



University of Glasgow | School of  
Computing Science

# **LangBuddy: A Study Web App Powered by ChatGPT for EFL Learners**

Tiansheng Chen

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow G12 8RZ

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

1 September 2023

### **Abstract**

This dissertation describes the design and development of "LangBuddy", a web application tailored for English as a Foreign Language (EFL) learners. Leveraging the capabilities of ChatGPT, LangBuddy aims to enrich the English learning experience and facilitate effective study journal management. The application requirements are gathered through an extensive literature review, comparative analyses of analogous applications, and interviews with prospective users. The progress of development including architecture design, user interface and database design are demonstrated. This dissertation also explained the implementation of a monitoring learning system with spaced repetition strategy and chatbots powered by ChatGPT and voice interaction. Manual testing and user testing were adopted to explore the potential limitations of LangBuddy. Achievements, reflections and prospective improvements are also discussed.

## **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.

Name: Tiansheng Chen

Signature: Tiansheng Chen

## **Acknowledgements**

I would like to thank Dr. Sham Puthiya Parambath for his invaluable advice and guidance throughout the project. I would also like to say thanks to all the testers who supported this project.

I'm especially grateful to Ada, Daniel, Claire, Mark, John, Liam, and Lloyd. As an international student, their continuous emotional support and sharing insights into the culture of the UK and Scotland were vital. Thanks for standing by me through thick and thin.

Lastly, I want to express my profound thanks to my family and parents. Your understanding and unwavering belief in me have been the foundation upon which I've explored the endless possibilities of life, growing each day to become an awakened one.

# Contents

<b>Chapter 1</b>	<b>Introduction.....</b>	<b>1</b>
1.1	Aims and Objectives.....	1
1.2	Report Structure.....	1
<b>Chapter 2</b>	<b>Analysis and Requirements.....</b>	<b>3</b>
2.1	Background and Literature Review .....	3
2.2	Requirements Gathering.....	4
2.2.1	Analysis of Existing Applications .....	4
2.2.2	Interviews.....	6
2.3	Application Requirements.....	7
2.3.1	Functional Requirements.....	7
2.3.2	Non-Functional Requirements.....	8
<b>Chapter 3</b>	<b>Design and Implementation .....</b>	<b>10</b>
3.1	System Architecture.....	10
3.2	User Interface.....	12
3.2.1	Colour, Font and Style.....	12
3.2.2	Components and Pages .....	12
3.3	Database Design .....	16
3.4	Function Implementation.....	17
3.4.1	Structured Learning Courses.....	17
3.4.2	Study Plan System .....	17
3.4.3	Review System with Spaced Repetition Strategy .....	17
3.4.4	Chatbots.....	17
3.4.5	Security and Authentication.....	22
3.4.6	User Experience and Error Handling .....	22
<b>Chapter 4</b>	<b>Testing and Evaluation .....</b>	<b>23</b>
4.1	Unit Testing.....	23
4.2	Manual Testing.....	23
4.3	User Testing.....	24
4.4	Evaluation.....	24
4.4.1	Extreme Cases.....	25
4.4.2	Button Clicking Conflict.....	25
4.4.3	Voice Recognition.....	25
4.4.4	Interaction with Chatbots .....	26
4.4.5	Content Display Format .....	26
<b>Chapter 5</b>	<b>Conclusion.....</b>	<b>27</b>
5.1	Achievement .....	27
5.2	Reflection and future works .....	27
5.2.1	ChatGPT in ESL education.....	27
5.2.2	Technical Debt .....	28
5.2.3	Handling Long-form Input and Large Set of Data .....	29

5.3	Summary .....	29
Appendix A	Interview Questions .....	1
Appendix B	MoSCoW Prioritisation .....	2
Appendix C	User Stories .....	4
Appendix D	Entities Attributes and Constraints .....	6
Appendix E	Code Samples – Send Chat Message.....	8
Appendix F	User Experience and Error Handling Examples	
12		
Appendix G	Manual Test Cases.....	14
Appendix H	User Testing Results.....	17
Appendix I	Ethics Checklist.....	19

# Chapter 1 Introduction

In recent years, large language models (LLMs) have become pivotal in advancing natural language processing (NLP) and revolutionizing research and development in the domain. ChatGPT, short for Chat Generative Pre-trained Transformer, is a remarkable LLM system developed by OpenAI, launched on November 30, 2022. It excels in language generation, capturing significant public interest. Distinguishing features of ChatGPT include its accessibility, personalized interactions, conversational structure, and economic feasibility (Farrokhnia *et al.*, 2023).

Within education, both educators and students benefit from ChatGPT's myriad capabilities. It has been employed in diverse applications such as essay generation, bot responses, answering questions, text-based language and language translation (Murad, Surameery and Shakor, 2023). ChatGPT promotes engagement, encourages autonomous learning, and offers personalized resource recommendations, enriching the learning environment (Božić, 2023).

Recent studies indicate ChatGPT's efficacy in enhancing language skills. It aids learners by providing language feedback and recommending language practice activities (Bin-Hady *et al.*, 2023). For EFL (English as a Foreign Language) students and educators, ChatGPT augments teaching and learning experiences (Alhalangy and AbdAlgane, 2023). However, the potential limitations of ChatGPT in educational contexts have also sparked interest (Tlili *et al.*, 2023). This project delves into the pragmatic integration of ChatGPT in education, aiming to harness its capabilities for an EFL-focused learning application.

## 1.1 Aims and Objectives

The objective of this project is to develop a ChatGPT-driven web application tailored to support EFL students in their English learning journey. Additionally, it should offer a comprehensive platform for effective study task management.

The application aims to provide a set of chatbots with voice integration and translation functionalities which are valuable for EFL learners. Furthermore, it will encompass a study and review management system, which empowers users to customize their learning path and employ spaced repetition learning strategies for improving outcomes. An in-depth feature list will be presented in Chapter 2.

## 1.2 Report Structure

Chapter 2 establishes the application requirements by literature review, comparing similar applications and collecting opinions from prospective users. The rationale behind the chosen application features is also discussed.

Chapter 3 demonstrates the project's development cycle, outlining the system architecture design and expounding on the core feature implementation.

Chapter 4 elaborates on the rationale of testing the software and offers a comprehensive evaluation of the test outcomes.

Chapter 5 reflects on the project's achievement, challenges, and potential future works.



## Chapter 2 Analysis and Requirements

### 2.1 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 they transform and redefine traditional pedagogical practices (Klyshbekova, 2023). Benefits attributed to ChatGPT in EFL encompass personalized and interactive learning, the generation of adaptive learning materials, the provision of continuous and immediate feedback, the fostering of 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 improve 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 an obvious 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 persist. Despite ChatGPT's aptitude for transforming auxiliary learning materials into interactive study resources, the quality and accuracy of such narratives remain under scrutiny (Diwan *et al.*, 2023). ChatGPT, on occasion, 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 providing structured learning materials to complement ChatGPT's capabilities and deploying a combination of hidden prompts to guide ChatGPT and open prompts for influencing 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.

## 2.2 Requirements Gathering

Based on the previous literature review, this section will investigate and compare the 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.

### 2.2.1 Analysis of Existing Applications

According to the result of the 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 are 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 not just to assimilate accurate pronunciation but also to self-assess their progress by recording and comparing their accents over time.

**Langotalk**, an innovative start-up, 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 be based 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 of Existing Applications**

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

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 applications, 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, weak quality assurance for learning materials, and a systematic review mechanism somewhat diminishes 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.

### 2.2.2 Interviews

To gain richer insights into the experiences of perspective learners using EFL education software and ChatGPT, four participants were recruited to join a 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.

## 2.3 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*.

### 2.3.1 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 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 a statistical overview of a user's learning trajectory, enhancing their ability to strategize and plan.

#### 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 on the review schedule and amend content in the integration process.
- **Customizability:** While the system offers pre-set 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 based on the information provided by the platform.

### **2.3.2 Non-Functional Requirements**

The project must also fulfil 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 the loading state, which refers to the waiting for a response from the chatbot and the 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**

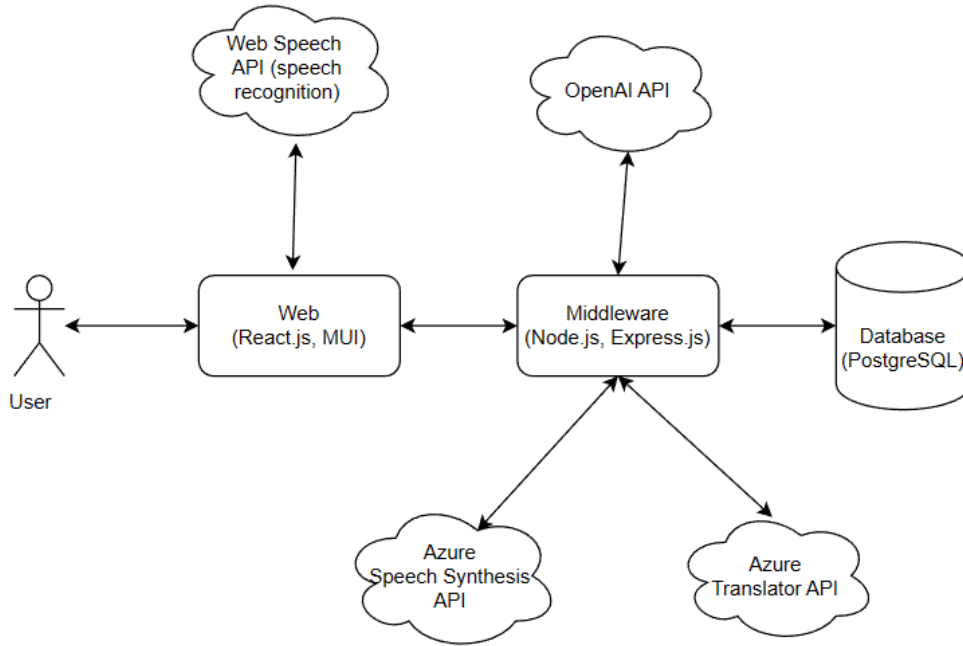
Given the reliance on third-party APIs, a reasonable restriction on user-chatbot interactions is enforced. As OpenAI API charges the usage based on tokens,

conversation points are proposed as a tool to limit the number of messages sent to chatbots. This will prevent abuse of these features and control the operational cost at an affordable level. Meanwhile, this limitation will also encourage the user to engage with the chatbot thoughtfully and responsibly.

