

<Title of project placed here>

Tiansheng Chen

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

A dissertation presented in part fulfillment of the requirements of the Degree of Master of Science at the University of Glasgow

1 September 2023

**Abstract**

<Abstract goes here…>

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic form.

<**Please note that you are under no obligation to sign this declaration, but doing so would help future students.>**

Name: Tiansheng Chen Signature: Tiansheng Chen

Acknowledgements

<Acknowledgements go here>

Contents

<Update the table of contents by right-clicking on it and selecting Update Field… and then select page numbers only.>

Chapter 1 Introduction <This is Heading 1> 1

1.1 Aims and Objectives 1

1.2 Report Structure 1

Chapter 2 Analysis and Requirements 2

2.1 Background and Literature Review 2

2.2 Requirements Gathering 3

2.2.1 Analysis of Existing Applications 3

2.2.2 Interviews 5

2.3 Application Requirements 6

2.3.1 Functional Requirements 6

2.3.2 Non-Functional Requirements 7

Chapter 3 Design and Implementation 9

3.1 System Architecture 9

3.2 User Interface 11

3.2.1 Colour, Font and Style 11

3.2.2 Components and Pages 11

3.3 Database Design 15

3.4 Function Implementation 16

3.4.1 Structured Learning Courses 16

3.4.2 Study Plan System 16

3.4.3 Review System with Spaced Repetition Strategy 16

3.4.4 Chatbot 16

3.4.5 Security and Authentication 21

3.4.6 User Experience and Error Handling 21

Chapter 4 Testing and Evaluation 22

Chapter 5 Conclusion 23

Appendix A <Name of appendix> 1

Appendix B <Another appendix> 2

# Introduction <This is Heading 1>

Introduce the project.

## Aims and Objectives

Please note your dissertation need not follow the included section headings – this is only a suggested structure. Also add subsections etc. as required.

## Report Structure

Try to avoid this too much, but it’s here if you need it.

# Analysis and Requirements

## Background and Literature Review

Since its inaugural release on November 30, 2022, ChatGPT amassed over a million subscribers within its first week. This generative AI tool dazzled many with its sophisticated ability to undertake intricate tasks, especially in the realm of EFL (English as a Foreign Language) education. Such extraordinary abilities have evoked a spectrum of reactions from educators, as it transforms and redefines traditional pedagogical practices (Klyshbekova, 2023). Benefits attributed to ChatGPT in EFL encompass personalized and interactive learning, the generation of adaptive learning materials, provision of continuous and immediate feedback, fostering cultural understanding, and its 24/7 availability (Kohnke, Moorhouse and Zou, 2023). For instance, ChatGPT has demonstrated proficiency in generating dialogues tailored for students similar to the Common European Framework of Reference for Languages (CEFR) levels A and B1 (Young and Shishido, 2023). Adaptations of ChatGPT have effectively performed the role of an EFL teacher, assisting students with their writing tasks in regions like South Korea and Indonesia (Fitria, 2023; Han *et al.*, 2023). Moreover, EFL instructors in Vietnam acknowledged ChatGPT's potential in generating teaching materials and structuring EFL courses (Nguyen, 2023). It has been established that ChatGPT is adept at detecting grammatical errors, assisting Japanese EFL students in grammar acquisition (Schmidt-Fajlik, 2023). Remarkably, ChatGPT has been successfully deployed as a reading comprehension exercise generation system, offering qualified and individualized reading materials for middle-school English students in China (Xiao *et al.*, 2023). Consequently, such advantages have amplified the motivation among EFL learners to hone their reading and writing skills (Ali *et al.*, 2023). Yet, the technology is not devoid of limitations.

While ChatGPT's foundation is predominantly text-based, its inability to address speaking and listening skills emerges as a palpable drawback. Although EFL students generally exhibit a positive motivation towards reading and writing exercises facilitated by ChatGPT, their sentiments regarding its efficacy in nurturing listening and speaking proficiencies remain ambivalent(Ali *et al.*, 2023). It is recommended to conceive a comprehensive foreign language learning tool, which includes Augmented Reality, Voicebots, and ChatGPT technologies, thereby delivering a unique experience for EFL learners (Topsakal and Topsakal, 2022). Consequently, the assimilation of voice input and output mechanisms becomes imperative to fully harness the capabilities of ChatGPT in language learning.

Yet, other concerns still persist. Despite ChatGPT's aptitude in transforming auxiliary learning materials into interactive study resources, the quality and accuracy of such narratives remain under scrutiny (Diwan *et al.*, 2023). ChatGPT, on occasions, might produce misinformation or biased perspectives. Predominantly, the data it relies upon is procured from English sources and subsequently translated into various languages, such as Chinese or Japanese (Diwan *et al.*, 2023). Users also face the challenge of designing prompts to guide ChatGPT towards desired outputs—a process often daunting for non-technical individuals(Woo, Guo, and Hengky Susanto, 2023). Possible remedies involve furnishing structured learning materials to complement ChatGPT's capabilities or deploying a combination of hidden prompts to guide ChatGPT and open prompts for user initiation, ensuring a more refined feedback loop.

Furthermore, ChatGPT does not encompass features to record learners' progression or prompt them for periodic reviews. The technique of spaced repetition review, however, stands as a pivotal, economical, and effective learning strategy with immense potential to augment educational results (Kang, 2016). This method, aimed at reinforcing long-term retention, relies on strategically timed reviews premised on the forgetting curve—a principle that underpins numerous study systems (Yang, Oh and Youm, 2016). Empirical evidence reveals that EFL students who engaged in spaced repetition exercises for merely three minutes daily witnessed a threefold surge in long-term vocabulary retention (Chukharev-Hudilainen and Klepikova, 2016). Hence, the integration of a monitoring mechanism and a spaced repetition review system in educational software becomes indispensable.

In summary, while ChatGPT holds numerous advantages for EFL learners, its optimal utilization necessitates a combination of supplementary learning materials, efficient prompt management, voice integration, a robust learning tracking mechanism and spaced repetition strategy.

## Requirements Gathering

Basing on the previous literature review, this part will investigate and compare functionality of the existing applications related to EFL education. Furthermore, to obtain a deeper understanding of user perspectives regarding EFL software and the incorporation of ChatGPT, interviews were conducted with four EFL learners.

### Analysis of Existing Applications

According to the result of literature review, five salient features intrinsic to EFL software were identified: integration of AI technology, learning material, voice integration, learning tracking mechanism and spaced repetition review system. While these form the primary attributes, other desirable features will also be examined in this analysis. For a comprehensive understanding, three popular EFL websites have been chosen for a thorough evaluation of the merits and potential shortcomings of these applications.

**Duolingo** is one of the most popular language learning applications. As presented in its introduction, it embraces the prowess of AI with established language pedagogies to curate tailored learning courses(Duolingo, 2023). The platform promises a comprehensive curriculum that covers crucial language domains, such as reading, writing, listening, and speaking. Although it also incorporates its voice function in courses and practices, the review system doesn’t follow spaced repetition rules. The website also emphasizes the gamified nature of learning. Earning points, unlocking levels, and interacting with their mascot, Duo the Owl, are elements that underscore Duolingo's commitment to making language acquisition both engaging and habit-forming.

**Glossika** positions itself as an innovative platform because of adaptive learning algorithms and an extensive database of handpicked sentences (Glossika, 2023). With a strong emphasis on customization, the platform crafts content in alignment with users' competency levels and interests. By immersing learners in full-sentence contextual practice, Glossika ensures a combination of vocabulary, pronunciation, listening, reading and writing practices. Another intriguing aspect of Glossika's approach is its promised use of a spaced repetition algorithm, designed to obviate the often-daunting task of memorization. This ensures that learners receive timely prompts on phrases they're easily to forget, offering a seamless learning curve. Furthermore, authenticity is emphasized by the platform's incorporation of native speaker audio for every sentence. This allows learners to not just assimilate accurate pronunciation but also self-assess their progress by recording and comparing their accents over time.

**Langotalk**, an innovative startup, harnesses the potential of ChatGPT technology to redefine EFL education (Langotalk, 2023). One of its standout features is the diversity of chatbots it offers, each built with a distinct personality tailored to imitate varied settings and specialized topics. This range enables learners to immerse themselves in a conversational context. While it doesn't offer a structured curriculum, Langotalk allows users to create notes and log study items. The chatbots can offer real-time, individualized feedback, acting not just as conversational partners but also as tutors to guide on expression choices and grammatical structures. However, the incorporation of voice input and output can only base on the conversation content. In addition, a noticeable gap in Langotalk's offering is the absence of a robust learning tracking mechanism and a systematic review system, which might be essential for EFL learners looking to monitor their progress methodically.

