MANCHESTER METROPOLITAN UNIVERSITY

FINAL YEAR DISSERTATION

Interactive Student Support Tool for Lecture Dissemination

Author:
Haseeb Shafiq
(21344273)

Supervisor:
Dr Naomi Adel

Degree:
Computer Science

Department:
Computing and Mathematics

Faculty:
Science and Engineering

May 2024

Declaration

No part of this project has been submitted in support of an application for any other degree or qualification at this or any other institute of learning. Apart from those parts of the project containing citations to the work of others, this project is my own unaided work. This work has been carried out in accordance with the Manchester Metropolitan University research ethics procedures and has received ethical approval number 77488**.**

Signed

Acknowledgements

I would like to thank my supervisor, Dr Naomi Adel, for supporting me through the development of my project.

Thank you to all the users for participating in my study.

I would also like to thank my family and friends for their constant support throughout university.

Abstract

The lack interactivity and efficiency of modern studying techniques result in students spending vast amounts of time upon a singular topic, constantly reiterating over the content. Educational videos on platforms such as YouTube allow for a slightly better method of teaching through visual aids. However, students still struggle for time when making notes a numerous number of times for different topics. This time could be allocated elsewhere, like answering questions on each topic to test their knowledge retention.

This project presents Scribble, an education tool which uses AI and is set out to aid students with the process of studying, saving them time and embedding effective practices to retain information. This tool uses the YouTube transcript API to extract a transcript from an educational video that contains captions. A dedicated Claude model is then passed the transcript to create a set of notes which include a summary, key terms, main takeaways, review questions and content outline section, providing the student with different methods of retaining the themes taught within the video. In addition, an implementation of a chatbot messenger has also been included for students who would like to ask questions to gain a deeper understanding of the video's topic.

The design and implementation of core features were evaluated in terms of effectively meeting the criteria of requirements set out within the initial stage of development.

To measure the effectiveness of Scribble, in comparison with traditional methods of revision, a study took place. This study consisted of 10 participants, split into two groups. One group was allowed to use Scribble to study for a quiz while the other used basic methods of note taking to undergo the quiz. The resulting performance metrics represented a 16% difference between the scoring of the experimental group and control group.

Limitations and future work were discussed as part of the concluding segment, to provide a reflection on what problems occurred and what future adjustments would take place to further improve the project.

Table of Contents

Abbreviations

Abbreviation	Description
AI	Artificial Intelligence
API	Application Programming Interface
CSS	Cascading Style Sheet
GPT	Generated Pre-Trained Transformer
HTML	Hypertext Markup Language
JSON	JavaScript Object Notation
ML	Machine Learning
POST	Power On Shelf Test
SDK	Software Development Kit
URL	Uniform Resource Locater
VS	Visual Studio

1. Introduction

1.1. Project Background

The rapid evolution of modern learning methods has changed considerably within the last decade. As new technologies emerge, the learning resources accessible to students advance, creating solutions to improve their learning capabilities. Online learning has become normalised since moving forward from COVID, which has reshaped the methods educational institutes teach content to students. This shift caused a large dependence on the production of online resources, leading to an increase in the production of educational content on platforms such as YouTube, hosting millions of videos.

Traditional methods of studying, where students refer to the notes taken during lectures or classes, can be time-consuming and inefficient due to the possibility of missing important content. The benefits of using video-based content to supplement a student's learning allow them to learn visually while covering more specific concepts regarding a topic.

However, students find themselves replaying and having to constantly rewind videos to grasp these concepts, actively creating their own notes while watching; adding to the ineffective practice of studying using traditional methods.

The integration of Artificial intelligence presents the opportunity to improve the process of studying for students. Modern large language models such as ChatGPT are dominating the technology industry, boasting impressive capabilities. The use of these models can enhance the traditional study methods by automating specific processes to reduce the wastage of time in tasks such as note-taking.

This means that students will obtain a set of results generated based on the educational content, providing them with a structured summary of different note formats. This implementation will lead to an increase in study effectiveness and efficiency, leading to an improved chance of better performance.

For this project, an HTML web page was created to implement a Claude AI model to generate a tailored set of notes for an educational video. Students can access this site and paste a link for an educational video for which they would like to receive notes. They can then watch or refer to the video while studying the provided notes. Five different sets of

notes are generated, each with their own learning outcome. Users are also presented with the option to interact with a chatbot messenger to ask any further inquiries about the topics within the video.

1.2. Aim and Objectives

1.2.1. Aim

This project aims to develop an AI driven support tool which aids students by the provision of tailored learning resources derived from an educational video.

1.2.2.

1.2.3. Objectives

The objectives of this project are as follows:

- Producing a backend which dynamically implements core features with relevant application of functions.
- The employment of user functionality and experience in the improvement of the development of the project.
- Creating an interactive and distinctive design which helps students to engage with the application itself

1.3. Dissertation Structure

Chapter 0. Introduces the related research into the background of AI, applications and implementations of AI, the role of AI in higher education, concerns of AI in academic settings, ethical considerations, and current AI assistants

Chapter 3. Provides an overview of the design, the requirements to implement, application architecture, user interface and methodology used within this phase

Chapter 4. Explores the implementation stage of the project, breaking down the features implemented and the methos of testing's conducted.

Chapter 5. Reviews the achievements made within the project, discussing whether the aim and objectives were met. The results of the comparative effectiveness quiz and usability feedback were analysed.

Chapter 6. Includes the summarisation of the work and achievements made by the end of the project, any limitations that were encountered and the palns for future implementations to improve the project further.

2. Literature Review

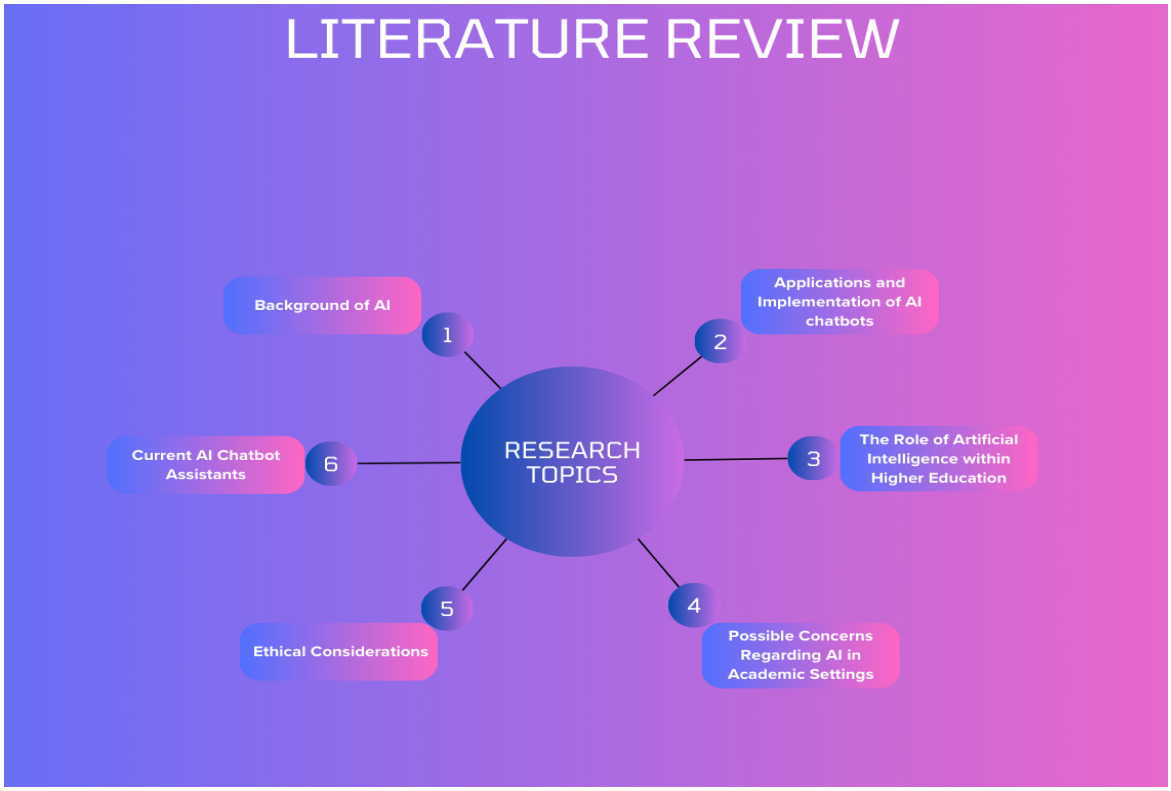


Figure 1 - Literature review research topics

2.1. Background of AI

Artificial Intelligence is produced with the science and engineering of creating intelligent machines (McCarthy, J. 1970) that is capable of performing complex tasks that utilise human learning capabilities such as, problem solving, decision making and creativity (Stryker, C. 2025). The development of artificial intelligence can be tracked and followed from 1942 onwards, with the questioning of creating intelligent machines beginning. An Inspirational story Runaround which was published by Issac Asimov (1942) gave future scientists a motivating mindset to the field of robotics (Haenlein and Kaplan (2019)). The story contained three 'Laws of Robotics', a robot may not injure a human being or, through inaction, allow a human being to come to harm; a robot must obey the orders given to it by human beings except where such orders would conflict with the First Law; and a robot must protect their existence as long as such protection does not conflict with the First or Second Laws. These laws set a foundational concept within many discussions surrounding both robotics and Artificial Intelligence.

Alan Turing was a key figure who provided explanations in 1950 on how to create an intelligent machine, while also producing a machine that was utilised by the British Government to decipher the German Enigma code. The level of intelligence that made this machine (The Bombe) complete a task that humans could not achieve led to Alan Turing pondering the topic of intelligent machines. The Turing test is still used today as a benchmark to tell the difference between machine and human, to determine a machines intelligence.

2.2. Applications and Implementation of AI Chatbots

Since then, we have seen that the rapid growth and application of Machine Learning (ML) integrated Artificial Intelligence (AI) has soared across many industries within recent years, allowing for businesses to reap the benefits of an enhanced intelligent tool that can elevate multiple areas of development, automation, and learning. The application of Artificial Intelligence in education has grown significantly, promoting its deep learning capabilities and providing users with an in-depth response to questions across a remarkable range of categories. The utilisation of these responses has gained the attention of many students within higher education, providing them with unique explanations and solutions to academic challenges, boosting their understanding, and leading to an increase in overall performance.

Artificial Intelligence, alongside Machine Learning, has shown the relevance within the pharmaceutical industry, providing improvements such as drug candidate selection and understanding disease mechanisms to the Research and Development side of this field (Kolluri et al., 2022). In the article ML/AI in Pharmaceutical research and development, the applications are employed across different stages of drug development.

In the early stages of drug discovery, AI and Machine Learning can aid in the identification of potential drug candidates based on predictions made on molecular properties and biological mechanisms, which would provide them with a higher probability of therapeutic benefit. Machine Learning modules analyse a patient's past medical history, their genetic information and patient data. The result of this is the ability to obtain an optimal match of patients that can then receive a personalised treatment, enhancing the treatment effectiveness and efficiency.

This study presents the variety of applications of different AI implementations, providing the pharmaceutical industry with a potentially powerful tool, as opportunities arise alongside the continuous increase of data influx. This article concludes that AI and machine learning implantations can provide substantial benefits within the industry but depending on the way it is implemented and ensuring that the algorithms utilised are well refined.

Another use case of Artificial Intelligence, specifically pre-existing chatbot models available to the public, has been investigated within the pharmaceutical industry to observe the accuracy of the chatbot in answering pharmaceutical enquiries. A study which explores the safety and quality of AI responses (Yasser Albogami et al., 2024) evaluated the performance of the different GPT models with questions regarding drug inquiries that would be deemed relevant to what you would find in a real-life scenario. The performance of different GPT models such as GPT 3, 3,5 and 4 was evaluated in comparison with licensed pharmacists, concentration of factors such as the level of detail, risk of harm and overall accuracy of what was being outputted was the main focus to determine this.

The study's results showcased the effectiveness of each model, with GPT-4 providing the most accurate responses (64.3%) in comparison to the pharmacists. However, the study also reveals that when inquiries were made that didn't require previous patient data, the AI model outperformed the human participants, while providing additional information, such as risk mitigation, within its responses.

This provides a strong basis for the advocacy of AI and its utilisation within industries to enhance human performance, as we can distinguish the number of advantages and disadvantages and conclude that Artificial Intelligence models can process vast amounts of information and provide the users on the receiving end with accurate information that can be deemed reliable.

The potential of Artificial Intelligence has been projected through its implementation within the pharmaceutical industry, providing its capability in handling the complex processing of pharmaceutical data. The architecture of artificial intelligence will assist the author with the production of a tool which will support students with an effective tool in boosting academic understanding through generating a structured set of notes.

2.3. The Role of Artificial Intelligence within Higher Education

Just as we have seen the pharmaceutical industry benefit from AI chatbots in various applications, students can also reap the benefits of these pre-existing intelligent models, processing academic information to provide students with in-depth explanations of new concepts, while providing reliable answers to complex questions.

There have been several studies that provide extensive insight into the significance of AI chatbots within education. An example where this has been represented was within Labadze et al, where various articles had been reviewed and analysed to provide the following results. There were numerous advantages for students via chatbots such as ChatGPT, which led to an increase in student engagement, understanding and efficiency (Chen et al). Students have the ability to take advantage of a personalised learning experience as AI chatbots can bestow detailed feedback to underline areas of improvement and explain methods to achieve this. Although being a strongly positive and effective tool at students' disposal, many also realise the implications of where Chatbots retrieve their information from, sometimes being inaccurate. However, many are aware of this and will follow up with supporting research to determine the chatbot responses accurately.

Within Labadze et al's results it was found that, AI Chatbots not only aid students within the education sector but also the educators, relieving them of time by providing support in activities such as grading (Dai et al, 2023), planning but also in producing questions for students that can be customised to fit the characteristics of the teacher following their personal methods of teaching.

A study by Dai et al, which utilised ChatGPT as an assessment tool, was conducted by evaluating data science assignments by postgraduate students and delivering feedback using the criteria of clarity of both project goals and the solution, business benefit details and the relevance of the chosen topic to data science. The feedback generated from the chatbot was assessed along with the instructor's feedback via three experts; it was found that ChatGPT's responses were adequate to where it presented highly detailed suggestions and feedback, showing its ability to analyse the student's execution within the delivery of their project, as well as indicating future improvements and solutions. The findings within this study fortify the value of AI chatbots and their ever-growing potential to be a major helping hand within the education sector.

It shows its worth as a learning enhancer while also being capable of providing guidance to point students in the direction of improvement and growth. To be able to integrate such tools and adapt to them to specific implementations such as an AI note making assistant as explored within this project, can provide the field of education with a cost-effective and scalable approach to improving academic performance.

2.4. Possible Concerns Regarding AI in Academic Settings

However, although there are many clear evidence-based reasons why AI chatbots can be used as an essential tool to aid with a student's learning capability, there are some concerns that arise. An article investigating the ethical implications of AI chatbots in education, written by Kooli, C, suggests from their results that education in the future may be at risk of students who have not developed such skills as critical thinking and problem solving. The leading cause of this stems from the concentrated reliance on Chatbots providing students with an answer without them doing any critical thinking themselves.

Another study which highlights some additional implications of AI chatbots, such as Chat GPT, within higher education is AI Chatbots in education: Challenges and opportunities, Davar, Dewan and Zhang (2025). A notable limitation that was underscored within this study was the concern for models such as GPTs' misinformation and reliability issues, depending on the information used to train these models. This can stem from using a vast number of studies and data without using filtering methods that ensure quality control of the sources, leaving room for ambiguity and the possibility of producing inaccurate results. Such concerns can lead to dire consequences for students, depending on the model's implementation, affecting a student's learning outcome.

