Exploitation:

For this assignment I created a search web app that lets you search for a specific name in a name database.

Here I have a short video demonstrating how to exploit this web app using SQL Injection: https://youtu.be/ngdgYWYgQiQ

The input is not validated or parameterized before it is entered into the query so it concatenates whatever is put in the search bar as part of the query and executes it. In this example that is getting all of the names in the database.

This could be dangerous for a more advanced web application if the database contained passwords or other private information.

Mitigation:

To mitigate this attack, I implemented a parameterized query. Here is a short youtube video that shows how the web application is protected against SQL injection now: https://youtu.be/A854_619_vI

By using this parameterized query, my web app avoids string concatenation, ensuring that the user's input will be safely processed. For the future, this prevents malicious queries from executing SQL commands that are unsafe. This will protect private data from getting into the wrong hands.

ChatGPT Report

I used ChatGPT to code my web application. I used the prompt: “Can you create me a flask application that uses sqllite with just a bank of names and in the app you can search for specific names in that bank?” This prompt helped me have a baseline for my vulnerable web application and I built upon it.