The issue here is the possibility that an LLM produces the wrong answer and the effects that this could have on the user. One example of this would be if a software engineer was using ChatGPT to help them code, but they receive a piece of code that is incorrect. They then use this code and their software crashes, leading to losses of time, effort, and/or money for their team and stakeholders. There are many types of these examples, where a person can really mess something up if they blindly trust an LLM output. In a way, LLMs are “human” because, like humans, they are not perfect and will make mistakes at times.

As far as the LLM developer’s responsibilities, I feel like ChatGPT’s disclaimer is enough to address this issue. As a user, it is your responsibility to ensure that what you take from an LLM is accurate or correct. Similarly to if you are using Google, the answer you get will likely be true, but there is also a very real possibility that it is not. LLMs should be used as suggestions instead of being blindly trusted because of this possibility of error. It is the user who should be accountable when it comes to what they actually take away from the LLM, as they should at least make an effort to try to understand the LLM outputs and distinguish correct from incorrect.

As a developer, I would take the same approach as ChatGPT’s developers and limit what I do to just a disclaimer. I feel like I would have done my part in that scenario by providing a warning, and it is up to the user as to whether they heed it or not. Of course, as a developer, one of my main goals would be to improve the accuracy of the LLM and hope that one day, the disclaimer would not be necessary. But until that point, I can’t really think of a way to protect user’s while still allowing them to utilize the LLM as much as possible, so I would keep my efforts to just a disclaimer about the model’s accuracy.