Ever since the invention of Charles Babbage's difference engine in 1822, computers have required a means to perform a specific task. This is known as a programming language. Computer languages were first composed of a series of steps to wire a particular program; these morphed into a series of steps keyed into the computer and then executed. Later these languages acquired advanced features such as logical branching, a feature that changes what question or page a respondent sees next based on how they answer the current question, and object orientation.

In the beginning, Charles Babbage's difference engine, an automatic mechanical calculator designed to tabulate polynomial functions, could only be made to execute tasks by changing the gears which executed the calculations. Thus, the earliest form of a computer language was physical motion. Eventually, physical motion was replaced by electrical signals when the US Government built the ENIAC, or Electronic Numerical Integrator and Computer, the first electronic general-purpose digital computer. It followed many of the same principles of Babbage's engine and hence, could only be "programmed" by presetting switches and rewiring the entire system for each new "program" or calculation. This process proved to be very tedious.

The first computer language for electronic devices was the Short Code, or Binary code, a series of numbers expressed in the base-2 numeral system, which uses only two symbols: "0" and "1." Still, it was the first step towards the complex languages of today. Grace Hopper, an American computer scientist, wrote the first compiler, A-0. A compiler is a program that turns the language's statements into 0's and 1's for the computer to understand. This led to faster programming, as the programmer no longer had to do the work by hand.

In the early 1990's, interactive TV was the technology of the future. Sun Microsystems, an American company that sold computers, computer components, software, and information technology services, decided that interactive TV needed a special, portable language. This language eventually became Java. In 1994, the Java project team changed their focus to the World Wide Web, which was becoming "the cool thing" after interactive TV failed. Netscape, an independent American computer services company, sued Java for using their internet browser, Navigator. At this point, Java became the language of the future and several companies announced applications which would be written in Java, none of which came into use.

Though Java had very lofty goals and is a textbook example of a good language that was easier to learn. It may be the “language that wasn’t”; but the programs written in it ran very slowly. And Sun has hurt Java’s acceptance by engaging in political battles over it with Microsoft. But Java may wind up as the instructional language of tomorrow as it is truly object-oriented, a programming paradigm based on the concept of "objects", which can contain data, in the form of fields, and code, and implements advanced techniques such as true portability of code and garbage collection, an automatic form of memory management.

We could say that it took humanity more than 150 years to create what is commonly known as the Internet. Nowadays, anyone can develop a webpage, especially a 17 year-old kid that does not know anything about programming or coding, like me.

What I have been achieving in my senior project is helping a small community of teachers by creating a simpler and faster way to create a report for their students. With the spare time that each teacher has after creating the transcript, schoolteachers can spend more time on making the world a better place.

It has been a very hard-working month for me and Josh Blaisdell, a software developer and my Subject Matter Expert, or SME. I started my project by understanding the basic concepts of coding, such as HTML and CSS. HTML, or Hypertext Markup Language, is the standard markup language for documents designed to be displayed in a web browser, while CSS, or Cascading Style Sheets, is a style sheet language used for describing the presentation of a document written in a markup language. Later on, I had to start working on Java and PHP, a general-purpose scripting language that is especially suited to web development.

After creating the alpha (an experimental version of a program) of a progress report and evaluation, I started creating a new website that would run the two types of evaluation. I initially thought that the website would be very simple: just a blank sheet of paper where each teacher can choose the student and subject, write whatever they wanted on the pupil evaluation and then simply submit it via email to the Administrative Director. Instead, I had to create a more complex site.

At the beginning of the project, I thought that programming was simply writing what the computer needed to do in a blank sheet. In some ways I was right: to program the computer to do something, you have to write in a special app (like BBedit) in a coded language, passing through different stylesheets using functions. For example, to store data inside a database (an organized collection of data) you start by writing in the controller (a stylesheet that tells the computer what to do, a “save data function”) which passes through a model (a stylesheet that tells the computer where to save the data) and then a view (a stylesheet that retrieves the information from the controller and displays it in the website).

Programming languages have been under development for years and will remain so for many years to come. They got their start with a list of steps to wire a computer to perform a task. These steps eventually found their way into software and began to acquire newer and better features. The first major languages were characterized by the simple fact that they were intended for one purpose and one purpose only, while the languages of today are differentiated by the way they are programmed in, as they can be used for almost any purpose. And perhaps the languages of tomorrow will be more natural with the invention of quantum and biological computers.

I could say that I did the same thing as Charles Babbage: while he started a domino effect that helped humanity achieve the creation of the World Wide Web, I also created a domino effect that could potentially, in a small way, save our planet.