

VIP-ITS Fall 2023 Project Proposal - ChatGPT Summarizer

Group Membership:

Member		
Alyssa Zhu	Skills	Java, Python, C, HTML, SQL, CSS, JavaScript, Swift
	Credits	2
	Responsibility	<ul style="list-style-type: none"> ● Read the previous code (ChatGPT ChatBot and ChatBot Recommender) <ul style="list-style-type: none"> ○ Brainstorm what can be reused/added to for summarizing ● Learn/understand ChatGPT API and connection functionality with frontend ● Research ChatGPT and other models ● Test different prompts to ChatGPT API to get best summarizing results ● Develop frontend functionalities as necessary, reinforcing Swift skills
Alex Tang	Skills	Java, Python, CSS, HTML, SQL, JavaScript, Swift
	Credits	2
	Responsibility	<ul style="list-style-type: none"> ● Read the previous code and understand how the ChatBot works ● Improve Swift skills and work on improving the frontend and integrating this with the QuizApp ● Work on developing ChatGPT prompts to deliver consistent, accurate summaries of texts ● Develop the UI accordingly in the frontend ● Research different ways to parse through PDFs and find the key topics of different files
Nathan Papa	Skills	Java, Python, HTML, CSS, JavaScript
	Credits	1
	Responsibility	<ul style="list-style-type: none"> ● Read previous code from all prior repositories ● Learn necessary tools for development (likely Swift) ● Research ChatGPT and other chatbots to develop a fundamental understanding of how they work and determine which model we want to use

- | | | |
|--|--|---|
| | | <ul style="list-style-type: none">• Integrate text summarizing chatbot into QuizApp |
|--|--|---|

Project Goals:

Currently we have two independent iOS applications (ChatBot and QuizApp). We want to integrate the two applications together for one cohesive product. Our main goal is to improve the apps (iOS) by implementing a **text summarizer** feature. We will start by reading through the previous code to understand what research and implementation has already done for the ChatGPT app and ChatBot Recommender textbook summary algorithm. There needs to be research done into PDF text parsing and how ChatGPT works. Some front end work also needs to be done to create a finished look for the application itself. The focus of this project is to create a function within the apps that will summarize text the user submits. Once the functionality is added, we can work on limiting the summarizer to class specific material. By the end of this project, we hope to gain more knowledge about ChatGPT and API calls.

Project Timeline:

Week 1-3: Team formation, proposing project ideas, planning the project.

Week 4-5: Read into the previous ChatBot code/documentation and the summarizer application. Research ChatGPT API calls, effective prompts, and ways to parse through PDF's

Week 6-9: Integrate the ChatBot into the iOS QuizApp application. Begin working on creating API calls to ChatGPT and test for consistent, accurate work. The frontend of this application should also be improved for a smoother experience

Week 10-12: Research into parsing through PDFs and automatically generating summaries/key topics and give the option to provide more information and practice exercises

Week 13-15: Refine the general application overall. Make sure all branches are properly merged and the user experience is as polished as possible.

Week 16-17: Wrap up the slides and app. Final Presentation.

Project Description:

Problem and Research Ideas

Our project aims to solve the problem of students struggling to be able to read a textbook or an article for a class and be able to easily gather the important information from the text. We plan to solve this by implementing a new text summarization feature in the existing ITS chatbot. Our new feature will allow a student to input a section of text that our chatbot will search through and extract much shorter and more readable key points so the user can read the key points instead of reading the inputted text. This way, they can gain the key knowledge they need without having to read through pages of text.

In terms of research, we definitely need to research ChatGPT itself more to gain an understanding of how it works. This way we can develop our chatbot with a fundamental understanding of what we should create. Additionally, we need to research the ChatGPT API so that we can actually develop our chatbot as none of us have much experience with the API. We also need to take the time to research methods of searching the text and determining what information we should return to the user. We also need to figure out how to make the information we return to the user more readable and comprehensible.

Foreseeable Problems and Potential Pitfalls

We will still need to spend some time getting comfortable with Swift, the previous ChatBot code, and making API calls to ChatGPT and eventually using other libraries to implement the PDF parser. There is also the issue of ChatGPT becoming more and more inaccurate in its answers lately, so we will need a way to consistently return accurate responses.

Another problem that we can foresee is that, since Nathan has a Windows computer, he needs to find a way to run the Swift Application and he might experience issues when testing the app itself. This could negatively impact his development experience, and thus we will have to work through such problems together. Also, this means that for the demo, we will not be able to use his machine and will have to rely on Alyssa and Alex's Apple computers. It could be possible that Nathan works on algorithms outside of the Swift app to be applied (since we simply just need the data returned from the API call).

Implementation, Tools, Resources, and Collaboration

This project is mainly developed in Swift with the backend API calls to ChatGPT using NodeJS, which we use as well as any additional technologies we need for our PDF parsing. We will fork the ITS-Swift-GPTChatBot repository on GitHub to use as the starting template that we can add the text summarizer to. Each member will create their own branches and commit their individual contributions to their respective branches. We will primarily communicate on Microsoft Teams and will meet weekly on Fridays 12:00 PM to ask questions, solve problems, and coordinate tasks. These meetings should last around 30 minutes.