## Chapter 3 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 also be elaborated.

### 3.1 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, which is 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 functions 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 modelled 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 the 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 to 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 the 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, a 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. As web speech API does not require authentication like an API key, it interacts with users in the frontend without the involvement of middleware.
- **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 outweigh 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.

## 3.2 User Interface

### 3.2.1 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 the 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 to reduce users' cognitive overload.

### 3.2.2 Components and Pages

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

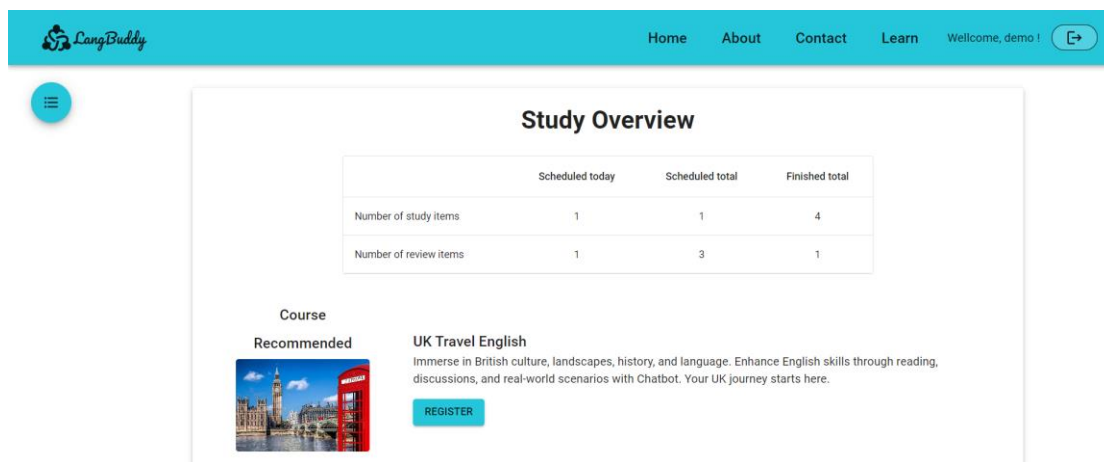


Figure 3: Dashboard Page

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.

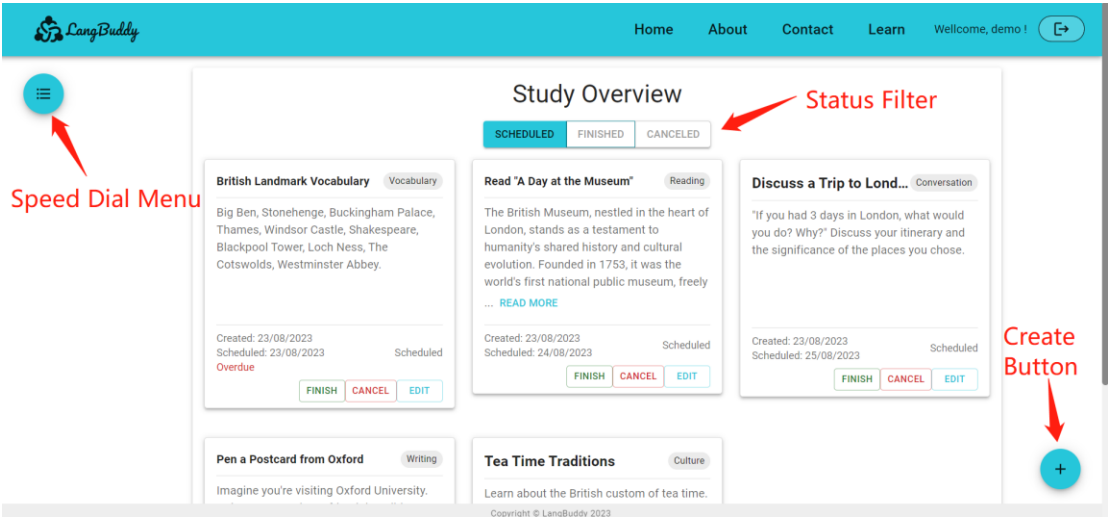


Figure 4: Study Overview Page

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 the 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.

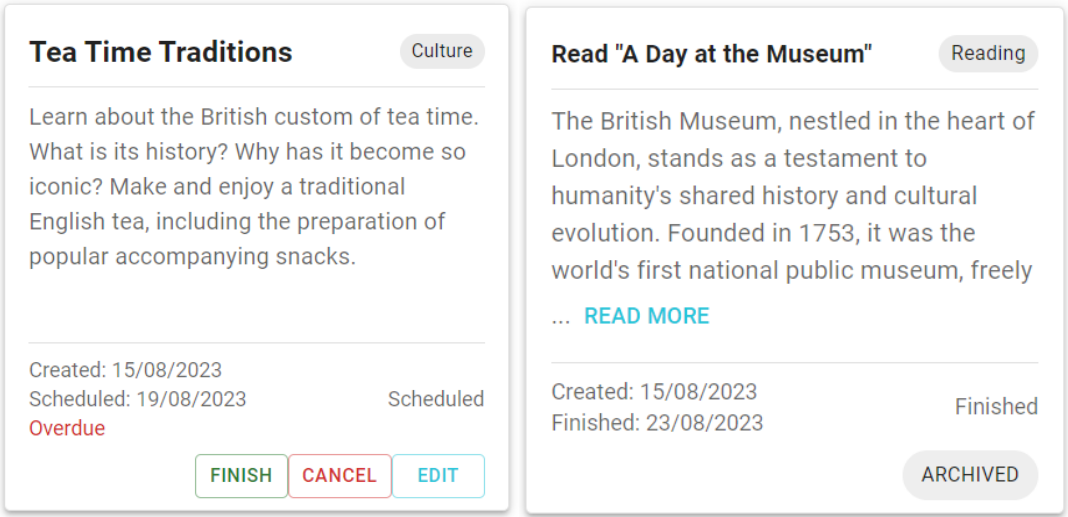


Figure 5: Study Card Component

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 functions, are available for user interaction. The study card will remind the user with red characters if it is

overdue compared to the scheduled date. For longer content, a "read more" option unfolds the content details in a modal view.

