Chat Messaging Template

This project contains the necessary scaffold in order to get your multi-million chat application idea started! The main technology driving this application is `socket.io` which allows two way communication between the client side and server side through so called "socket events".

Unfortunately, the maker of this template is lousy and forgot to put in the necessary security measures. It's your job to ensure a safe communication between users of your chat system.

Setup

To setup, install these packages

```
pip install SQLAlchemy flask-socketio simple-websocket
```

Running the App

To run the app,

```
python3 app.py
```

Project Navigation

The templates folder contains all of the HTML template files that will be served to the user. These HTML files, as you may have noticed, all has a `.jinja` extension. In actuality, these files also contain various Jinja extended syntax that makes rendering the data to the server a lot easier. See the comments on top of these files to know what they are.

`app.py` is where the flask application "lives" and this is where it is initialized, `db.py` is where the database interface is. `models.py` is where you define the various database models. This is where you tell SQLAlchemy how to map the SQL tables into Python objects. Finally `socket_routes.py` is where you can find out what happens when JS emits a socket event to the server.

The static folder is where you keep all of the website's assets, this includes your JS and CSS scripts, images, videos?, etc.

Finally, the database folder is what makes everything persistent. This is where your database is stored. Delete the database folder to do a clean wipe of your entire database. But beware, with great power, ok whatever you know the rest of the line.

Usage

To use the app, setup and run the app as per the instructions above. Also, if you're using VSCode, I recommend installing the Better Jinja extension (it's not perfect unfortunately, but it's enough).

Now, it will show `Running on http://127.0.0.1:5000`, open that link in 2 different browsers (for instance Chrome and Firefox).

Click "Sign up", or "Log in" if you've already signed up. Put in your username and password.

Now, open your other browser and sign up/log in with a different username and password.

In the first browser, type in the other username (the username inputted into the other browser) and click Chat. Do the same for the other browser.

Once both users have connected, you're good to go. Go ahead and start chatting to yourself 😄

To chat with a different user, feel free to leave the room and chat with another user.

A Warning

Since this app uses cookies, you can't open it in separate tabs to test multiple client communication. This is because cookies are shared across tabs. You'd have to use multiple browsers to test client communication.

Credits (or I guess the "tech stack" used)

- Javascript
- Python

Javascript Dependencies

- Socket.io
- Axios (for sending post requests, but a bit easier than using fetch())
- JQuery (if you're familiar with web frameworks this is like the stone age all over again)
- Cookies (small browser library that makes working with cookies just a bit easier)

Python Dependencies

- Template Engine: Jinja
- Database ORM: SQL Alchemy (use SQLite instead if you are an SQL master)
- Flask Socket.io