Another interesting theme regarding the idea of using chatbots as an educational tool is that although they provide students with the capability of accessing all types of information and provide in-depth explanations to aid learning, traditional educators go beyond the concept of just teaching. Chatbots give the feeling of human-like responses but truly do not understand human emotions; where a teacher would aid a student and build a trustworthy relationship, providing that student with an education that is taught in a way which is adapted towards the student's understanding. This human-to-human connection cannot be replicated by AI; they can certainly enhance a person's learning experience and are always available to be utilised, but we can conclude that the most effective route for the future

would be to have an approach where both teachers and AI are used in conjunction to provide the most efficient method of learning.

As a result, we can conclude that AI chatbots such as ChatGPT have exhibited a wide range of benefits for not just students but also their educators, offering them with a tailored support mechanism, leading them to progression while offering teachers with the ability to supplement their learning methods and materials based on the suggestions made by AI. However, we can also perceive that these tools do not come without their limitations in terms of considerations such as the development of critical thinking, credibility of information and emotional understanding. These hindrances can be mitigated with the right measures and in time, as Artificial Intelligence chatbots continue to advance.

The project acknowledges these concerns and accommodates them through the features which have been implemented. The in-built AI assistant can assist students while providing a thorough explanation, allowing students to have a conversational element as they would with an educational tutor, preserving the human experience of learning and adding a technological aspect. In addition, the structure and architecture of this web application allow students to learn in an interactive manner, which encourages them to engage with the material they are learning, helping them to embed the idea of critical thinking.

2.5. Ethical Considerations

After analysing the duality of chatbots within education, another subject matter to consider is the ethical implications of using this technology. Within Kooli (2023), a significant point regarding ethics is brought to attention, suggesting that chatbots can result in the bias towards certain groups or individuals, for example within a political setting, if users ask a question regarding sensitive subject matters such as political conflicts; the chatbot may have been developed/programmed to respond to this query adhering to the beliefs of the developer, feeding a misleading opinion that would favour one side than the other. The example presented within this study was the Ukraine and Russia conflict; if developers decided to, it would serve their own political interests to train the chatbot to portray the opposing side in a negative light. If certain viewpoints like this are being promoted and some disregarded public opinion on similar matters can be swayed. This is a dangerous consideration and must be handled effectively so public ideologies and beliefs are not

manipulated and influenced, as major societal impacts could be made, producing negative opinions of certain individuals and communities.

In addition, the general idea of bias being present as an issue within a chatbot can cause multiple problems in many of the responses it makes; to solve/reduce the occurrence of this problem is for governments and institutions to provide ethical guidelines for developers to follow to ensure the deployment of AI chatbots will not lead to any misinformation and bias, supplying facts and statistics that will hold no opinion.

Due to the complex technicality of AI chatbots, to function accordingly, they require personal user data to withhold saved dialogues between the user and chatbot, users' geographical and browser history (Tawfeeq, Awqati and Jasim, 2023) are examples of some of the information collected. This accumulation can bring about data misuse when information is utilised without the user's consent, raising concerns for privacy. Specifically, within the education sector, students may share personal information without acknowledging how it is being collected and used; models such as ChatGPT are being pushed to sustain and practice more accountability and transparency. The need for stricter data protection regulations is being called into question, AI chatbots' technological advancement is still new, safeguarding user data through solutions of encryption will allow users to operate without the worry of misuse. Notifying how information will be employed so students know how and what their data is being used for, therefore promoting ethical regulation.

Another interesting ethical consideration is the possible future impacts upon societies ability to interact alongside the social skills they develop; it can be argued that AI chatbots may cause a hinderance to these aspects as they become more comparable to a human interaction within its discussion functionality, simulating a two-way conversation, providing the feeling of speaking to a human. Studies such as (Alotaibi and Alshahre, 2024) provides a strong basis for the support of AI chatbots in offering a social connection, where 85% of participants within the study felt that they were given a sense of companionship. Whereas in the long run, this may have a reverse effect on these abilities, as individuals may place a

large amount of reliance upon these models. Within an educational setting, a student may only utilise these tools rather than confide in a teacher, resulting in the potential diminishment of their social engagement skills and surrendering any chance of a relationship with a mentor, meaning that they wouldn't have that genuine human interaction needed for professional environments when looking for jobs. AI chatbots offer temporary comfort and support for users, but we must consider the long-term effects on social behaviours and concentrate on reinforcing these abilities by promoting a hybrid environment where AI works alongside teachers to form a collaboration so that these concerns can be eased.

When considering the use of AI chatbots within the area of scientific research, which is commonly referred to within higher education as part of assignments for students, the use of such mechanisms brings about important concerns in relation to the information students receive. AI can benefit students in generating new ideas, and in our case, to summarise literature; however, questions surrounding who owns the credits to the information provided become relevant as chatbots are not deemed to be human, so can not claim authorship over the content they produce and supply (E. Sobaih, 2024). This issue remains unanswered and can lead to issues regarding plagiarism. If students rely on information generated for them by chatbots without checking its source and reliability, the distinction between the students' original thoughts and AI content becomes vague. Students may unknowingly violate this ethical issue if educational institutions do not make their guidelines clear about how and if they can use AI chatbots within their higher education.

To ensure responsible and ethical AI implementation within the project, the chatbot model for user questions will be trained to only respond to questions regarding the educational video they have submitted for notes, this minimises any chance of a bias response to a topic not in relation to an educational matter but also in the matter of plagiarism as the information provided via notes will only derive from the information collected from the video. To combat the privacy concerns outlined by Tawfeeq, Awqati and Jasim (2023), the web application will hold no personal data and will require no user information for its use; This means that there is no chance that the data within the user session will be held and tied to an individual.

2.6. Current AI Chatbot Assistants

With the new arrival of AI chatbots swarming the technological industry, many companies have produced their own iteration of an artificially intelligent assistant, which aims to provide answers to a wide variety of questions across almost any topic. Some mainstream models include Open AI's Chat GPT, Anthropic's Claude and DeepSeek's R1 and V3 models; these high-performing models have become widely adopted in higher education settings, aiding students. Within a study evaluating different models in complex medical decision making, AI models such as Claude, Chat GPT and Google Bard were compared using their performance based on descriptive statistics, which measured accuracy, relevance, clarity and completeness. The study analysed that Claude AI held the most dominance with its completeness and relevance, whereas Chat GPT held the most clarity (Vamsi Krishna Uppalapati and Deb Sanjay Nag, 2024); this supports the authors use of Claude as it is proven to provide the most appropriate and well explained responses which is vital when supplying students with a summarisation of educational content.

This literature review critically evaluates relevant sources of references to emphasise previous findings of how Artificial intelligence affects academic performance of individuals and its significance in the possibility of becoming an ideal tool within education.

3. Design

3.1. Design Contents

This section will outline the requirements that were used during the production of the project, the reasoning behind choosing them and the way in which it was implemented, the choice of methodology and ensuring that the development phase would allow enough time to meet all the demands and aims set out in the initial stages of planning.

It will present the inner workings and components of the web application's architecture alongside the decisions made regarding the configuration, layout and overall user interface and how the author was able to implement features accommodating user satisfaction.

3.2. Requirements to Implement

The main feature of the system to implement, within the web application, is the ability to provide users with a service which extracts material from an educational video and converts it into a concise but informative set of summary notes. This will be formatted differently to provide a varied range of notes which will each have their own purpose. The project is set out to include six different categories in which the user can navigate between with the use of tabs, this will comprise from a main summary, key terms, main takeaways, review questions, content outline and a chatbot assistant. The final tab, consisting of a chatbot assistant, will provide an interactive interface to ask questions specific to the video content, upon which users will receive formative responses back.

The project will place a deep focus on the user interface and website design; this is a valuable element to a system to make sure core functionality is in place, as well as providing the user with an interface that can easily be accessed and is self-explanatory to use. Maintaining a good standard of design throughout the project creates a visually engaging atmosphere with the intention of creating a memorable brand identity to differentiate itself from similar educational tools. The thought processes and steps taken to achieve this will be discussed as well as why the author has decided to include such characteristics.

3.3. User Stories

The following user stories directed and influenced the design decisions, developmental processes from the tabbed content organisation to integrating an AI assistant.

1. As a student, I want to efficiently extract key information from educational videos so I can study without having to watch them repeatedly.

2. As a visual learner, I want educational content to be well-organised and visually distinct to enhance my information retention.

3. As a user with specific questions, I want to engage with an AI assistant that uses video context to provide in-depth explanations without the need to rewatch.

4. As a user with diverse study requirements, I want options for different educational content formats (summaries, key terms, questions) to select the one that best fits my current study objectives.

3.4. Web Application Architecture

This section will include the overall layout of the system's architecture, outlining the variety of components in both the frontend and backend segments. To begin, the entirety of the project shall be produced and managed within Microsoft's Visual Studio (VS) Code, which provides a great amount of support for multi-language programs, making it highly supportive in the development of Scribble due to the use of JavaScript, HTML, CSS and Python. In addition, VS Code allows developers to integrate third-party extensions which the project needs for chatbot functionality and web design features such as animations; these can either be downloaded or imported through the terminal, allowing for a seamless experience.

The web application backend utilises Flask to host a server for the project; this is achieved using Python. Python was chosen as the primary backend language due to the substantial support for AI applications in projects; integration is made seamless for the YouTube API and Anthropic's Claude AI SDK, which is Python-based, making it an optimal choice for the AI models.

Flask is a framework aimed at making the process of creating and managing web applications simpler. This is being used in the process of implementing the AI chatbot model, handling the API requests through a HTTP POST method along with creating any routes to different pages within the web application.

The project is utilising Anthropic's Claude AI model, more specifically Claude 3.7 sonnet, Anthropic's best performing model with the capability of producing In-depth explanations which will be best suited for the web applications ability to produce five set of notes.

The Claude 3-Haiku model has been chosen for the complimentary AI assistant when the user has any further questions regarding the video. This was decided as the model offers faster responses in comparison to the 3.7 Sonnet model, which is more befitting as the author not only desires the chatbot to generate responses but also make the user feel like they are having a reciprocal conversation to maintain engagement.

Both models are fed a prompt which is comprised from information training them and providing them with the context of what their role is, what information they will receive and how they should respond to it. Each model's prompt has been tailored to fit the desired requirements in meeting the initial aims.

The application concentrates on the dissemination of educational content found on YouTube; this was chosen due to the platform's ever-expanding library of material catered towards students while covering a variety of topics for different age ranges, and not just higher education. Articles evaluating YouTube's impact on education, such as Shoufan (2022), show strong results through the use within classrooms, leading to a positive impact (of 18.7%) on the performance and grades achieved by students. This shows how the platform has established itself to be useful within educational environments in teaching relevant material to students. The accessibility of YouTube is a significant factor in the choice of being the main platform chosen for the application's main video source, as it allows for students across the world to utilise and enhance their learning potential.

However, despite the potential, older educational content can be seen as unorganised and make it difficult to grasp concepts if only watched once; with the leveraging of the authors AI tool, we can address this problem by producing this educational content into a structured format, preventing the user having to repeatedly watch video, allowing for a efficient learning method which promotes retention and comprehension.

The figure below presents the functioning of Scribble, laying out the architecture revealing the process of both the frontend and backend.

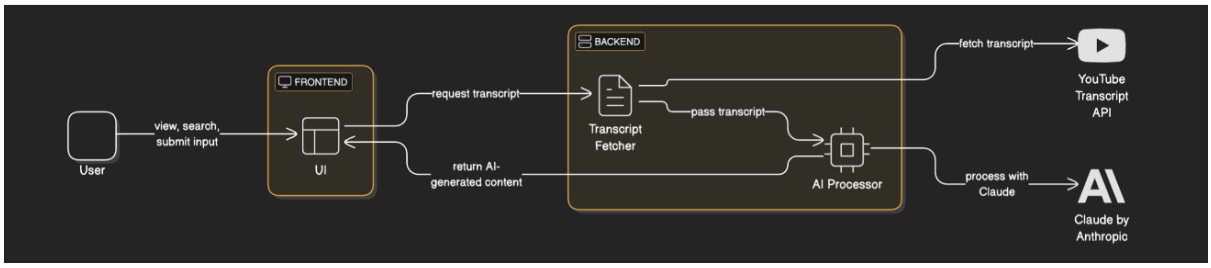


Figure 2– Simple System Architecture Diagram

JavaScript was chosen as the preferred language to process frontend interactivity for the application. JavaScript allows for dynamic updates to the webpage without having to reload; this can be used for the generative content and displaying this to the user without having to navigate to a different page. JavaScript also possess the capability of formatting these responses into structured elements which can be easily read. The error handling mechanisms can also be appreciated within JavaScript, as it provides an indication of when an error has occurred within the main functions of the project.

3.5. Web Application Format

For the design and parsing of information the frontend is built using HTML, CSS and JavaScript. HTML and CSS is used to design and create any web elements styling the Scribble platform. HTML separates the page into organised sections each serving a different purpose. The first being the header, allowing users to navigate between different pages and access an information button which will hold a small set of instructions on how to use Scribble.

The middle section contains the search bar for users to paste their YouTube link within, along with a search button to activate the YouTube transcript API and Claude 3.7 Sonnet Post request. Once the user has pressed this button, a video preview appears below, giving the user the option to reference back to the video while making use of their notes.

The final component encapsulates the chatbots summary responses accompanying the AI assistant; this is organised through the use of buttons which will act as navigation between the different categories.

3.6. User Interface

The design and interface produced for the project were crucial to create appeal and add an identity for the platform. This section will discuss certain design features and their appearances, the inspiration behind them and the reason why they were chosen to be implemented that way.

To begin, a significant amount of research was conducted in order to find inspirations of pre-existing platforms to analyse and evaluate them to create a list of content, which will be utilised within the interface and design of the Ai tool.

3.7. Brand Identity

To stand out and create a unique and engaging artificially intelligent application, which differentiates to its counterparts; a brand identity was crafted. This showcases users with a distinctive presence, to which users will have a memorable perception of Scribble, which they resonate with when in need of notes. Different attributes, including animations, colour schemes, and brand name, all collaborate towards pushing a positive image to appeal to students.

The name ‘Scribble’ was selected as the application’s name to evoke the brand's main purpose of providing a set of educational results, while remaining short and memorable. The positioning within the centre of the page serves both functional and aesthetic purposes, setting the tone for the user experience by creating a focal point representing the brand

identity.

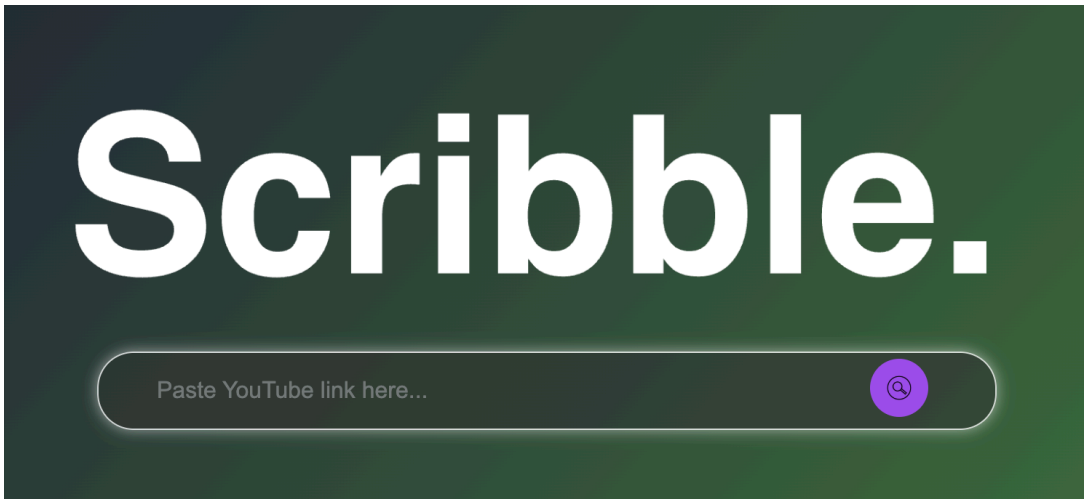


Figure 3– Scribble Logo

3.8. Web Application Elements

Scribble combines a variety of dynamic colours transitioning through a gradient animation for the background to establish a distinctive visual image. In addition, animated cube components that enlarge and rotate were added to complete Scribble's background.

