TOEFL-Modder

Group 9 沈皓天，翁歌华，蔡贺丞，胡烁怡，姜林楚

# Background

Mr. Cai, a student at Westlake University, is facing the TOEFL exam and feeling overwhelmed by the lack of effective English learning resources. Existing resources are limited, filled with advertisements, and lack a comprehensive approach. As his friends, we want to create an interactive English learning software that is tailored to Westlake University students, helping them improve their English skills anytime, anywhere. Our goal is to support Mr. Cai and other students in conquering the challenges of the TOEFL exam while uncovering its true nature.

# Software Design

TOEFL-Modder is built using the most popular PC software development technologies and frameworks: - Electron: A framework that allows developers to build cross-platform desktop applications

- Vue: A progressive JavaScript framework that simplifies the development of user interfaces

- Vite: A build tool for web applications that offers fast development and optimized production builds

- Naive UI: A Vue-based UI library that provides a set of customizable components for building desktop applications

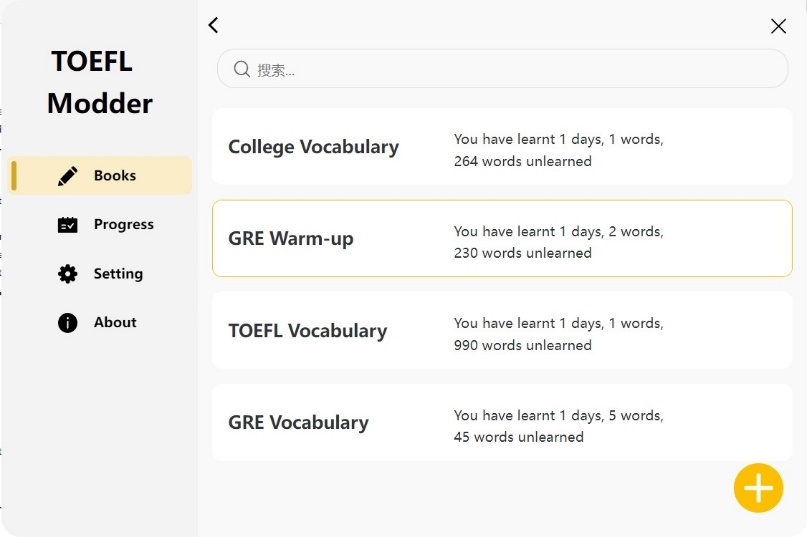
- SQLite: A lightweight, serverless, self- contained, and embeddable database engine

