If presented with a problem or challenge, how confident are you that this course has equipped you with the skills to appropriately respond? If a particular topic has not been covered, to what extent has this course equipped you with the ability to expand your knowledge by learning on your own? Justify your rationale.

Professor and class,

I feel that this course has adequately equipped me for overcoming challenges, but not just from the programming experience presented in this course. Let me explain. Before this course, I would freeze up at the thought of asking for help and even considered the very idea something that would earn shame in a software development job. I had struggled in this course and kept getting behind on work. I don’t want to get too detailed but I’ve just got a lot on my plate lately so small bumps in the road like having problems with the assignment can easily lead to me having to turn in work late and over time it just gets worse as more programming issues pile up. This exact problem led me to the sink or swim decision of either just allow myself to fail this course or swallow my pride and ask for help. I decided to seek help with some things I struggled with and not only did it help me catch up on all the work I was behind on, it also gave me a much greater interest in the topics as well. I’ve also realized that asking for help is likely a real part of programming jobs- especially for new hires. After all, I’m sure nobody expects us to know everything about programming. Languages and software change so quickly that it’s nearly impossible to keep up with everything.

As for the actual course work, I feel it’s helped me gain a good understanding of JavaScript frameworks and how they are used to speed up the development process of projects on the web. I did learn an awful lot about routers and routes and found the data flow between database, API, and app to be enlightening. It motivates me to experiment with JavaScript frameworks and create different personal projects in my spare time- well, whenever I will have spare time at least.

Professor,

I see you’re also a Nikola Tesla fan. His work with the Tesla Tower very nearly caused wifi to be invented long before Ethernet so he was almost the true inventor of the modern internet as well. Something to note is that he didn’t actually become well known as an inventor until after his death. It’s a little grim, but it’s also motivating in the sense that our hard work may one day be recognized even if it’s not in our own lifetimes. His rivalry against Edison is a great example of how the most popular technology isn’t always the best.

What I’m trying to say is that maybe we shouldn’t feel pressured to go with the popular technology and that we as developers should go with our instincts. Java having a sudden resurgence after starting to die off shows that we can’t predict the future of technology as well as we think we can. Learning and using popular software can help us get jobs, but trusting our instincts can help us stand out as competent and skilled programmers.

Scott,

I can already see that self-learning will be a necessity even after we graduate. Programming languages and frameworks change every day so a lot of what we learn in these courses may be depreciated by the time we get jobs which may make these courses seem pointless on a surface-level basis. However, we aren’t just learning the languages and frameworks in these courses, we’re also learning how to research, troubleshoot, and think outside the box when we reach challenges. We’re training ourselves to move mountains instead of giving up when things seem impossible. It seems that programming not only requires a lot of concentration but also a positive mindset and no short amount of confidence in our own abilities.