Supporting attributes, including pulsating animated borders around information containers, including the chatbot assistant messages are integrated as part of Scribbles aim to provide users with an ‘energetic’ and ‘interactive’ environment, as well as clearly outlining any important information.

Tab based navigation categorises the different forms of notes for users to interact between the information they require, providing a structured method which separates the generated notes from each other rather than having all information displayed in one section which may cause students to be overloaded with information.

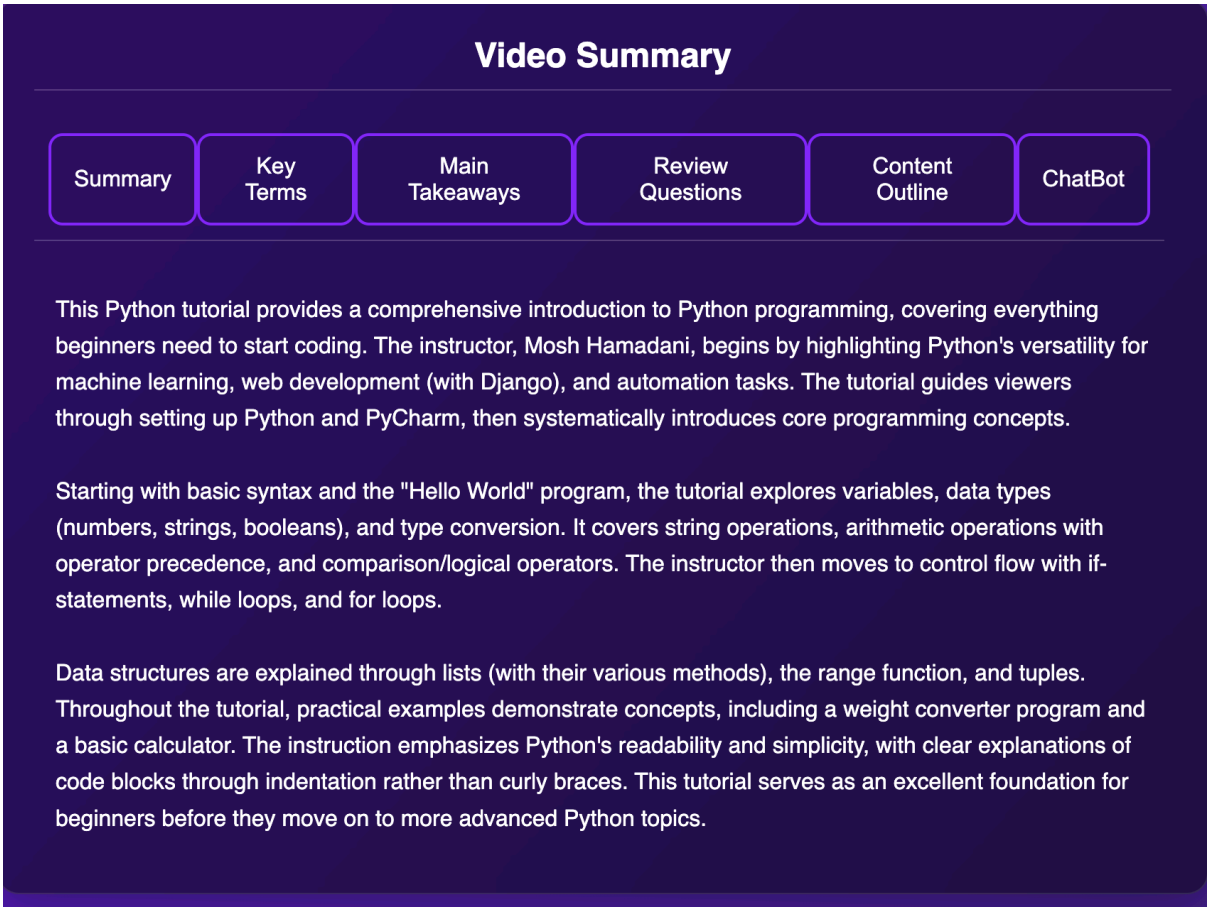


Figure 4– Screenshot of tab navigation

A video playback feature, which displays as soon as the user presses the search button, gives them the option of referring back to the video along with their notes.

Figure 5– Screenshot of video player

All containers within Scribble are styled in the same way. They provide a blurred, semi-transparent background, enhancing the contrast between the text and the background to ensure the application is accessible to everyone and remains clear and visible.

Scribble sets itself apart from existing AI tools with design choices and user interfaces that lack distinctive features, resulting in a uniform appearance. This consistency can be found throughout the entire webpage to create an element which will aid in the brand identity.

3.9. Methodology

The development of Scribble incorporates the combination of methodologies including iterative and prototyping, utilising features of both to ensure development requirements and deadlines are managed and met on time.

The project implements an iterative development approach, breaking down the complex integration features of the project into incremental steps. This methodology works by creating the initial steps of development by first setting out any project requirements, then designing the solutions to meet those requirements, leading to the first iteration. After this, the development will go through this cycle while also putting each iteration through testing and reviewing to continuously improve each iteration.

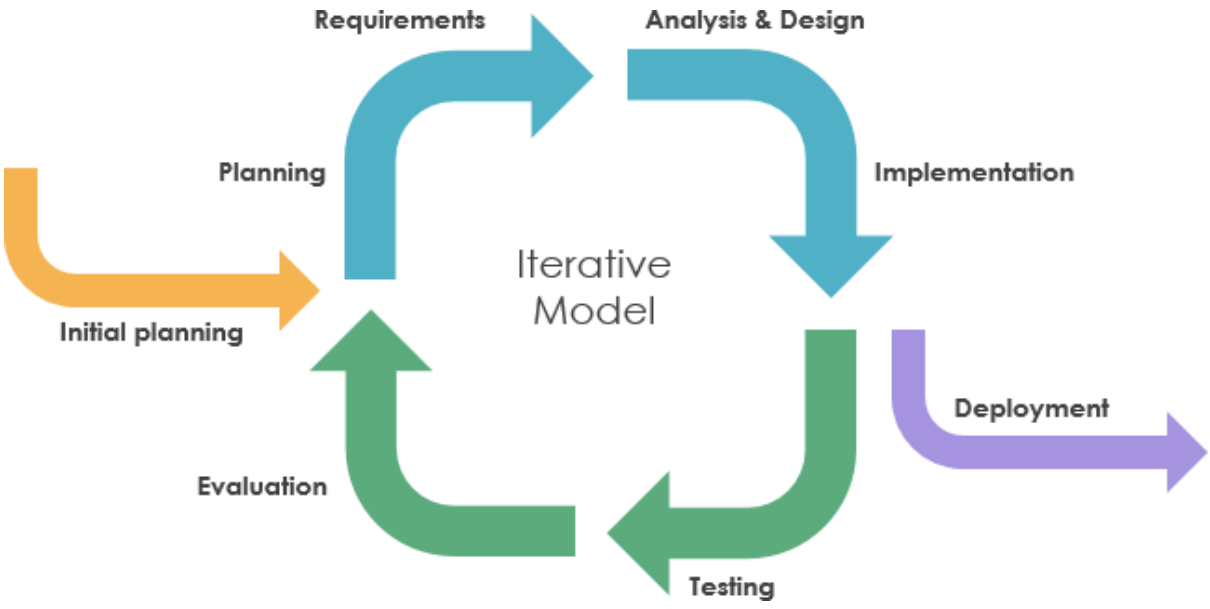


Figure 6– Iterative Model Diagram

JIN (2023). *Software Development Framework — Iterative Model*. [online] Geek Culture. Available at: https://medium.com/geekculture/software-development-framework-iterative-model-68584bfad773.

This diagram shows a thorough example of the iterative model and the processes required to improve each cycle's iteration.

3.9.1. Advantages of Using the Iterative Model

The decision to choose this methodology stemmed from the beneficial feature of testing each implementation before embedding it within the project, producing prototypes, and reviewing and revising them, ensuring that all problems are addressed and any features users disliked are adjusted accordingly.

This methodology contributes to several advantages through the implementation to provide a positive outcome for the project. Each iteration cycle will refine the project leading to the quality control of content produced will be of a higher calibre, as user feedback will inform the author of possible improvements and desires, they require to meet an adequate satisfactory level.

The model decreases in the amount of risk faced during the later stages of development. Testing in the early stages of development mitigates any chance of recurring issues or larger problems caused by not addressing issues at the beginning, leading them to grow worse.

The flexibility of the iterative methodology allowed new discoveries and implementations to be incorporated into the project without disrupting the development process. An example of this is the optimisation of each Claude AI prompt, making minor adjustments to improve the generative capabilities and higher quality responses provided to the user.

The iterative model directly supports the fulfilment of meeting the project requirements as it ensures that both functional and non-functional requirements are accounted for through cycles. Appropriate content generation capabilities were refined in the production of note summaries through regular testing, verifying the quality standard of educational content created. Design and user interface elements were progressively enhanced while making sure the core functionality of features remained.

Reflecting on the use of iterative methods during the project's development stages, the methodology demonstrated the value in providing a standardised approach to enhancing the project while clearly indicating progression within a short time frame.

4. Implementation

This chapter will explore the implementation stage of Scribble, outlining and presenting the project's technical features and design concepts previously mentioned in the previous section. This portion will also cover the methods used to assess the web application's effectiveness and the resulting performance. The implementation process will introduce the backend and frontend elements of the project while providing explanations of the components within these frameworks. Any complications faced during this phase of development will also be discussed, along with the solutions devised to address these issues. Additionally, testing methodologies will be analysed to demonstrate the effectiveness of the final implementation of Scribble.

4.1. Project Setup

The Scribble application was developed as a web-based platform that uses Flask as the main framework for the backend and hosts the website. HTML and CSS are employed to create the layout and styling of each page, implementing distinctive features such as the primary containers surrounding the information provided for the user, tab navigation to segregate different types of notes and information, and an information button that guides users on how to use Scribble. JavaScript formats frontend responses generated by each Claude AI model to improve user readability, but also holds the necessary backend functions which validate, alter and extract for the different elements of the application. Python's main purpose is to integrate the YouTubeTranscriptAPI, the Claude API for content summarisation and the Claude API for the chatbot messenger.

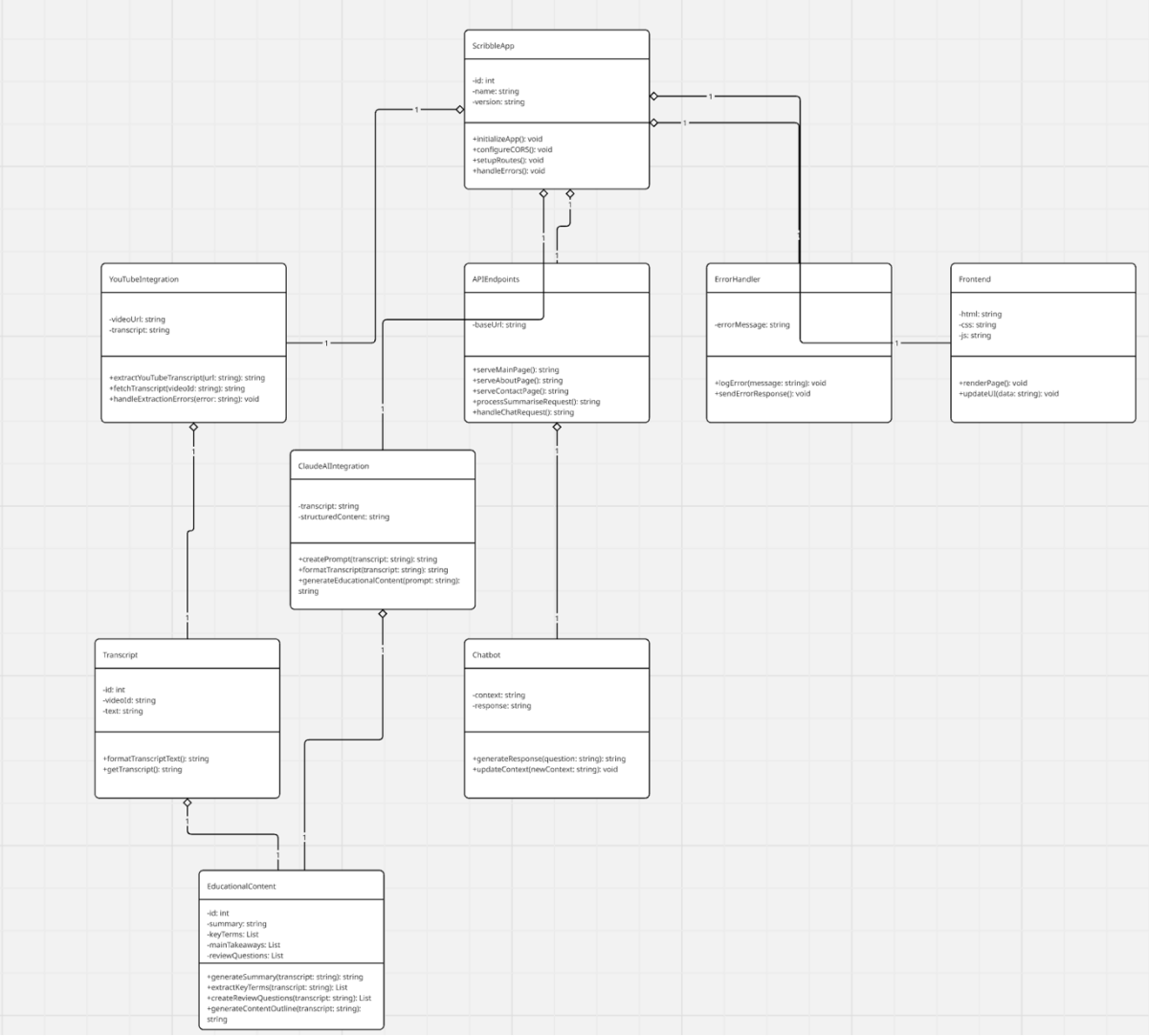


Figure 7- In-depth Code Architecture structure.

4.2. Backend implementation

4.2.1. Flask Structure

The backend functionality was implemented via the Flask framework due to the simplicity of deployment, allowing for quick validations of new application iterations. This means that any new changes made could be tried and tested regularly.

```
from flask import Flask, request, jsonify, render_template #import flask libraries
from flask_cors import CORS #for requests across origins
import anthropic # for claude chatbot model
from youtube_transcript_api import YouTubeTranscriptApi #transcript
import os
from dotenv import load_dotenv #used for api key (not working)

load_dotenv()

app = Flask(__name__, static_folder='static')
CORS(app)
```

Figure 8– Screenshot of Flask Imports

```
@app.route('/')
@app.route('/home')     #diff routes to render pages
def index():
    return render_template('ChatBotTutor.html')

@app.route('/about')
def about():
    return render_template('AboutUs.html')

@app.route('/contact')
def contact():
    return render_template('Contact.html')
```

Figure 9– Screenshot of Flask Routing

This structure of code represents the initialisation of the different components required to enable the use of the Flask application.

