**Introduction**

The introduction of Generative AI has allowed for a renaissance of productivity, creativity, and collaboration. ChatGPT, the most popular of the Gen AI tools, has been quickly adopted by the software development community and is making its mark in the industry. With the ability of generative AI tools to create, analyze, and refactor code, the productivity of developers has the ability to flourish if these tools are used effectively. This possibility of enhanced productivity has motivated this research. Others have looked into this developing field and have published their findings that are referenced in this research.

In the study *Can ChatGPT Support Developers? An Empirical Evaluation of Large Language Models for Code Generation [1]*, the researchers also use the same dataset as the one used in this paper, though they used data from all five Github features which are Code Files, Commits, Issues, Pull Requests, and Discussions, which broadens their scope of study while instead we narrow it to examine the field of sharing. Though this works for their RQs, it is too broad for the questions we pose about the sharing feature and its correspondence to ChatGPT.

In another study done by Dr. Eman AlOmar and others called *Exploring ChatGPT’s code refactoring capabilities: An empirical study [2]*, they looked at ChatGPT and how reliable it is when prompted to refactor and change code. They told ChatGPT to refactor forty Java segments and found that thirty-nine times out of forty the refactor code was done correctly and reliably. A concern with this use of ChatGPT is that it is not consistent, where the same prompt done twice will give two different responses. Yes, it will do what is asked correctly but it is not consistent.

Recent related studies have shown that only a median of 54% of ChatGPT-generated code snippets are directly implemented and integrated by developers. In addition, ChatGPT-generated code typically remains in the developer’s workspaces for almost three months, emphasizing the importance of effective refactoring by ChatGPT and efficient workspaces. Out of these integrated code snippets, it was also found that 82% of code snippets generated are modified at some point. Our research also found supporting results, showing that in 48% of cases, programmers implement ChatGPT-generated code, while in 38% they do so indirectly. So, it is crucial to understand that regardless, developers frequently review AI-generated code.

**Results**

**Background:**

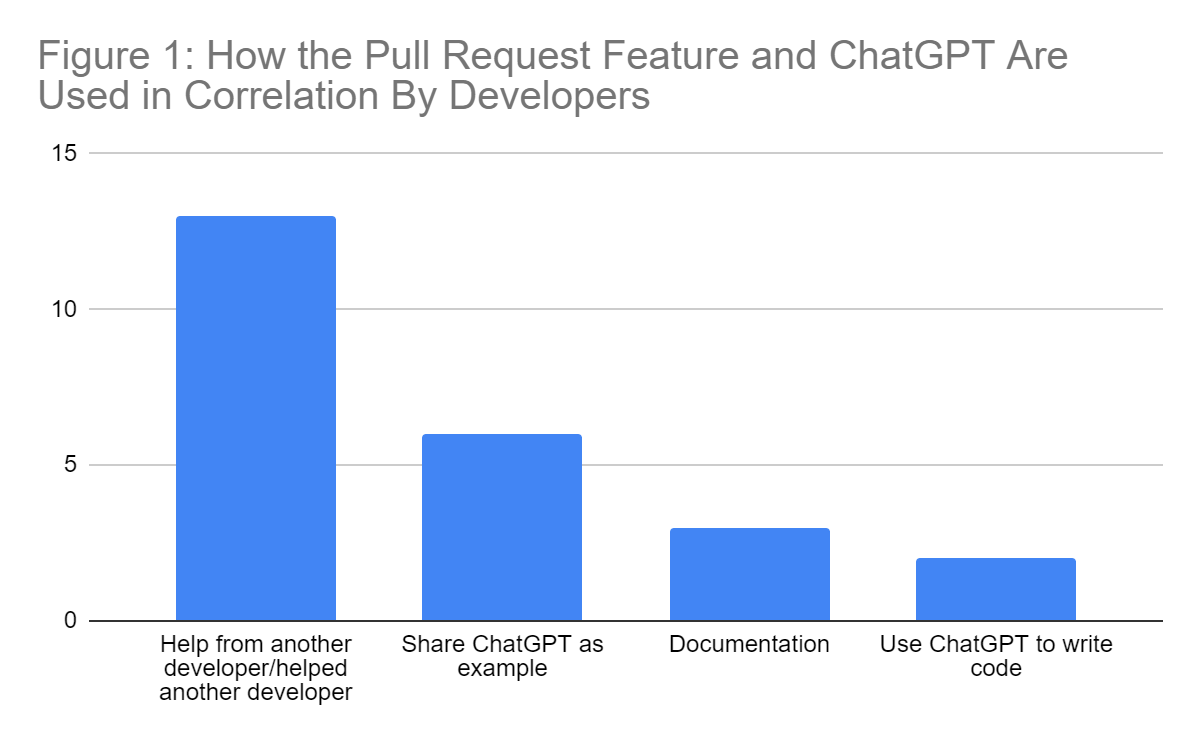
DevGPT is a database that contains conversations of ChatGPT that are used by developers to aid them in creating projects that include and utilize the benefits of AI. It allows users to share their work and ideas in order to improve the productivity and efficiency of a program. An Excel Spreadsheet was given that contains around seventeen hundred rows of Github entries that involve the use of DevGPT or ChatGPT between developers. This Excel Spreadsheet was used for both Research Question #1 and Research Question #2.