Figure 1: Comparison among Existing Applications

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Duolingo** | **Glossika** | **Langotalk** |
| **AI Technology** | AI-powered course recommendation and assessment. | AI-powered voice recognition and review algorithm basing on learning behavior and spaced repetition | ChatGPT |
| **Learning Material** | Well-structured and interactive courses martials for different level. | The sentences recommended by AI basing on user’s ability and preferences. | Not provided. Users can interact with chatbots directly and produce study content. |
| **Voice Integration** | Voice input and output for materials and review exercises. | Voice input and out for vocabulary and sentences practices. | Voice input and output for conversation texts. |
| **Learning Tracking Mechanism** | Record the courses which have accomplished. | Well recorded study progress and review schedule. | Not provided. But users can make study notes. |
| **Spaced Repetition Review System** | Review system but does not mention using spaced repetition strategy. | Individualized review schedule using spaced repetition algorithm. | No review system. |
| **Other** | Gamification and virtual rewards for achievements. | Weekly study report. | Various chatbots with different personality. |

The comparative analysis of the three EFL education software, as presented in the table above, highlights distinct strengths and areas of improvement for each. Traditional application, Duolingo and Glossika, prioritize high-quality learning materials, underscored by their robust learning and review mechanisms. However, their structured approach may somewhat scarify the flexibility and tailored experiences many modern learners seek. In contrast, Langotalk, capitalizing on its innovative approach, outstands in delivering rich, personalized learning experiences tailored to individual EFL learners. Meanwhile, its conspicuous absence of a learning progress monitoring system, week quality assurance for learning materials, and a systematic review mechanism somewhat diminish its overall user experience. Noteworthy across the board, though, is the seamless integration of voice features, enhancing the user's learning experience in all the platforms.窗体顶端

### Interviews

To gain richer insights into the experiences of perspective learners using EFL education software and ChatGPT, four participants were recruited to join an structured interview, the details of which are provided in Appendix A. These participants, native speakers of either Mandarin or Cantonese, reside in the UK. Findings indicated that while all participants had utilized EFL software for English language learning, their familiarity with ChatGPT's interactive features didn't necessarily sway their preference. Despite recognizing the potential benefits of ChatGPT in facilitating English acquisition, traditional EFL education software remained their primary choice. Their preference was attributed to two notable factors: the lack of voice integration in ChatGPT and its absence of a systematic learning and review monitoring system. These reasons echo the sentiments expressed in our earlier discussions. The insights garnered from the interviews contribute to the establishment of application requirements, which will be elaborated upon in the subsequent section.

## Application Requirements

Through the above-mentioned research, a compilation of vital features has been identified. These features were prioritized using the MoSCoW methodology (refer to Appendix B) and substantiated with user stories (refer to Appendix C). This section delves into the key functional and non-functional requirements for the proposed application, *Langbuddy*.

### Functional Requirements

For an adaptive and holistic EFL learning experience, the application is designed around four main functional requirements:

**Structured Learning Material**

* **Diverse Course Offerings**: The platform should provide a spectrum of EFL courses, allowing users to select specific areas and levels they wish to bolster.
* **Seamless Integration**: Learning materials can be effortlessly integrated into the study plan system, auto-generating a structured study tasks with a pre-built schedule.

**Study Plan System**

* **Customization**: Users can craft and schedule their study tasks to fulfill their unique learning needs. This fosters autonomy, enabling learners to manage their educational journey flexibly.
* **Adaptability**: Users can inspect and modify their learning content and schedule as required.
* **Analytical Overview:** The platform offers statistical overview of a user's learning trajectory, enhancing their ability to strategize and plan ahead.

**Tailored AI Chatbot with Voice Interaction Ability**

* **Multi-interaction**: Users can communicate with the AI chatbot using both text and voice inputs. The chatbot, in turn, is equipped to respond vocally if required by users.
* **Versatile Chatbot Personalities**: Recognizing the diversified needs in EFL learning, distinct chatbot personalities should be curated for varied learning contexts and requirements.
* **Review Integration**: Conversational content can be flagged and seamlessly integrated into the review mechanism automatically, ensuring consistent learning progression.
* **Translation Capability**: For optimal comprehension, conversation content can be translated into the user's mother language when required.

**Review Mechanism with Spaced Repetition Strategy**

* **Automatically Integration**: The finished study task and noteworthy points from chatbot interactions can be seamlessly transitioned into the review system if required. Users can decide the review schedule and amend content in the integration process.
* **Customizability**: While the system offers preset spaced repetition schedules, users also have the autonomy to devise unique review timelines. Additionally, they can create and modify content and timelines as they progress.
* **Analytical Overview**: Users can see their review performance and progress through statistical insights. They can adjust their learning process basing on the information provided by the platform.

### Non-Functional Requirements

The project must also fulfill three non-functional requirements as part of its design and implementation.

**Usability**

The application user interface should be user-friendly and fast-loading. The design style should enable users to identify information easily and focus on learning tasks. Furthermore, the management of loading state, which refers to the waiting for response from chatbot and transformation between text and voice should be considered. Moreover, in instances of errors or unexpected user actions, clear and instructive hint messages should be presented, guiding users to rectify issues or understand the context better.

**Security and Privacy**

Before any personal information or voice data is accessed or processed, users should be informed and their consent is obtained, especially when data needs to be shared with third-party API providers to enable specific functionalities. The application should strictly maintain compliance with GDPR (General Data Protection Regulation) and ensure that personal and sensitive user information is safeguarded at the adequate level of security measures.

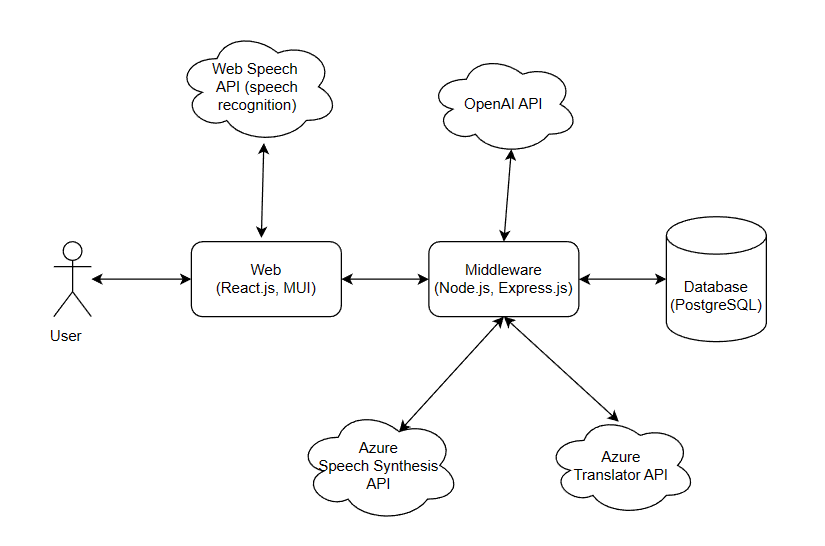
**Daily Interaction Limitation with Chatbot**

Given the reliance on third-party APIs, a reasonable restriction on user-chatbot interactions is enforced. This will prevent abusing these features and control the operational cost at an affordable level. Meanwhile, this limitation will also encourage the user to engage with Chatbot in a thoughtful and responsible manner.

# Design and Implementation

This part will illustrate the system architecture of this application. Then the user interface design and database design will be explained. Finally, the implementation of functionalities will be also elaborated.

## System Architecture



**Figure 2:** system architecture

This application system includes four parts: frontend, middleware, database and external API.

The frontend is developed using **React**, a widely-used JavaScript library known for building interactive user interfaces for single-page applications and maintained by Facebook (React, 2023). React's component-based approach ensures modularity, facilitating easier updates and maintenance. Moreover, it allows for the creation of dynamic and responsive web pages, enhancing the user experience, which is suitable for rendering the study items, review items and chat boxes repeatedly. Paired with React, the popular UI framework **MUI (Material-UI)** which implements Google’s material Design is chosen. MUI offers a set of React components tailored for faster and more efficient web development (MUI, 2023a). It's designed for creating visually pleasing and intuitive user interfaces, making it especially suitable for an education-focused application that requires clear and accessible UI elements.