- Flask – Used to create an instance of a Flask application
- Request – To gain access to incoming information (API calls)
- Jsonify – converts Python to JSON to return formatted data.
- Render template – Rendering of different pages

Flask enables the ability to manage the navigation between the different pages within the project to provide the user with separated content which is structured into logical sections. This is achieved by Flask's template rendering to construct the accompanying web pages.

4.2.2. YouTube Transcript API

For the AI models to process and generate effective responses, it requires the use of this API tool in order to provide it with the context of the video.

```python
def extract_youtube_transcript(url):
    try:
        video_id=url.replace('https://www.youtube.com/watch?v=', '') #get youtube link
        transcript = YouTubeTranscriptApi.get_transcript(video_id)  #get youtube transcript

        output=''
        for x in transcript:
            sentance = x['text']    #format traanscript into string
            output += f'{sentance}\n'

        return output
    except Exception as e:
        print(f"Error with exraction: {str(e)}") #error handling
        return None
```

Figure 10– Screenshot of YouTube Transcript API

This function extracts the video transcript using the video link pasted in by the user. This link is then used by the API to fetch the transcript and format it into a string. This implementation also accounts for any case of an error occurring during the retrieval of the transcript which will then prompt the user with an error message.

During this phase, some challenges regarding the YouTube Transcript API were faced; it was found that there are some rare occurrences where videos did not have captions, meaning that there was no possibility of extracting the content from the video for video summarisation. To address this issue, the user will be prompted with an error message to notify them that they should choose another video. Although this challenge may occur, the majority of educational content on YouTube has supported captions; there are also many substitute videos on the same topics, which provide the user with an option of a different video to use in Scribble.

4.2.3. Claude AI implementation

The project's most important feature is to create summarised notes of educational content through the integration of Anthropic's Claude AI.

```python
@app.route('/api/summarise', methods=['POST']) #post request to endpoint
def summarize_video():
    try:
        data = request.json
        video_link = data.get('video_link', '') #gets video link

        if not video_link:
            return jsonify({"error": "No video link provided"}), 400 #error if an invalid link entered
        transcript = extract_youtube_transcript(video_link)

        if not transcript:
            return jsonify({"Error: no link provided"}),400 #error if there is not available transcript

        prompt_ai="""You are an AI assistant specialized in creating educational summaries and revision
        prompt =prompt_ai.replace("{VIDEO_TRANSCRIPT}", transcript)
        message = client.messages.create(
            model="claude-3-7-sonnet-20250219", #best performance model for long prompt
            max_tokens=4000,
            temperature=0.7,
            messages=[
                {
                    "role": "user",
                    "content": [
                        {
                        "type": "text",
                        "text": prompt
                }
                    ]
                }
            ]
        )
        return jsonify({"response": message.content[0].text}) #json ai responce

    except Exception as api_error:
            print(f"API error: {str(api_error)}")
            return jsonify({"error": f"API error: {str(api_error)}"}), 500 #errors
```

Figure 11– Screenshot of Claude AI 3.7 Sonnet

This set of code checks to see whether a valid video link has been entered, then extracting the video transcript and sending it to Claude AI. This information is sent alongside a customised prompt which instructs the AI model to generate their response in a certain manner which should allow for the extraction of 5 set of notes.

Anthropic's Claude 3.7 Sonnet model has been optimised to provide the best responses while being concise through the adjustment of maximum tokens. It has also been refined to produce a balanced approach in the response by changing the temperature to 0.7.

The speed of generated responses from the Claude 3.7 Sonnet model proved to cause some issues, as the dependence on video duration led to response time increasing massively. This directly effects the useability and experience as users should not have to wait for an extended period to receive their notes. This was the main motive to optimising this model further along with refining the prompt, so that responses could be generated faster.

Anthropic provides a comprehensive amount of documentation to assist the implementation of their chatbot. This includes a chatbot builder within Anthropic's Console; a sandbox like environment allowing developers to create and customise their prompts to tailor their chatbot in processing and generating information.

The environment also offers the opportunity to test the functionality before having to apply the API within any code, permitting any adjustments or fixes, to ensure a ready to use scenario. The console also contains the initial starting code to implement Anthropic's API, this can be seen within the figure below.

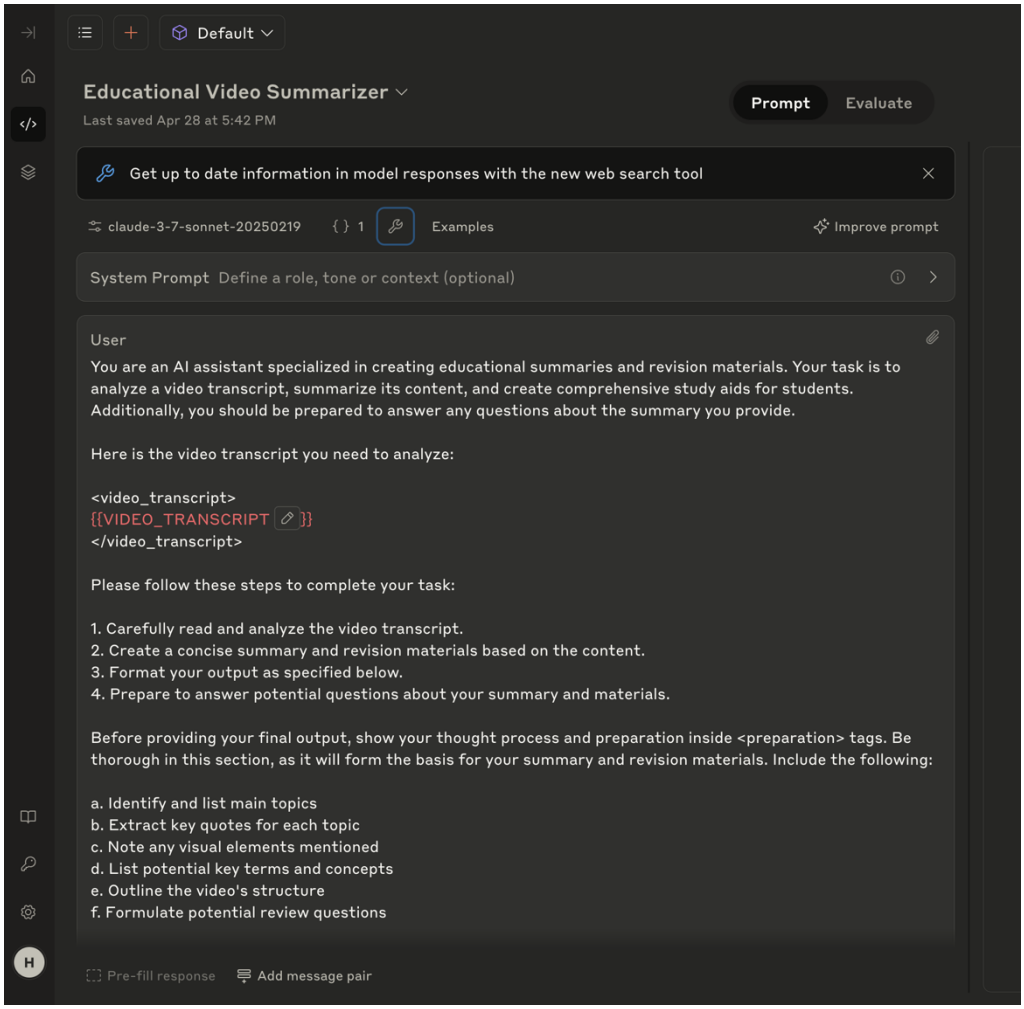


Figure 12– Screenshot of Anthropic Chatbot Builder

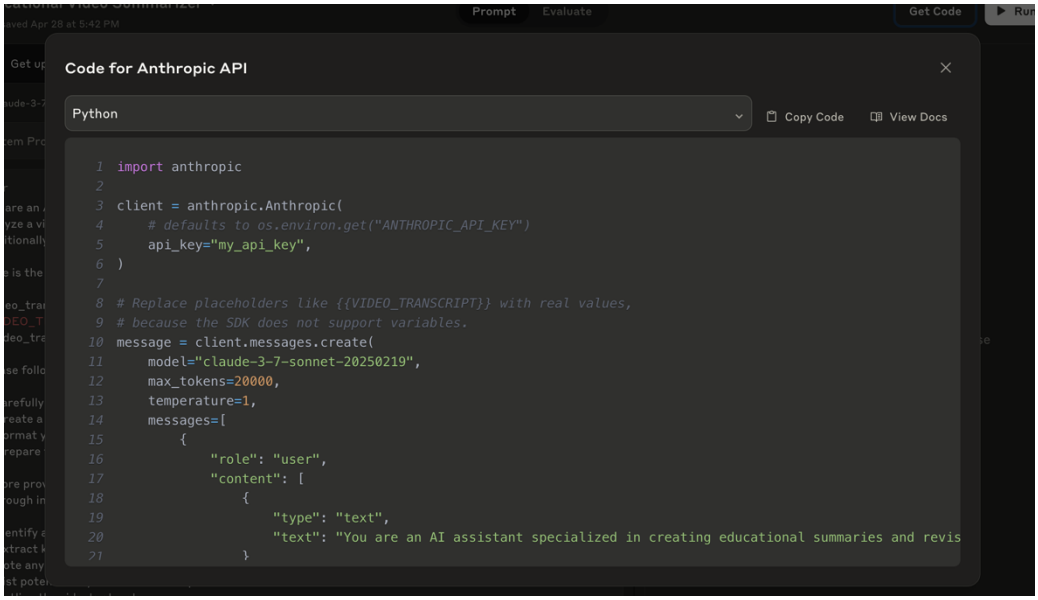


Figure 13– Screenshot of Anthropic Starter Code

The author was able to take advantage of this to effectively integrate this within the Scribble platform. The thorough support for developers in Anthropic's documentation was a significant reason for the choice as Scribble's main chatbot, with supporting evidence of the outstanding performance in delivering relevant and clear responses.

4.2.4. Claude AI Messenger

```python
@app.route('/api/chat',methods=['POST']) #post req
def message_chatbot():
        try:
            data= request.json
            userMessage=data.get('message', '') #get users submitted message
            context=data.get('context', {}) #uses summary as context

            if not userMessage:
                return jsonify({"error": "No Message"}), 400 #message error
            messengerPrompt = """You are a helpful AI assistant that helps students understand

            if context.get('summary'):
                messengerPrompt +=f"\n\n SUMMARY:\n{context['summary']}" #contect info for bot

            if context.get('keyTerms'):
                messengerPrompt +=f"\n\n Keyterms:\n{context['keyTerms']}"

            if context.get('mainTakeaways'):
                messengerPrompt  +=f"\n\n MainTakeaways:\n{context['mainTakeaways']}"

            message = client.messages.create(
            model="claude-3-haiku-20240307", #faster model for quick responces
            max_tokens=1000,
            temperature=0.7,
            system=messengerPrompt,
            messages=[
                {
                        "role": "user",
                        "content": userMessage
                }
            ]
        )
            return jsonify({"response": message.content[0].text}) #ai responce

        except Exception as api_error:
            print(f"API error: {str(api_error)}")
            return jsonify({"error": f"API error: {str(api_error)}"}), 500
```

Figure 14– Screenshot of Claude AI 3 Haiku

A technical obstacle within this implementation was maintaining the context between the user's query and the video submitted. This implementation uses a different Claude model (3 Haiku), which provides context when the user submits a message, to provide responses that are in relation to the set of notes produced by the previous Claude model, mitigating the previous issue where the chatbot may provide generic answers.

The context is passed through via the stored information sets (Summary, Key Terms, Main takeaways). The ‘users' message will be submitted alongside the video context and prompt engineered for this Claude model.

Such as the previous model, the Haiku model has been optimised to provide swift responses to user questions to provide an improved user experience. This model differs from the 3.7 Sonnet model as it is designed for less complex user demands which suits it purpose as a chatbot messenger.

4.3. Frontend Implementation

4.3.1. JavaScript

Core functionality components were implemented through JavaScript, allowing for formatting methods to be utilised for the content generated by the Claude AI models. Two JavaScript scripts, for each Claude model handled the data received and displayed this data in an organised manner within the frontend of the application.

```
function formatResponse(response){      //formating ai response
    if (!response) return 'Content not available';

    return response
        .replace(/\n\n/g, '<br><br>')
        .replace(/\n/g, '<br>')
        .replace(/^\d+\.\s+(.*?)$/gm, '<li>$1</li>')
        .replace(/^[\-\*]\s+(.*?)$/gm, '<li>$1</li>');
}
```

Figure 15– Screenshot of JavaScript for Chatbot Messenger

```
function formatContent(content){
    if (!content) return 'Content not available';

    return content
        .replace(/\n\n/g, '<br><br>') //formatting to html elements
        .replace(/\n/g, '<br>')
        .replace(/^\d+\.\s+(.*?)$/gm, '<li>$1</li>')
        .replace(/^[\-\*]\s+(.*?)$/gm, '<li>$1</li>');
}

summaryElement.innerHTML=formatContent(getSection(data.response,'summary'))     //display res
keyTermsElement.innerHTML=formatContent(getSection(data.response,'key_terms'))
mainTakeawaysElement.innerHTML=formatContent(getSection(data.response,'main_takeaways'))
reviewQuestionsElement.innerHTML=formatContent(getSection(data.response,'review_questions'))
contentOutlineElement.innerHTML=formatContent(getSection(data.response,'content_outline'))


openSection('summary');
resultContainer.style.display='block';
resultContainer.scrollIntoView({behavior: 'smooth'});
```

Figure 16 – Screenshot of JavaScript for Chatbot Summariser

The format content function arranges the generated content into a readable manner, ready to be displayed to the user. Below, formatted content is then displayed in the allocated section within the tab navigation area.

The initial implementation of displaying and organising this information led to an overload of chunky blocks of text, which displayed all 5 sets of notes within one container. The figure above represents the solution that followed this problem to improve the frontend layout.

4.3.2. Display User Submitted Video

```
embedVideoLink= videoLink.replace('/watch?v=', '/embed/'); //convert link with embed
const iframe = document.createElement("iframe");
iframe.src=embedVideoLink;                    //place in new link to work with iframe
videoContainer.appendChild(iframe);           //add in iframe
}
```

Figure 17– Screenshot of iframe implementation

This short section of code renders the video player by converting the pasted link from the search bar to enable its use within the webpage. Adjusting the '/watch?v' to '/embed/ 'in the YouTube link fixed the issue of displaying the video using the original URL.

4.3.3. Chatbot Messenger

```
function addMessage(message, isOutgoing = false, formatted = false){ //add msg to chat
    const li =document.createElement('li');
    li.className = isOutgoing ? 'outgoingChat chat': 'incomingChat chat'; //class dependant on if its
    const p = document.createElement('p');
   if(formatted){
    p.innerHTML=message; //for ai
   }
   else{
    p.textContent = message; //for user
   }

    li.appendChild(p);
    chatbox.appendChild(li);
    chatbox.scrollTop=chatbox.scrollHeight; //scroll down to bottom
}

async function sendMessage(message){ //post request user msg
    try{
    const videoSummary = document.getElementById('summary').textContent; //get content from these se
    const videoKeyTerms = document.getElementById('keyTerms').textContent;
    const videoMainTakeaway = document.getElementById('mainTakeaways').textContent;

    const response = await fetch('/api/chat', { //post req to ai
        method: 'POST',
        headers: {'Content-Type': 'application/json'},
        body: JSON.stringify({
            message: message,
            context:{ summary:videoSummary,keyTerms:videoKeyTerms,mainTakeaways:videoMainTakeaway} //

        })
    });
```

Figure 18– Screenshot of AI messenger

Figure 18 shows the two main functions representing the inner workings of the methods used to submit a user's message to the Claude model while also updating the user interface. Every time a user sends a message, the chatbot will automatically update with the corresponding styling adding to the chatlog. The same scenario occurs for any responses made by the chatbot messenger.