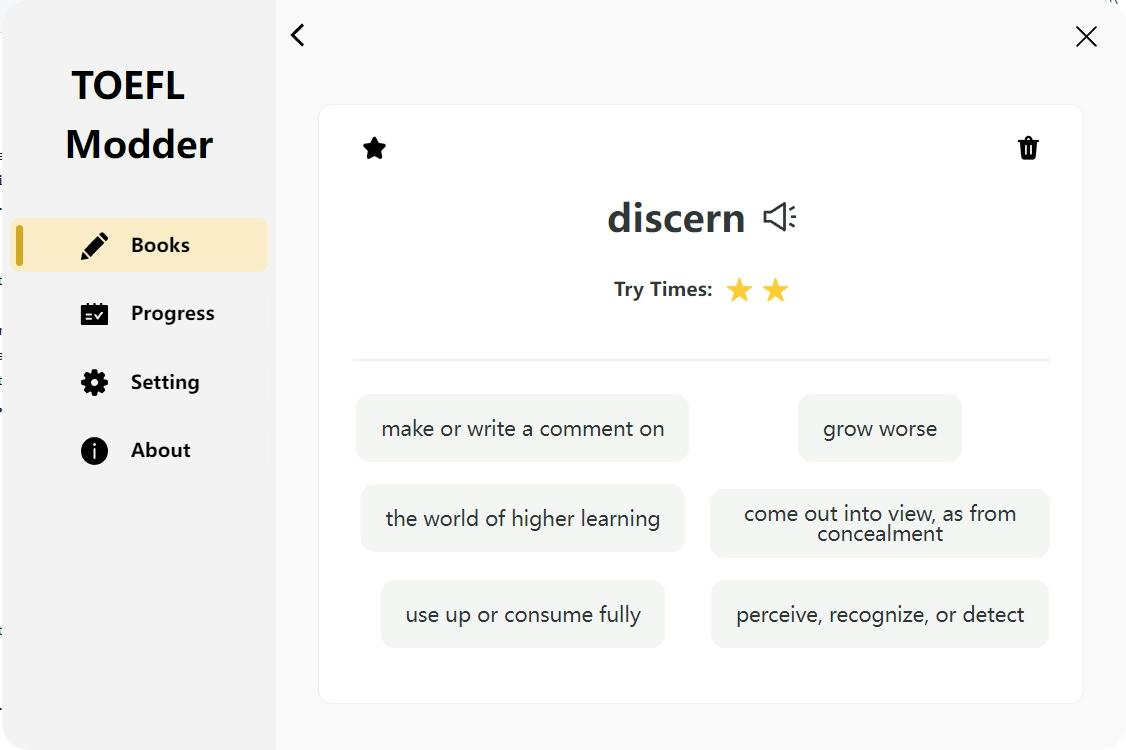
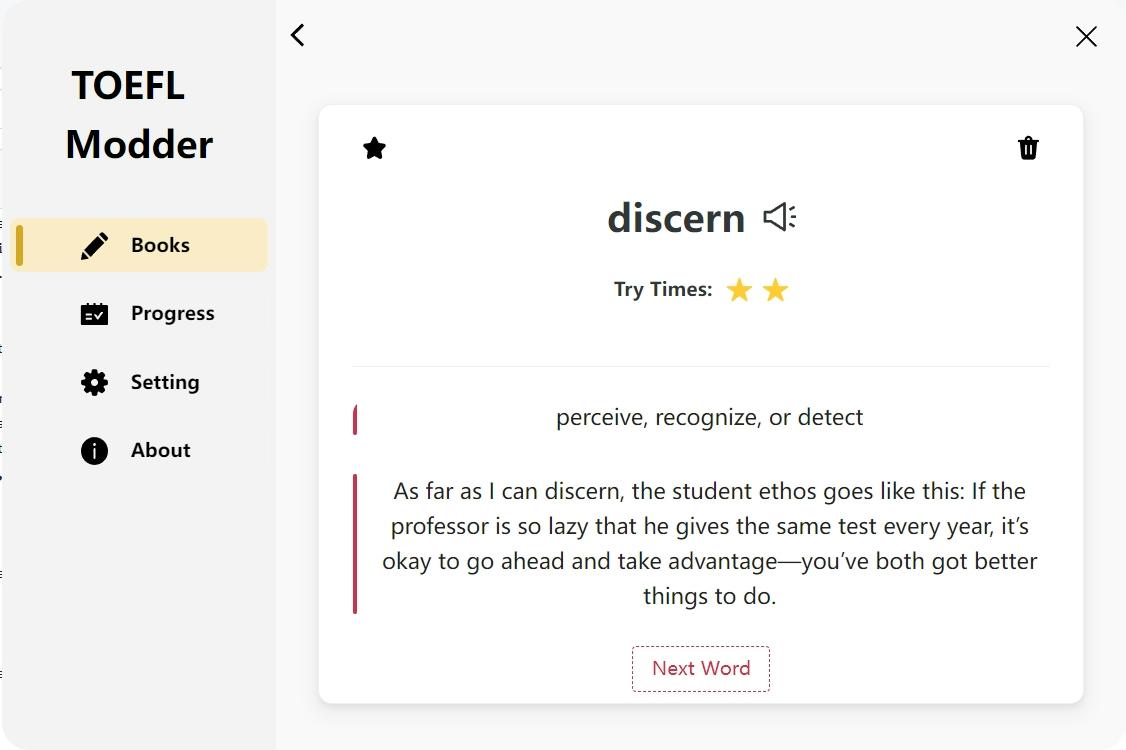
# Requirements

