

Courtroom Clash: How We Built an AI Lawsuit App

1. What This App Is All About

This report explains how we built "Courtroom Clash," a simple app that makes two AI lawyers argue a case. We wanted to see if we could make one of the AI lawyers smarter by giving it a list of facts to work with. The whole app is a good example of how to make an AI use real information to create a better answer.

2. How Our App is Built

We built our app using three different files. Think of it like building a car: each part has a different job. This makes our code neat and easy to fix.

- **frontend.py:** This file is just the screen you see. It creates all the buttons, the text boxes, and the different sections for the lawyers. It doesn't do any of the thinking—it just shows what the other files tell it to.
- **backend.py:** This is the brain of our app. It does all the hard work. It's the file that talks to the Gemini AI to get the arguments for the lawyers. It also has our list of legal cases.
- **app.py :** This is the main file that runs everything. It connects the frontend to the backend. When you click a button, this file sees that you clicked and tells the backend to start working. It also remembers things like the case scenario and the lawyers' arguments.

3. How the "Smart" Lawyer Finds Its Facts

We wanted one of our lawyers, the RAG Lawyer, to be super prepared. To do this, we gave it a small, fake list of legal cases to use as a database.

- **Our Fake Database:** We have a list of cases, and each case has a title, the story, and some extra information (metadata). This extra information is really important! It tells us things like the **type of case** (like "Negligence") and the main **legal principles** (like "Duty of Care").
- **Finding the Right Case:** When a new case is announced, our app looks through the database to find the best match. It does this in two ways:
 1. **It looks for matching words.** For example, if the case is about a "wet floor," the app will look for cases that have the words "wet" or "floor" in their story.
 2. **It looks at the extra info.** If it can't find a good word match, it will look at the metadata. For example, if the case is about a company being careless, the app will look for cases with the case_type "Negligence."

This way, the RAG Lawyer can always find a relevant fact to use in its argument, making it seem much smarter than the Chaos Lawyer.