For the middleware, **Node.js** stands out as an ideal choice. This open-source, cross-platform JavaScript runtime environment is known for its efficiency in handling concurrent requests, essential for a responsive chatbot experience. Furthermore, its non-blocking I/O model ensures real-time updates, a crucial feature for any interactive and data-intensive platform. Node.js also has rich libraries which can be easily integrated with various function of the application. Complementing Node.js, **Express.js** is utilized. Express.js, being a minimal and adaptable Node.js web application framework, simplifies the process of crafting server-side applications, making the integration of features and APIs seamless and straightforward (Express, 2023).

For the application's data management needs, **PostgreSQL** is selected. Renowned as an advanced open-source relational database system, PostgreSQL offers key features such as extensibility, data integrity, and adherence to SQL standards. Within this application, the relationships between users, study tasks, review sessions, and chat history are intricate and tightly-knit. Such relationships are best modeled using a relational database paradigm. Relational databases, like PostgreSQL, excel in representing data entities and the connections between them in a structured manner. They are particularly adept at handling complex queries and ensuring data consistency. Given that application demands a clear representation of the relationship dynamics among its data elements, the feature ACID (Atomicity, Consistency, Isolation, Durability) of PostgreSQL is beneficial and essential for this software.

The application's capabilities are further magnified through a selection of external APIs:

* **OpenAI API**: Considering the chatbot's intelligence, OpenAI's API allows this software leverage the GPT’s advantage to generate human-like and qualified interactions. Given the nature of this application, it's vital that users get accurate and interactive EFL learning guidance and assistance from AI model, making this API an apt choice.
* **Web Speech API**: The Web Speech API encompasses a suite of tools and protocols offered by contemporary web browsers, enabling developers to integrate both speech recognition and speech synthesis (text-to-speech) capabilities seamlessly into web applications (Mozilla, 2023). Within the application, speech recognition feature is incorporated, primarily due to its ready accessibility and rapid response attributes. Facilitating interactions between users and chatbots through voice not only elevates the intuitiveness of the experience but also enhances user-friendliness, making the learning process more engaging.
* **Azure Speech Synthesis**: Aiding in the text-to-voice functionality, Azure Speech Synthesis offers voice outputs that closely mirror natural human speech. This enriches user interactions, making digital learning more immersive. Furthermore, the azure speech Synthesis API provides diverse accents and tones, which helps users learn different accents (Microsoft Azure, 2023a). The vivid output speech and various choices of accents outweighs the speech synthesis function provided by web speech API.
* **Azure Translator API**: The Azure Translator is employed for real-time, multi-language translations (Microsoft Azure, 2023b). This ensures that users, regardless of their linguistic background, can access content in their mother language when needed in the process of interacting with chatbots.

## User Interface

### Colour, Font and Style

The visual aesthetics of the application are predominantly shaped by the theme colour functions provided by MUI. The main colour is set to cyan, a hue known for its refreshing, calming, and modern characteristics (Canva, 2023). Cyan is associated with clarity and stimulates mental focus, which aligns with the educational intent of the platform. To complement this primary shade, light blue is selected as the secondary colour. This colour not only harmonizes well with cyan but also further emphasizes the cool, tranquil atmosphere conducive to focused learning. For certain interface elements, such as disabled buttons and components, the application employs a neutral grey. This default colour subtly signifies non-active or inaccessible features without detracting from the overall visual appeal.

As for the typography, the application utilizes the Roboto font. Given that our application's primary objective is to facilitate effective language learning, it's crucial to ensure readability and legibility in every textual element. Roboto achieves this by offering clear letterforms, making it easier for learners to engage with the content.

The style of application's user interface is rooted in simplicity and directness. By adopting a minimalist approach, any potential distractions are eliminated, ensuring that users can focus wholeheartedly on their studies. Every element, from the layout to the interactive components, is crafted to be intuitive and self-explanatory in order to reduce users’ cognitive overload.

### Components and Pages

This section highlights the crucial components and four primary pages of the application: the dashboard, study overview, review overview, and chatbuddy.

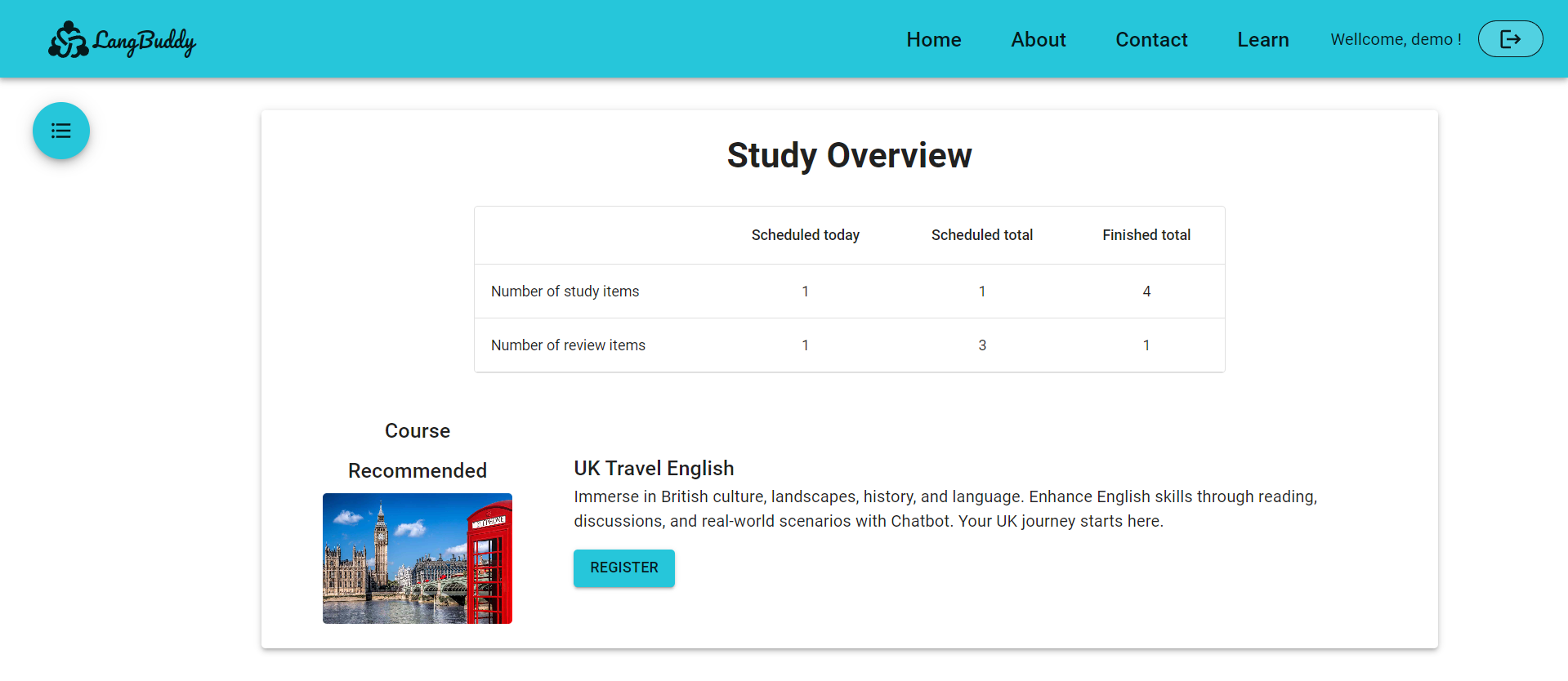


Figure 3: Dashboard Page

The dashboard showcases statistics for both study and review items, providing users with a clear insight into their learning progress. Additionally, a structured course is presented for demonstration purposes, enabling users to enrol and automatically generate default study items.

