2. Initially the history showed that I had never visited the website before. After refreshing the page a few times, I was able to see the date and time of the last visit, as well as the number of times I had visited.

3. By using the developer tools and the information in step 1, I was able to see that the cookies for this page were storing the last visit date, as well as the number of visits that I had made to the website.

4. After deleting the cookie association, the counters were both reset for number of visits and last visit. As a result, reloading the page showed that I had never visited the website before in the history section.

5. By examining my recently saved cookies, I was able to see multiple cookies stored from different websites such as GitHub, blackboard, Stack Overflow, google and YouTube. This makes sense because for the past week I have been working hard on trying to get our website working for our project, and all of these websites have been constantly visited for troubleshooting my code.

6. Moving the code makes it so that even though the cookies are updating the information, as can be seen by the updating information in the cookies from the dev console, the information is not being displayed in the website history anymore, since the cookie runs after the code for the website display.

8. Clicking the link without putting in any password takes us to session-cookies-3.php and shows an error that states that there was an authorization failure.

9. Clicking the link after putting in a wrong password takes us to session-cookies-2.php and shows the same error as the previous page that states that there was an authorization failure.

10. Upon looking at the visit counter, we can see that the counter increases even if the wrong password is given.

11. For the session, the settings for the variables are as follows:

Expires: Session

HttpOnly: False

Secure: False

SameSite: None

HostOnly: True

12. After supplying the correct password, I was able to successfully reach the link.

13. After examining the code, it seems the session-cookies-2.php is used to handle the login functionality. Clicking the link directly in session-cookies-1.php takes us to session-cookies-3.php, however this will not load if the session has not been authorized by authorizing the session cookie in session-cookies-2.php, which can only be done by a correct login.

When we enter the wrong password in the login field, the password cannot be authorized, and the session cookie is not updated. However, as soon as we enter the preset password ‘Test’, the session cookie is updated and authorized by session-cookies-2.php as shown by line 15 and line 16:

$processingOK = 'ok';

$\_SESSION["authorized"] = 'ok';

In line 58 of the same session-cookies-2.php, we can see that there is an if statement that runs if the value of $processingOK is not ‘ok’, which is the case when the correct password is not entered and the value of $processingOK is ‘not yet’, which has been set at the beginning of the file.

After the correct password has been put in, we can click the link to go to session-cookies-3.php. This file uses the same logic. $processingOK is set to ‘not yet’ at the beginning of the file, and then there is an if statement which checks if the session has been authorized. If so, the statement sets the value of $processingOK to that of the \_SESSION[“authorized”] which is set to ‘ok’ by the successful login, as seen in session-cookies-2.php. As a result, user can be verified in session-cookies-3.php as they had previously logged in.