

AI Tutor

Overview

Spencer Thompson

December 03, 2025

Table of Contents

I The Beginning	3
An Idea	3
AI AI AI	3
Natural Evolution	3
II The Journey	4
Burn it to the Ground	4
Roadblocks	4
Scrappy Solutions	4
Oh hi canvas	5
III The Existential Crisis	5
Suffering from Success	5
It's Concious?	5
Unforeseen Consequences	6
Don't you know AI will replace you?	6
IV The End	6
Success?	6
Lessons Learned	7
An Exciting Future	7

I | The Beginning

The AI Tutor project started out of conversations shared with [Dr. Ahmed Alsharif](#) and [Dr. Armen Ilikchyan](#) regarding the *realistic* capabilities of a new and exciting technology, AI. This was ChatGPT, released by OpenAI right at the end of 2022, quickly taking the world by storm.

An Idea

Rather quickly after the release of ChatGPT, I started experimenting with the APIs, *code that talks to other code*, used by ChatGPT simply out of sheer curiosity. In the process of my experimentation, I was rather quickly starting to realize the power and capability that could be harnessed with the OpenAI API and just a small amount of code.

Given the novelty of this new artificially intelligent technology, there were those who didn't know it existed, those who didn't believe in it, and those who had tried it and been fascinated. I was enthralled by this new technology along with Dr. Alsharif and Dr. Ilikchyan here at UVU. Spurred by a shared interest, we promptly began discussing how we had begun using it in our day to day work. We shared our perspectives, fears, and excitements.

It didn't take long before we came up with the idea to use this new and exciting technology to build a custom AI assistant for students at UVU.

AI AI AI

ChatGPT was **exploding** in popularity, it was for *better or worse*, being used more and more frequently by students. While this was happening, I was beginning to realize that using these APIs with some custom code, I could get much more precise and controlled outputs out of these AI systems. Using the API, and creating custom "AIs" seemed to provide overall better quality of responses.

The AI Tutor project started right around the beginning of 2024. The Technology Management Department at UVU was interested in learning more about AI, and helped to put together a team to work on this project. Without the Tech Management Department this project would not exist.

Natural Evolution

Our team very quickly created a rough prototype, that showed promising success almost immediately. The initial prototype was a relatively simple tutor that was built for only one class, TECH 1010 Understanding Technology.

With this rough prototype, it was surprisingly quickly inserted into several courses within Canvas at UVU. Just as quickly as it made its way into classes, new ideas for features or capabilities started popping up. This was both daunting and exciting at the same time.

Before we knew it, this rough prototype of the AI Tutor:

- Knew the course syllabus.
- Knew about upcoming assignments.
- Had a unique UVU identity.
- **Helped** students learn **instead** of just answering their questions.

II | The Journey

After this first rough prototype was created, we had some students leave the team and other students join. There was a problem though, given how fast this first iteration of the AI Tutor was created, there were some clear problems. It did indeed work, and it worked well, although it was confined to one specific course.

Burn it to the Ground

As the Tutor began to grow in complexity, we had accumulated some unpleasant technical debt in the code due to the rapid tempo of development. I realized that part of the process of development is learning how it **should** be built only happens after it has been built.

I knew I wanted to add more features with a better design, not to mention many students and other professors (even of other departments) wanted new features. So, *without telling anyone*, I re-wrote all the code for the AI Tutor in about a week. This eliminated the technical debt, and provided a much more solid foundation to build a new version of the tutor.

Roadblocks

One core feature that we wanted for the AI Tutor was the ability to interact with Canvas. This would give the AI Tutor the ability to provide custom fine-tuned tutoring for **any** UVU student in **any** course. In order to do this, we needed access to the Canvas API.

This API, of course, is protected by several layers of security. In order to use it, we would need a valid API key which could only be provided by the digital transformation department at UVU. We spent an incredible amount of time and effort working with them in an attempt to get a developer API key. Despite this, they declined our requests for two reasons. First that we were students, but perhaps more significantly that they already had a team (with UVU students) working on their own “AI Tutor”.

Scrappy Solutions

This was very disheartening and frustrating. One thing about me is that I am stubborn, and given that I felt dismissed, I was determined to find a solution.

I thought deeply about this problem and spent time digging around the internet for ideas. I am not sure how or when I came up with this idea, but I realized that a browser extension would allow us the ability to communicate with Canvas without needing an API Key. Perhaps you could call this a security vulnerability, or simply just a web browser, but sure enough, when I created our browser extension, I was able to communicate with the canvas API.

Not only did this solve the *unsolvable* problem, but it also provided us with bulletproof security without having to do much ourselves. If the browser extension was working, that meant that the student **was** logged into Canvas.

This was one of several very clever solutions that I came up with to solve very tricky problems encountered during development of the AI Tutor. Without these solutions, the AI Tutor would certainly not exist as it does today.