The procedure of supplying context to the model is also represented here in the send message function, where the summary, key terms and main takeaways are retrieved to be sent with the POST request.

4.3.4. HTML Elements Functionality

Features such as tab navigation and the information pop-up use JavaScript for their functionality.

```
let popup = null; //global

function openPopup(){
    popup.classList.add("open-popup")
}

function closePopup(){
    popup.classList.remove("open-popup")
}
```

Figure 19– Screenshot of the open/close Function for the Pop-up

```
function openSection(contentName){ //function for tab navigation
    var i;
    var x = document.getElementsByClassName("tab-content");
    for (i = 0; i < x.length; i++) {
        x[i].style.display = "none";
      }
      document.getElementById(contentName).style.display ="block";
}
```

Figure 20– Screenshot of Tab Content Display

Information Pop Up – Allows the opening and closing of the pop-up.

Tab Navigation – Displays the relevant information dependent on which tab the user has interacted with.

4.4. HTML and CSS

The design and layout of the webpage are implemented through both HTML and CSS.

The overall HTML structure is clear and concise; all functionality and adaptations to the webpage are completed through JavaScript.

```
<section id="search-wrap">
    <div class="search-content">
      <h1 class="animate__animated animate__pulse animate__infinite">Scribble.</h1>  <
      <div class="search-container">
        <div class="SearchBar">
          <form id="videoForm">
            <input type="text" id="videoLink" placeholder="Paste YouTube link here...">
            <button type="submit" id="submitBtn">
              <img src="static/Search Button White Circle 512.webp" alt="Search">
            </button>
          </form>
        </div>
      </div>
    </div>
  </section>
</div>

<div class="video-Container">  <!--iframe video-->
</div>
```

Figure 21– Screenshot of Search bar Layout

This structure embodies the main layout of the webpage that users are introduced to as soon as they load the page.

```
<div id="resultContainer">
    <h2>Video Summary</h2>
    <div id="tabContainer">
        <div class="tabs">
            <button class="tabButton" onclick="openSection('summary')">Summary</button>  <!--tabs with js
            <button class="tabButton" onclick="openSection('keyTerms')">Key Terms</button>
            <button class="tabButton" onclick="openSection('mainTakeaways')">Main Takeaways</button>
            <button class="tabButton" onclick="openSection('reviewQuestions')">Review Questions</button>
            <button class="tabButton" onclick="openSection('contentOutline')">Content Outline</button>
            <button class="tabButton" onclick="openSection('ChatBot')">ChatBot</button>
        </div>
    </div>


    <div id="summary" class="tab-content"><h2>Summary</h2></div>            <!--content for each section
    <div id="keyTerms" class="tab-content"><h2>Key Terms</h2></div>
    <div id="mainTakeaways" class="tab-content"><h2>Main Takeaways</h2></div>
    <div id="reviewQuestions" class="tab-content"><h2>Review Questions</h2></div>
    <div id="contentOutline" class="tab-content"><h2>Content Outline</h2></div>
    <div id="ChatBot" class="tab-content">
     <div class="botChat">              <!--chatbot assistant section-->
```

Figure 22– Screenshot of Tab Navigation Set Up

This figure shows the area content is displayed to within the page which is located below the search bar but is hidden until the generated response from the chatbot is received.

4.4.1. Important CSS implementations

This portion of the implementation phase will cover important CSS features which adds uniqueness to the webpage or implements smart components to improve the overall interface.

```
body, html{
    font-family: sans-serif;
    height: 100%;
    margin: 0;
    padding: 0;
    position: relative;
    scroll-behavior: smooth;
    overflow-x: hidden;
    background: linear-gradient(132deg, #ff0022, #591BC5, #212335, #60ce3f);
    background-size: 400% 400%;
    animation: Gradient 15s ease infinite; /*background gradient animation*/
}
```

Figure 23 – Screenshot Gradient Background Implementation

Figure 23 shows the inclusion of the gradient background which expands itself throughout the whole webpage to give Scribble a unique visual identity.

```css
.cube{
  border: solid 1px #D7D4E4;
  transform-origin: top left;
  transform: scale(0) rotate(0deg) translate(-50%, -50%);
  animation: cube 12s ease-in forwards infinite;
  z-index: 2;
}

.cube:nth-child(2n){
  border-color: #FFF;
}

.cube:nth-child(2){
  animation-delay: 2s;
  left: 20vw;
  top: 40vh;
}

.cube:nth-child(3){
  animation-delay: 4s;
  left: 70vw;
  top: 50vh;
}

.cube:nth-child(4){
  animation-delay: 6s;
  left: 90vw;
  top: 10vh;
}

.cube:nth-child(5){
  animation-delay: 8s;
  left: 10vw;
  top: 80vh;
}
```

Figure 24– Screenshot Cube elements

The cube elements, which are animated, are also placed in via CSS

```css
@keyframes spin{      /*aniamtion for cube*/
  0% { transform: rotate(0deg); }
  100% { transform: rotate(360deg); }
}
```

Figure 25– Screenshot of Spin animation

```css
@keyframes cube{ /*fade effect for cube*/
  from {
    transform: scale(0) rotate(0deg) translate(-50%, -50%);
    opacity: 1;
  }
  to {
    transform: scale(20) rotate(960deg) translate(-50%, -50%);
    opacity: 0;
  }
}
```

Figure 26– Screenshot of Fading Animation

The two figures above represent the animation effects of the background cubes, producing a fade and a spinning effect. These elements are important to the Scribble platform as they reinforce the application's interactivity and energetic feeling, providing students with an engaging learning experience.

4.5. Error-Handling

For Scribble to provide a reliable user experience, the application of error handling within the project is a necessity. The author has taken precautionary steps to enforce this in the architecture to ensure that both the user and developer are notified of any errors occurring within the functionality of the YouTube Transcript API and Claude API. The following applications of error handling will alert users if there is a problem with the process of the API, allowing for an effective way to diagnose problems within the code in order for it to be rectified.

```python
def extract_youtube_transcript(url):
    try:
        video_id=url.replace('https://www.youtube.com/watch?v=', '') #get youtube link
        transcript = YouTubeTranscriptApi.get_transcript(video_id)  #get youtube transcript

        output=''
        for x in transcript:
            sentance = x['text']    #format traanscript into string
            output += f'{sentance}\n'

        return output
    except Exception as e:
        print(f"Error with exraction: {str(e)}") #error handling
        return None
```

Figure 27– Screenshot of Error handling implementation for YouTube Transcript API

This function shows that if there is an issue with the retrieval and formatting of the transcript, an error prompt will appear.

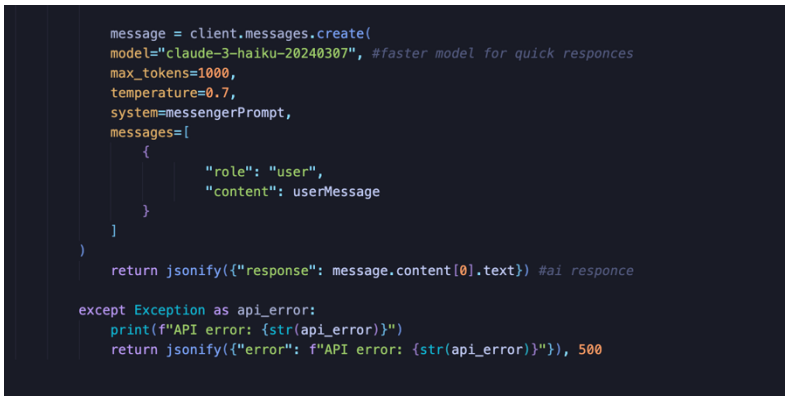


Figure 28– Screenshot of Claude Haiku Error Handling

Figure 29– Screenshot of Claude Sonnet Error Handling

Here, two points are referenced within the architecture, which will process a video transcript and user messages through Claude 3.7 Sonnet and Claude 3 Haiku. After each implementation, there is a validation section, which notifies the user of any API errors occurring.

4.6. Testing Implementation

This phase of the project will be outlined by providing a panoramic approach to how the application was tested. This includes the assessment of the main individual components (functional testing) to ensure correct functionality. Usability testing will discuss the general feedback received from the application's target audience, providing an in-depth presentation of characteristics that needed to be amended to meet user requirements and deliver sufficient usability satisfaction. The results from a comparative effectiveness

study which tests the effectiveness of Scribble shall be included within this analysis of testing.

4.6.1. Functional Testing

During iterations of development, new functions and components were implemented within the project. Components such as the chatbot summariser and messenger are the core implementations in meeting the aims and requirements set out to fulfil the applications' main purpose. These features were tested upon their installation to verify their functionality.

4.6.2. Testing Flask Functionality

Functionality testing was conducted based on the technology stack of the application. The first major implementation was the Flask framework; this was tested by importing and producing the necessary sets of code to initialise the local server, which hosted a simple HTML file which only included a ‘Hello world’ statement. The verification of initial hosting was completed once the terminal displayed the HTTP address and prompted the author with the message “serving flask app ‘chat’”.

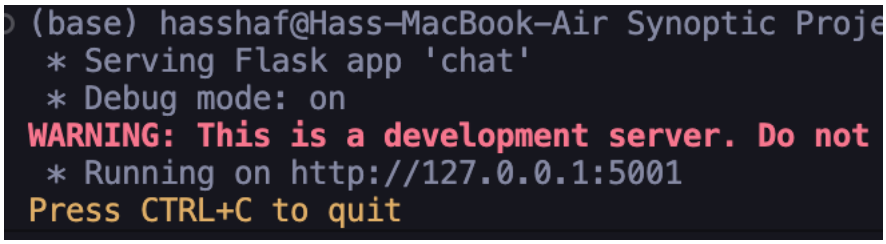


Figure 30– Screenshot of Flask Test

This was then extended by producing the accompanying ‘About Us’ and ‘Contact’ pages to validate the routing process.

Both methods of testing worked upon their first implementation, resulting in the webpage displaying each page when adjusting the main app route. Once the application progressed in development, the application routes within the Flask platform were tested once again to verify that the navigation buttons present within the header would each route to their corresponding pages.

```python
@app.route('/')
@app.route('/home')     #diff routes to render pages
def index():
    return render_template('ChatBotTutor.html')

@app.route('/about')
def about():
    return render_template('AboutUs.html')

@app.route('/contact')
def contact():
    return render_template('Contact.html')
```

Figure 31– Screenshot of Flask Routing for Testing

4.6.3. YouTube Transcript API Testing

YouTube's transcript API was tested through the use of the main page search bar, where users would paste their link. Extracting this link from the search bar and passing it into the API, the author checked that the transcript had been successfully received by printing a validating statement to the terminal. This was achieved through a basic implementation of the API, which maintained a simple print function and printed the transcript to the terminal.

```python
def extract_youtube_transcript(url):
    try:
        video_id=url.replace('https://www.youtube.com/watch?v=', '')
        transcript = YouTubeTranscriptApi.get_transcript(video_id)

        output=''
        for x in transcript:
            sentance = x['text']    #format traanscript into string
            output += f'{sentance}\n'

        print (output)
        return output
    except Exception as e:
        print(f"Error with exraction: {str(e)}") #error handling
        return None
```

Figure 32– Screenshot of YouTube API Output Test

However, upon further investigation of testing different types of content from YouTube; videos with no captions meant that they could not be summarised. The justification of the API use still withstands as there were many cases where videos of similar content which had captions could have been used instead.

4.6.4. Claude AI Summariser

Once the necessary HTML content had been integrated into the webpage, the functionality of the summary chatbot was tested. This was done to ensure that data appeared within the allocated container.

Upon the first test, a few notable observations were made: the long duration of loading times (1-minute 20s) which caused the response to time out, the formatting of Claude's response and the single page container layout were not adequate in promoting usability.

Through enforcing the use of error handling, the author was able to identify an issue occurring within the initial tests, which was due to a large token exception exceeding the API processing limits.

```
except Exception as api_error:
        print(f"API error: {str(api_error)}")
        return jsonify({"error": f"API error: {str(api_error)}"}), 500 #errors
```

Figure 33– Screenshot of Claude Token Error Test

After multiple testing cycles of altering the code, implementing formatting functions, chat bot optimisations and the implementation of a tab navigation, the summary function not only worked correctly but represented a structural improvement in the web applications practicality.

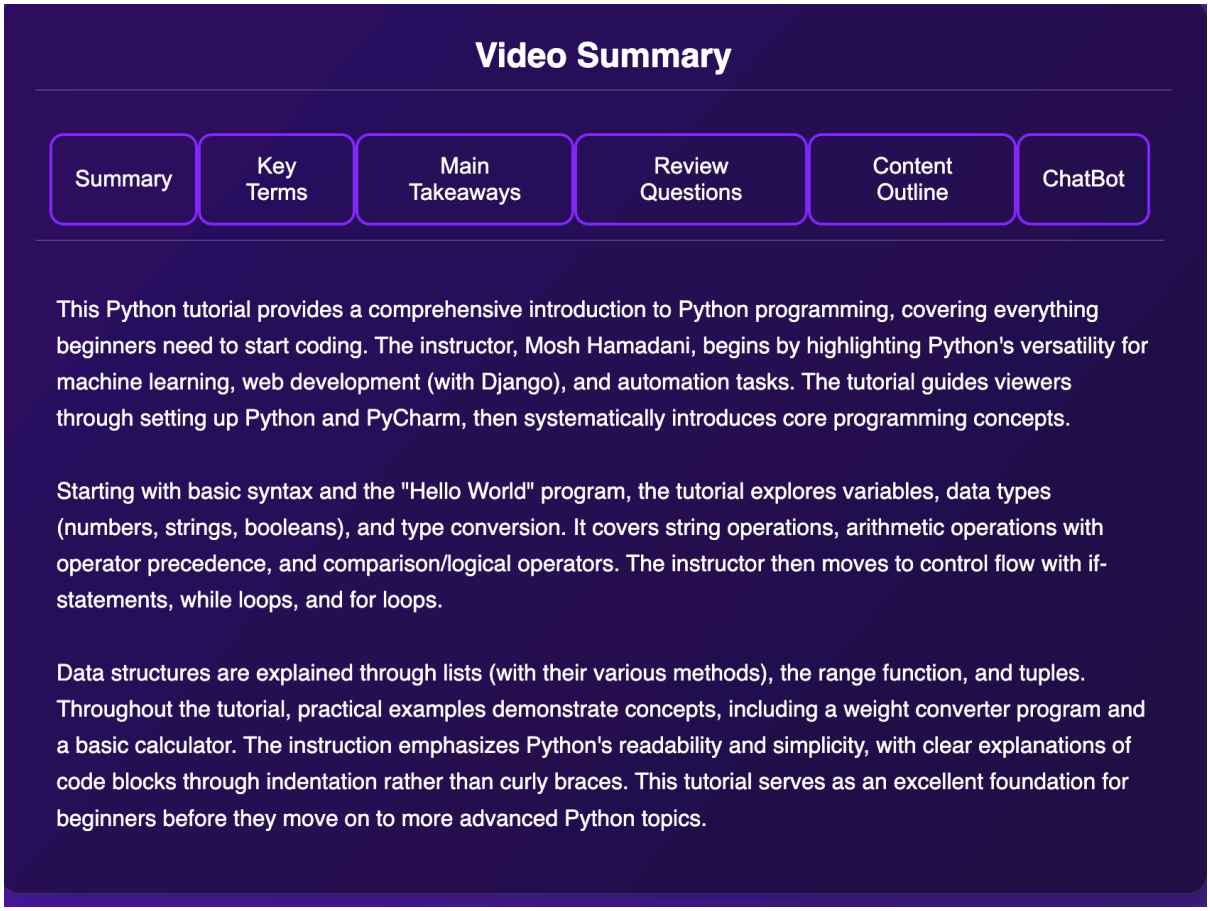


Figure 34– Screenshot of Tab Navigation Implementation

```
model="claude-3-7-sonnet-20250219", #best performance model for long prompt
max_tokens=4000,
temperature=0.7,
messages=[
```