***RQ #1: How does the introduction of sharing feature technology in accordance with ChatGPT impact the ways that developers are using ChatGPT?***

**Method:**

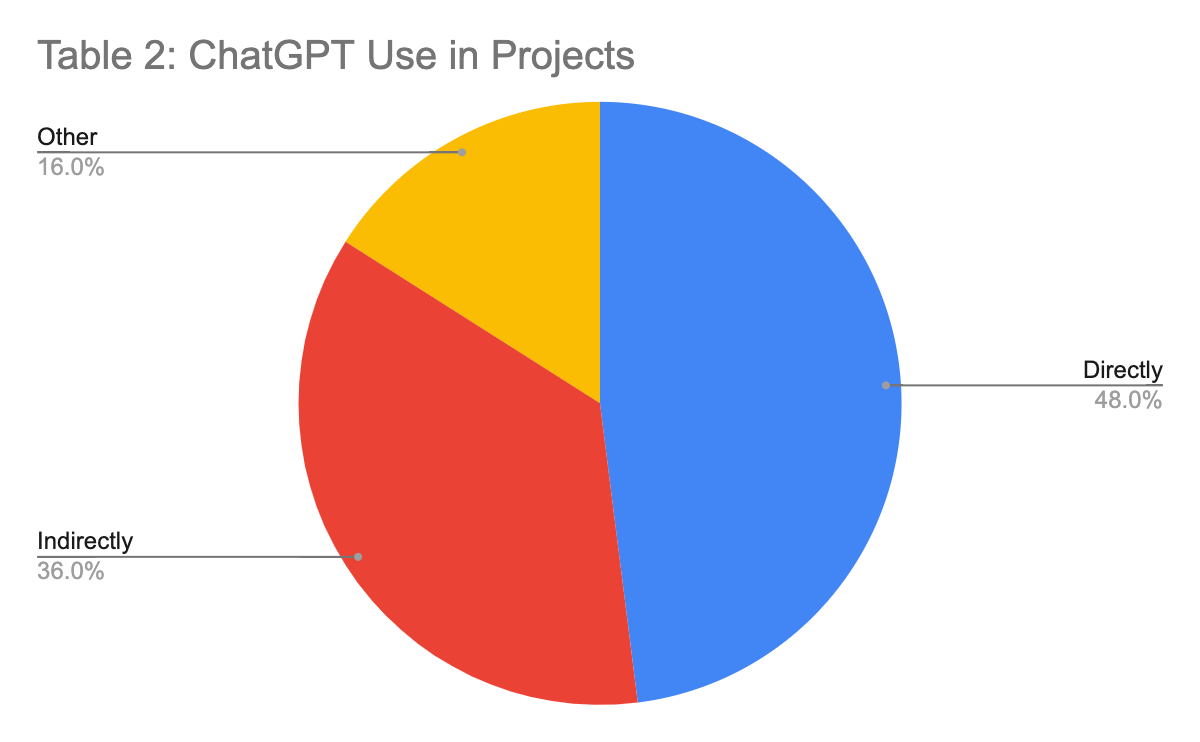
Using a qualitative method, our group selectively chooses five different conversations based on guidelines. We used Excel’s find feature to find conversations based on Python and merged Github files. Using the same feature, we documented repeated versions of the same issue so they were not double-counted during analysis. To gain the most information about each correlation, we read through the chats on GitHub between sharing developers as well as the ChatGPT conversations to conclude why both modes of communication were used. Then, we separated them based on commonalities, and if they were outliers, a separate section was made for them. Later, they were manually filtered in order to find more commonalities.

