

Our topic of interest is in the development of a study tool app that could automatically help students with finding the mistakes in their work or help them solve problems, similar to applications such as Photomath. As of right now we have decided to put our focus on the subject of Math, as subjects such as languages are a lot more subjective and are hard to code for.

In this time period we have researched our topic extensively and developed a basic outline as to how we plan on continuing with our project. In simplicity, we're following the path of retrieving information (input) → understanding information → connecting the information to a larger dataset (the math) → providing feedback (output). We're using Python for the, as of right now, majority of our coding. The information retrieval is going to be done using OCR and the language comprehension, which is especially prominent in word problems, is going to be accomplished through NLP. We've spent the past few months learning the basics of Python coding and can code simple programs, and currently we are in the process of learning how to incorporate NLP into Python coding.

The biggest problem that we foresee is the connection between the input (the math problem) and the database (how, for example, algebra is done). From a purely logistical point of thought, we are attempting to think of mathematical knowledge as a stream. For example, typically quadratic functions are during algebra I or II, but this is all built off of solving basic algebraic equations that are taught in earlier grades. Thus, if a student repeatedly shows trouble in solving algebraic portions of the problem, then the algorithm notices that and recommends them more practice problems that are centered around solving algebraic equations. However, if the student is showing a lack in understanding the conceptual parts of the problem, then the algorithm would instead give it a tutorial on the context behind the formula. So, in simplicity, by somehow coding school-taught mathematical knowledge into a stream that has a direction of flow, we can connect the input to the correct "position" within the stream of knowledge, thus giving the correct output feedback.

Although the task sounds daunting enough, we have faith in producing a well-functioning and polished end product. Our next step is to finish learning about the incorporation of NLP and then proceed onto tackling the "connection" step. We hope that we will have that main portion figured out and coded by the deadline, as that is the most crucial part of our application that we are developing.

Referenced information:

1. Researching the biggest problems faced in education (educational gap/inequalities): [The 10 Education Issues Everybody Should Be Talking About \(edweek.org\)](#)
2. W3Schools, used for most of Python learning: [W3Schools Online Web Tutorials](#)
3. Understanding what NLP is: [What is Natural Language Processing? Definition and Examples | Coursera](#)
4. Codebasics, used their tutorial for NLP learning: [YT In-feed - 2A \(youtube.com\)](#)
5. Understanding how apps such as Photomath are coded: [How To Create An App Like Photomath? \(appicsoftwares.com\)](#)