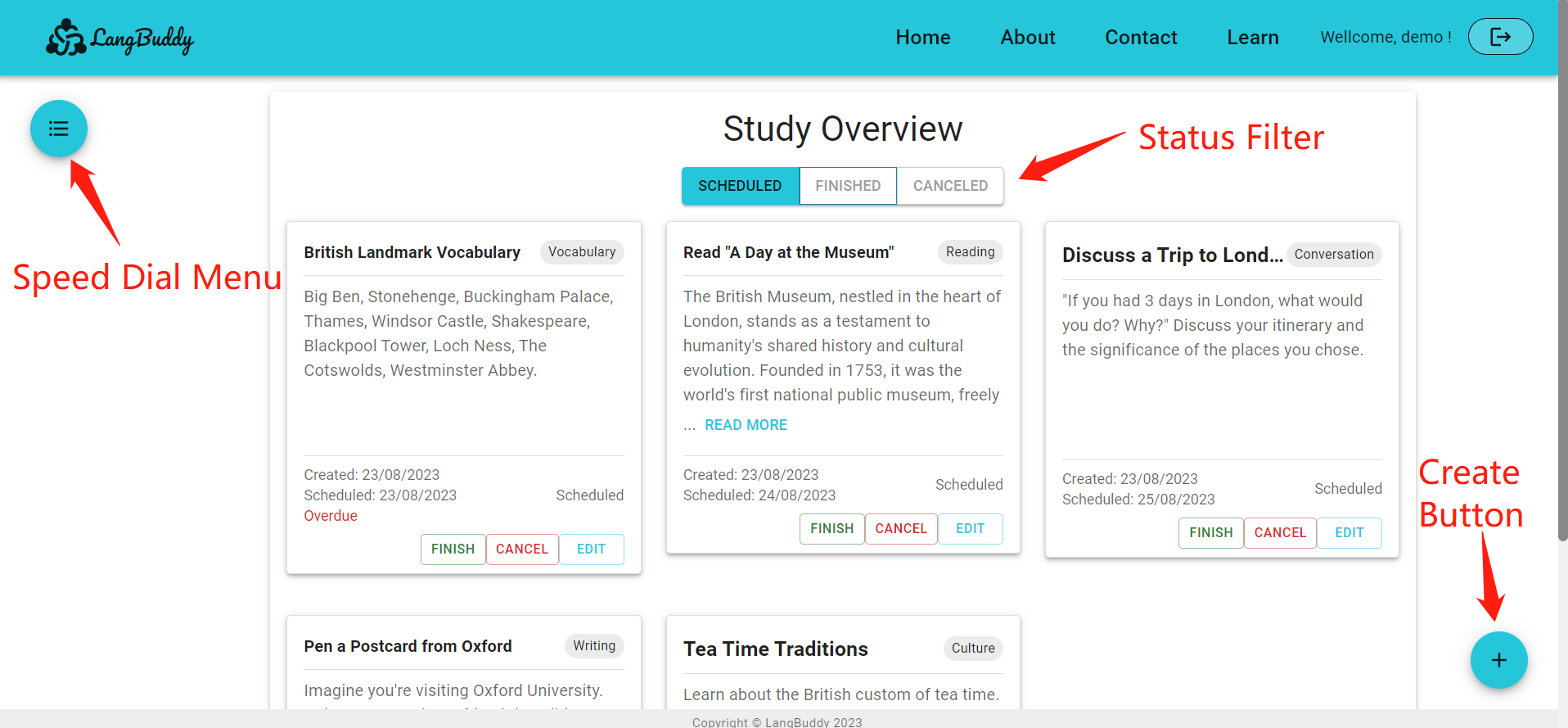


Figure 4: Study Overview Page

The study overview page displays study items through the study card component. These study cards can be organized based on their status: scheduled, finished, or cancelled. For scheduled items, cards are arranged in ascending order by scheduled date, while for finished and cancelled items, they are in descending order by completion date. This arrangement ensures that users can easily track and manage their study plans. Users can navigate to other pages using the speed dial menu and add new study items with the "create new study item" button.

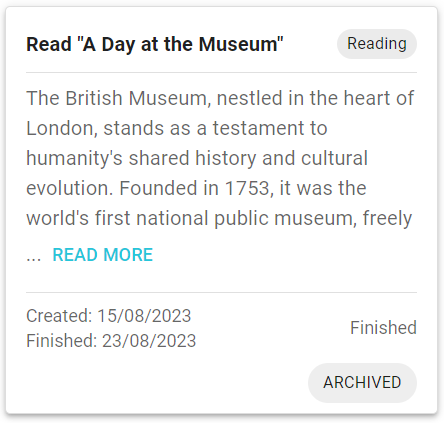
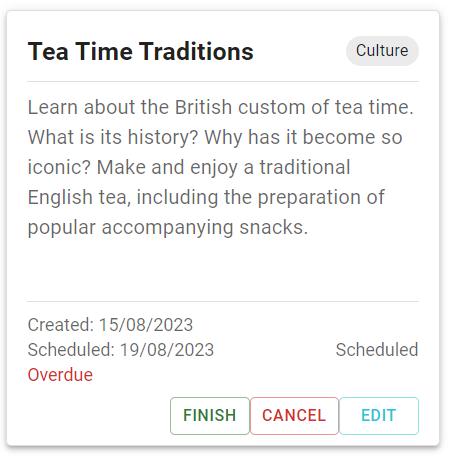


Figure 5: Study Card Component

Each study card offers essential details such as the title, category, content, creation date, and scheduled date. Depending on the status, additional features or buttons, including status changing and editing function, are available for user interaction. The study card will remind the user with red characters if it is overdue comparing to the scheduled date. For longer content, a "read more" option unfolds the content details in a modal view.

