

1. Team details: Clearly state the names and netids of your team members (there are 2 of you).
 - a. Srinidhi Ayalasomayajula (sa1783), Palak Mehta (pm862)
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.
 - a. We collaborated with each other on the entirety of this project. There were no outside websites that we consulted in reference to our project because this project was more straightforward than the previous. In order to understand the project, we did print out the variables that were holding content. For example, we printed the contents of the body and the header to find out what they contain and saw that the user input was a part of the body message. Since we had to parse data from the body, we were able to identify the information we needed easily by splicing the string message. However, we did get a lot of assistance and guidance from the TA's. The concept of clearing cookies from the browser by using the settings or using curl commands to test was new to us, so with the TAs help, we were able to effectively test our project. We also asked them a lot of questions to clarify the test cases and the expected output, as well as questions about the general way to approach certain questions.
3. Is there any portion of your code that does not work as required in the description above? Please explain.
 - a. All our code works correctly.
4. Did you encounter any difficulties? If so, explain.
 - a. We encountered several difficulties in our project. After building and testing our code, we came across a problem in storing and removing cookies from the established database for when users login and log out twice. We were ultimately able to resolve the issue by deeming that under any other circumstance when the cookie has been stored, it will be stored indefinitely. The only way for it to be removed from the database is if a person explicitly logs out (thus, the cookie times out by itself). This means that the test case we were envisioning of an account logging in twice and then afterwards logging out twice is not possible. Another difficulty we faced was while logging into an account on a given browser, then logging on through another browser with no user name and no password. In reality, it should show the basic login page, however, based on our code, it was simply logging in in the second opened browser. To combat this, we added an extra if statement condition to check to see if there were cookies enabled and the login was blank. If it was, then the login page was called. On top of this, another issue we were facing is that after logging in once, and then opening another browser, we were immediately shown the success page without interacting with the html page at all. Since we knew we needed to test incorrect

username/password pairings for valid cookies, we decided to change the logic for a valid cookie. Instead of directly showing the success page, we first called the login page for the new browser display. Then since the cookie in the header and the cookie token stored in the database (dictionary) matched, we then showed the success page. Being able to work with real time user data was another new task. Another challenge with this was being able to handle different types of requests, such as blank username and passwords as well as invalid credentials. A majority of the difficulties we encountered were because there are many edges in this project that we wanted to ensure we were completing correctly. By testing our project from the start as we were building the logic and design, we were able to find issues quickly and fix them sooner rather than later.

5. Describe 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.
 - a. One thing that we learned about HTTP is what type of information is stored in the header and body of each message request and how each component plays a part in a task, for logging in a user, for this project, for example. As for understanding the HTTP Get and Post requests, we observed the output in the terminal which gave us a better idea of the difference. Only the Post request contains the contents of the username and password in the body and also details of the contents in the Post header fields. Whereas, the Get Request has an empty body and no header fields for details about the content. It was engaging to be able to parse the HTTP POST request to recover the username and password, which is fed by the user on the browser. The username and password is in the body of the POST request and not the GET request since the POST request ensures for more security and allows the information to be parsed as sub-content to the host url. Seeing this in action helped solidify our understanding of the different messages that the Professor mentioned during lectures. Another thing that we learned is that uncovering the cookie token for cookie-based authentication was also new to work with and important to see because it made the login process a lot more efficient and reduced latency. By being able to compare the Cookie header in the POST request to the already existing cookies in the database, we were able to validate and login the request immediately. Since the cookie is going to be stored in the POST request's header information until it expires, we can repeatedly log in until we log out (which would remove the stored cookie). Another observation made was the power/importance of cookies and how valid cookies essentially ensure success for a login. Once the cookie has been validated, even if the username or password is empty fields or there is a non existing username/password combination, the login page should still lead to a successful login. This eliminates the need to input the correct username and password repeatedly on the same browser where we previously logged in. This was initially surprising, but very cool since it shows

the practical scenario of cookies in the real world. Having access to cookies for a user's activity on a browser can provide a lot of data on the preferences of the user. In the case of this project, once a user is logged in, their information, such as the secret data is stored. In the case of the real world, once a user clicks around on a browser, their activity can be stored and can be analyzed to understand what the user deems important. This concept can assist with targeted ads and marketing techniques. Overall, this project gave a really good glimpse into how HTTP and Cookies work with each other to make a user's experience on a website better.