# Unreadable Code

Last quarter, I joined a development team of an almost finished project, and I found it almost impossible to contribute anything because the code is not documented and stacked together. We were developing a cloud storage platform for medical research with Django framework. I was assigned to implement the private access to files and folders function. Catching up was already difficult because I was new to Django, and it has been harder when I saw the code. The code was chunked together with no documentation, and I have to read and understand this kind of code for the views, models, and controllers.

I started with trying to read the code. This can give me a very clear sense of what the code is doing but it is too time consuming. Then I tried to talk to the developers, this can help me catch up faster, but they had already forgot why they did what they did for some essential parts. After that, I decided to take a greedy approach where I started to think of my implementation first and only look for what I need in the code. My ideal implementation is to give all the files and folders an attribute with the accessible id to get private access. Then I specifically look for how they implemented their database and added an attribute to it and changed the views according to it. This largely decreases the lines of code I need to understand. However, this method is useful only because I only need to implement this one function. If there is more, then this method is going to result in more bugs.

There are some auto comment tools like “I Programmer” but these tools are only generating a template for you to write on during the developing phase, not helping you to read afterwards, and you still need to fill in the contents on your own. There is no such tools exists that helps you read a chunk of uncommented code, which is why we are all told to document our codes when we are writing them.

If there is a tool that solves this problem, it would take in the uncommented chunk of code and separate them into parts and add descriptions of the functionalities to each parts. This can be possibly done by machine learning techniques or even simply by separating by condition operators like if else statements or for loops. The contents can be predicted by the code, also by machine learning, and the content do not need to be very specific, an abstract helps a lot already. The challenge is to design such a model for machine learning which is super complicated.