Github: https://github.com/Tesla-SHT/TOEFL-Modder

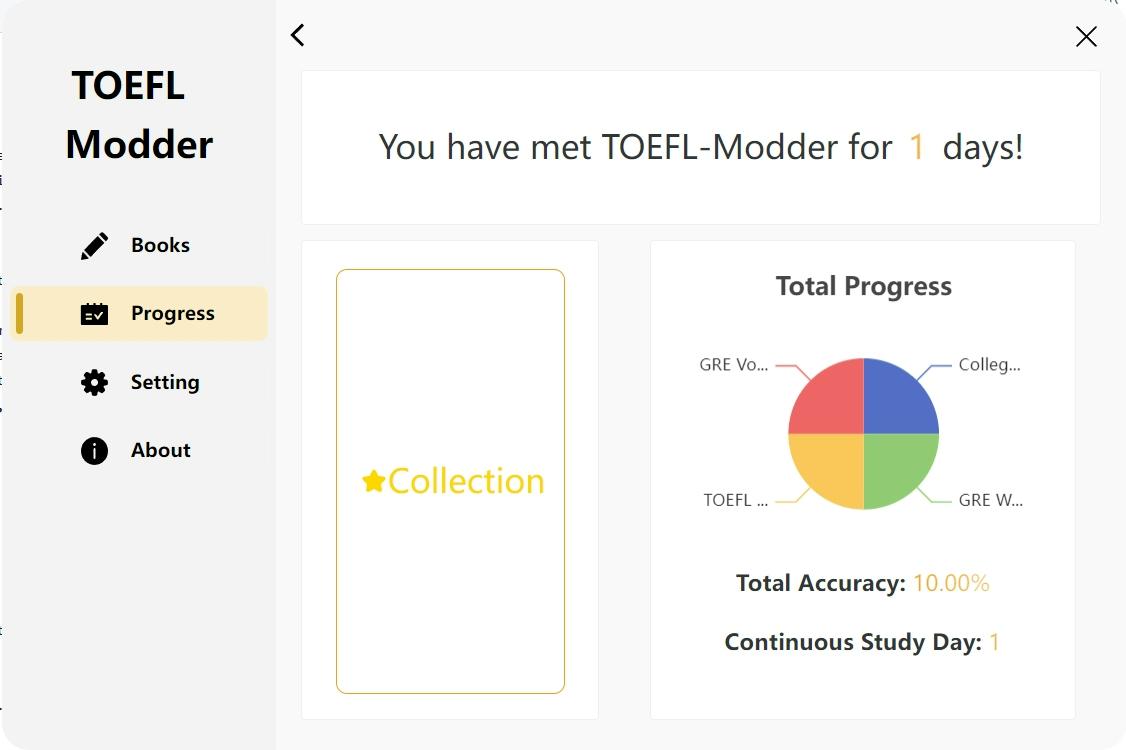
1. Quick Start
   1. Just install our released installation package
2. Used for further development
   1. Clone the repository
   2. Download Nodejs
   3. npm install
   4. npm run dev
   5. npm run start (open in another cmd window)