**Conclusion:**

****

**Figure 1 shows the ways ChatGPT was used by developers and why they may have shared the ChatGPT link to the conversation.**

After analyzing many entries on Git Hub, there are correlations between some entries to form categories where developers help other developers, developers use ChatGPT as examples to allow other developers to understand what is being done and written, and other categories as seen in Table 1. Through ChatPGT, a monotonous process of research is cut out from developing project timelines by using the platform to easily explain concepts to other developers as well as ask for help, advice, and coding excerpts in order to either learn the basis of a data structure or algorithm or have it written for them in correspondence with the project and format they are working with.

**Figure 2 shows how the information from ChatGPT was implemented into the developers’ projects.** 

***RQ #2: Is ChatGPT used as a direct reference to the work being done as developers or as an indirect assistant (conceptual, side tasks)?***

**Method:**

Similar to the method we implemented for Research Question 1, we qualitatively analyzed the same subset of conversations and sorted them into categories (based on what we saw as the most frequent situations) to derive quantitative results as seen in Table 2. We strove to see whether ChatGPT was used by the developers in order to hard-write their code for them or use the AI program for things related to their project such as explaining concepts and asking for advice. Similarly, we did check for duplicates as the same data set was used. Though some of the prompts blurred the lines between indirect and direct, after careful consideration of not only the ChatGPT conversation but also the chats on GitHub, we chose which category the conversation belonged to. One example was when a developer asked [3] ChatGPT if a certain implementation of code was more beneficial than the other. They were asking for advice but used their code directly. In this case, we decided to say this was a direct use.

**Conclusion:** After collecting data to figure out the distribution of how ChatGPT is used mostly, it was found that users implement ChatGPT into their work directly more than indirectly, for example, writing code for them versus giving an explanation of something they give to the AI. There were some entries that did not use ChatGPT at all but the majority of the data do use ChatGPT somehow in the entries on GitHub. This research was very interesting and we learned a lot about AI, ChatGPT, and DevGPT. The use of GitHub is very important for developers to share work and implementing ChatGPT into this is even more beneficial for developers. The creation of DevGPT, the database that was used, was very crucial for this paper to be created.

**Takeaways:**

**RQ#1:** A main takeaway from this question is that users are more likely to use ChatGPT to edit code rather than create new code in their personal programs. Today, ChatGPT is not the best at creating new code from scratch to do functions that a user gives, but it is very good at finding and figuring out errors and problems that are halting the progress of users in their projects; these projects can be personal ones or given to them for work. It is also truly astonishing how ChatGPT can understand code and the outputs like it is a computer that can compute the code that it is giving, but being able to comprehend it and give accurate solutions is very impressive.