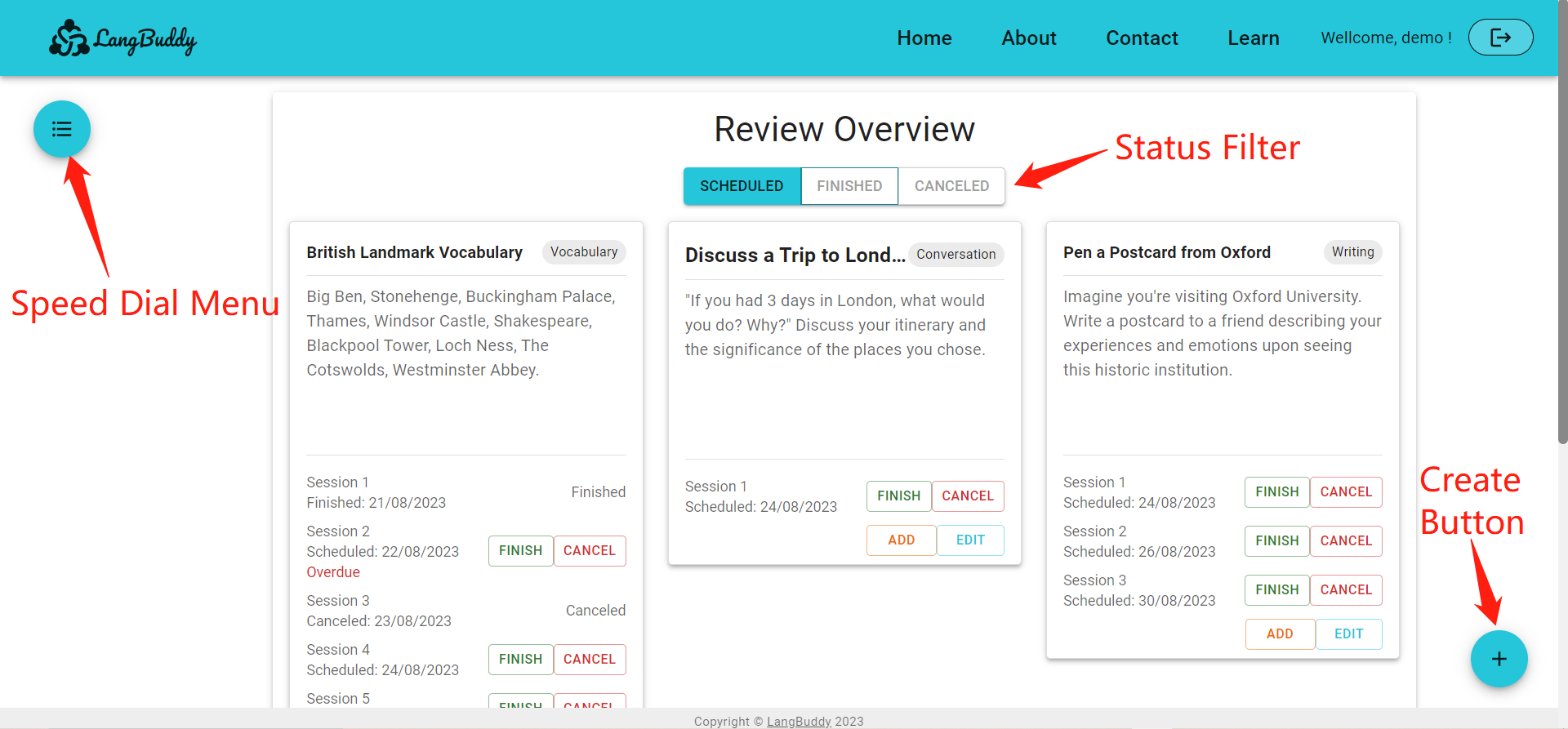


Figure 6: Review Overview Page

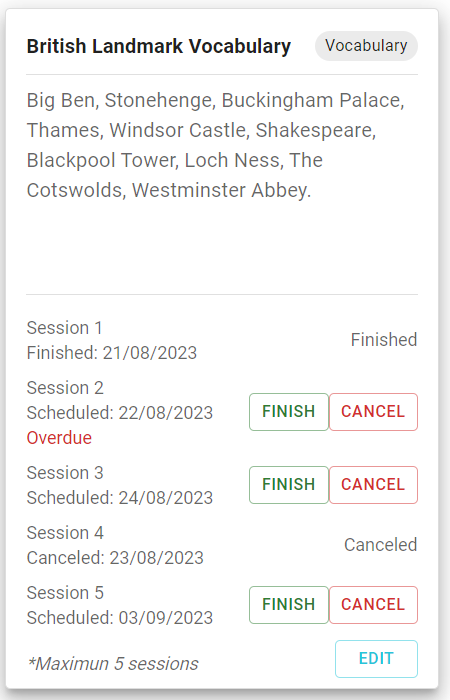
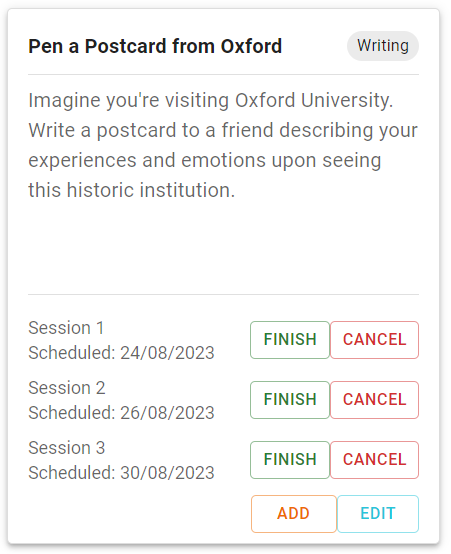


Figure 7: Review Card Component

The review overview page mirrors the organizational structure of the study overview page, listing all review cards based on their status. However, the layout for individual review cards differs from that of study cards. In addition to standard details (title, category, content and etc.), review cards showcase a set of session components that detail the scheduled date, completion date, status, and possible actions for each review session. Users have the flexibility to add or edit review item details and sessions, with a cap of five sessions for each review item. The overall status of a review item adjusts based on the completion or cancellation of all its sessions.

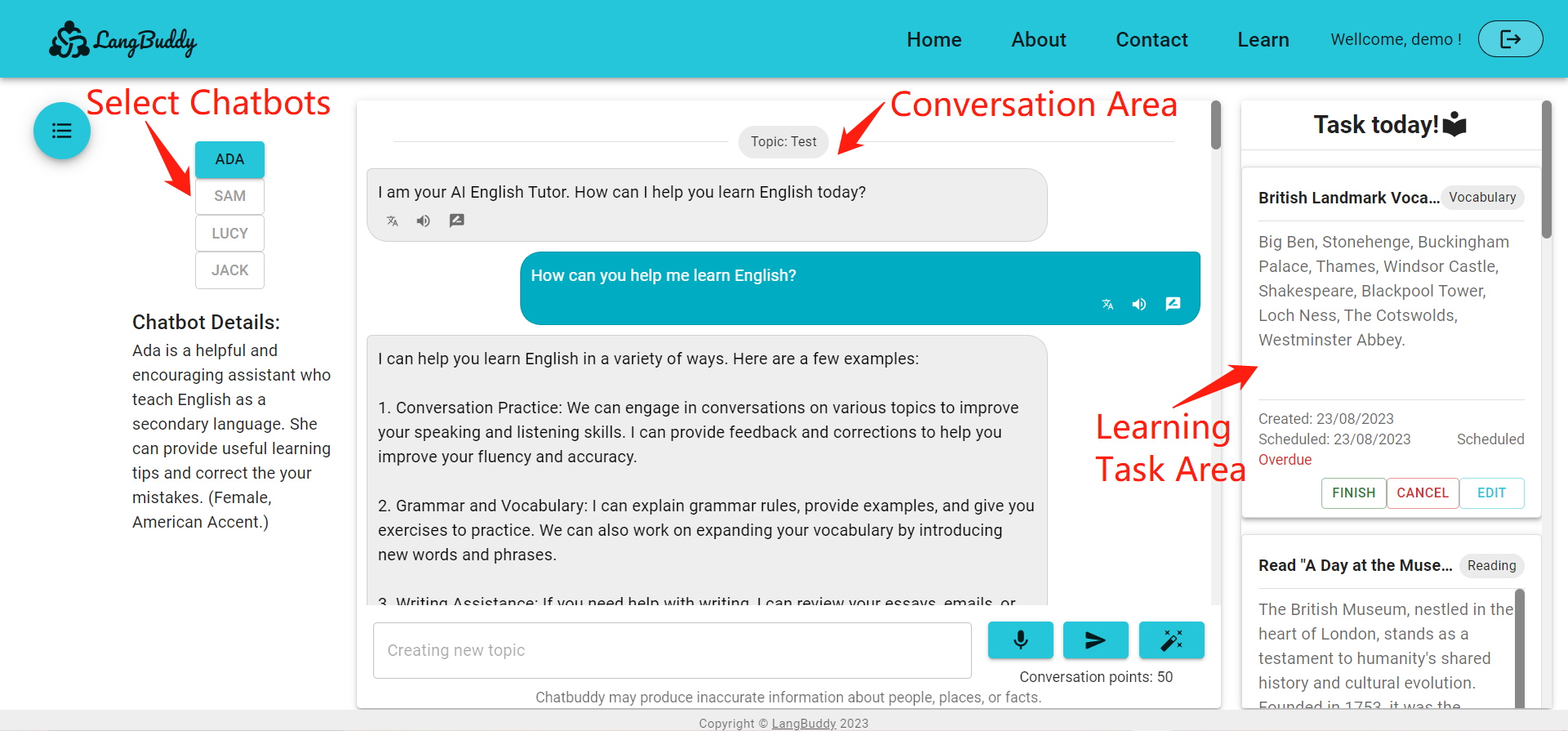
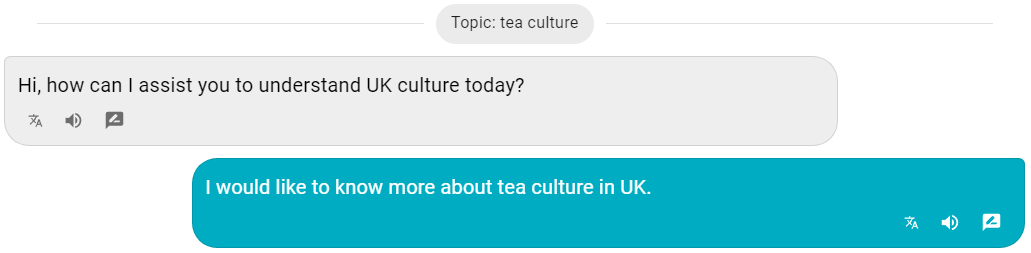


Figure 8: Chatbuddy Page

The chatbuddy page is structured into three main sections: chatbot selection, conversation area, and learning task area. In the chatbot selection section, users can choose a chatbot tailored to their specific learning objectives. The conversation area is where users interact with their chosen chatbot. Conversational history is displayed through a series of chatbox components. Users can type messages in a dedicated input area. This area also features three buttons: one for speech recognition, another for sending messages, and a third for initiating new topics. Beneath these buttons, conversation points are highlighted for user reference. Additionally, the learning task area showcases study tasks and review sessions scheduled for the current day, using the previously described study cards and review cards.



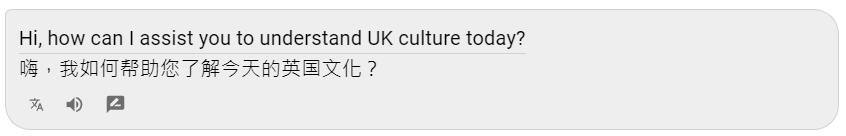


Figure 9: Chatbox Component

Diving deeper into the chatbox component, its appearance and features are determined by the 'role' prop. If its role is set as 'topic', it serves as a divider showcasing the topic's value. If the role is 'assistant', the chatbox appears on the left with a grey background. When the role is 'user', the chatbox is positioned on the right, boasting the theme's main color. Moreover, three functional buttons are present at the base of the chatbox for translation, text-to-speech, and review item creation. These functionalities are activated upon user interaction.