Figure 35– Screenshot of Tab Navigation Implementation

4.6.5. Claude AI Messenger

A similar approach in testing was applied to this Claude model, as all HTML elements, including the chat box, were set up prior to the integration within the project. The process of implementing the Claude model was very similar to the one utilised for producing summaries, leading to an easier process of setting up this model within the application. During testing, the main error that occurred was the formatting of messages within the chatbot; the functionality of sending and receiving responses worked. This meant that the only adjustments needed to be made were to the CSS and JavaScript.

```
function formatResponse(response){      //formating ai response
    if (!response) return 'Content not available';

    return response
        .replace(/\n\n/g, '<br><br>')
        .replace(/\n/g, '<br>')
        .replace(/^\d+\.\s+(.*?)$/gm, '<li>$1</li>')
        .replace(/^[\-\*]\s+(.*?)$/gm, '<li>$1</li>');
}
```

Figure 36– Screenshot of Claude Haiku Formatting

```
.chatBox li{
    margin-bottom: 10px;
}

.chat{
    display: flex;
    max-width: 100%;
}

.chat p{
    padding: 10px;
    margin: 0;
    border-radius: 10px;
    max-width: 80%;
    word-wrap: break-word; /*if msg is too long*/
}

.outgoingChat{
    justify-content: flex-end; /*user msg to the right*/
}

.outgoingChat p{
    background-color: #822afd;
    color: white;
}
```

Figure 37 Screenshot of Message Styling

5. Evaluation

This evaluation chapter will provide an overview of the achievements and completion of the project, reflecting on the aims and objectives set out within the initial stages of development. The effectiveness of the project's purpose in aiding students' learning outcomes will be analysed by reviewing and assessing an effectiveness study, which was performed to compare data between students using traditional methods of information retention, against students utilising the tool produced within this project. Discussion regarding user feedback on user experience, indicating the elements within Scribble that needed to be altered or enhanced, and what implementations were appreciated.

5.1. Achievement of Aims

The original aim the author set out to achieve by the end of the project was the development of an AI-driven support tool which aids students by the provision of tailored learning resources derived from an educational video. The Scribble application successfully converts YouTube videos into a systematically organised structure including summaries, key terms, main takeaways, review questions and content outlines. This provides students with a number of options to utilise to reinforce their learning. This achievement was met with minor exceptions; students are provided with learning material produced from a YouTube video rather than a video link from any platform.

5.2. Achievement of Objectives

The objective of producing a backend which dynamically implements core features with relevant application of functions can be assumed to be fully achieved. Within the Scribble web application architecture, there is evidence of fully functional implementations of elements that meet the project requirements. This includes the YouTube transcript API that takes the user's link and extracts the transcript to pass onto the program's Claude AI summariser, which in turn produces the desired set of notes for students.

The secondary main feature of the Claude chatbot messenger, effectively responds to user question regarding the topic at hand. In addition, the backend also contains function that improve user readability through the formatting of generated responses by the Claude models.

Other attributes of the HTML webpage, such as the functioning of opening and closing the information pop-up, appearance of the summary loading indicator and tab navigation, have all been implemented successfully as they operate within the demands of the project.

5.2.1. Objective 2

The employment of user functionality and experience in the improvement of the development of the project is an additional objective which has been adequately achieved. The feedback received from the experimental group who took advantage of Scribble was used to make further enhancements to the project, leading it to the final form and pushing for a higher standard of development.

5.2.2. Objective 3

The final objective of creating an interactive and distinctive design which helps students to engage with the application itself can also be considered a success. The design of Scribble inherits unique features that set it apart from existing artificial intelligence chatbot platforms. Using the leverage of an engineered scheme of producing a brand identity, Scribble utilises animations and a tasteful gradient of background colours as part of the functional design components. Additions where information containers are highlighted, loading indicators are spinning and the way generated content is displayed, cementing the solid design of Scribble as an innovative solution enabling the ease of interactivity.

5.3. Comparative Effectiveness Study

A comparative effectiveness study took place to evaluate the educational impact of Scribble. The Study employed the following structure:

Participants: Ten students over the age of eighteen from different educational backgrounds.

Method of testing: Participants were randomly assigned to two groups, consisting of five students in each. Each group would undergo the same quiz, which is based on the same topic that stems from a chosen YouTube video (Computer Hardware).

Control Group – Uses traditional revision/note-taking methods

Experimental group – Uses Scribble and traditional methods

The aim of this study was to find a distinctive comparison between the results achieved by each group to represent the effectiveness of Scribble on education.

Quiz – Consisted of ten questions based on the content taught within the video.

Feedback Form – Participants within the experimental group were asked to complete a short questionnaire about the overall usability and experience of using Scribble.

5.3.1. Quiz Topic Selection

Computer hardware was the chosen topic for the quiz; this was decided by the author due to each key component having a different purpose, which can be understood from the definitions alone, and the large library of videos on YouTube with similar content. This means that the retention of this information can be applicable to real world scenarios to a certain extent.

5.3.2. Learning durations

The Control group was provided with a duration of thirty minutes, eight minutes to watch the video and 22 minutes to write and revise their notes in preparation for the quiz.

The Experimental group was also given the duration of thirty minutes, eight minutes to watch the video, but had the remaining twenty-two minutes to utilise Scribble in the

generation of their notes. Their only job was to utilise the learning materials to retain the information from the video.

This amount of time was selected as it is optimal for measuring the effectiveness of information retention in a short time to simulate studying under pressure.

Results:

Control Group Average: 66%

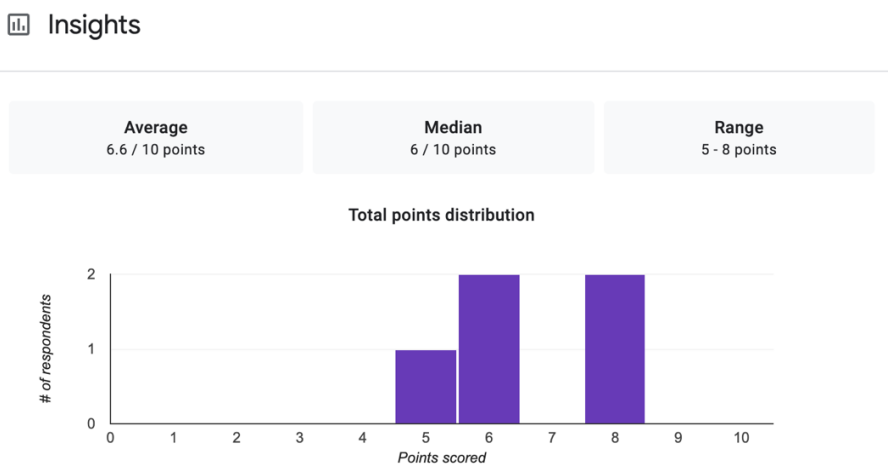


Figure 38– Screenshot of Control Group Results

Experimental Group Average: 82%

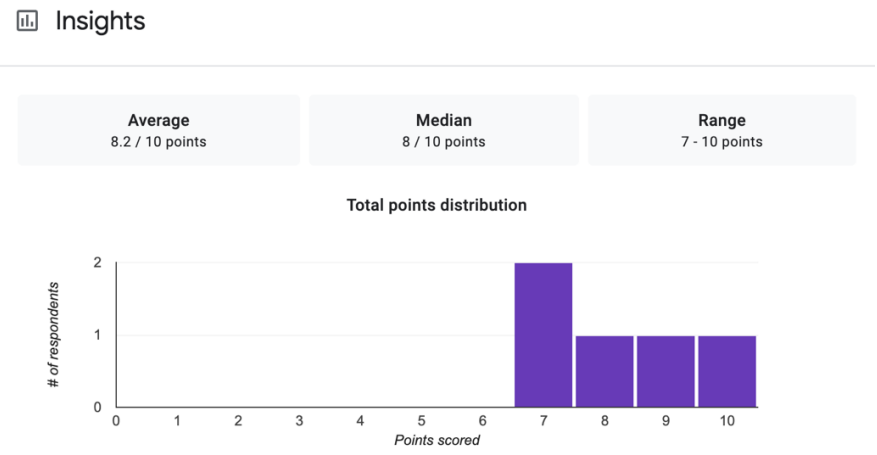


Figure 39– Screenshot of Experimental Group Results

From observing the results of the effectiveness study, there is a significant difference between the two groups. The quantitative findings present a 16 % difference in the average score achieved by each group. This improvement in performance suggests that tools such as Scribble, which combine pre-existing forms of educational content and artificial intelligence, can lead to an enhancement in student learning outcomes. The experimental group's results provide support for the advocacy of using AI tools in an educational setting, as long as they provide support in teaching and learning materials, instead of providing direct answers to questions.

5.4. User Feedback

The experimental group participants were all asked to complete an anonymous questionnaire, asking how they felt about the different aspects of the web application. Elements such as design and functionality of features were assessed.

The questions asked within this form can be viewed within appendices A,B,C.

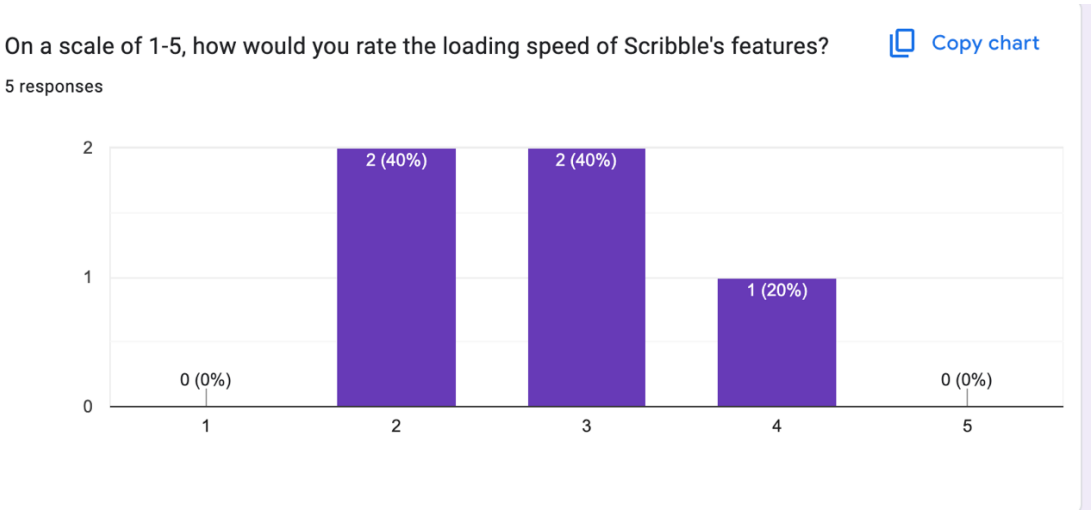


Figure 40 – Screenshot of Questionnaire Assessing Loading Speed

The general conclusions from the feedback include the consensus that the speed of some Scribble features is considerably slow. This is in relation to the loading of the summarised set of notes; this feedback let to the adjustment of the prompts length in training the Claude AI model when it produces the notes.

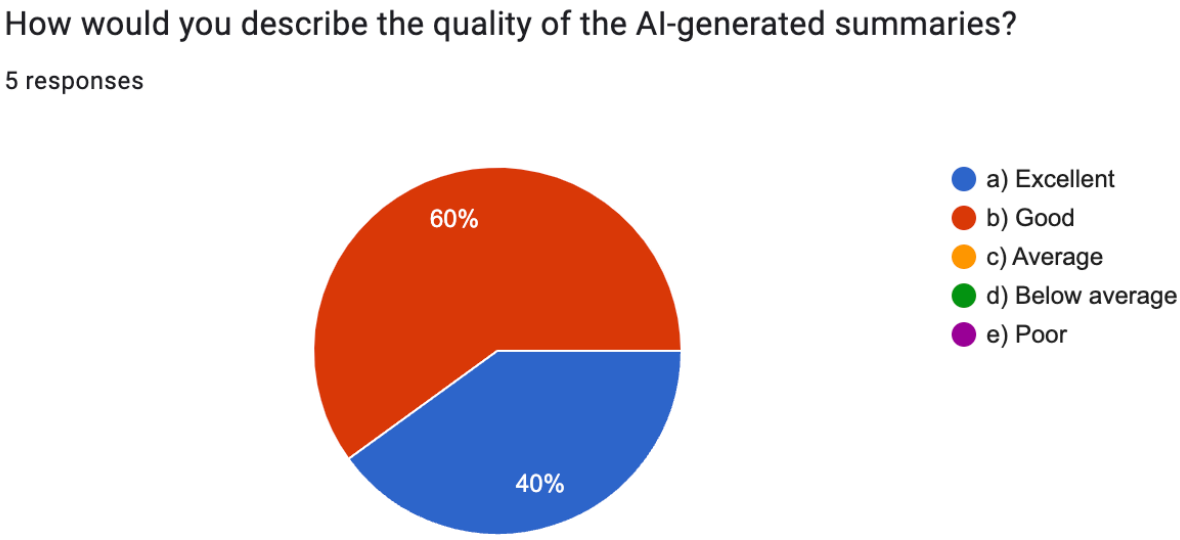


Figure 41– Screenshot of Questionnaire Assessing Quality of Generated Summaries

Although some participants found the performance of certain elements to be slow, they showed a likeness towards the quality level of the responses from the Claude model. This shows that although the prompt may be long, it provides an effective response that makes up for the long wait times.

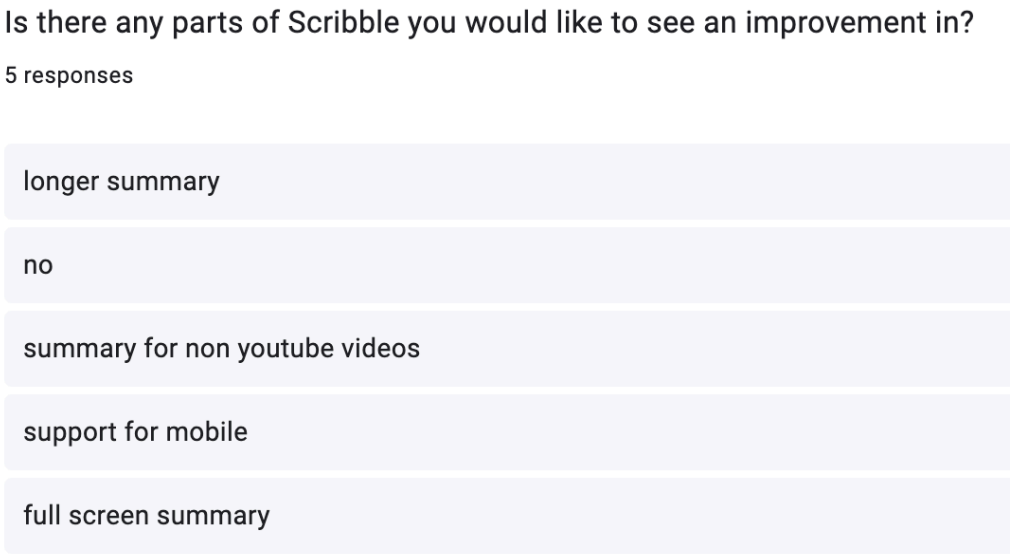


Figure 42– Screenshot of Questionnaire Project Improvements

The feedback included some suggestions from the participants in what they would like to see as an improvement in the platform. These suggestions can be viewed as possible future implementations to further elevate Scribbles wholeness as an educational tool.