A second takeaway found was that the sharing feature encouraged users to get help from other users instead of giving ChatGPT prompts for help with their projects. Before, users were either unable to share their projects and get feedback or had to use inefficient ways to share data like creating a PDF file of code and emailing other known people to help them. Now, they can post their project on a forum and ask for help, and strangers who are passionate about coding can comment on solutions and tips to keep progress moving on the project.

Another takeaway was that developers use ChatGPT to communicate with each other and help understand improvements by using AI. Users can give prompts to ChatGPT and use the output as a way to translate what code does so that other users, developers, and helpers are able to understand what is trying to be done with the given code. Before, users had to figure this out by hand and communicate this to other users who were offering help.

By incorporating the sharing feature into ChatGPT as per software development, it prompts a new fostering sense of interaction in collaboration between developers which allows for an easier-for-use skill exchange and communication among developers. In addition, it enhances developers' ability to access and utilize ChatGPT's capabilities for project development and problem-solving. The sharing feature has been very beneficial for GitHub users and developers, although some negative aspects can be taken away.

After looking at how the sharing feature works, it can be seen that no feature keeps users from arguing and criticizing work just to be harsh to each other. Another can give their opinion and input onto a project that is posted, meaning anything can be said. A fix for this is a community that does not encourage this type of behavior and encourages helpful, constructive criticism rather than harsh and bashing criticism.

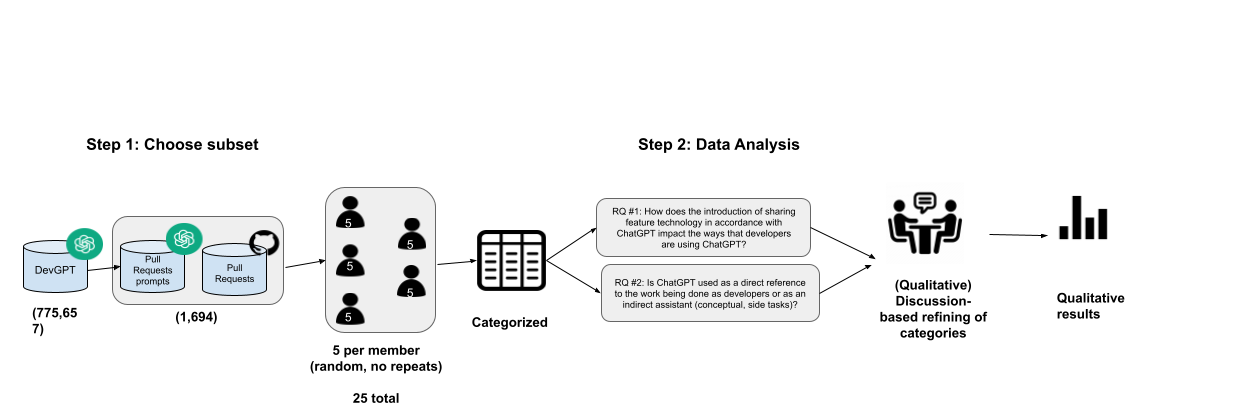
With all the takeaways, the idea and creation of the sharing feature in GitHub is very beneficial for users and guides users to use ChatGPT in responsible ways like using it as a tool rather than using it to create code for them. A community that encourages constructive criticism is needed for the sharing feature to keep being a beneficial tool for GitHub users and developers.

**RQ#2:** The biggest takeaway from the results of the second research question is that users are more likely to use ChatGPT directly in their projects to aid them in their main focus on projects, instead of using it indirectly for a purpose that beats around the bush of their project, for example conceptually in aiding for development of ideas. By utilizing this information found in the research, developers at OpenAI can pay more attention to the direct applicational features of ChatGPT in the way that it develops code and puts more research and training into them as well.

A second takeaway is that most GitHub sources use ChatGPT to produce results either directly or indirectly to benefit developers to improve their code using AI. Those that did not use ChatGPT at all though do benefit the user by using other libraries that are beneficial to the developer.