## Database Design

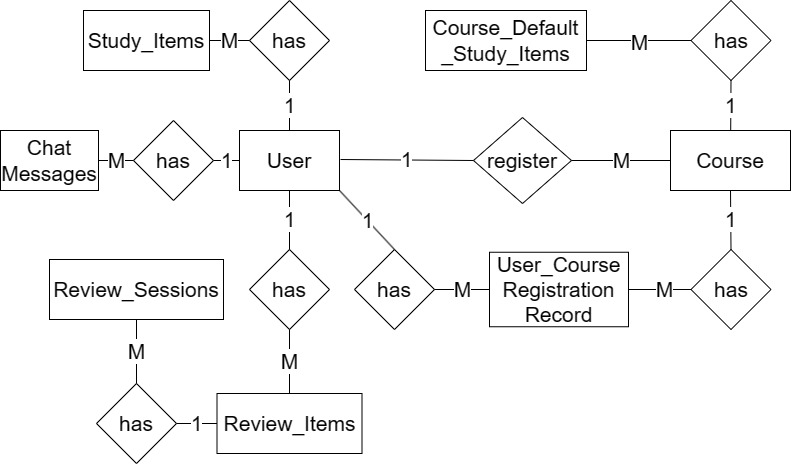


Figure 10: Entity Relationship Diagram

The provided diagram offers a comprehensive view of the project's entity relationship. Identifying the entities and their relationships between each other is essential for good database design. Each entity also has associated attributes and constraints, these can be found in Appendix D. Several crucial aspects of this design merit attention:

**Course Default Study Items**: These are static and pre-configured within the database. The platform should provide a range of courses, each containing a set of default study items. When users enrol in a course, these default items will be imported to the dynamic 'study items' category. Unlike the pre-set defaults, study items can be modified by users basing on their needs. This function requirement makes it reasonable to separate tables for these two types of study items.

**Review Items and Review Sessions**: Contrasting with study items, each review item comprises multiple review sessions. This structure empowers users to formulate a spaced repetition pattern tailored to their learning style, affording them control over the quantity and scheduling of review sessions for each review item.

**Data Harvesting**: Being a learning-centric platform, user study behaviours are paramount for refining the services provided. Data attributes such as content, category, created date, scheduled date and completion date can be valuable insights for service enhancement. Furthermore, token usage during user-chatbot interactions is vital and monitored by the chatmessage table. Since the OpenAI API determines charges based on token consumption, monitoring this can not only understand user behaviour but also prevent misuse and effectively managing operational expenses.

## Function Implementation

### Structured Learning Courses

Courses encompass a variety of default study items covering multiple topics and proficiency levels. When users enrol in a course depending on their individual learning objectives, the system seamlessly incorporates a set of study items from the course's predefined data which are saved in the database. This effectively constructs a user-specific study plan. Users retain the option to modify both the content and scheduling of these integrated study items at any future point in the study plan system.

### Study Plan System

Besides creating new study items by registering the pre-set courses, the user can also craft the new study plan tasks by themselves. The dashboard statistics and study overview page demonstrate the learning task plan by days and status categories. With scheduled study items arranged in chronological order, users can effortlessly access upcoming learning tasks. Users also have full flexibility to modify all attributes of each study task. If a task's status is changed to "finished," a modal will appear for users to select a review pattern and consequently create review items and sessions if desired.

### Review System with Spaced Repetition Strategy

As mentioned previously, the finished study items can be migrated into review system with a selected spaced repetition strategy by users. Two standardized spaced repetition methods are available: a simple model and a normal model. The simple model suggests reviews on the 1st, 3rd, and 7th days, while the normal model schedules sessions on the 1st, 2nd, 4th, 7th, and 14th days. Furthermore, the user can customize their unique review strategy and review sessions for each review item.

The user can also monitor the review progress through the overview statistics. Emulating the study plan system, review items are showcased chronologically based on their upcoming review session dates, facilitating users in anticipating imminent learning activities. Moreover, even after setting up the review plan, this system allows user to edit every aspect of the review items and sessions, especially for the scheduled dates. This enables users to deploy the spaced repetition strategy in alignment with their individual pace. The blend of predefined and customizable review patterns empowers users to chart their academic trajectory and optimize learning outcomes.

### Chatbot

**Integration with study system and review system**

The chatbot enhances learning outcomes through seamless integration with the study and review systems. On the chatbuddy page, the user can view the study item and review sessions which are scheduled at today on the right hand, facilitating effortless input into the chatbot conversation area. Additionally, users can create review items and sessions directly from chatbot conversations by clicking the creating review icon. By default, though the conversation content serves as the review item's content, users have the flexibility to customize review attributes and pattern using a spaced repetition strategy.

**Chatbot personalities and prompts design**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Temperature | Presence Penalty | Frequency Penalty | Speciality |
| Ada | 0.3 | 0.2 | 0.2 | General and Helpful Tutor |
| Sam | 0.1 | 0.1 | 0.1 | Vocabulary and Spaced Repetition Review Strategy |
| Lucy | 0.5 | 0.5 | 0.5 | UK culture |
| Jack | 0.3 | 0.2 | 0.2 | Job Hunting and Career |

Figure 11: Chatbots Attributes and Speciality

This platform offers four distinct chatbots, all of which utilize the gpt-4 model without max\_token restrictions. However, the combined length of input and generated tokens adheres to the model's context length constraints (OpenAI, 2023).

Given the specialized roles of these chatbots, they are endowed with unique attributes. Three key attributes are associated with chatbot behavior. The temperature, ranging between 0 and 2, governs the randomness of generated text. Higher values introduce unpredictability, whereas lower values ensure consistency and focus(OpenAI, 2023). The frequency and presence penalties can be used to adjust the possibility of sampling repetitive sequences of tokens. Presence penalty, with value between -2.0 and 2.0, influences new tokens based on frequency in the text so far, encouraging the model to talk about new topics (OpenAI, 2023). Meanwhile, frequency penalty, with a similar value range as the presence penalty, influences new tokens based on frequency in the text so far, reducing the model's possibility to repeat the same line verbatim (OpenAI, 2023). As Ada is performed as a general assistant, the attributes are set be in an intermediate level, which means it will appear new topic and a little randomness in the reply. However, as Sam focuses on explaining the word, it should follow the instruction from the user strictly. The randomness of Sam should be low and new topics should not be encouraged. Lucy is specialized in UK culture. It is encouraged to add more randomness, new topics and various vocabulary. For Jake, it is neutral to the randomness and penalties so the attributes are set to the middle level.

Prompts function as vital directives for the LLM, ensuring adherence to rules, process automation, and output specificity. They shape LLM interactions and outcomes, effectively programming the model's outputs (White *et al.*, 2023). This application employs both hidden (or system messages) and open prompts to guide chatbot outputs. System messages prime the model with context or instructions without being visible to users (suhridpalsule and eric-urban, 2023). The system message can be used to define the assistant’s personality, adjust what the content and format of model responses. The system message usually not be shown to the users but send to the API with conversation history together. Conversely, the open prompts are visible and show the interaction between chatbots and users. In this application, the initial open prompt is pre-set by the system and give a clear guidance to the users how they should interact with the chatbots. For example, as a general purpose chatbot like Ada, the system message and initial open prompt is set to be a more general guidance with encouraging character. The details of the prompt structure are shown in the figure below.

|  |
| --- |
| System message: [        { "role": "system", "content": "You are a helpful and encouraging assistant, who teaches English as a Second Language in an interesting way. The user is learning English. You can provide useful learning tips and correct the user's mistake in a polite way." }  ] |
| Initial prompt: " Hi, how can I assist you with learning English today?" |

Figure 12: Prompts for chatbot Ada

Another technique employed is few-shot learning, where training examples provide the model with added context (OpenAI, 2023). For chatbot Sam, which is specialized in vocabulary and spaced repetition review strategy, few-shot examples is integrated in the system message as shown below. Moreover, Lucy adopts the few-shot prompts pattern similar to Sam and Jack use a no-shot prompt pattern similar to Ada.