Oh hi canvas

Another issue that continues to haunt me to this day is the inconsistencies and *interesting* design of the Canvas API. On more than one occasion, we had severe bugs due to incorrect documentation or strangely designed database schema inside Canvas.

One rather notorious issue is that every course by default has a `course_name`. At the time, I figured that did not change, because none of my classes had any other identifying data coming back from the API. But sure enough, if a student changes their course's name inside canvas, it changes the `course_name` and a new field appears, `course_code`, with the original course name.

Dealing with the Canvas API really opened my eyes to the fact that most corporate software almost certainly has crippling amounts technical debt.

III | The Existential Crisis

In the process of developing the AI Tutor, I had several significant realizations. Some obvious, some not so obvious. Some of them even caused me existential dread about my career, existing software or even just my future as it relates to technology.

Suffering from Success

Something that became apparent rather quickly was that students and teachers both really liked using the AI Tutor. The first time we checked how many users we had, expecting something like 10 or 20, we were surprised to see we had **more than 300**. This was exciting and a huge success, but it came with some interesting downsides.

I quickly realized that if there were bugs in this code, or if I made a mistake, it would impact **several hundreds** of students. Further, I couldn't just re-write the whole thing again so easily. We now had a database, and this was a foundation of the AI Tutor project that could not be so easily changed without disrupting students use of the AI Tutor.

It's Conscious?

Perhaps the most frightening aspect of developing this project was and still is, the code that lets the AI make decisions. To explain this simply, essentially in the code, The AI Tutor has the ability to execute its own code, make decisions for itself in the future, and plan for itself how to talk and think, **with each response**.

I oftentimes think that being scared of AI "taking over" is foolish, and while I still feel the same, I just can't help but reflect on this code. Watching the AI plan, talk to itself and use tools is just uncanny. It

almost feels as though there is some argument to be made that the AI Tutor is **conscious** even if only for a moment.

Unforeseen Consequences

During my time at UVU I took a couple courses from Craig Bell. He is the Database Administrator over at Adobe, and teaches here at UVU as an adjunct from time to time. During one of his classes he mentioned something along the lines of, "if you write it, you own it".

At the time, I didn't think much of that statement, although I have since realized that it is very true. Ultimately, I am responsible for the code I write and the users experience. As the team for this project has changed multiple times, I have realized that new developers need me to explain how things work. If I don't document the code well, I have to be the one to fix it when it breaks.

This might seem obvious, although it has come back to haunt me, and taught me some truly valuable lessons.

Don't you know AI will replace you?

Towards the end of this project, I have heard more and more talk along the lines of, "AI will replace all software engineers". I have spent my time worrying, thinking, and ruminating, although that time is over. I am no longer worried.

I have used AI myself on other projects, and it is a big part of my workflow. But generally, I don't like to just copy and paste AI generated code, I often tell people rather jokingly, "I like to put the bugs in myself".

Jokes aside though, even on this project, there have been instances where a team member has used AI without really checking or thinking, just *vibe coding*. This in itself has proven that AI will not be taking my job. I have fixed countless severe bugs introduced by AI, and I am confident that AI will not be taking away true software developers.

IV | The End

This project has taught me so much, not only about writing code, but also how to effectively work in team, how to communicate technical concepts to engineers and managers. I have learned the hard way why we are taught to do some of the things we do, and I have learned that sometimes raw ingenuity can solve seemingly *impossible* problems.

Success?

The big question, was creating the AI Tutor, and working on it for nearly two years a waste of time? I think the answer is most definitely yes.

At the time of writing this we have 660 lifetime users. Not only this, we track the analytics of our users. Students and teachers alike use the AI Tutor **multiple times a day, every day**.

Those who use the AI Tutor truly like it, and they really think that it helps them to **actually learn** instead of just getting their questions answered in a robotic and impersonal way.

Lessons Learned

I have learned so much from this project. To sum up the most important lessons I have learned during the development of the AI Tutor it would be these:

- Code in production with active users is entirely different than code that isn't. Planning for and knowing this ahead of time saves countless headaches.
- Schemas, or the way you store and organize data in a database, is incredibly important. No matter how much you plan or hard you try, it will inevitably cause issues.
- Documentation is really important, perhaps even more important than the code itself. Without proper documentation, teamwork is a hindrance and maintenance is impossible.
- Communication is never enough, particularly with technical projects and endeavors. Communicating effectively is the most difficult part of good technology.
- Determination can solve impossible problems.

An Exciting Future

In the end, this project has been the most edifying and valuable pursuit of my entire bachelors degree. It is not often that a coding project has users, or sees the kind of success that we have been lucky enough to find.

The AI Tutor as my capstone is the best possible way to wrap up my time at UVU. I leave feeling like I have provided something valuable, learned invaluable lessons and accomplished something truly noteworthy.

I cannot wait for the next chapter.