Corey Carrington

N13307613

Lab1B: Fix your web app

My web app fixes:

During Lab1A user were allowed to see the images from every other user in the database who uploaded photos. This action would clearly violate the privacy of every user who visited the site. Also, the dashboard.html file was not hidden. You did not see it in the top links whoever you could still manually type it in and go to the page in order to be able to upload images to the database.

Another, feature that I fixed was also with the images and the deletion of images. Originally any user was not only able to view everyone else's images that they posted to there account. Any user could delete all of the images that was ever uploaded to the database from any user. During the fix I change it so that a user can only delete images that they have posted.

Fixes for other students web app:

XSS:

The first vulnerabilities I tried to find was XSS because it is top vulnerabilities for web apps. For XSS to be prevalent there needs to be authentication on the web app. I would be able to change the code on the index.html page but the effects would only be visible on my web page. It would not show up on another user of the website.

SQL Injection:

```
def gallery():
    cur.execute("SELECT * FROM test_photos ORDER BY timestamp DESC LIMIT " +
    str(0) + "," + str(10))
```

```
    cur.execute("""INSERT INTO test_photos VALUES (%s,%s,%s)""",
                (url, photo_caption, timestamp))
```

The students sql queries were written in a way that does not allow for a sql injection attack. The most you could do to the site is upload a picture with with vulgar text. However, that's not an attack.

Conclusion:

The creator of the site didn’t make it if vulnerable in order to do attacks on the site that will affect the system or will be visible by other user on the site.