Please answer the following questions for the project report.

1. Team details: Clearly state the names and netids of your team members (usually there are 2 of you).

Richard Li - rl902
Wesley Zhou - wgz4

2. Collaboration: Who did you collaborate with on this project? What resources and references did you consult? Please also specify on what aspect of the project you collaborated or consulted.

Beyond the scope of collaboration between us two, we heavily referenced the TA's lecture slides on the intricacies of each exit case, and some further research into the python socket library and the documentation for the curl command. Richard handled the majority of the main feedback loop while Wesley did the cookie parsing and dictionary creation for authentication and secret maps.

3. Is there any portion of your code that does not work as required in the description above? Please explain.

Through our testing (primarily through curl commands), all of our code works as described from the project description. Our code also works with multiple machines through our optional command line reading. When running the server on the first machine, please enter along with the port number the name of the machine that the server will run on like so:

python3 server.py 45006 ilab2.cs.rutgers.edu

Then when formatting a curl command from another machine, you can format it like this:

curl -v http://ilab2.cs.rutgers.edu:45006

4. Did you encounter any difficulties? If so, explain.

Yes, I encountered several challenges throughout the project. One major difficulty was managing connection issues when running the server on a remote virtual machine and accessing it from my local browser. I spent way too much time trying to figure out how to get it to work (before even starting the bulk of the project mind you) before I realized that I totally didn't have to do any of that and dived into learning about the curl command to test our code locally within the machine to finish the project. Additionally, we faced hurdles with cookie management—stale cookies from previous sessions would interfere with the authentication flow. This required careful handling of cookie creation, validation, and expiration (particularly for logout actions) so that an invalid session token would not mistakenly override valid login attempts.

5. Please discuss two observations or facts you learned about HTTP and cookies in the process of working on this project. Please be specific and technical in your response.

One of the key observations was that HTTP, by its very nature, is stateless, meaning each request from a client is independent of others. To maintain session state (like user login sessions), the server and client must use mechanisms such as cookies. In this project, when a user successfully logs in, the server generates a random 64-bit token and sends it back using a Set-Cookie header. This token allows subsequent requests to be associated with the logged-in user without requiring re-authentication, effectively allowing the server to “remember” the user across multiple requests.

Another important fact learned is how cookies are managed throughout their lifecycle. Once set by the server, cookies are automatically sent by the browser in subsequent requests via the Cookie header. In our implementation, upon successful authentication, the server sets a cookie that is stored on the client-side. The server checks this cookie in every request to validate the session. Also, the method of invalidating the cookie—by sending a Set-Cookie header with an expiration date in the past (e.g., expires=Thu, 01 Jan 1970 00:00:00 GMT)—was crucial in implementing the logout functionality. This ensures that once a user logs out, the stale or invalid cookie is removed from the browser, preventing unauthorized access using an expired session token.