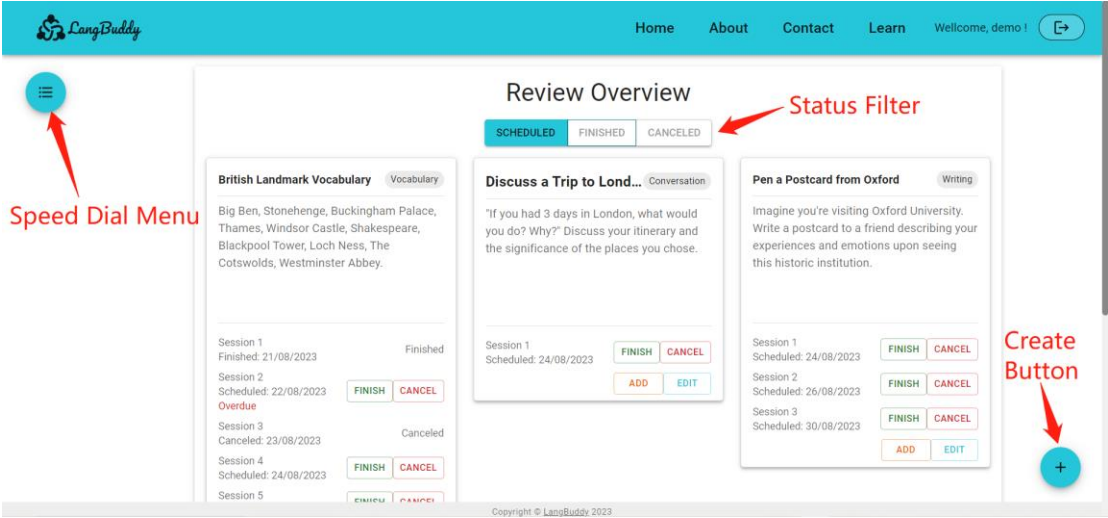


Figure 6: Review Overview Page

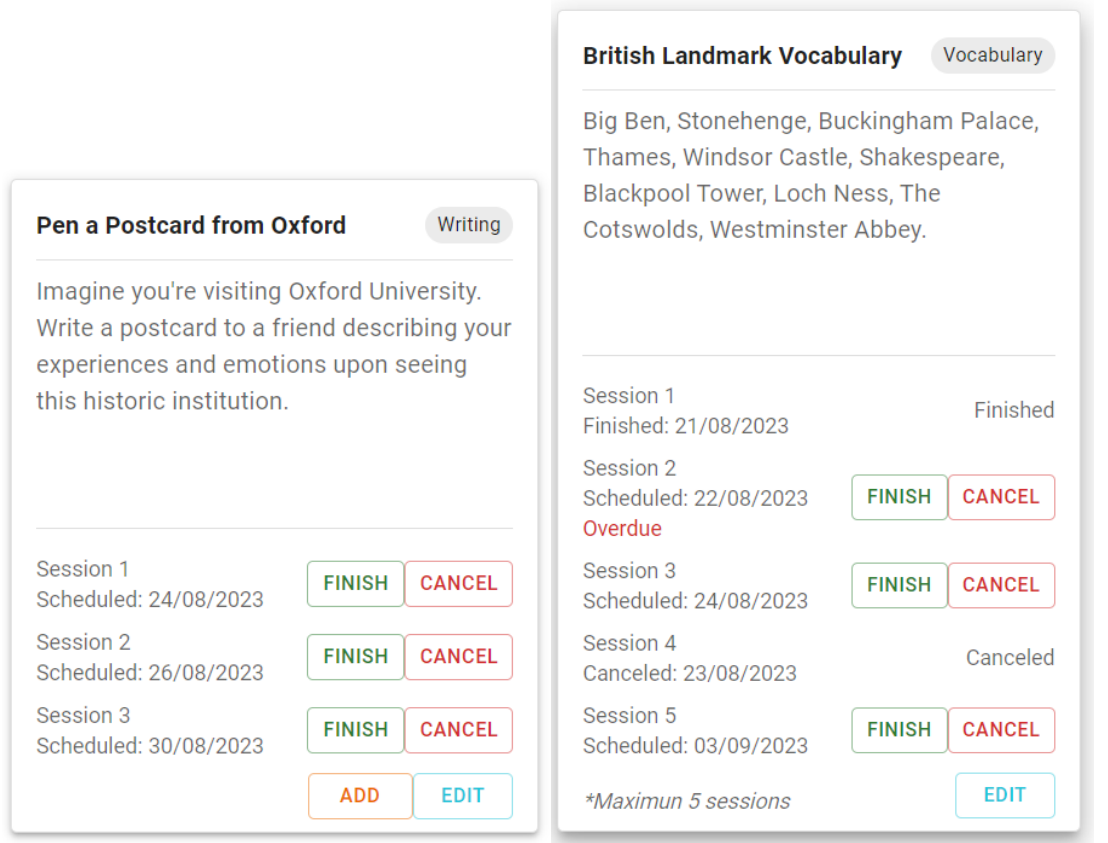
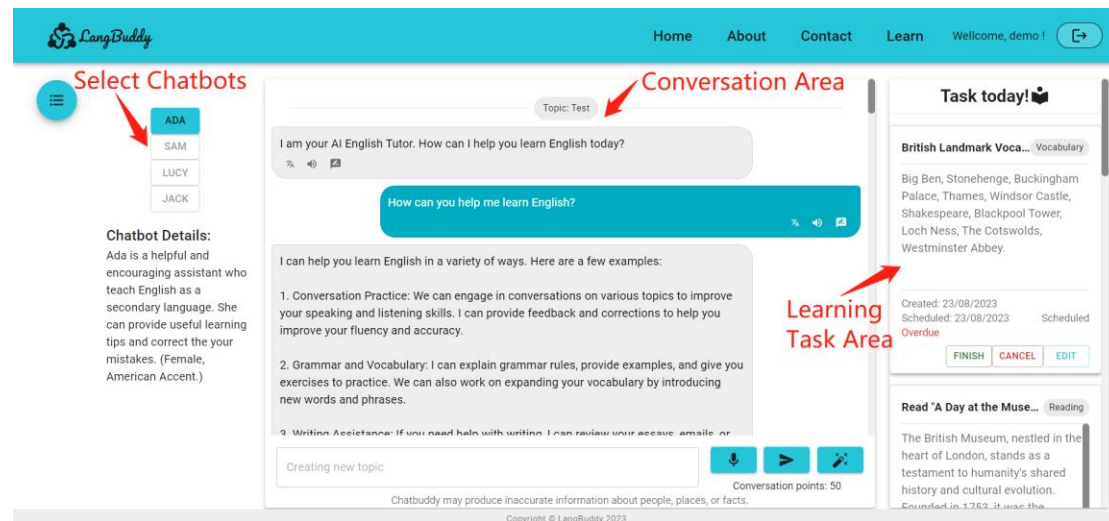


Figure 7: Review Card Component

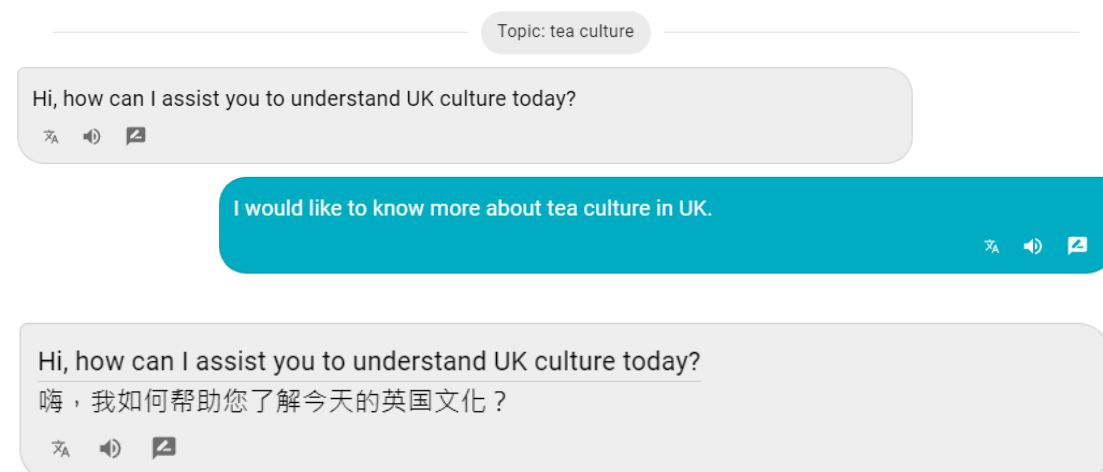
The review overview page mirrors the organizational structure of the study overview page, listing all review cards based on their status. However, the layout for individual review cards differs from that of study cards. In addition to standard details (title, category, content etc.), review cards showcase 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.

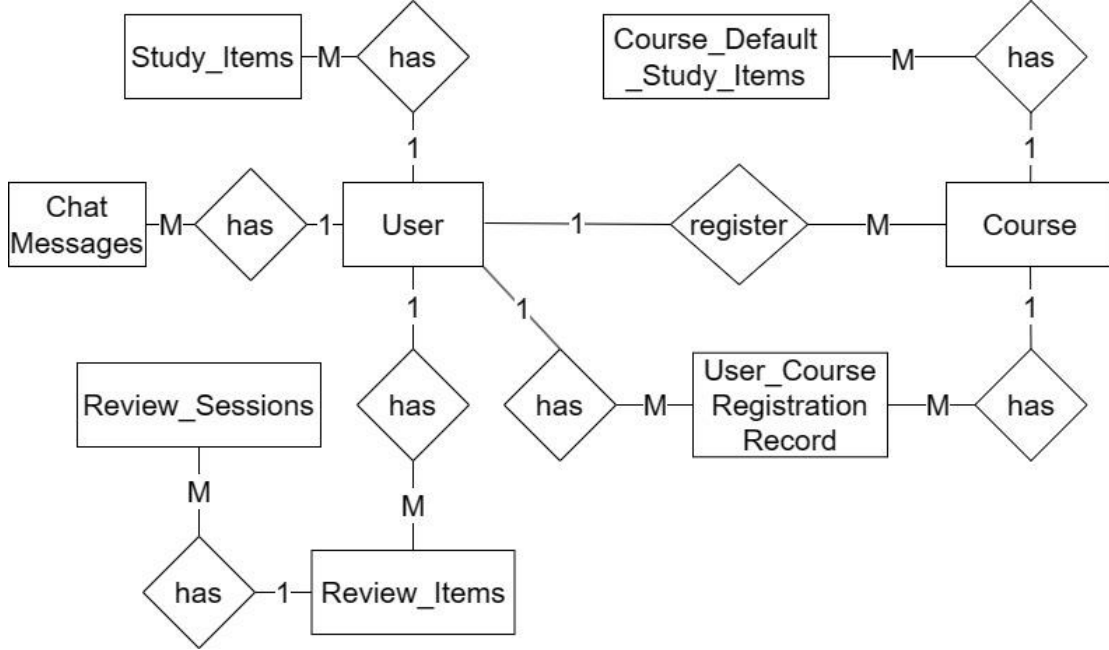


**Figure 9: Chatbox Component**

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 colour. 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.

### 3.3 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 with 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 based 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 manage operational expenses.

## **3.4 Function Implementation**

### **3.4.1 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 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.

### **3.4.2 Study Plan System**

Besides creating new study items by registering for 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.

### **3.4.3 Review System with Spaced Repetition Strategy**

As mentioned previously, the finished study items can be migrated into the 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 users 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 pace. The blend of predefined and customizable review patterns empowers users to chart their academic trajectory and optimize learning outcomes.

### **3.4.4 Chatbots**

#### **Integration with the 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 that are scheduled for 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 patterns 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 behaviour. The temperature, ranging between 0 and 2, governs the randomness of the 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. The 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, the 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 at 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 instructions from the user strictly. The randomness of Sam should be low and new topics should not be encouraged. Lucy specialises 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 and adjust the content and format of model responses. The system message usually not be shown to the users but is sent to the API with the 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 gives clear guidance to the users on how they should interact with the chatbots. For example, as a general purpose chatbot like Ada, the system message and initial open prompt are set to be more general guidance with encouraging character. The details of the prompt structure are shown in the figure below.