|  |
| --- |
| messages: [        { "role": "system", "content": "You are a helpful assistant, who specialized in teaching English as a secondary language and explaining vocabulary with examples. You also suggested repetitive learning strategy to learn English." },        { "role": "system", "name": "example\_user", "content": "What does the word evoke mean?" },        { "role": "system", "name": "example\_assistant", "content": "Evoke means to bring to mind or to recall a feeling, memory, or image. For example, the smell of freshly baked bread might evoke memories of your grandmother's kitchen." },        { "role": "system", "name": "example\_user", "content": "So, if I listen to a song which makes me think of a past event, it evokes that memory? Is this sentence correct?" },     { "role": "system", "name": "example\_assistant", "content": "Exactly!Remember, to effectively learn and remember vocabulary, use a repetitive learning strategy. This means, try to use evoke in sentences several times this week." },        { "role": "system", "name": "example\_user", "content": "Can you tell me more about how to use spaced repetition strategy to learn the vocabulary?" },        { "role": "system", "name": "example\_assistant", "content": "To use spaced repetition for vocabulary learning, start with a manageable set of words. Initially, review the new words after a short interval, such as 20 minutes, and then progressively extend this gap (e.g., 1 day, 2 days, 4 days) as you remember them. Introduce new words daily while reviewing older ones, and for better retention, try to use these words in your own sentences. Consistency is key, so make it a daily habit, adjusting intervals based on your personal recall rate." }      ] |
| Initial prompt: "Hi, how can I assist you with learning English vocabulary today? Please give me a word or phrase you want to learn. " |

Figure 13: Prompts for chatbot Sam

**Conversation message management**

Chat messages are securely stored in the ChatMessage table, serving as the authoritative data source. The management of these conversation messages encompasses two main areas: interaction with the OpenAI API in the backend and frontend presentation.

Upon receipt of a message sent by the user from the frontend, alongside the associated chatbot name, the message is initially stored in the database with the role of "user." Subsequently, the system retrieves the last 20 conversation records under the current topic from the database. This collection of past messages, combined with the chatbot's system message which is identified by chatbot name, is integrated into a new message array and transmitted to the OpenAI API for a response. This response, along with token data, is recorded in the database under the "assistant" role and send to the frontend for display. The details of code is shown in the Appendix E.

On the frontend, React's useState hook is utilized to oversee the chatbot's conversation history and the name of selected chatbot. The history is maintained under the messages state, which is rendered through mapping the chatbox component, while the actively selected chatbot is controlled by the selectedChatbot state, with "Ada" as its default value. Upon loading the chatbuddy page, a React useEffect activates, rendering the most recent 20 conversation records of "Ada". Another useEffect handles any changes to the selected chatbot, contingent on changes in the selectedChatbot state. This ensures that when a different chatbot is selected, the most recent 20 conversation records for that particular chatbot are retrieved and displayed. Furthermore, as users dispatch messages, both the user's message and the OpenAI API response are added in the messages state, enabling immediate rendering on the interface.

**Speech and translation integration**

Speech recognition is implemented through the web app API. When users click the recording button, the input text area is temporarily disabled, capturing the spoken content. Once recording ends, the transcribed text appears in the input text area which allows users to modify it before submitting it to the backend.

Regarding speech synthesis and translation, they are activated when users the respective buttons within the chatbox. The text within the chatbox is dispatched to the appropriate API, with the resulting synthesized speech or translation then presented on the frontend.

**Fine-tuning**

Fine-tuning serves as a technique to elevate chatbot response quality, delivering superior results compared to prompting and with reduced token consumption. Nonetheless, this application doesn’t employ fine-tuning for two reasons.

First, on July 06, 2023, OpenAI announced the upcoming retirements of older GPT-3 and GPT-3.5 models accessible via the completions endpoint and their first-generation text embedding models (OpenAI, 2023). These models will be shut down on January 04, 2024 without alternative option provided at that time. Later, on August 22nd, 2023, OpenAI announce availability for the new fine-tuning API  for GPT-3.5 Turbo (OpenAI, 2023). The timing didn't favour integrating the fine-tuning model into this application.

Another reason is the training data. It’s hard to generate and collect high-quality training data for producing a finely-tuned model that fulfil the requirement of this application. So the fine-tuning method to build a chatbot is not adopted for this application.

### Security and Authentication

To ensure the user security and authentication, this application use the bcrypt package. This involves salting each password 10 times and then hashing it to ensure its protection against potential breaches.

Upon successful login, the system generates a JSON Web Tokens (JWT) for authentication. This token remains valid for 24 hours. This token is then dispatched to the frontend encapsulated within a cookie. This methodology guarantees both security and seamless user experience.

For every subsequent interaction between the frontend and backend post-login, the frontend forwards the token to the backend. This token is verified at the backend. Upon successful decoding, the backend retrieves the user ID for further operation. This arrangement ensures that every transaction or request is both legitimate and secure.

### User Experience and Error Handling

Visibility of system status, error prevention and guidance for handling error are the foundational principles for interactive application design (Jakob, 2020). This application adheres to these heuristics to ensure good user experiences. Key implementations concerning these principles are elaborated below, with visual demonstrations available in Appendix F.

As this application involves interaction with external API service, users may frequently wait for the response from these API services. Recognizing the importance of communication during these waiting periods, the application should manage the loading state and provide users with unambiguous indicators regarding system status. To achieve this, the frontend uses a set of state hooks to monitor and reflect the loading status. When the loading state is triggered, distinct icons are dynamically presented to the user, signifying both the start and end of the loading process.

As the application adopts the minimalist design principle and reduces unnecessary disrupts for users, some buttons are presented without any explicit explanation. However, to prevent potential user confusion, hovering over these buttons activates a tooltip. This concise textual description helps users understand the button's functionality and avoid misunderstanding.

User data entry, another vital part of this application, requires validation procedure to mitigate input errors. To facilitate this, the frontend uses the react-hook-form library to manage form creation and validation. When the user input invalid data, the form shows hint message which assists the user to deal with error.

Furthermore, Moreover, the application employs the "try-catch" methodology in both frontend and backend frameworks for error management. If error happens, an informative error message is rendered on the frontend via a snack bar positioned at the screen's lower-left. The user will be informed the reason of the error from user’s perspective and possible solution for recovering from error. Moreover, for critical operation, this snack bar will also be shown as a hint message for successful outcome, which help the user know the potential result.

# Testing and Evaluation

## Unit Testing

Unit testing focuses on verifying individual units of code in isolation to confirm they function as anticipated. In this web application, both the frontend (developed with React) and the backend (built using Node.js) are distinct units and thus, require separate testing.

For the React, it is typically recommended to use Jest, a renowned JavaScript testing framework, and React Testing Library, which facilitates React component testing. However, as this application uses MUI library component, it is generally advisable not to tie the tests too closely to Material UI (MUI, 2023b). Furthermore, given that the application comprises fewer than 10 pages, the decision was made to implement frontend unit testing in favour of manual methods.

In the case of Node and Express.js, tools like Jest and Supertest – a library tailored for testing Node.js HTTP servers – are suggested. The Supertest provides a high-level abstraction for testing HTTP assertions, making it an ideal choice for Express.js applications. Combined with Jest, this suite enables the comprehensive testing of APIs, ensuring that both endpoints and integrated functions work as expected. However, this application involves interactions with third party APIs and the intricacies of voice data testing, setting up a mock API environment might be necessary. This approach has a steep learning curve, which could potentially reduce frequent testing.

Considering the factors mentioned above, due to the constrains of development time and inherent learning curve, emphasis should be placed on manual testing and user testing. However, it would be good practice to consider these testing in the future iterations for efficient and thorough testing of all aspects of the application. Furthermore, end-to-end testing presents a viable option for future automated testing endeavours, which ensures the application works as expected when frontend and backend are integrated together.

## Manual Testing

Manual testing involves a tester executing tests on software by adhering to a predetermined set of test cases. This approach requires testers to define test cases for specific code segments and functions and then evaluate the software accordingly. The methodology adopted here is white box testing. In white box testing, testers possess knowledge of the internal mechanics of the system under review. Besides the functionalities proposed, this method also takes into account the internal architecture and operations of the application. Positive test cases are proposed for verifying the intended functionalities under normal conditions, while negative test cases are designed to ensure the system can handle invalid inputs or unexpected user behaviour appropriately. Boundary test cases, on the other hand, specifically target the system's limits, ensuring that it operates correctly at its input and output boundaries. These varied testing cases ensure a comprehensive evaluation of the application and capture potential vulnerabilities. Detailed test case specifications and outcomes are shown in Appendix G.