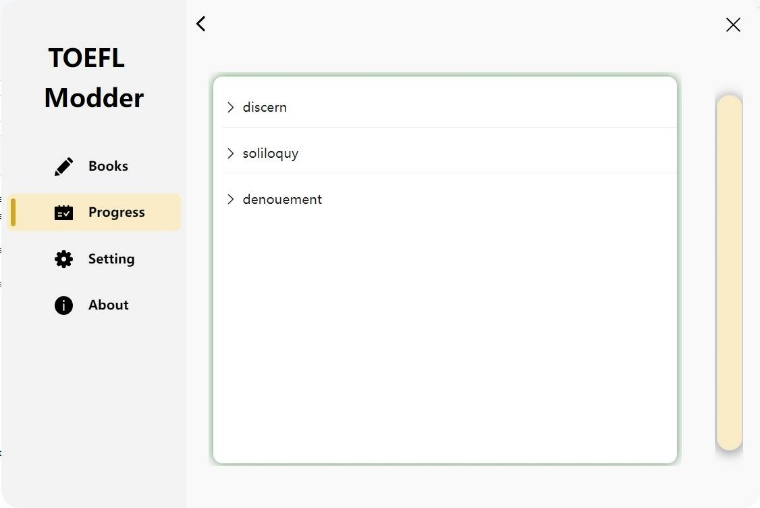
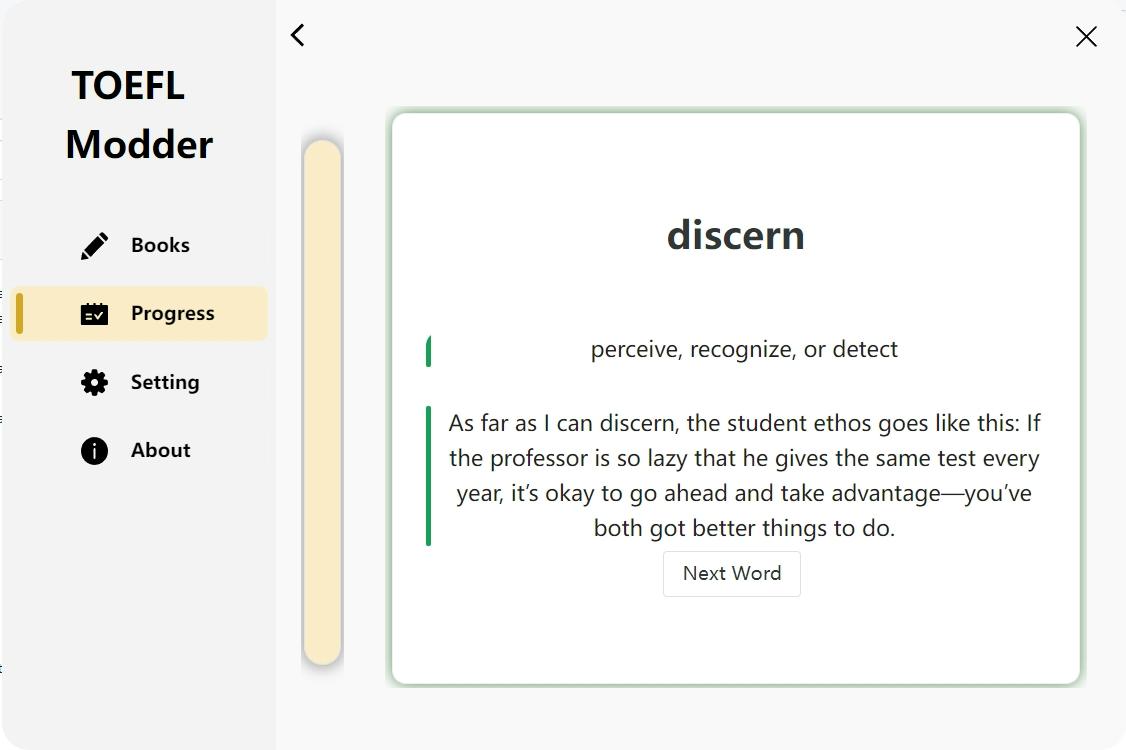
# Description

1. **Books**: This is the central page of our software, in which users could select the vocabulary books that they want to learn by clicking the “+” button in the lower right place of the page. Here we already have four vocabulary books all set. And the user could input the books that they wish to work on. Till now, only csv. files are allowed to be imported with words on column one, explanations on column two and example sentences on column three.

