AI Tutor

Technical Report

Spencer Thompson

Table of Contents

1. Introduction

This project, the AI Tutor, is a project that has now been under development for quite some time. We have seen some exciting success regarding utilizing artificial intelligence in real world application.

This report outlines Why, How, and What this project is all about, with a focus on the technical design, implementation and solution.

1.1. Background

As previously mentioned, the AI Tutor project started back in early 2024. In discussions with the excellent faculty in Tech Management Department at Utah Valley University, we hypothesized that using the new and exciting technology of large language models, we could provide excellent, *personalized* tutoring to students 24/7.

So we set out to develop a simple application to accomplish this goal. The original AI Tutor was indeed a success. Quickly the project gained attention among multiple departments and more than a handful of students. Although, given the rather breakneck pace of development, there was a rather rapid accumulation of technical debt. Some of the core features that we wanted had become quite difficult.

After the original scope of the project had been completed, we still had a desire to have a system that was better suited for both students and professors. As the lead developer, I had the feeling that a fresh start might be better than attempting to pay down a significant amount of technical debt. Therefore, this report is particularly focused on:

- The evolution of the project as a whole.
- The features that gives the AI Tutor an advantage over **every other alternative**.
- The design and implementation of the tutor, as well as the challenges faced along the way.

1.1.1. Context

A rather interesting piece of context to keep in mind while reading this report is that: during the duration of this project, every single piece of technology has changed **drastically**. Oftentimes the APIs or services that we were utilizing evolved or changed overnight. This could possibly be attributed to the incredible amount of development and hype around the use of generative AI.

The point being, many other competitors both at our own university and others, were quickly building and iterating on similar ideas. Our team built and deployed **two** complete iterations, while other teams have yet to deploy their projects.

1.1.2. Purpose

From the beginning our project was focused on providing a rather niche ideal. We all had this idea of a tutor or assistant that had intimate knowledge of a student's courses. This would give the tutor the ability to provide:

- 24/7 access to students to assist with coursework.
- Personalized responses to student questions that were unique to *each* student.
- Answer questions about the syllabus, upcoming assignments, grades and more.

The hope and idea being that, this could be an incredibly valuable resource for students, faculty and the university as a whole.

1.2. Scope

1.2.1. Goals

1.2.2. Limitations

- Other opportunities
- A very difficult semester

2. Design

- Browser Extension

2.1. Data

2.1.1. Analytics

2.1.2. Pipeline

2.1.3. Storage

2.2. Security

- JWT
- Keys
- SSH
- 2FA?

2.3. AI Behavior

- Class specific behavior / answers

3. Implementation

- Lessons Learned from first iteration

3.1. Infrastructure

- Hetzner VPS
- Cost reduction
- Docker & Compose

3.2. Frontend

- User Interface
- Prioritize good User Experience

3.2.1. Browser Extension

- Side panel
- Cookies

3.2.2. Chat Interface

- Streamlit

3.3. Backend

- FastAPI

3.3.1. AI

- OpenAI API

3.3.2. Tools

- Catalog
- Upcoming Assignments
- Grades

3.3.3. Telemetry

- Plausible Analytics

3.4. Database

- MongoDB
- Document Based

4. Challenges

4.1. Permissions

- UVU
- Competing Team

4.2. API Inconsistency

4.2.1. Canvas LMS

- Name vs Course Code

4.2.2. Plausible Analytics

- Weirdest API I have ever seen
- Cartesian products galore

4.3. Messy Data

- Canvas
- My own Design

4.3.1. Live Service

- Developing / Adding features for an application in use is difficult

5. Solutions

5.1. Workarounds

- Browser Extension

5.2. Clever Tricks

5.2.1. Authentication

- JWT / Cookie

6. Conclusion

6.1. Findings

6.2. Implications