## User Testing

User testing is a technique employed to assess a product by having real users interact with it. The main aim of user testing is to gain insights into the genuine user-product interaction, identify any challenges faced, and explore solutions to enhance the user experience. This methodology is an irreplaceable usability practice as it offers direct feedback on real-world system usage.

Due to time constraints, only a small group of ten users were recruited for usability assessment and feedback provision on the application. All participants were ESL learners residing in the UK, with English proficiency ranging between intermediate to advanced. Their native languages were either Mandarin or Cantonese. The demographic spanned young adolescents to adults and encompassed a spectrum from students to working professionals. Moreover, their familiarity with computer science and ChatGPT varied. This diverse participant background provided a useful insight into how the application was perceived by users across different profiles.

Before the beginning of the test, participants were briefed on the testing purpose, procedures, and their rights. They were presented with a series of tasks aligned with application requirements. After the participants completed these tasks, a debrief session was arranged for them to address any concerns and provide clarity for filling out a questionnaire. The evaluation question was divided into three parts. The first section evaluated task completion ease on a 1-5-point scale, with higher scores indicating greater ease. The second segment, also on a 1-5-point scale, aimed to understand the participants’ opinions about the dimensions of this application, including design appealing, ease of use, helpfulness, efficiency and overall satisfaction. The third segment comprised two open-ended questions to collect participants’ opinion about the positive feedback and suggestions on this application. The questionnaire and results are shown in Appendix H. All data has been stored anonymously. The appropriate signed ethics checklist is appended in Appendix I.

## Evaluation

In the aspect of manual testing, most of the function test were passed successfully. However, extreme test case like long-form input data are frequently found that are not be dealt with appropriately. Other bugs, including lack of dealing with invalid URL address and buttons conflicts are also found.

Regarding the user testing, all participants gave positive comments about the ease of implementing required tasks and overall dimensions’ performance. They didn’t experience any bugs which seriously influence the user experiences. Meanwhile, some users expressed confusion with the usage of speech recognition function and how to interact with the chatbot effectively. Some important issues and possible solutions are discussed as below. Most of these errors had been fixed after the testing phase.

### Extreme Cases

As an interactive application, this application requires users to input lots of data. Data validation is a key aspect of this application. The Manuel testing found that the application didn’t handle the large amount data input (long-form data) input in some field. For example, as for OpenAI API, different model has different limitations on the number of tokens. When user send a long message to OpenAI API which exceeded the token limits, the endpoint will send an error message which is not well managed by the backend. The solution to this problem may be set a character limits in the frontend input area. Furthermore, there should be a mechanism to catch the error message from the external API.

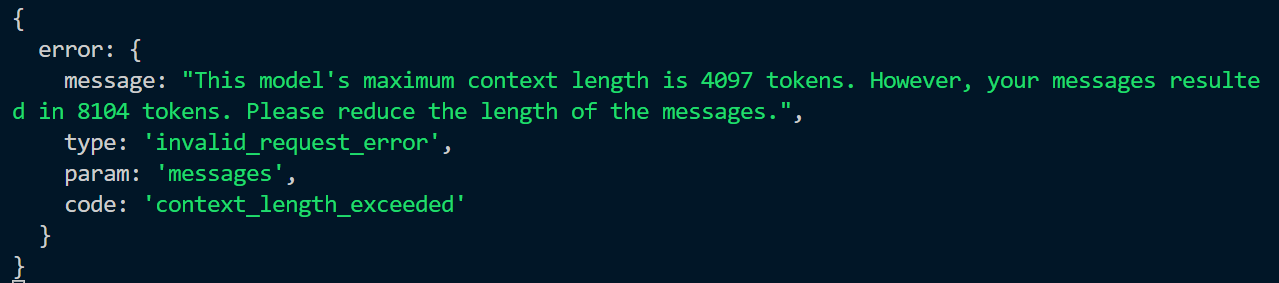


Figure 14: Error message for context\_length\_exceeded from OpenAI API

Another common long-form data challenge is found in the study item and review item management. When the user input long-form data in the title or content area, the form will show red colour to inform the user status of error. But no error message is shown to inform the user how to handle this error. A hint message for users to handle this error should be added.

### Button Clicking Conflict

As the loading state frequently happens for the buttons, it is essential to test any side effect happened when clicking other buttons and the loading state button itself. Although the text input area is disabled when recording, the previous message is still saved in the text state hook and can be sent by clicking sending message button. To solve this issue, the text state hook should be set to null when the recording function is called.

### Voice Recognition

Most of participants felt it’s hard to identify when the voice recognition ended after clicking the recording button. The recording function is set to be ended by the API automatically recognition. However, the user feel the user interface didn’t show the progress clearly. In order to solve this issue, the loading state button of recording should be changed to stop recording icon. When recording, the user can click the stop recording icon and terminate the recording process.

### Interaction with Chatbot

Some users who are not familiar with ChatGPT feel confused with producing effective prompts to instruct the chatbot to give the desired result. They also held unrealistic expectation about the chatbot, such as think chatbot can remember what has been discussed before or provided extreme qualified tests for them. Besides user education, a prompt manager is suggested to incorporate in the system.

### Content Display Format

Another issue found is the display format of review and study content. Despite users entering their content with line breaks or in separate rows, the displayed text appears as a continuous block without respecting these line breaks. The solution of formats tantalization need to be explored.

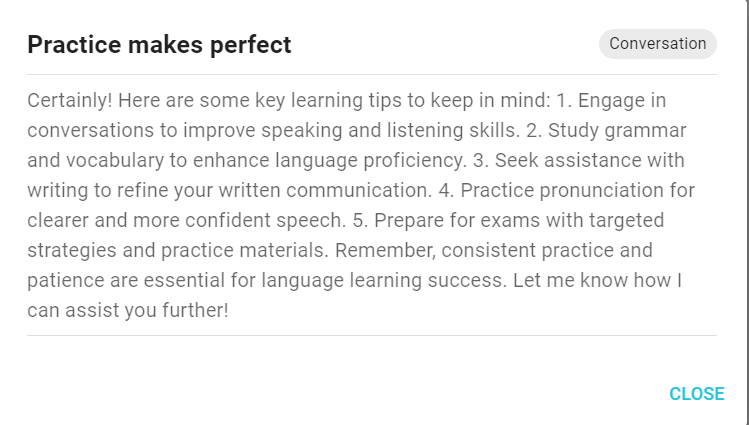
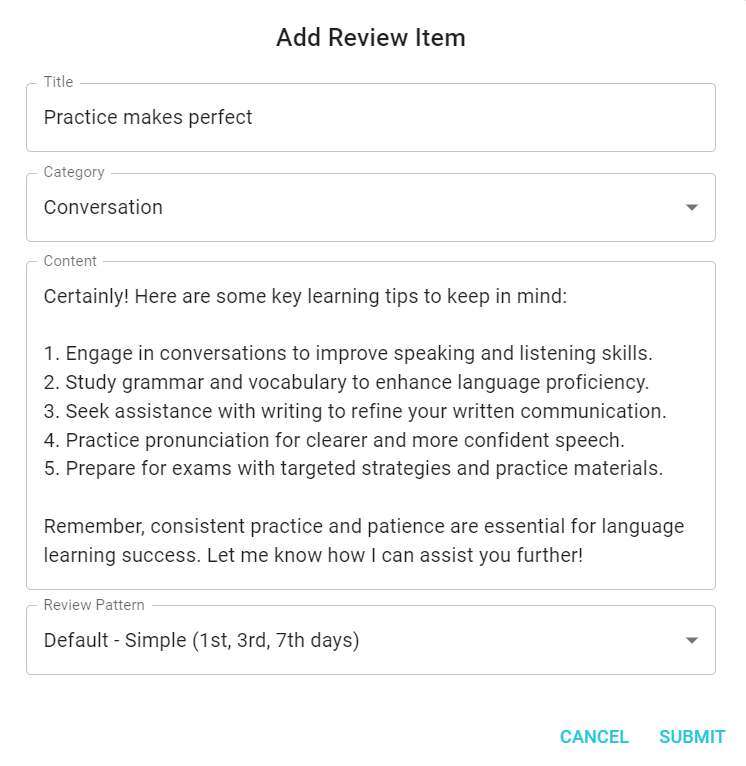


Figure 15: Content Display Format

# Conclusion

1.GPT and English learning: response time, voice template,

[1] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.

###### <Name of appendix>

<Use Heading 6 for the Appendix heading>

###### <Another appendix>