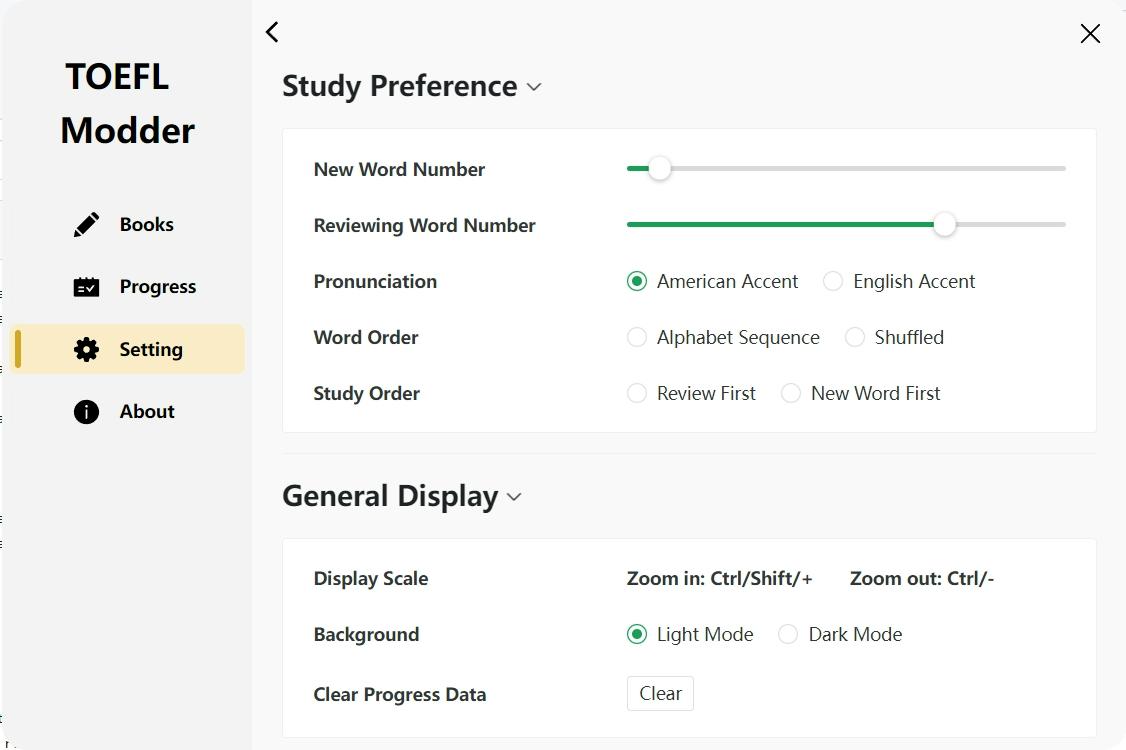
And the click action on each book would direct to the reciting page of each vocabulary. In which users would select the explanation of the word and get a response. Also, there were two more functions including adding the word into the user’s collection list (which could be further accessed on the Progress page) and removing the word from the whole book.

1. **Progress**: Display the learning data and some analysis:

Here you could see the progress that you made after you met our software, including the days you spent on each book, your accuracy in selecting the explanation of the word, and continuous study. And also, the word number that you recite each day.

