1. Christine Yue 207000676, Zahir Musliwala 209007164
2. Each other, Project03 Slides, and Piazza for more guidance and clarity to project instructions. ChatGPT to troubleshoot test case and other difficulties (further explain in question 4)
3. No. However, we did leave print statements to help guide users/TAs to see which case (ex, Case A,B,C,D,E) they are accessing and a "---" line to help separate Header/Entity and Response outputs in the terminal.
4. We did encounter a couple of difficulties:
   a. Originally, we tested the code on our local machines but realized Mac is not compatible to run server.py unless we hardcoded http://localhost:%s instead of http://:%s. We moved our development to ilab.
   b. We did not understand how to test the test cases provided in the instructions.pdf. More specifically number 6, 7, 8, 9, 10, and 11. These test cases required us to use "Inspect Element" in FireFox (Developer Tools). ChatGPT provided instructions on how to add cookies to FireFox (change name to 'token', and value to 'garbage' for invalid cookies and the actual cookie value for valid cookies).
5.
   a. We learned how cookies are used to create stateful sessions in HTTP that are stateless. By generating a random 64-bit token (*random.getrandbits(64)*) after successful authentication, storing it in a *sessions* dictionary mapped to the username, and sending it as a "Set-Cookie" header, subsequent requests that include this cookie, allow the server to "remember" the user and its respective password and secret, via the token number (Case C). For logout (Case E), the server expires the cookie by setting its expiration to *Thu, 01 Jan 1970 00:00:00 GMT*, forcing the client to delete it so that each new login after the first successful one has no cookies (unless manually added through Developer Tools). Implementing these allowed us to learn how exactly cookies are used and managed.
   b. We also learned that cookies are strictly bound to specific domains, and mismatched domains break authentication flows. Initially hardcoding *localhost* in *form actions* worked locally but would fail other machines (like ilab servers) because browsers only send cookies to the exact dominant they were set for. Proper domain handling (using a dynamic hostname) ensures cookies work across testing environments while maintaining security boundaries.