Hidden prompt:

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 politely." }
```

]

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 specialises in vocabulary and spaced repetition review strategy, few-shot examples are integrated into the system message as shown below. Moreover, Lucy adopts the few-shot prompts pattern similar to Sam and Jack uses a no-shot prompt pattern similar to Ada.

Hidden prompt:

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 a 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 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, if the system has over 20 conversation records for the current topic in the database, it retrieves the most recent 20 records. For fewer than 20 records, it retrieves all available conversations. It is assumed that the most recent 20 records can provide enough information for ChatGPT to

generate a qualified response in case of a long conversation history under the current topic. This collection of past messages, combined with the chatbot's system message which is identified by the 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 sent to the frontend for display. The details of the code are shown in Appendix E.

On the frontend, React's `useState` hook is utilized to oversee the chatbot's conversation history and the name of the 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 to 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 options provided at that time. Later, on August 22, 2023, OpenAI announced the availability of 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 fulfils the requirements of this application. So the fine-tuning method to build a chatbot is not adopted for this application.

### **3.4.5 Security and Authentication**

To ensure user security and authentication, this application use uses 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 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 a 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.

### **3.4.6 User Experience and Error Handling**

Visibility of system status, error prevention and guidance for handling errors 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 services, 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 disruptions 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 a 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 inputs invalid data, the form shows an error handling message which assists the user in dealing with the error.

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

## Chapter 4 Testing and Evaluation

### 4.1 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 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 the 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, so 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 constraints of development time and the inherent learning curve, emphasis should be placed on manual testing and user testing. However, it would be good practice to consider these tests in 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 the frontend and backend are integrated.

### 4.2 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.

### 4.3 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 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 complete these tasks, the participants are also allowed to explore the application based on their interests. Then a debrief session was arranged for them to address any concerns and provide clarity for filling out a questionnaire. The evaluation questionnaire 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 appeal, ease of use, helpfulness, efficiency and overall satisfaction. The third segment comprised two open-ended questions to collect participants' opinions 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.

### 4.4 Evaluation

Within the realm of manual testing, most of the positive test cases were successfully passed. Nevertheless, challenges arose with extreme test cases, especially with long-form input data that wasn't handled appropriately. Additionally, issues about the lack of validation for invalid URL addresses and button conflicts were identified.

Regarding the user testing, participants predominantly provided positive feedback concerning the ease of task execution and the application's overall performance dimensions. No major bugs that could significantly hamper user experience were reported. However, certain users articulated their confusion regarding the speech recognition functionality and optimal interaction methods

with the chatbot. After the testing phase, most of these issues were rectified. Specific concerns and potential solutions are discussed below:

#### 4.4.1 Extreme Cases

As an interactive application, this website demands significant user data input. Ensuring data validation is paramount. During manual testing, it was observed that the application didn't handle the large amount of data input (long-form data) input appropriately in certain fields. For instance, OpenAI's API, depending on its model, has token limitations. If a user's message surpasses this limit, the API responds with an error that isn't efficiently handled by the backend. The solution to this problem may be imposing character restrictions within the frontend input space and instituting a mechanism to capture error responses from external APIs.