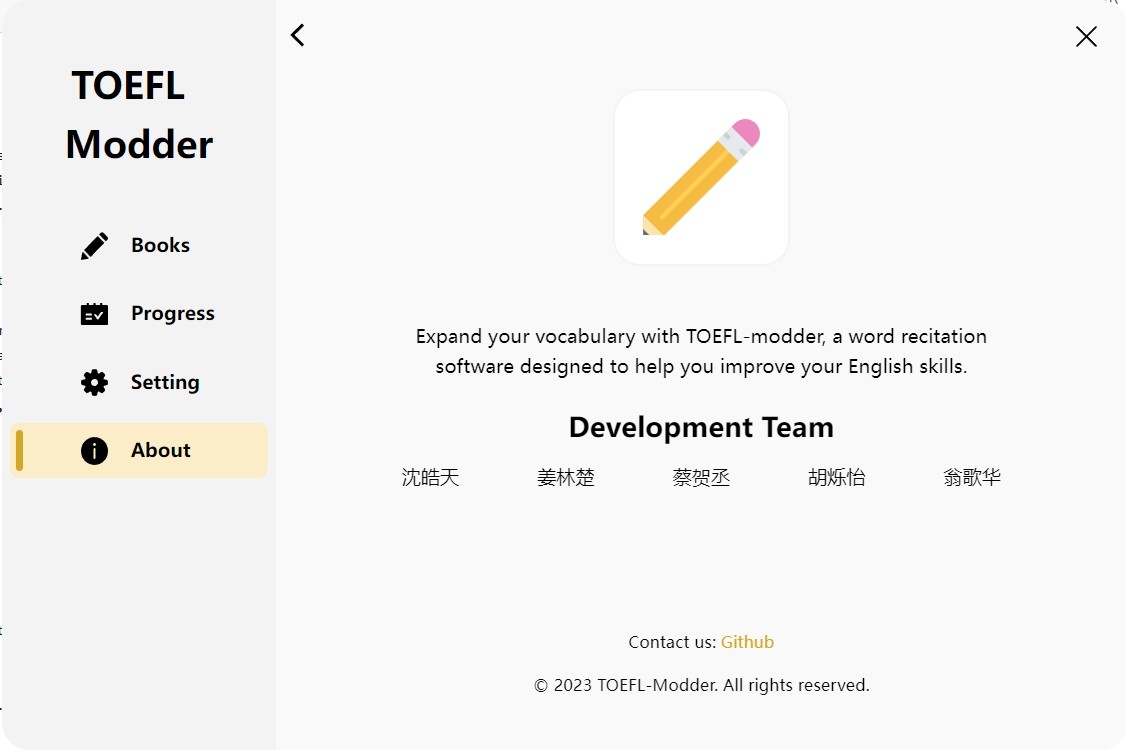
Collection button will direct to the collected word list. And in that page you can find the words in your collection list, and there also have one subsection of the page for you to review.

1. **Setting**: Divided into two parts: Study and General

In study section, you could personalize your choice of how many words you want to recite or review. Also, the pronunciation mode, word order and the study order of words could be selected.

In general display section, you could change the display scale and choose the background display mode. Also, the progress data could be cleared out if the user wishes to start again.