Evidence of positive feedback also emerged from the questionnaire; appreciating the visual design of the platform and the organised layout within the web application.

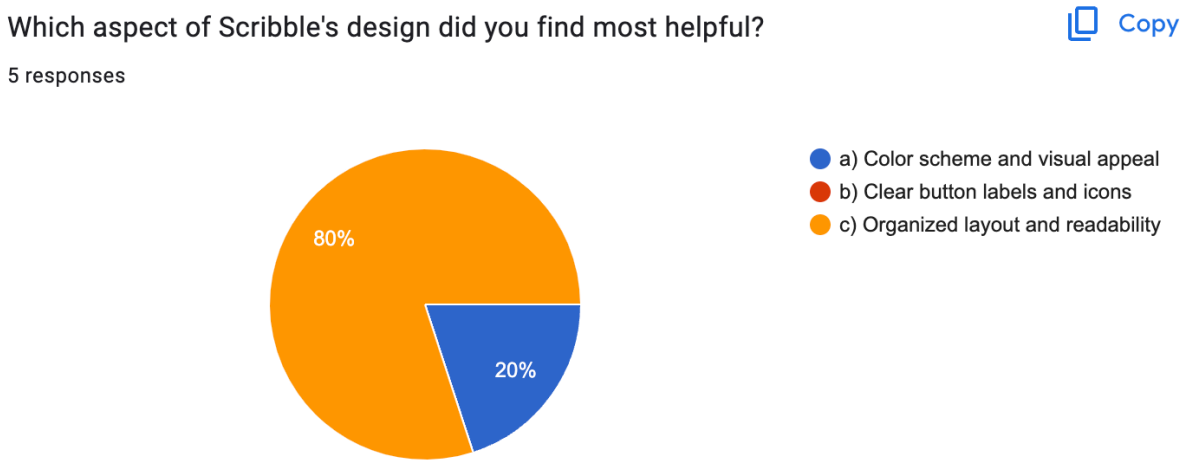


Figure 43– Screenshot of Questionnaire Assessing Helpful Design Features

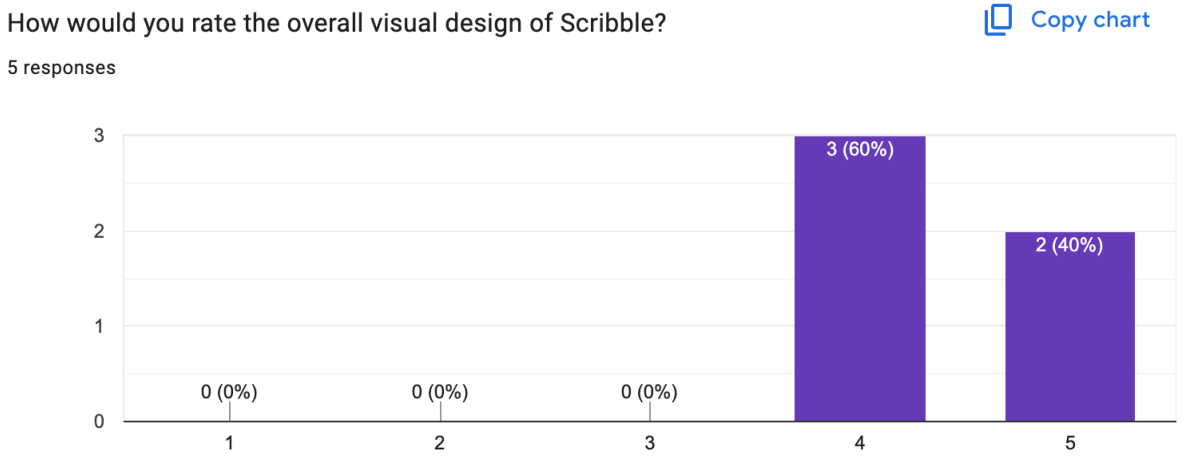


Figure 44– Screenshot of Questionnaire Assessing Helpful Design Features

5.5. Meeting Requirements

The requirements set out in the design phase of this project are all linked to the main aim and objectives achieved. Design and usability have proven to be implemented productively to provide students with an interface which is clear, organised and engaging. Functionality of core components are all working accordingly, completing the achievement of meeting all functional and non-functional requirements.

However, although these requirements have been met, the user feedback questionnaire showed the potential improvements that could be integrated into the project to remove any aspects that are holding the existing features back from being perfected. These improvements provide an opportunity to refine the application by addressing these minor performance concerns. These concerns could not be addressed within the project's timeframe, requiring more research to optimally implement solutions.

5.6. Summary

This evaluation shows that Scribble has fully achieved the majority of the aims and objectives planned within the design. By accomplishing this goal, the platform enables students to use an AI education summary tool to improve their learning capabilities and performance. The tools educational value has been validated through the effectiveness study, showing the potential effect of using it within an educational environment. Although user feedback highlighted possible areas of improvement, it also showcased the quality of the visual design and extensiveness of the AI generated notes.

6. Conclusion

This final chapter will attempt to draw together the key themes and arguments presented within earlier chapters. The conclusion will summarise the achievements and positive aspects of the project as well as the problems faced during the development period. These problems will be broken down and explained how it was solved or why it could not be solved. Any approaches for future enhancement will be discussed and explained why they are chosen.

6.1. Completed Work Review

The project was aimed at creating a tool that aids students with the process of revising educational content. The traditional methods of revision and note-taking are usually a lengthy process that can cause unwanted pressure for students. Many students find it difficult to find the best way possible to retain the information they learn, leading to the idea of producing an AI-powered web application which aims to provide students with a starting point in their revision phase.

Beginning with the concept of developing a web application similar to other AI-driven platforms, but centred around creating notes for students. Through the utilisation of the research conducted within the literature review, the majority of objectives were met, as previous studies provided the necessary methods to implement and create an environment which will have a positive outcome on a student's educational ability. The extensive research into ethical considerations led to careful thought processes during the developmental and design phase of the project. This helped to achieve an application that does not need user login details, which could be withheld within the application's backend database and guarantees that no inappropriate questions can be asked of the chatbot messenger.

This project was able to achieve its desired goal in creating an AI note-making tool that takes a YouTube video and converts it into educational revision material. This was achieved along with the representation of the effectiveness, beating traditional methods by 16%, confirming the tool’s value. Core functionality requirements consisting of three APIs (YouTube Transcript, Claude 3.7, Claude 3 - Haiku) and the displaying of the data that is received from them have all been successfully implemented.

6.2. Limitations

Although the requirements of the project have been fulfilled, during the iterations of development, the author was exposed to some limitations. These limitations did not affect the overall usability of the web application, but would have rather strengthened the project's universal application. A limitation lies within the API for the extraction of transcripts. During the testing phase of the Claude 3.7 Sonnet model, it was found that not all YouTube videos have captions to extract, meaning the user must find a substitute video to use instead. Some videos have captions but lack the quality needed to provide the Claude model with a worthwhile transcript, as it can lead to misleading information being a part of the summarised set of notes.

A previously mentioned limitation was the platform's limitation to only providing educational content for YouTube videos. This problem occurred through the realisation that models such as the Claude chatbots implemented within this project can not directly 'watch' the videos when a link is sent to them. Placing the reliance of functionality towards the use of third-party APIs like YouTube transcript to fetch the content inside the video. Although this is a limiting factor, it was decided to be limited to YouTube videos, due to the estimated time it would take to research and implement a feature which could withdraw content from any type of video.

6.3. Future Work

Although Scribble is a well-functioning application, it does require the need of future modifications to push it to a professional level which could be published and used around the world.

Future work mainly includes fixing existing limitations and areas of improvement, which were highlighted within the usability feedback form. With more time and resources, research can be conducted to find possible solutions in implementing a way to provide summaries for videos deriving from any source. This would not limit students to one platform, allowing them to use educational content from places such as university websites.

The way Scribble currently stands, it is limited in real-world application as it currently only supports English educational content. By implementing support for multiple languages, the web application will be one step closer to publication.

The final future implementation is a suggestion found within the usability feedback form, suggesting support for mobile hardware. This is achievable, with the only exception of carrying over the design elements into a mobile application. However, the choice of having the Scribble platform on mobile devices allows for a huge increase in the amount of accessibility, as more students are likely to have access to a smartphone rather than a desktop or laptop.

References

1. Ait Nasser, A. and Akhloufi, M.A. (2023). A Review of Recent Advances in Deep Learning Models for Chest Disease Detection Using Radiography. *Diagnostics*, 13(1), p.159. doi:https://doi.org/10.3390/diagnostics13010159.

2. Alotaibi, J.O. and Alshahre, A.S. (2024). The role of conversational AI agents in providing support and social care for isolated individuals. *Alexandria Engineering Journal*, [online] 108, pp.273–284. doi:https://doi.org/10.1016/j.aej.2024.07.098.

3. Benko, A. and Sik Lányi, C. (2019). History of Artificial Intelligence. *Encyclopedia of Information Science and Technology, Second Edition*, [online] pp.1759–1762. doi:https://doi.org/10.4018/978-1-60566-026-4.ch276.

4. Daphna Keidar, Yaron, D., Goldstein, E., Shachar, Y., Blass, A., Leonid Charbinsky, Aharony, I., Lifshitz, L., Dimitri Lumelsky, Neeman, Z., Mizrachi, M., Majd Hajouj, Nethanel Eizenbach, Sela, E., Weiss, C.S., Levin, P., Ofer Benjaminov, Bachar, G., Tamir, S. and Rapson, Y. (2021). COVID-19 classification of X-ray images using deep neural networks. *European Radiology*, 31(12), pp.9654–9663. doi:https://doi.org/10.1007/s00330-021-08050-1.

5. Davar, N.F., Dewan, M.A.A. and Zhang, X. (2025). AI Chatbots in Education: Challenges and Opportunities. *Information*, [online] 16(3), p.235. doi:https://doi.org/10.3390/info16030235.

6. Dhar, T., Dey, N., Borra, S. and Sherratt, R.S. (2023). Challenges of Deep Learning in Medical Image Analysis -Improving Explainability and Trust. *IEEE Transactions on Technology and Society*, pp.1–1. doi:https://doi.org/10.1109/tts.2023.3234203.

7. E. Sobaih, A.E. (2024). Ethical concerns for using artificial intelligence chatbots in research and publication: Evidences from Saudi Arabia. *Journal of Applied Learning & Teaching*, 7(1). doi:https://doi.org/10.37074/jalt.2024.7.1.21.

8. Ghauri, M.S., Reddy, A.J., Tak, N., Tabaie, E.A., Ajay Ramnot, Parsa Riazi Esfahani, Neel Nawathey and Siddiqi, J. (2023). Utilizing Deep Learning for X-ray Imaging: 9. Detecting and Classifying Degenerative Spinal Conditions. *Cureus*. doi:https://doi.org/10.7759/cureus.41582.

9. Haenlein, M. and Kaplan, A. (2019). A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. *California Management Review*, [online] 61(4), pp.5–14. doi:https://doi.org/10.1177/0008125619864925.

10. JIN (2023). *Software Development Framework — Iterative Model*. [online] Geek Culture. Available at: https://medium.com/geekculture/software-development-framework-iterative-model-68584bfad773.

11. Kolluri, S., Lin, J., Liu, R., Zhang, Y. and Zhang, W. (2022). Machine Learning and Artificial Intelligence in Pharmaceutical Research and Development: a Review. *The AAPS Journal*, [online] 24(1). doi:https://doi.org/10.1208/s12248-021-00644-3.

12. Kooli, C. (2023). Chatbots in education and research: a critical examination of ethical implications and solutions. *Sustainability*, [online] 15(7), p.5614. doi:https://doi.org/10.3390/su15075614.

13. Shoufan, A. (2022). *YouTube and Education: A Scoping Review*. [online] ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/abstract/document/9965379.

14. Tawfeeq, T., Awqati, A. and Jasim, Y. (2023). The Ethical Implications of ChatGPT AI Chatbot: A Review. *Journal of Modern Computing and Engineering Research*, [online] 2023, pp.49–57. Available at: https://jmcer.org/wp-content/uploads/2023/08/The-Ethical-Implications-of-ChatGPT-AI-Chatbot-A-Review-1.pdf#page=3.53 [Accessed 23 Apr. 2025].

15. Vamsi Krishna Uppalapati and Deb Sanjay Nag (2024). A Comparative Analysis of AI Models in Complex Medical Decision-Making Scenarios: Evaluating ChatGPT, Claude AI, Bard, and Perplexity. *Cureus*. doi:https://doi.org/10.7759/cureus.52485.

16. Yasser Albogami, Almaha Alfakhri, Abdulaziz Alaqil, Aljawharah Alkoraishi, Heba Alshammari, Elsharawy, Y., Abdullah Alhammad and Abdulaziz Alhossan (2024). Safety and quality of AI chatbots for drug-related inquiries: A real-world comparison with licensed pharmacists. *Digital Health*, [online] 10. doi:https://doi.org/10.1177/20552076241253523.

Bibliography

https://stackoverflow.com/questions/76856230/how-to-extract-youtube-video-transcripts-using-youtube-api-on-python

https://github.com/anthropics/anthropic-sdk-typescript

https://www.geeksforgeeks.org/create-working-chatbot-in-html-css-javascript/

https://www.anthropic.com/api

https://www.w3schools.com/hTml/html_youtube.asp

https://unused-css.com/blog/css-outer-glow/

https://boxicons.com/?query=bot

https://www.w3schools.com/cssref/tryit.php?filename=trycss_anim_box-shadow

https://www.w3schools.com/howto/tryit.asp?filename=tryhow_css_contact_form

Appendices

Appendix A

Computer Hardware Quiz- Control Group

Form description

1. What is the main function of the CPU (Central Processing Unit) in a desktop computer?

- A) Providing storage for files and programs
- B) Rendering visual output to the display
- C) Processing data and running programs
- D) Supplying power to the computer components

2. Which component provides temporary, fast-access storage for active programs and data?

- A) Hard drive
- B) Motherboard
- C) RAM (Random Access Memory)
- D) Graphics card

Appendix B

3. Where are installed programs and user data files permanently stored in a desktop computer?

- A) RAM
- B) Motherboard
- C) Power supply
- D) Hard drive

4. What is the primary role of the graphics card (GPU) in a desktop computer?

- A) Connecting all the internal components
- B) Converting data into visual output for the display
- C) Providing power to the entire system
- D) Storing large amounts of user data

5. Which component acts as the central connection point for all the other internal parts of a desktop computer?

- A) Power supply
- B) CPU
- C) Motherboard

D) Hard drive

What is the process called when data is transferred from the hard drive to RAM for faster access by running programs?

A) Rendering

B) Loading

C) Calculating

D) Powering

7. How are the storage capacities of RAM and hard drives typically different?

A) RAM has larger storage capacity than hard drives

B) RAM and hard drives have equal storage capacities

C) Hard drives have larger storage capacity than RAM

D) Storage capacity is not a differentiating factor

Which of the following is not considered one of the seven essential components of a desktop computer system?