```
{
  error: {
    message: "This model's maximum context length is 4097 tokens. However, your messages resulted in 8104 tokens. Please reduce the length of the messages.",
    type: 'invalid_request_error',
    param: 'messages',
    code: 'context_length_exceeded'
  }
}
```

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

Another prevalent challenge with long-form data emerged in managing study and review items. Inputting extended data in the title or content fields triggers a red colour in the form, indicating an error. However, the application fails to display a specific error message, guiding users on rectification. Incorporating such an informative prompt would enhance user experience.

Lastly, if there is a large set of data in study and review tasks, the browser may experience performance issues. Implementing a pagination function is recommended to alleviate this issue in the application.

#### 4.4.2 Button Clicking Conflict

As the loading state frequently happens for the buttons, it is essential to test any side effects that happen 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 the sending message button. To solve this issue, the text state hook should be set to null when the recording function is called.

#### 4.4.3 Voice Recognition

Most of the participants found it challenging to identify the termination point of voice recognition after activating the recording button. The recording function is designed to conclude based on web speech API's automatic recognition. Nonetheless, users felt the interface lacked clarity in this progression. An effective remedy would involve transitioning the loading icon button to a stop recording icon during the recording phase and granting users autonomy to manually terminate the recording process.

#### 4.4.4 Interaction with Chatbots

Certain users unfamiliar with ChatGPT encountered difficulties in crafting effective prompts to elicit desired responses from the chatbot. Misconceptions, such as expecting the chatbot to recall prior interactions or generate highly specialized responses, were also noted. Besides educating users, embedding a prompt manager within the system could improve this experience.

#### 4.4.5 Content Display Format

Another issue found is the inconsistent display format of review and study content. Even when users structured their input with line breaks, the rendered content manifested as an uninterrupted block, disregarding these breaks. Exploring solutions to standardize content formatting is recommended.

The figure displays two side-by-side screenshots of a user interface. The left screenshot shows a form titled "Add Review Item". It contains the following fields: "Title" (Practice makes perfect), "Category" (Conversation), "Content" (Certainly! Here are some key learning tips to keep in mind: 1. Engage in conversations to improve speaking and listening skills. 2. Study grammar and vocabulary to enhance language proficiency. 3. Seek assistance with writing to refine your written communication. 4. Practice pronunciation for clearer and more confident speech. 5. Prepare for exams with targeted strategies and practice materials. Remember, consistent practice and patience are essential for language learning success. Let me know how I can assist you further!), and "Review Pattern" (Default - Simple (1st, 3rd, 7th days)). At the bottom are "CANCEL" and "SUBMIT" buttons. The right screenshot shows the content of the form displayed in a card format. The title "Practice makes perfect" is at the top right, next to a "Conversation" category tag. The content area contains the same text as the form, but the list of five tips is rendered as a single, uninterrupted block of text. At the bottom right of the card is a "CLOSE" button.

**Figure 15:** Content Display Format



## **Chapter 5 Conclusion**

### **5.1 Achievement**

The primary aim of this project was to create a ChatGPT-powered web application designed to assist EFL students in their English learning journey and provide an integrated platform for effective study journal management. This application has successfully met all the "must", "should", and "could" requirements as outlined in the MoSCoW analysis and user stories in the appendix.

The platform offers structured sample courses tailored to users' individual learning needs. Upon enrolment, default learning tasks from these courses are imported into the study system. Users have the flexibility to create, modify, and reschedule these tasks. Once a study task is completed, it can be transitioned to the review system, where a spaced repetition strategy can be applied for effective retention. The review system also offers customization options, and users can monitor their learning progress statistics.

To enrich the learning experience, diverse chatbots are integrated into this application. Each chatbot serves specific learning objectives. Users can engage using voice for both input and output for better learning experiences. The translation capability augments the application functionality. Daily study and review items are seamlessly presented, linking the chatbots page with the learning system. Additionally, users have the option to integrate chatbot conversations into the review system when needed.

### **5.2 Reflection and future works**

#### **5.2.1 ChatGPT in ESL education**

Recent studies have shed light on the potential of ChatGPT in language teaching and learning. Since ChatGPT's beta release in November 2022, its implications for education have been widely debated. Many see its transformative potential in education, while others emphasize potential drawbacks, such as ethical concerns, accuracy, and cultural biases (Kohnke, Moorhouse and Zou, 2023). However, the project's development experience has enriched my understanding of ChatGPT's role in ESL education.

#### **Digital Competency and Prompt Engineering**

To effectively utilize ChatGPT in education, users need to cultivate specific digital competencies. This includes mastering interactions with ChatGPT and facilitating pedagogically beneficial tasks while remaining conscious of its limitations. Non-technical users might undergo a learning curve to draft appropriate prompts for language learning with ChatGPT (Woo, Guo, and Hengky Susanto, 2023). Therefore, it's essential for our application to offer guidance on interacting with ChatGPT and managing potential issues.

In addition to providing educational resources on how to use ChatGPT in this application, prompt engineering can enhance user experience. It is proposed to add a prompt manager to the application. The prompt manager offers users structured and predefined prompts, aiding effective interactions with different chatbots for desired responses.

### **Fine-tuning the GPT Model**

The application's learning content dictates a particular learning scope. Given the predictability of user interactions and the content covered, it is possible to create bespoke datasets based on this content to fine-tune a custom GPT model. This refinement can significantly enhance the GPT model's responsiveness and elevate user experience.

While the fine-tuned model caters to specific learning content, chatbots addressing broader learning needs are equally crucial. Given the predictability of user learning behaviour, prompting strategies like few-shot learning and chain-of-thought can further enhance the GPT model's response quality.

### **Voice Interaction**

As language learning encompasses speaking and listening, the voice integration with ChatGPT strongly impacts learning experiences. User testing revealed ambiguities in the audio recording conclusion process. It is suggested to change the loading icon to a stop recording icon and allow users to control the recording process autonomously.

Currently, this application offers four voices across two accents (UK and USA). For a richer experience, future iterations should integrate a wider array of voices, making interactions more engaging and lifelike.

### **5.2.2 Technical Debt**

Technical debt, in software development, describes the potential costs associated with opting for a quicker, simpler approach now, rather than adopting a more appropriate or sophisticated solution that might take longer. Accumulated technical debt can result in additional work in the future, making subsequent changes more challenging to implement. This application, developed in JavaScript without an external React State management library, began to exhibit symptoms of growing technical debt as the codebase expanded and as more states were needed to fulfil application requirements.

### **JavaScript vs TypeScript**

JavaScript is a high-level, interpreted scripting language characterized by its dynamic typing. This dynamism means that variables don't have a fixed type; a variable that is initially assigned a string can later be assigned a number or an object. Although JavaScript is robust and versatile for web application development, its lack of type annotations becomes a limitation. Especially as the codebase enlarges, maintenance becomes complicated due to its dynamic nature.

To solve the code maintenance problem for the large codebase, TypeScript is proposed as a potential tool for refactoring. TypeScript extends JavaScript by

introducing optional static typing, interfaces, generics, enums, decorators, and other features that streamline development. As this application becomes increasingly complex, TypeScript can enhance maintainability, offering tools that aid in implementing design patterns to solve complex problems.

## **React State Management**

State management is a critical aspect of React applications, especially as they grow in complexity. While React has its own built-in `useState` and `useContext` hooks for state management, these tools can be inadequate for more intricate applications. As development progressed, the application involved a huge amount of state management, which caused prop drilling (such as review session component) and numerous shared states (such as snack bars) among components. Furthermore, frequent syncing of frontend states with backend data also makes state management more complex.

The React State management library, like Redux or MobX, can be introduced to simplify and structure state handling as a solution in the future. These libraries centralize state storage, utilize actions to describe state changes and employ reducers to manage those changes. As the application becomes larger and more complex in the future, the adoption of a management library will undoubtedly aid in maintaining the codebase more efficiently.

### **5.2.3 Handling Long-form Input and Large Set of Data**

As addressed in the evaluation section, the application exhibits challenges when managing long-form input data. Beyond imposing restrictions and validating input length, it's essential to provide users with clear instructions on rectifying errors. Prioritizing improvements for long-form input data should be scheduled for future development. A large set of data in study and review tasks can impact the browser's performance. Implementing a pagination function in this application is recommended.

## **5.3 Summary**

This project explored the possibility and defined the application requirements for constructing a ChatGPT-powered web application aimed at aiding EFL students in effectively managing their study journals. The application was developed using React and MUI for the frontend, while employing Node.js, Express, and PostgreSQL for the backend. The structured course, study system and review system provide a robust ecosystem for users to efficiently navigate their learning journey. Enhanced by external APIs, the integrated chatbots support voice interactions and provide a translation function to assist users in learning English. Users can seamlessly transition between learning tasks, review systems, and chatbot interactions. Manual testing and user testing are implemented to explore the limitations of this application. Reflection and future works are also discussed.

## Bibliography

Alhalangy, Abdalilah.G.I. and AbdAlgane, M. (2023) 'Exploring the Impact of AI on The EFL Context: A Case Study of Saudi Universities', *Journal of Intercultural Communication*, pp. 41–49. Available at: <https://doi.org/10.36923/jicc.v23i2.125>.

Ali, J.K.M. *et al.* (2023) 'Impact of ChatGPT on Learning Motivation:: Teachers and Students' Voices', *Journal of English Studies in Arabia Felix*, 2(1), pp. 41–49. Available at: <https://doi.org/10.56540/jesaf.v2i1.51>.

Bin-Hady, W. *et al.* (2023) *Exploring the Dimensions of ChatGPT in English Language Learning: A Global Perspective*.

Božić, V. (2023) 'Chat GPT and education'. Available at: <https://doi.org/10.13140/RG.2.2.18837.40168>.

Canva (2023) *Everything about the colour Cyan*, *Canva's Design Wiki*. Available at: <https://www.canva.com/colors/color-meanings/cyan/> (Accessed: 23 August 2023).

Chukharev-Hudilainen, E. and Klepikova, T.A. (2016) 'The effectiveness of computer-based spaced repetition in foreign language vocabulary instruction: a double-blind study', *CALICO Journal*, 33(3), pp. 334–354.

Diwan, C. *et al.* (2023) 'AI-based learning content generation and learning pathway augmentation to increase learner engagement', *Computers and Education: Artificial Intelligence*, 4, p. 100110. Available at: <https://doi.org/10.1016/j.caeai.2022.100110>.

Duolingo (2023) *Learn a language for free*, *Duolingo*. Available at: <https://www.duolingo.com/> (Accessed: 21 August 2023).

Express (2023) *Express - Node.js web application framework*. Available at: <https://expressjs.com/> (Accessed: 23 August 2023).

Farrokhnia, M. *et al.* (2023) 'A SWOT analysis of ChatGPT: Implications for educational practice and research', *Innovations in Education and Teaching International*, 0(0), pp. 1–15. Available at: <https://doi.org/10.1080/14703297.2023.2195846>.

Fitria, T.N. (2023) 'Artificial intelligence (AI) technology in OpenAI ChatGPT application: A review of ChatGPT in writing English essay', *ELT Forum: Journal of English Language Teaching*, 12(1), pp. 44–58. Available at: <https://doi.org/10.15294/elt.v12i1.64069>.

Glossika (2023) *Glossika Language Learning | Speak with Confidence*. Available at: <https://ai.glossika.com/> (Accessed: 21 August 2023).

Han, J. *et al.* (2023) 'RECIPE: How to Integrate ChatGPT into EFL Writing Education', *Proceedings of the Tenth ACM Conference on Learning @ Scale*, pp. 416–420. Available at: <https://doi.org/10.1145/3573051.3596200>.

Jakob, N. (2020) *10 Usability Heuristics for User Interface Design*, Nielsen Norman Group. Available at: <https://www.nngroup.com/articles/ten-usability-heuristics/> (Accessed: 27 August 2023).

Kang, S.H.K. (2016) 'Spaced Repetition Promotes Efficient and Effective Learning: Policy Implications for Instruction', *Policy Insights from the Behavioral and Brain Sciences*, 3(1), pp. 12–19. Available at: <https://doi.org/10.1177/2372732215624708>.

Klyshbekova, M. (2023) 'ChatGPT and Language Learning'. Rochester, NY. Available at: <https://doi.org/10.2139/ssrn.4488587>.

Kohnke, L., Moorhouse, B.L. and Zou, D. (2023) 'ChatGPT for Language Teaching and Learning', *RELC Journal*, p. 00336882231162868. Available at: <https://doi.org/10.1177/00336882231162868>.

Langotalk (2023) *Langotalk: Learn Languages 6x Faster with AI*, Langotalk. Available at: <https://www.langotalk.org> (Accessed: 21 August 2023).

Microsoft Azure (2023a) *Text to Speech – Realistic AI Voice Generator | Microsoft Azure*. Available at: <https://azure.microsoft.com/en-us/products/ai-services/text-to-speech> (Accessed: 23 August 2023).

Microsoft Azure (2023b) *Translator – Translation Software as a Service | Microsoft Azure*. Available at: <https://azure.microsoft.com/en-us/products/ai-services/translator> (Accessed: 23 August 2023).

Mozilla (2023) *Web Speech API - Web APIs | MDN*. Available at: [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Speech\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API) (Accessed: 23 August 2023).

MUI (2023a) *MUI: The React component library you always wanted*. Available at: <https://mui.com/> (Accessed: 23 August 2023).

MUI (2023b) *Testing - Material UI*. Available at: <https://mui.com/material-ui/guides/testing/> (Accessed: 27 August 2023).

Murad, I.A., Surameery, N.M.S. and Shakor, M.Y. (2023) 'Adopting ChatGPT to Enhance Educational Experiences', *International Journal of Information Technology & Computer Engineering (IJITC) ISSN: 2455-5290*, 3(05), pp. 20–25. Available at: <https://doi.org/10.55529/ijitc.35.20.25>.

Nguyen, T.T.H. (2023) 'EFL Teachers' Perspectives toward the Use of ChatGPT in Writing Classes: A Case Study at Van Lang University', *International Journal of Language Instruction*, 2(3), pp. 1–47. Available at: <https://doi.org/10.54855/ijli.23231>.

OpenAI (2023) *OpenAI Platform*. Available at: <https://platform.openai.com> (Accessed: 23 August 2023).

React (2023) *React*. Available at: <https://react.dev/> (Accessed: 23 August 2023).

Schmidt-Fajlik, R. (2023) 'ChatGPT as a Grammar Checker for Japanese English Language Learners: A Comparison with Grammarly and ProWritingAid',

*AsiaCALL Online Journal*, 14(1), pp. 105–119. Available at: <https://doi.org/10.54855/acoj.231417>.

suhridpalsule and eric-urban (2023) *Prompt engineering techniques with Azure OpenAI - Azure OpenAI Service*. Available at: <https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/advanced-prompt-engineering> (Accessed: 25 August 2023).

Tlili, A. *et al.* (2023) ‘What if the devil is my guardian angel: ChatGPT as a case study of using chatbots in education’, *Smart Learning Environments*, 10(1), p. 15. Available at: <https://doi.org/10.1186/s40561-023-00237-x>.

Topsakal, O. and Topsakal, E. (2022) ‘Framework for A Foreign Language Teaching Software for Children Utilizing AR, Voicebots and ChatGPT (Large Language Models)’, *The Journal of Cognitive Systems*, 7(2), pp. 33–38. Available at: <https://doi.org/10.52876/jcs.1227392>.

White, J. *et al.* (2023) ‘A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT’. Available at: <https://doi.org/10.48550/ARXIV.2302.11382>.

Woo, D.J., Guo, K., and Hengky Susanto (2023) ‘Cases of EFL Secondary Students’ Prompt Engineering Pathways to Complete a Writing Task with ChatGPT’. Available at: <https://doi.org/10.13140/RG.2.2.31464.85762>.

Xiao, C. *et al.* (2023) ‘Evaluating Reading Comprehension Exercises Generated by LLMs: A Showcase of ChatGPT in Education Applications’, in *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*. *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, Toronto, Canada: Association for Computational Linguistics, pp. 610–625. Available at: <https://doi.org/10.18653/v1/2023.bea-1.52>.

Yang, J., Oh, K. and Youm, K. (2016) ‘English Vocabulary Learning System Based on Repetitive Learning and Rate-Matching Rule’, in A. De Gloria and R. Veltkamp (eds) *Games and Learning Alliance*. Cham: Springer International Publishing (Lecture Notes in Computer Science), pp. 189–195. Available at: [https://doi.org/10.1007/978-3-319-40216-1\\_20](https://doi.org/10.1007/978-3-319-40216-1_20).

Young, J.C. and Shishido, M. (2023) ‘Investigating OpenAI’s ChatGPT Potentials in Generating Chatbot’s Dialogue for English as a Foreign Language Learning’, *International Journal of Advanced Computer Science and Applications*, 14(6). Available at: <https://doi.org/10.14569/IJACSA.2023.0140607>.

## Appendix A Interview Questions

Questions used to conduct User Requirement interviews.

1. Could you briefly introduce yourself and describe your current fluency level in English?
2. What challenges have you encountered while learning English?
3. Which EFL (English as a Foreign Language) education software or apps have you used in the past?
4. Can you share your experience using these EFL education software or apps? Were they helpful? Why?
5. In what ways do you think EFL education software could be improved?
6. Are you familiar with or have you used ChatGPT? If yes, have you used it to assist with your English learning? If not, why?
7. In your opinion, how could ChatGPT be further tailored to support English learners?
8. What specific features do you desire in an EFL education software?

## Appendix B MoSCoW Prioritisation

Below are the prioritized requirements for the application:

### **Must have:**

- Users must be able to register/log in to the account to track their learning process.
- Users must be able to practice English with AI Chatbot.
- Users must be able to use the text and audio input to interact with AI Chatbot.
- The Chatbot must be able to generate responses in text and audio formats.
- Users must be able to make CRUD(create, read, update, delete) operations in the study plan systems.
- Users must be able to make CRUD(create, read, update, delete) operations in the review mechanism systems.
- The review system must be able to generate a default spaced repetition strategy review schedule for users.
- The review system must be able to remind users to review content upon logging into the website.

### **Should have:**

- Users should be able to select their preferred input format (text or audio) for the AI chatbot.
- User should be able to select their preferred output (text or audio) from the AI chatbot.
- The web app should incorporate a prompts management mechanism to improve the performance of the AI chatbot.
- The users should be able to select diverse chatbots for various learning purposes.
- The review system should provide overview statistics of review tasks.
- The study plan system should provide overview statistics of study plans.



- The website should feature a user-friendly interface and design.
- The system should monitor the usage frequency of the AI chatbot.
- Both visitors and users could be able to view product features, contact methods, basic policy information and other relevant details.
- The user should be able to register for courses and create study tasks automatically.
- The user should be able to decide on the review item when the study task is finished.
- The user should be able to add the conversation content to the review system.

**Could have:**

- The users could be able to be reminded of what study and review task should be finished today.
- The users could be able to view the overview statics of the study plan categorized by status.
- The users could be able to view the overview statics of review items categorized by status.

**Would have:**

- The system would provide a wide spectrum of EFL courses.
- The system would provide various user interface themes.
- Users would be able to select different study task and review item according to customized criteria.

## Appendix C User Stories

### Must-Have:

- As a user, I want to register and log in with my personal details to track my learning process.
- As a user, I want to practice English with an AI Chatbot.
- As a user, I want to interact with the AI Chatbot using both text and audio inputs.
- As a user, I want to receive the responses from Chatbot in both text and audio formats.
- As a user, I want to create, read, update, and delete items within the study plan system.
- As a user, I want to perform CRUD operations (create, read, update, delete) within the review mechanism system.
- As a user, I want to create a default spaced repetition strategy schedule for my reviews.
- As a user, I want to be reminded to review content upon logging into the website.

### Should Have:

- As a user, I want to choose whether I input text or audio when interacting with the AI chatbot.
- As a user, I want to select if I receive chatbot responses in text or audio format.
- As a user, I want to see the prompts management mechanism function to enhance the performance of AI chatbot.
- As a user, I want to choose from various chatbots tailored for different learning purposes.
- As a user, I want to view comprehensive statistics of my review tasks in the review system.
- As a user, I want to see an overview of my study plans in the study plan system.
- As a user, I want to enroll in courses and have study tasks created for me automatically.

- As a user, I want to decide which items to review after completing a study task.
- As a user, I want to add content from my chatbot interactions to the review system.

**Could Have:**

- As a user, I want to see reminders about study and review tasks due for the day.
- As a user, I want to view statistics of my study plan broken down by status.
- As a user, I want to view statistics of my review items, categorized by their statuses.

**Would Have:**

- As a user, I want to select a broad range of EFL courses through the system.
- As a user, I want to be offered multiple user interface themes for personalization.
- As a user, I want to select study tasks and review items based on custom criteria.

## Appendix D Entities Attributes and Constraints

The entities' attributes and constraints are shown in SQL format:

```
CREATE TABLE Users (  
  id_user UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  user_email VARCHAR(40) NOT NULL UNIQUE CHECK (user_email LIKE  
'%@%'),  
  password_hash VARCHAR(60) NOT NULL,  
  created_date TIMESTAMP NOT NULL,  
  updated_date TIMESTAMP,  
  mother_language VARCHAR(20) NOT NULL,  
  username VARCHAR(40) NOT NULL,  
  conversation_points INT DEFAULT 50,  
  last_message_date DATE  
);
```

```
CREATE TABLE Study_Items (  
  id_study SERIAL PRIMARY KEY,  
  id_user UUID NOT NULL REFERENCES Users(id_user) ON DELETE  
CASCADE,  
  category VARCHAR(20) NOT NULL CHECK (category IN ('Vocabulary',  
'Reading', 'Listening', 'Conversation', 'Writing', 'Culture', 'Other')),  
  title VARCHAR(50) NOT NULL,  
  content VARCHAR(2000),  
  created_date TIMESTAMP NOT NULL,  
  scheduled_date DATE NOT NULL,  
  finished_date DATE,  
  status VARCHAR(20) NOT NULL CHECK (status IN ('Scheduled', 'Finished',  
'Canceled'))  
);
```

```
CREATE TABLE Review_Items (  
  id_review SERIAL PRIMARY KEY,  
  id_user UUID NOT NULL REFERENCES Users(id_user) ON DELETE  
CASCADE,  
  category VARCHAR(20) NOT NULL CHECK (category IN ('Vocabulary',  
'Reading', 'Listening', 'Conversation', 'Writing', 'Culture', 'Other')),  
  title VARCHAR(50) NOT NULL,  
  content VARCHAR(2000),  
  created_date TIMESTAMP NOT NULL  
);
```

```
CREATE TABLE Review_Sessions (  
  id_session SERIAL PRIMARY KEY,  
  id_review INTEGER NOT NULL REFERENCES Review_Items(id_review) ON  
DELETE CASCADE,  
  created_date TIMESTAMP NOT NULL,  
  scheduled_date DATE NOT NULL,  
  finished_date DATE,
```

```

    status VARCHAR(20) NOT NULL CHECK (status IN ('Scheduled', 'Finished',
'Canceled'))
);

```

```

CREATE TABLE ChatMessages (
    id_message SERIAL PRIMARY KEY,
    id_user UUID NOT NULL REFERENCES Users(id_user) ON DELETE
CASCADE,
    created_date TIMESTAMP NOT NULL,
    chatbot_name VARCHAR(100) NOT NULL,
    role VARCHAR(50) NOT NULL CHECK (status IN ('user', 'assistant', 'topic')),
    content TEXT NOT NULL,
    prompt_tokens INT,
    completion_tokens INT
);

```

```

CREATE TABLE Courses (
    id_course SERIAL PRIMARY KEY,
    title VARCHAR(100) UNIQUE NOT NULL,
    description VARCHAR(1000) NOT NULL,
    created_date TIMESTAMP NOT NULL
);

```

```

CREATE TABLE Course_Default_Study_Items (
    id_default_study SERIAL PRIMARY KEY,
    id_course INTEGER NOT NULL REFERENCES Courses(id_course) ON
DELETE CASCADE,
    category VARCHAR(20) NOT NULL CHECK (category IN ('Vocabulary',
'Reading', 'Listening', 'Conversation', 'Writing', 'Culture', 'Other')),
    title VARCHAR(50) NOT NULL,
    content VARCHAR(2000) NOT NULL,
    created_date TIMESTAMP NOT NULL,
    scheduled_date_offset INTEGER NOT NULL
);

```

```

CREATE TABLE User_Courses (
    id_user_course SERIAL PRIMARY KEY,
    id_user UUID NOT NULL REFERENCES Users(id_user) ON DELETE
CASCADE,
    id_course INTEGER NOT NULL REFERENCES Courses(id_course) ON
DELETE CASCADE,
    registration_date TIMESTAMP NOT NULL,
    UNIQUE (id_user, id_course)
);

```

## Appendix E Code Samples – Send Chat Message

The following code shows how to send chat message to OpenAI API and handle the response in the backend.

