

## Achievements

In this project we demonstrated an understanding of safe multithreading practices through our use of mutex locks. A second achievement is that we implemented persistent browsing that works between server restarts and client reconnections through the use of saved session information and client-side cookies. We also were able to collaborate using GitHub and achieved a good workflow using branches and pull requests to safely merge all code smoothly. We also were able to pay attention to detail with bug fixing, including the invalid input handling listed in the assignment details, but also unmentioned issues such as server freezing on a sudden client disconnect.

## Questions (Part 2)

1. *Why do we need locks for `browser_List` and `session_List` in `browser_handler()`? What could happen if there is no lock?*

If the index into either list shifts or the state of a session/browser changes during the execution of this function, it could result in unexpected or undefined behavior.

2. *Can we use multiprocessing instead of multithreading here? What are the differences between multiprocessing and multithreading?*

We can; we could, for instance, spawn an entirely separate server for each client. However, this would be less efficient since we would be creating an entirely new copy of the program versus just "extending" one program.

## Contributions

Joseph Judge: Part 2, Part 3.1

Kristen Goebel: Part 1, Git setup

Stephen Miner: Part 3.1, Part 3.2