A third takeaway is a contradiction. Though ChatGPT was used to share examples of code and to explain concepts, most of the examples ended up using the code ChatGPT generated directly in their code. This is because ChatGPT can be used to not only understand a concept better but to learn the concept as a whole, and then apply it directly to the code they are working with. This may be a hint towards learning through application. Once they see the concept actualized in the code, they can then see its usage and better understand the application. Another less optimistic approach is that they may not want to learn the concept and instead have ChatGPT generate the code for them.

An additional takeaway is the way that developers use ChatGPT indirectly in the aid of projects. They often use ChatGPT to show examples of similar questions and/or projects that can be used to address the developer's questions. They also often requested code chunks from ChatGPT for parts of the project that the developers will then read, and take inspiration from. They will not use the code as is, but instead use it to generate new ideas.

**Methodology**

To carry out this research, a group of 5 student researchers carefully selected a subset of the DevGPT data, specifically data from the Pull Request subset. We used not only the conversations between developers and ChatGPT for our analysis but also any messages exchanged on GitHub between other developers to get a full understanding of the motivation of the AI’s use. When each group member initially selected their data to analyze, our research team encountered issues with group mates selecting duplicates of each. As a result, we instructed each group member to pick 5 pieces of data and note down repeats in the dataset so others wouldn’t pick the same one. As a result, there was much more effective communication within the group in delegating our task, and each group member got to more efficiently and accurately focus on their area of study within the scope of the project. We then individually qualitatively analyzed the 5 data and categorized them into different motivations and uses. Afterward, we had a group discussion to further refine the data into either new categories or existing ones to resolve confusion, and conflicts, as well as end up with fewer outlying pieces of data. We did this process for both research questions and ended up with our qualitative data.

**Conclusion**

Throughout the process and scope of our research project, we came to numerous conclusions regarding the impacts of DevGPT and the new sharing feature with ChatGPT, and its consequences on developers. We found that users are more inclined to utilize ChatGPT for code editing rather than manually editing the code themselves. This is because of ChatGPT’s ability to provide feedback and suggest improvements quickly and easily. This is proven by the given examples from the first research question. This problem is that ChatGPT does everything perfectly and prolongs issues. This can lead to suggestions that are unnecessary and do not resolve issues. Therefore, developers should prevent this issue and manually try to resolve their issues themselves. Evaluation is key, especially for code editors. In addition, it was discovered and concluded by our team that users are more likely to use ChatGPT to do all the work. Many people know how ChatGPT can provide effective and relevant code suggestions, to speed up the process of coding and resolve the project quickly. Because ChatGPT can be a powerful tool in the right circumstance, it should only be used then. As we concluded in the scope of the second research question,many code editors will address the main issue with ChatGPT when it is not effective enough to be able to do all of the work, therefore causing issues in the future.

**Future Work**

While our research has given software developers valuable information and takeaways regarding sharing features and code implementation, it also allows developers to pave the way for related studies and projects in the field, incorporating the results from our study. For example, future researchers could explore how developers utilize ChatGPT to develop new skills, understand more complex concepts, and reach new levels of team understanding and connection. By evaluating learning outcomes such as knowledge retention and skill acquisition in addition to the results of our research, future researchers can assess the educational effectiveness of AI-powered tools in helping developers foster new skills and learning. In addition, by further researching the implications of ChatGPT-assisted collaborative development, future researchers could further increase the efficiency of development teams utilizing ChatGPT and change the computer science and IT landscape regarding their approaches to ChatGPT and team development projects.

**References:**

[1] <https://arxiv.org/pdf/2402.11702.pdf>

[2] <https://asgaard.ece.ualberta.ca/papers/Conference/MSR_2024_Grewal_Analyzing_Developer_Use_of_ChatGPT_Generated_Code_in_Open_Source_GitHub_Projects.pdf>

[3]

| <https://github.com/aiplanethub/genai-stack/pull/21> |
| --- |