```
//Send message to openAI API and get Chatbot response
app.post('/chatbot/:selectedChatbot', async (req, res) => {
  const client = await pool.connect();
  try {

    const token = req.cookies.token;

    if (!token) {
      return res.status(401).json({ message: "No token
provided" });
    }

    // Verify token and get user id
    const decoded = await jwtVerify(token, process.env.JWT_SECRET);
    if (!decoded || !decoded.user_id) {
      return res.status(401).json({ message: "Invalid token" });
    }
    const userId = decoded.user_id;

    // Check current conversation points
    const pointCheck = await client.query(
      "SELECT conversation_points, last_message_date FROM Users
WHERE id_user = $1",
      [userId]
    );

    const currentPoints = pointCheck.rows[0].conversation_points;

    if (currentPoints <= 0) {
      throw new Error("Insufficient conversation points");
    }

    // Get Last message date and update conversation points if it's
a new day
    const lastMessageDate = pointCheck.rows[0].last_message_date;
    const currentDate = new Date();

    if (!lastMessageDate || lastMessageDate.toDateString() !==
currentDate.toDateString()) {
      // If it's a new day, reset the conversation points to 50
and update the last_message_date
      await pool.query(
```

```

        "UPDATE Users SET conversation_points = 50,
last_message_date = NOW() WHERE id_user = $1",
        [userId]
    );
}

const { content } = req.body;

if(!content){
    throw new Error("No content provided");
}

const { selectedChatbot } = req.params;

if(!selectedChatbot){
    throw new Error("No chatbot provided");
}

//Get the chatbot model information by name
const chatbotModel = getChatbotModel(selectedChatbot);

//Begin transaction
await client.query('BEGIN');

//insert user message into database
const insertText = 'INSERT INTO ChatMessages(id_user,
created_date, chatbot_name, role, content) VALUES($1, NOW(), $2, $3,
$4)';
const insertValues = [userId, chatbotModel.name, 'user',
content];
await client.query(insertText, insertValues);

// Fetch the Last 20 messages in current topic from the
database
const fetchText = `
    SELECT role, content
    FROM ChatMessages
    WHERE id_user = $1
    AND chatbot_name = $3
    AND created_date > (
        SELECT MAX(created_date)
        FROM ChatMessages
        WHERE id_user = $1
        AND role = $2 And chatbot_name = $3
    )
    ORDER BY created_date DESC, id_message DESC
    LIMIT 20;
`;

```

```

const fetchValues = [userId, 'topic', chatbotModel.name];
const fetchResult = await client.query(fetchText, fetchValues);

//add the system messages to the messages array and format the
messages to be sent
let formattedMessages = fetchResult.rows.reverse().map(row =>
({ role: row.role, content: row.content }));
const systemMessage = chatbotModel.messages;
formattedMessages = [...systemMessage, ...formattedMessages];

//send the messages to openAI and get the reply
const completion = await openai.createChatCompletion({
  model: chatbotModel.model,
  messages: formattedMessages,
  temperature: chatbotModel.temperature,
  presence_penalty: chatbotModel.presence_penalty,
  frequency_penalty: chatbotModel.frequency_penalty,
})

// get the tokens information
const { prompt_tokens, completion_tokens } =
completion.data.usage

//insert chatbot response and tokens into database
const chatbotMessage =
completion.data.choices[0].message.content;

if(!chatbotMessage){
  throw new Error("No response message provided");
}

const insertBotText = 'INSERT INTO ChatMessages(id_user,
created_date, chatbot_name, role, content, prompt_tokens,
completion_tokens) VALUES($1, NOW(), $2, $3, $4, $5, $6)';
const insertBotValues = [userId, chatbotModel.name, 'assistant',
chatbotMessage, prompt_tokens, completion_tokens];
await client.query(insertBotText, insertBotValues);

// Decrease conversation points by 1 after sending the message
await client.query(
  "UPDATE Users SET conversation_points = conversation_points
- 1 WHERE id_user = $1",
  [userId]
);

// Fetch updated conversation points
const updatedPointsResult = await client.query(
  "SELECT conversation_points FROM Users WHERE id_user = $1",

```



```

        [userId]
    );

    const updatedPoints =
updatedPointsResult.rows[0].conversation_points;

    // Return both AI message and updated points to frontend
    res.json({
        aiMessage: completion.data.choices[0].message,
        updatedConversationPoints: updatedPoints
    });