A) Case

Appendix D

B) Sound card

C) Power supply

D) Motherboard

What is the primary function of the power supply unit in a desktop computer? *

A) Connecting all internal components

B) Providing electricity to the other components

C) Storing data and running programs

D) Rendering visual output to the display

10. How do the CPU and graphics card differ in terms of their specialised roles within the computer system?

Multiple choice

A) They serve identical functions

B) The CPU handles visual rendering, the graphics card handles processing

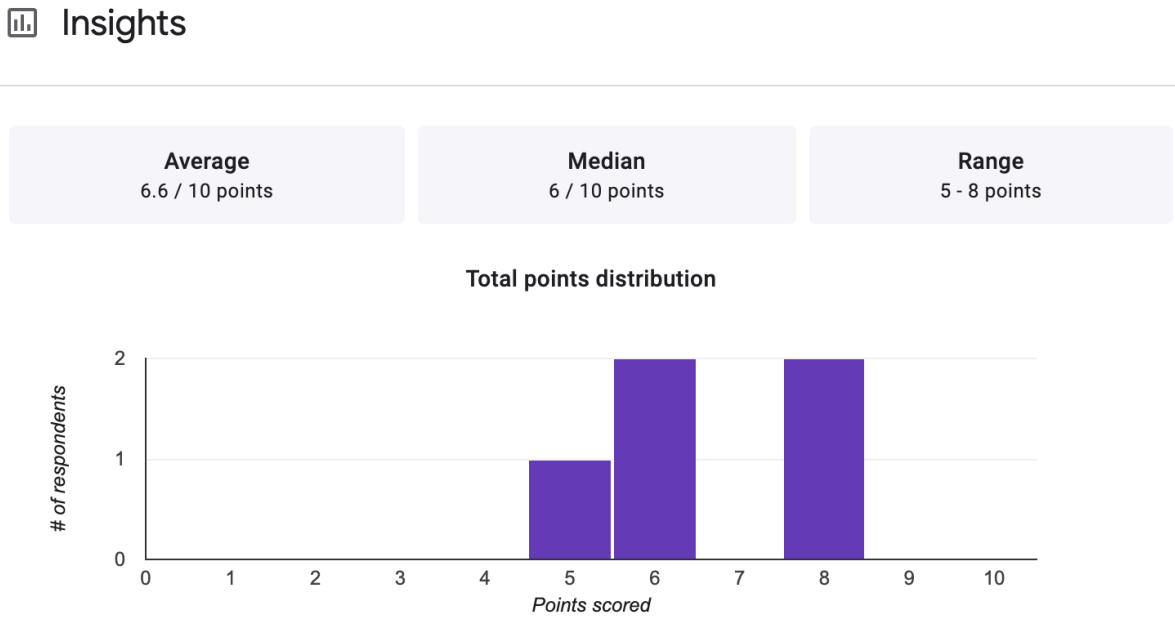
Add image

C) The CPU handles processing, the graphics card handles visual rendering

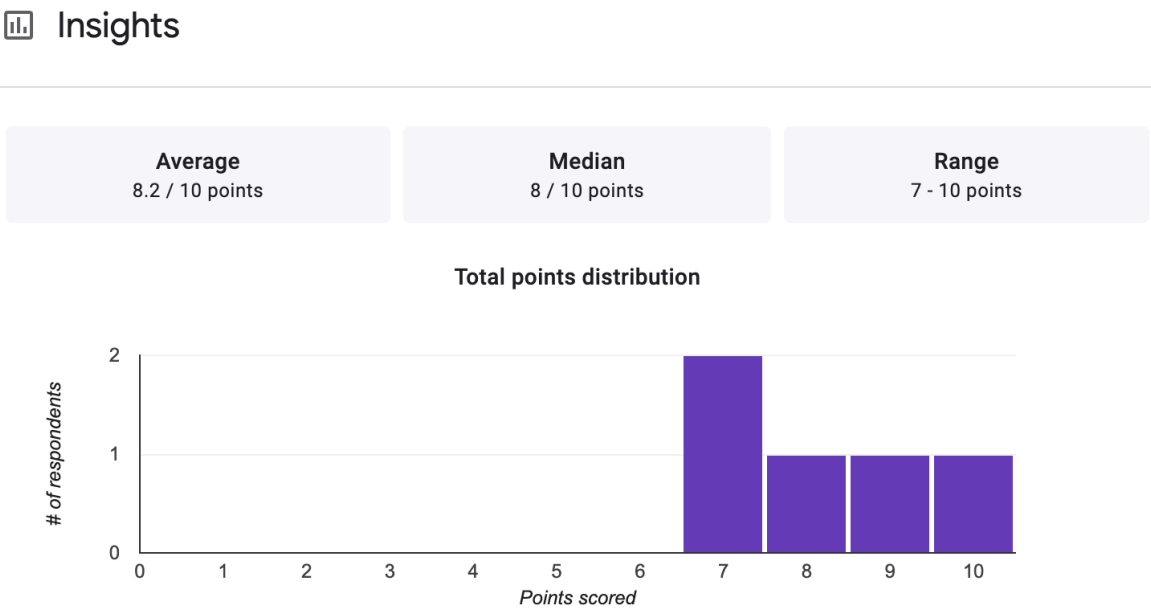
D) The CPU and graphics card have no distinct roles

Add option or add "Other"

Appendix E



Appendix F

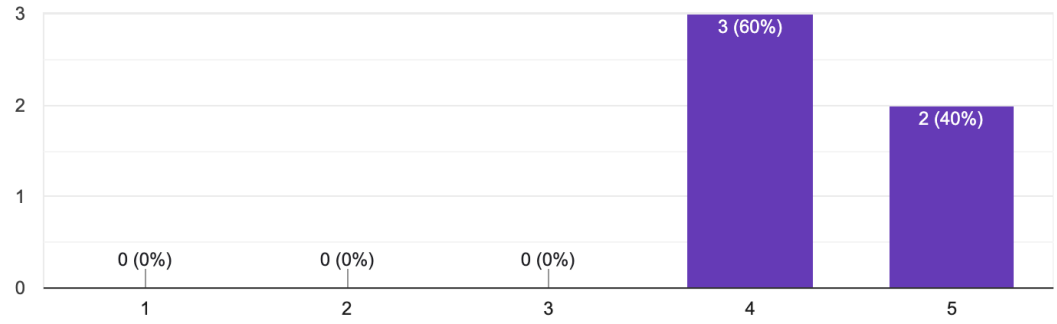


Appendix G

How would you rate the overall visual design of Scribble?

Copy chart

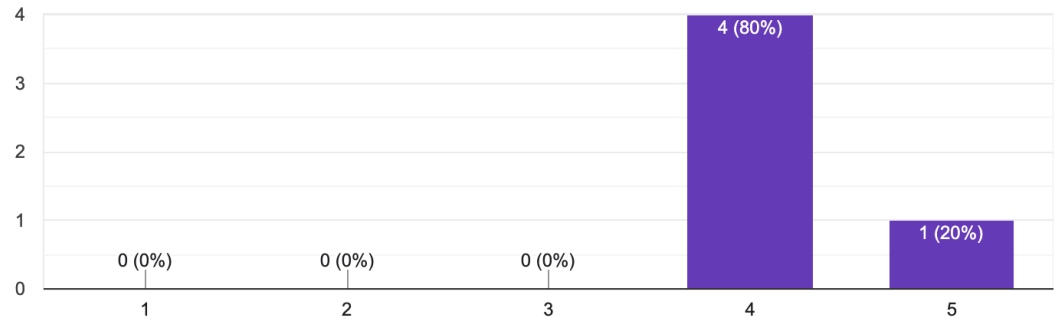
5 responses



How easy was it to navigate between different sections of the application?

Copy chart

5 responses

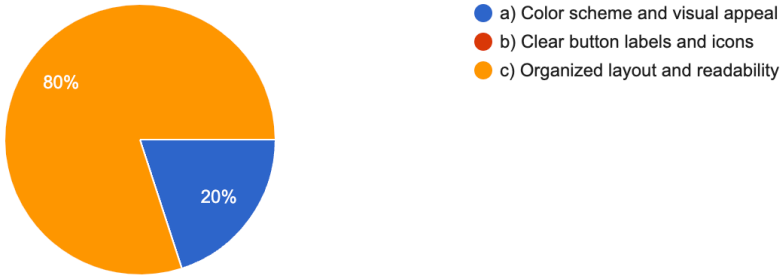


Appendix H

Which aspect of Scribble's design did you find most helpful?

Copy chart

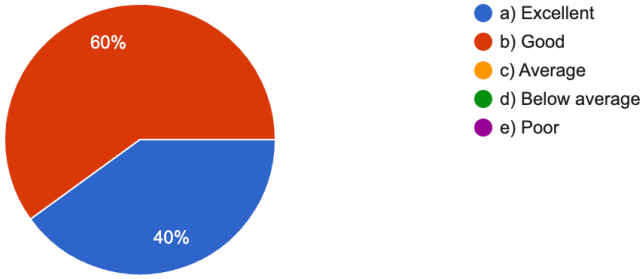
5 responses



How would you describe the quality of the AI-generated summaries?

Copy chart

5 responses

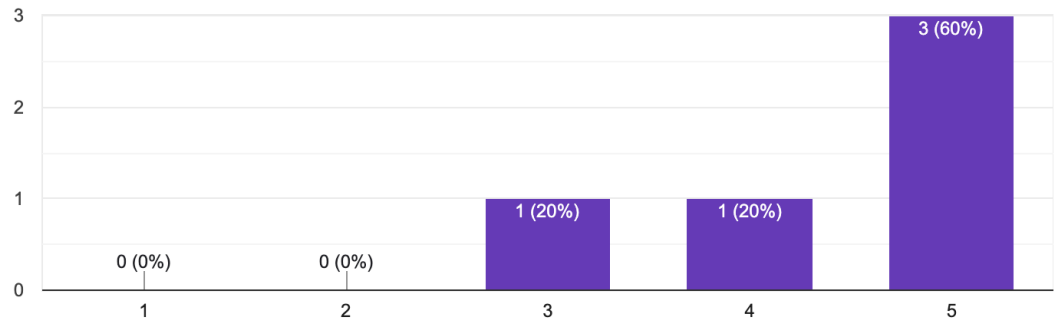


Appendix I

How useful did you find the chat interface for asking follow-up questions?

Copy chart

5 responses



Is there any parts of Scribble you would like to see an improvement in?

5 responses

longer summary

no

summary for non youtube videos

support for mobile

full screen summary

Participant Information Sheet

Interactive student support tool for lecture dissemination

Invitation to research

I am currently a student studying computer science at Manchester Metropolitan University. I invite you to participate within this research, which will test the usability of the interactive support tool and its effects on student learning. Two sets of parties have been invited to participate within this investigation; both students and educators within higher education such as universities.

Why have I been invited?

The decision to invite students and educators within higher education was made due to the level of complexity and the amount of content involved within a course, making the process of completing assignments and exams a difficult time which takes a large amount of dedicated time and effort to complete.

Do I have to take part?

The decision for a participant to participate within this research is completely voluntary and is up to the volunteers to decide to share their results.

What will I be asked to do?

Participants will be split into to two groups, experimental and control.

The experimental participants will be asked to use the provided tool to help assist with their educational needs in completing an assigned task that will help determine the effectiveness of the interactive support tool compared to traditional methods. Volunteers within the control will be required to complete the same set of questions but using standard studying methods without the use of the tool.

Are there any risks if I participate?

There are no potential risks in participating within this research. Regarding data privacy and protection, all participants data will be kept confidential and held within MMU encrypted OneDrive servers.

Are there any advantages if I participate?

There are no financial benefits or rewards that participants will receive from their participation within this research.

Informed consent

By agreeing to partake within this study, you are consenting to take part in an experiment that will aid to evaluate an AI student support tool which is designed to enhance a student's academic performance. Participants will be randomly assigned to either a control group, which will use traditional methods of study, or an experimental group, which will utilise the AI tool and results will be collected and evaluated. Each participant will be anonymous, with only the results being collected. The decision to participate is completely voluntary and participants may choose to leave at any point without any consequences. If this occurs all data tied to that person will not be used and will remain confidential.

What information about me will you collect and why?

The required information that we will need to clarify, consists of the participants age (over 18) and if they speak English as they're native language, however, this will not be collected as it will not be of use within the rest of the study. No names or sensitive information will be held and is not required for this study, every participant will be taking part anonymously.

How will my information be stored and how will you look after it?

Data collected within this research will be obtained anonymously, meaning no names will be tied to the results that will be recorded during the experimental stage. There will be no identifiable piece of information that can be connected back to the participant. All data recorded will be stored upon the Manchester Metropolitan Universities OneDrive which is an encrypted safe space against any data breaches.

How will you use my information?

All participants will be anonymous and unidentifiable, the data will only be used for internal purposes and will not be found on any publications or alternative outputs.

Will my data be sent anywhere else, or shared with other people or organisations?

The data collected will not be shared among any other people that are not part of the management side of the study, data will not leave the country or the premises of MMU.

When will you destroy my information?

All data collected will be held for the duration of the study. (approx. 1 month). Once the analysis and evaluation of the data has been completed, the data will be destroyed on both physical and digital storages to ensure confidentiality.

Data Protection Law

Data protection legislation requires that we state the 'legal basis' for processing information about you. In the case of research, this is 'a task in the public interest.' If we use more sensitive information about you, such as information about your health, religion, or ethnicity (called 'special category' information), our basis lies in research in the public interest. Manchester Metropolitan is the Controller for this information and is responsible for looking after your data and using it in line with the requirements of the data protection legislation applicable in the UK.

You have the right to make choices about your information under the data protection legislation, such as the right of access and the right to object, although in some circumstances these rights are not absolute. If you have any questions, or would like to exercise these rights, please contact the researcher or the University Data Protection Officer using the details below.

You can stop being a part of the study at any time, without giving a reason. You can ask us to delete your data at any time, but it might not always be possible. If you ask us to delete information within 2 weeks', we will make sure this is done. If you ask us to delete data after this point, we might not be able to. If your data is anonymised, we will not be able to withdraw it, because we will not know which data is yours.

What will happen to the results of the research study?

The results collected from this study will be used to evaluate the effectiveness of the Interactive student support tool in improving academic performance and learning. These results will only be used within in one report which will analyse the research as a whole and will not be accessible by the public. All participation will continue to be anonymous within this report, with only the statistical outcomes being represented.

Who has reviewed this research project?
The responsible institution for reviewing this research project will be Manchester Metropolitan University. They will be responsible for handling and storing all information and data regarding this study.

Who do I contact if I have concerns about this study or I wish to complain?
For any uncertainty or complaints please contact Haseeb Shafiq, TEL: 07493881787, Email: HASEEB.SHAFIQ@stu.mmu.ac.uk.

Alternatively, you may contact Dr Naiomi Adel, Tel: 01612471111, Email: n.adel@mmu.ac.uk

Manchester Metropolitan Data Protection Officer our Data Protection Officer can be contacted using the dataprotection@mmu.ac.uk e-mail address, by calling +44 (0)7584 330586or in writing to: Data Protection Officer, Legal & Governance, Ormond Building, Lower Ormond Street, Manchester, M15 6BX

UK Information Commissioner's Office
You have the right to complain directly to the Information Commissioner's Office if you would like to complain about how we process your personal data:
https://ico.org.uk/global/contact-us/

THANK YOU FOR CONSIDERING PARTICIPATING IN THIS PROJECT