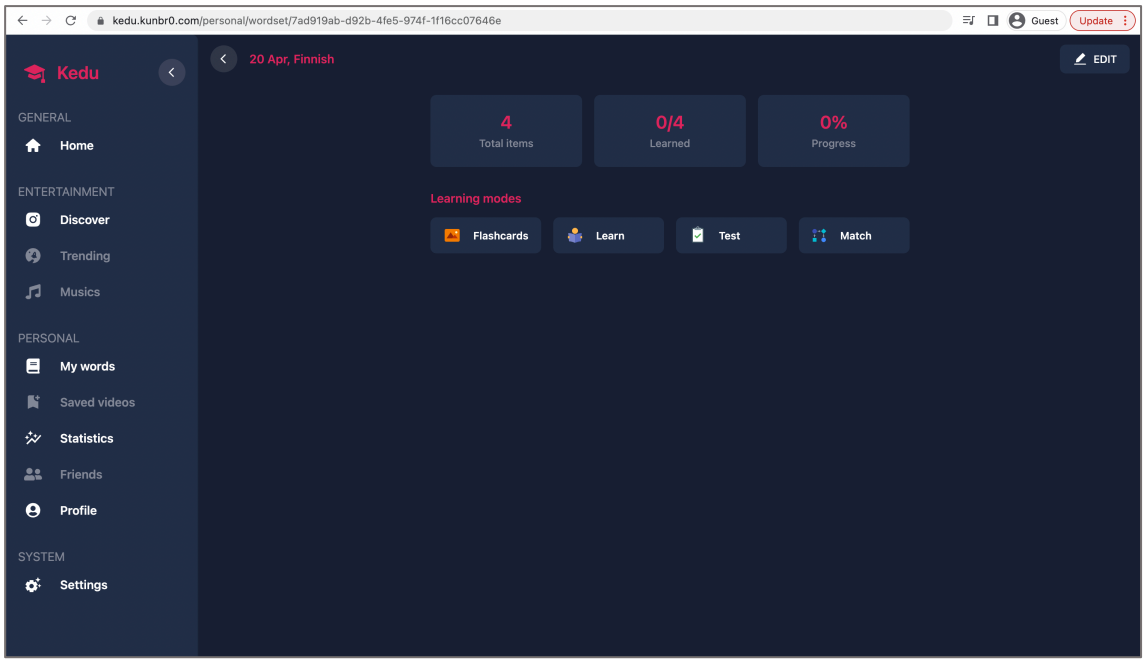


FIGURE 60: User interface after changing theme color (Red)

5 CONCLUSION

This thesis explored the effectiveness of various foreign language learning methods and strategies, pedagogy, as well as criteria for evaluating educational and user-friendly websites. It also covered how full-stack technologies and the use of cloud services from AWS can be applied to develop a software product based on the serverless architecture.

The main product of this thesis, Kedu, was designed with the goal of facilitating language self-studying, including vocabulary practice with flashcards, improving reading and listening skills by watching videos, and many useful built-in features such as translation, word and sentence pronunciation, and so on, making learning foreign languages easier and more convenient. In addition, software product quality, user interface (UI), and user experience (UX) were also concentrated during the deevelopment process of Kedu. The website had a clean, intuitive, and user-friendly interface; it could be responsive and operate smoothly on many screen sizes, from mobile to tablet and desktop.

Although Kedu has not been widely tested and used, initial reviews from acquaintances and a presentation to software engineering students and teachers at the university have been positive, with a lot of constructive feedback to improve Kedu in the future. Suggestions include features such as importing or exporting vocabulary from or to CSV and Excel files and adding more learning modes and games. Moreover, learning foreign languages through reading newspapers from public sources such as BBC News, Yle, and more, is also a wonderful feature that can be considered and applied in the future.

REFERENCES

Lin, Phoebe. 2019. “Vocabulary Learning in the Age of Internet Television.” *54th RELC International Conference and 5th Asia-Pacific LSP and Professional Communication Association Conference.* Singapore: Phoebe Lin.

Lin, Phoebe. 2022. “Developing an intelligent tool for computer-assisted formulaic language learning from YouTube videos.” Accessed 02 2023. https://doi.org/10.1017/S0958344021000252.

Komachali, Maryam, and Mohammadreza Khodareza. 2012. “The Effect of Using Vocabulary Flash Card on Iranian Pre-University Students’ Vocabulary Knowledge.” Accessed 02 2023. http://dx.doi.org/10.5539/ies.v5n3p134.

Spiri, J. 2008. “Online study of frequency list vocabulary with the Word Champ website.” *English Language Teaching* 7(1), 21-36.

Hung, Hsiu-Ting. 2015. “Intentional Vocabulary Learning Using Digital Flashcards.” Accessed 02 2023. https://doi.org/10.5539/elt.v8n10p107.

Serverless. n.d. *Serverless: Develop & Monitor Apps On AWS Lambda.* Accessed 02 2023. https://www.serverless.com/.

Serverlessland. n.d. *Serverless Land | Resources for learning about AWS serverless technology.* Accessed 02 2023. https://serverlessland.com/ .

Amazon. n.d. *Cloud Computing Services - Amazon Web Services (AWS).* Accessed 02 2023. https://aws.amazon.com/.

AllCode. n.d. *Top AWS Services List 2023.* Accessed 02 2023. https://allcode.com/top-aws-services/.

Okta. 2023. 10 04. https://www.okta.com/identity-101/serverless-computing/.

Stackoverflow. 2022. “Stack Overflow Developer Survey 2022.” Accessed 04 2023. https://survey.stackoverflow.co/2022/.

Hexaware Technologies Limited. 2020. *Serverless Computing, the mantra for business transformation and efficiency.* 04. Accessed 02 2023. https://hexaware.com/blogs/serverless-computing-the-mantra-for-business-transformation-and-efficiency/.

SolGuruz. 2022. *Top 10 ReactJS benefits for your web project.* 11. Accessed 02 2023. https://solguruz.com/blog/top-10-reactjs-benefits/.

n.d. “Codes for the Representation of Names of Languages.” *Library of Confgress.* Accessed 04 2023. https://www.loc.gov/standards/iso639-2/php/code_list.php.

M. Jones, Microsoft, J. Bradley, Ping Identity, N. Sakimura, NRI. 2015. “RFC 7519 - JSON Web Token (JWT).” Accessed 04 2023. https://datatracker.ietf.org/doc/html/rfc7519.