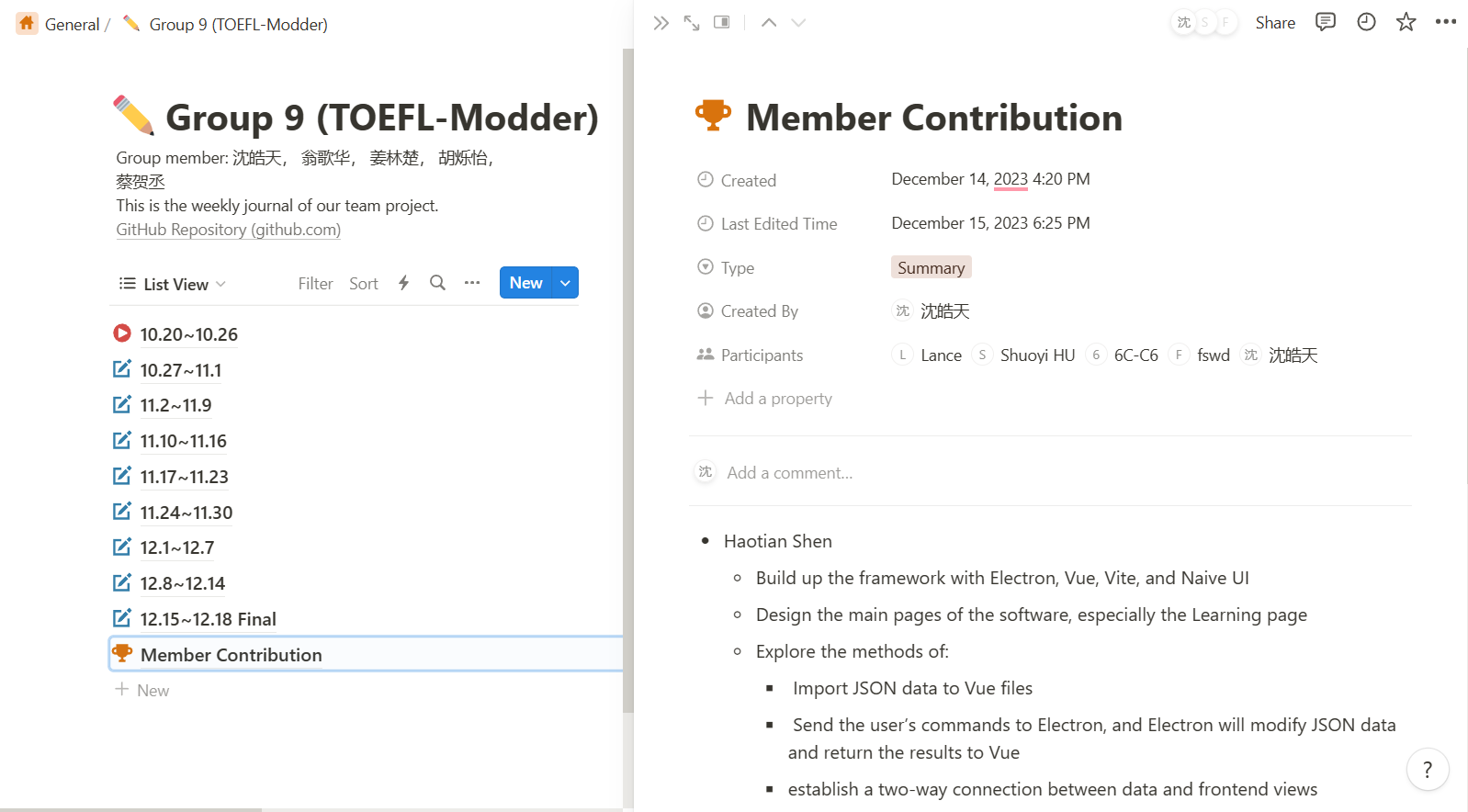
1. **About**: Basic information about our software



# Weekly Journal

For better inter-group communication, we created one notion page for recording the weekly progress. Since different people are in charge of different parts of the work, the potential blocks and bugs of the program are also recorded for more efficient debugging. And also, this notion page enabled us to check each person’s contribution to the project. Check this page out by the link below:

[Link to our journal in Notion](https://www.notion.so/77f99992c04b4985bba37d818bcd17b7?v=5ca4aee77cf440c0b3d5acf09f4c8a8a) (https://www.notion.so/77f99992c04b4985bba37d818bcd17b7?v=5ca4aee77cf440c0b3d5acf09f4c8a8a)



# Team Assignment

# Haotian Shen

# - Build up the framework with Electron, Vue, Vite, and Naive UI

# - Design the main pages of the software, especially the Learning page

# - Explore the methods of:

# - Import JSON data to Vue files

# - Send the user’s commands to Electron, and it will modify JSON data and return the results to Vue

# - Establish a two-way connection between data and frontend views

# Gehua Weng

# - Conducted market research from other apps

# - Design Progress page

# - Program the Collection page

# Linchu Jiang

# - Designed and implemented the database with SQLite

# - Package the software into an exe file

# Shuoyi Hu

# - Use a Python script to download dictionaries from a website

# - Design Setting page

# Hechen Cai

# - Program a Python file to transform excel files to JSON files

# - Design the reviewing mechanism

# - Debug and fix the program bugs

# Tips

1. How can users upload their own dictionaries?

First, the user should scratch a dictionary from the Internet and transform it to a csv file.

Second, the first row of the file should be Words, Definitions, Examples, and corresponding data should be list in three columns.

Finally, click “+” at the lower right of the books page and choose it!

1. What is our future plan for the database?

We can let users to sign up various accounts, and each account will have a set of json settings. So the database will manage these and overwrite the using json files with the specified user’s files.

1. Any comment or advice? Feel free to contact us!