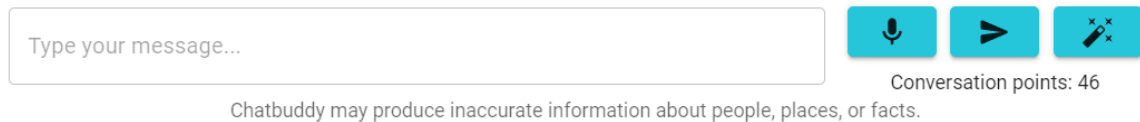
    await client.query('COMMIT');
} catch (err) {
    if (err.response) {
        console.log(err.response.status)
        console.log(err.response.data)
    } else {
        console.log(err.message)
    }
    res.status(500).json("Server error");
} finally {
    client.release();
}
});

```

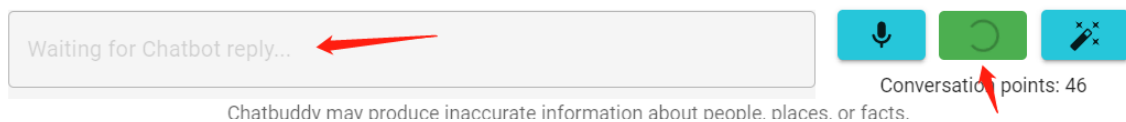
## Appendix F User Experience and Error Handling Examples

### 1. Loading state for chatbot input area and buttons

Not Loading:

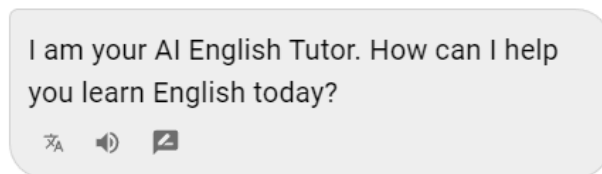


Loading:

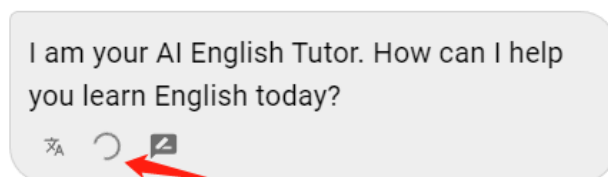


### 2. Loading state for chatbox:

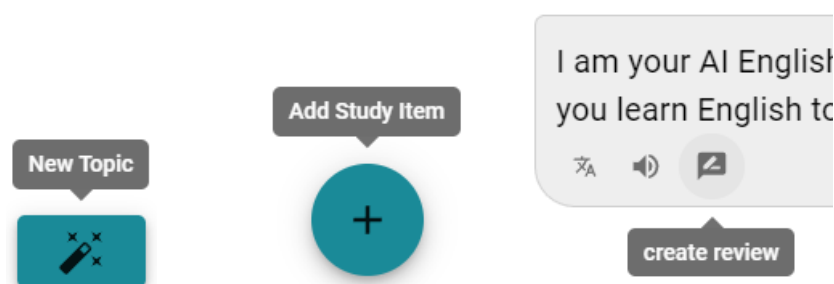
Not Loading:



Loading:



### 3. Tooltip for buttons without explicit text explanation



#### 4. Form validation

### Add Study Item

Title

Evoke

Category

▼

Category is required

Content

Content is required

Scheduled Date

25/08/2023

📅

The date should not be in the past

CANCEL

SUBMIT

#### 5. Snack bar at the left bottom of the screen to show error or success result of users' interaction:

✔ Tranlated successfully

✕

⚠ Please enter a message.

✕

## Appendix G Manual Test Cases

A specific script was employed to facilitate systematic manual testing of the application. This testing encompassed positive, negative, and boundary test cases.

Scenarios	Test Cases	Result
Visitors	<ul style="list-style-type: none"> <li>User should be able to view all pages except pages which require authorization.</li> <li>The links on the page should work correctly.</li> <li>When a user visit the wrong address, he should be able to redirect to page with error handling instruction.</li> </ul>	Partly passed. Failed test details: 1. If the user input the wrong URL address, there is no any information provided except an empty page.
Registration	<ul style="list-style-type: none"> <li>User should be able to register an account if input valid data.</li> <li>If user input invalid data, including wrong email address, same email address, weak password, wrong password for verifying identical password, empty input, the user should not be allowed and suitable error handling information should appear.</li> </ul>	All passed.
Log in and log out	<ul style="list-style-type: none"> <li>If the user input correct information, the user should be able to log in.</li> <li>If the user provides wrong information, including both email and password, it should not be allowed to log in and error message should appear.</li> <li>When the user click log out button, the user should be logged out and the token should become invalid.</li> </ul>	All passed.
Dashboard	<ul style="list-style-type: none"> <li>The user should be able to see the correct statistics of learning record.</li> <li>The user should be able to register the sample course if he has not registered it before.</li> <li>The user should see register successfully chip if he has registered the sample course.</li> </ul>	All passed.
Chatbuddy	<ul style="list-style-type: none"> <li>The user should be able to see the today's study task and review task correctly.</li> <li>The user should be able to select various chatbots.</li> <li>The conversation history and chatbot introduction should be shown correctly for all chatbots.</li> <li>The voices of chatbots should be played with the correct accent, even for long form content.</li> </ul>	Mostly passed. Failed test details: 1. If the user sends a long message to the backend, it may bring up more tokens than the API allowed. The API sends an error message back but

Chatbuddy	<ul style="list-style-type: none"> <li>• When the voice content is playing, clicking text-to-voice button again will replay the audio file.</li> <li>• The translation of chatbox content should be correctly shown, even for long form content.</li> <li>• When the translation is shown, clicking the translation button again will hide the translation.</li> <li>• When the user clicked the create review button in the chatbox, the create review form can appear with the text of chat box as default content.</li> <li>• The create review form triggered by chatbox icon clicking can create review and handle error (including extreme case and empty data) correctly.</li> <li>• The voice recognition can get the user input even for the speaking time more than 2 minutes.</li> <li>• When the user sends a valid message, the chatbot will give a reply.</li> <li>• When the user sends an invalid message (empty message and long message), error handling information should appear.</li> <li>• When the user clicks the new topic button, the new topic input dialog should appear.</li> <li>• After a user input a new topic, the new topic divider should appear in the conversation area.</li> <li>• Invalid input for new topic should be handled with clear error prevention and helpful instruction.</li> <li>• Destructive click testing. When one button is the loading state, the user clicks this button and other buttons. The loading state button should not work. Other button should work well at the same time.</li> <li>• When the user sends a message, the conversation points should reduce by one.</li> <li>• If the conversation points are zero, an error message should appear when user clicks send message button.</li> <li>• The conversation points should be update to default value 50 in a new day.</li> </ul>	<p>the code doesn't handle this error appropriately.</p> <p>2. The create review form can prevent the long content to create a review item. However, no error handling message is provided for user to take action except red colour hint.</p> <p>3. Speech recognition can only recognize speech nearly one minute.</p> <p>4. For the new topic, long input is not handled correctly. The long topic will break the conversation area interface.</p> <p>5. Although previous text in the message input area is hidden when user is recording, clicking the send message button can still send the hidden text.</p>
Study Overview	<ul style="list-style-type: none"> <li>• When the user clicks the status buttons, the study items can be displayed correctly.</li> <li>• The user can create/update a study item with valid input.</li> </ul>	<p>Mostly passed.</p> <p>Failed test details:</p> <p>1. All the form failed in handle long-form input</p>

	<ul style="list-style-type: none"> <li>• The form can show error handling message when user input invalid data (empty data and extreme data).</li> <li>• When user registered sample course successfully, a set of default study items should be imported to the study plan system automatically.</li> <li>• When the study task is finished, the user should be able to decide the review arrangement with valid input. This process should also handle invalid input appropriately.</li> </ul>	<p>data. The form only show red colour as a warning sign, but no error handle message appeared.</p> <p>2. A large set of data causes performance problems in the browser.</p>
Review Overview	<ul style="list-style-type: none"> <li>• When the user clicks the status buttons, the review items can be displayed correctly.</li> <li>• The user can create/update a review item with valid input.</li> <li>• The form can show error handling message when user input invalid data (empty data and extreme data).</li> <li>• The user can add new review sessions for review items with less than 5 review sessions.</li> <li>• When the study task is finished, the user should be able to decide the review arrangement with valid input. This process should also handle invalid input appropriately.</li> </ul>	<p>Mostly passed.</p> <p>Failed test details:</p> <p>1. All the form failed in handle long-form input data. The form only show red colour as a warning sign, but no error handle message appeared.</p> <p>2. A large set of data causes performance problems in the browser.</p>

## Appendix H User Testing Results

The evaluation questionnaire 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, with higher scores indicating higher agreement with the statement. The third segment included two open-ended questions to collect participants' opinion about the positive feedback and suggestions on this application.

#	Task Description	Average Score
1	Navigate to Home , About, Contact pages.	4.8
2	Create a user account.	4.8
3	Log in your account.	5
4	Visit dashboard and enroll the sample course.	4.8
5	Create two new study tasks in the study overview page.	4.6
6	Edit the title, content, category, scheduled date of a study task.	4.6
7	Set three scheduled study tasks to finished status and select different review pattern.	4.7
8	Set a scheduled study task to canceled status.	4.8
9	Use the study status filter and find the study tasks with different status.	4.8
10	Create a new review item in the review overview page.	4.6
11	Edit the title, content, category, review sessions of a review item.	4.6
12	Add a new review session into a study item which has less than five review sessions.	4.7
13	Change the status of review sessions in a review item to finished and canceled. Please make sure at least one review session was marked as finished.	4.8
14	Change the status of all review sessions in a review item to canceled status	4.8
15	Use the review status filter and find the review items with different status.	4.8
16	Send at least one chat message to every chatbots.	4.8
17	Interact with one of the chatbots with more than 10 conversations.	4.8
18	Find the study tasks and review tasks for today on the chatbuddy page.	4.8
19	Edit and change the status of one study tasks and one review task on the chatbuddy page.	4.6
20	Translate a chat message into Chinese by clicking translation button.	4.9
21	Play a chat message in audio by clicking text-to-speech button.	4.9
22	Create a review item by clicking create review button in the chatbox.	4.5
23	Input a text by clicking recording button.	3.8
24	Create a new topic by clicking new topic button.	4.6

25	Find the conversation points. You will also find the conversation points will reduce one after you send the message.	5.0
26	Select different chatbots to view the previous chat history.	4.8

**Table 1: Task Description and Result**

#	Statement	Response (Yes)
1	The design and user interface is appealing for you.	4.6
2	The website is user-friendly and intuitive.	4.3
3	This application is helpful for you to learn English.	4.4
4	The application allows you to accomplish core tasks efficiently.	4.5
5	Overall, you are satisfied with the application's performance and functionality.	4.6

**Table 2: Statement and Result**

<p><b>What aspects of this website do you appreciate most?</b></p> <p>Diversified options for the chatbots. The reply fits the personality and functions.</p> <p>Manage study tasks and review sessions easily.</p> <p>User can set the review pattern when the study task is accomplished.</p>
<p><b>What suggestions do you have for enhancing this application?</b></p> <p>The voice recognition is confusing. I don't know when I should stop recording.</p> <p>The chatbots can't remember previous topic and information. Sometimes I didn't get the desired reply. For example, I asked the chatbot give a test for me but the quality of the test is bad.</p> <p>The display format of the content for some study items and review items is not correct.</p> <p>The response from the chatbots sometimes are too slow.</p>

**Table 3: Open-ended questions and Result**



# Appendix I Ethics Checklist

## School of Computing Science

## University of Glasgow

### Ethics checklist form for 3<sup>rd</sup>/4<sup>th</sup>/5<sup>th</sup> year, and taught MSc projects

This form is only applicable for projects that use other people ('participants') for the collection of information, typically in getting comments about a system or a system design, getting information about how a system could be used, or evaluating a working system.

**If no other people have been involved in the collection of information, then you do not need to complete this form.**

If your evaluation does not comply with any one or more of the points below, please contact the Chair of the School of Computing Science Ethics Committee ([matthew.chalmers@glasgow.ac.uk](mailto:matthew.chalmers@glasgow.ac.uk)) for advice.

If your evaluation does comply with all the points below, please sign this form and submit it with your project.

- 
1. Participants were not exposed to any risks greater than those encountered in their normal working life.  
*Investigators have a responsibility to protect participants from physical and mental harm during the investigation. The risk of harm must be no greater than in ordinary life. Areas of potential risk that require ethical approval include, but are not limited to, investigations that occur outside usual laboratory areas, or that require participant mobility (e.g. walking, running, use of public transport), unusual or repetitive activity or movement, that use sensory deprivation (e.g. ear plugs or blindfolds), bright or flashing lights, loud or disorienting noises, smell, taste, vibration, or force feedback*
  2. The experimental materials were paper-based, or comprised software running on standard hardware.  
*Participants should not be exposed to any risks associated with the use of non-standard equipment: anything other than pen-and-paper, standard PCs, laptops, iPads, mobile phones and common hand-held devices is considered non-standard.*
  3. All participants explicitly stated that they agreed to take part, and that their data could be used in the project.  
*If the results of the evaluation are likely to be used beyond the term of the project (for example, the software is to be deployed, or the data is to be*

*published), then signed consent is necessary. A separate consent form should be signed by each participant.*

*Otherwise, verbal consent is sufficient, and should be explicitly requested in the introductory script.*

4. No incentives were offered to the participants.

*The payment of participants must not be used to induce them to risk harm beyond that which they risk without payment in their normal lifestyle. No information about the evaluation or materials was intentionally withheld from the participants.*

*Withholding information or misleading participants is unacceptable if participants are likely to object or show unease when debriefed.*

5. No participant was under the age of 16.

*Parental consent is required for participants under the age of 16.*

6. No participant has an impairment that may limit their understanding or communication.

*Additional consent is required for participants with impairments.*

7. Neither I nor my supervisor is in a position of authority or influence over any of the participants.

*A position of authority or influence over any participant must not be allowed to pressurise participants to take part in, or remain in, any experiment.*

8. All participants were informed that they could withdraw at any time.

*All participants have the right to withdraw at any time during the investigation. They should be told this in the introductory script.*

9. All participants have been informed of my contact details.

*All participants must be able to contact the investigator after the investigation. They should be given the details of both student and module co-ordinator or supervisor as part of the debriefing.*

10. The evaluation was discussed with all the participants at the end of the session, and all participants had the opportunity to ask questions.

*The student must provide the participants with sufficient information in the debriefing to enable them to understand the nature of the investigation. In cases where remote participants may withdraw from the experiment early and it is not possible to debrief them, the fact that doing so will result in their not being debriefed should be mentioned in the introductory text.*

11. All the data collected from the participants is stored in an anonymous form.

*All participant data (hard-copy and soft-copy) should be stored securely, and in anonymous form.*

---

Project title LangBuddy: A Study Web App Powered by ChatGPT for EFL learners

Student's Name Tiansheng Chen

Student Number 2807300c

Student's Signature Tiansheng Chen

Supervisor's Signature Sham Puthiya Parambath

Date 29/08/2023