Name : Linson Peter Rodrigues

LAB: 04 SQL Injection Attack Lab

Task 1: Get Familiar with SQL Statements

- dcbuild

```
[10/27/23]seed@VM:~/.../Labsetup$ dcbuild
Building www
Step 1/5 : FROM handsonsecurity/seed-server:apache-php
 ---> 2365d0ed3ad9
Step 2/5 : ARG WWWDir=/var/www/SQL_Injection
 ---> Using cache
 ---> 5eff9a6e105e
Step 3/5 : COPY Code $WWWDir
 ---> Using cache
 ---> a833d2aa94d2
Step 4/5 : COPY apache_sql_injection.conf  /etc/apache2/sites-available
 ---> Using cache
 ---> 7765df297359
Step 5/5 : RUN  a2ensite apache_sql_injection.conf
 ---> Using cache
 ---> becb5aabae1d

Successfully built becb5aabae1d
Successfully tagged seed-image-www-sqli:latest
Building mysql
Step 1/7 : FROM mysql:8.0.22
 ---> d4c3cafb11d5
Step 2/7 : ARG DEBIAN_FRONTEND=noninteractive
 ---> Using cache
 ---> 0ba37b6d1045
Step 3/7 : ENV MYSQL_ROOT_PASSWORD=dees
 ---> Using cache
 ---> 7f0bc397b820
Step 4/7 : ENV MYSQL_USER=seed
 ---> Using cache
 ---> 486efd1643a7
Step 5/7 : ENV MYSQL_PASSWORD=dees
 ---> Using cache
 ---> a400779bdd81
Step 6/7 : ENV MYSQL_DATABASE=sqllab_users
 ---> Using cache
```

- dcup

```
[10/27/23]seed@VM:~/.../Labsetup$ dcup
www-10.9.0.5 is up-to-date
mysql-10.9.0.6 is up-to-date
Attaching to www-10.9.0.5, mysql-10.9.0.6
mysql-10.9.0.6 | 2023-10-27 18:25:57+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2023-10-27 18:25:57+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
mysql-10.9.0.6 | 2023-10-27 18:25:57+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2023-10-27 18:25:57+00:00 [Note] [Entrypoint]: Initializing database files
mysql-10.9.0.6 | 2023-10-27T18:25:57.250328Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.22) initializing of server in progr
ess as process 44
mysql-10.9.0.6 | 2023-10-27T18:25:57.254564Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
mysql-10.9.0.6 | 2023-10-27T18:25:59.976842Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
mysql-10.9.0.6 | 2023-10-27T18:26:01.807434Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an empty password ! Please consi
der switching off the --initialize-insecure option.
mysql-10.9.0.6 | 2023-10-27 18:26:04+00:00 [Note] [Entrypoint]: Database files initialized
mysql-10.9.0.6 | 2023-10-27 18:26:04+00:00 [Note] [Entrypoint]: Starting temporary server
mysql-10.9.0.6 | mysqld will log errors to /var/lib/mysql/e8c10703532b.err
mysql-10.9.0.6 | mysqld is running as pid 91
mysql-10.9.0.6 | 2023-10-27 18:26:06+00:00 [Note] [Entrypoint]: Temporary server started.
mysql-10.9.0.6 | Warning: Unable to load '/usr/share/zoneinfo/iso3166.tab' as time zone. Skipping it.
mysql-10.9.0.6 | Warning: Unable to load '/usr/share/zoneinfo/leap-seconds.list' as time zone. Skipping it.
mysql-10.9.0.6 | Warning: Unable to load '/usr/share/zoneinfo/zone.tab' as time zone. Skipping it.
mysql-10.9.0.6 | Warning: Unable to load '/usr/share/zoneinfo/zone1970.tab' as time zone. Skipping it.
mysql-10.9.0.6 | 2023-10-27 18:26:09+00:00 [Note] [Entrypoint]: Creating database sqllab_users
mysql-10.9.0.6 | 2023-10-27 18:26:09+00:00 [Note] [Entrypoint]: Creating user seed
www-10.9.0.5 |  * Starting Apache httpd web server apache2                    AH00558: apache2: Could not reliably determine the server's f
ully qualified domain name, using 10.9.0.5. Set the 'ServerName' directive globally to suppress this message
www-10.9.0.5 |  *
www-10.9.0.5 |  * Starting Apache httpd web server apache2                    AH00558: apache2: Could not reliably determine the server's f
ully qualified domain name, using 10.9.0.5. Set the 'ServerName' directive globally to suppress this message
www-10.9.0.5 |  *
mysql-10.9.0.6 | 2023-10-27 18:26:09+00:00 [Note] [Entrypoint]: Giving user seed access to schema sqllab_users
mysql-10.9.0.6 |
mysql-10.9.0.6 | 2023-10-27 18:26:09+00:00 [Note] [Entrypoint]: /usr/local/bin/docker-entrypoint.sh: running /docker-entrypoint-initdb.d/sqll
ab_users.sql
```

- Dockps

```
[10/27/23]seed@VM:~/.../Labsetup$ dockps
2da89b9da63d  www-10.9.0.5
e8c10703532b  mysql-10.9.0.6
[10/27/23]seed@VM:~/.../Labsetup$ docksh e8
root@e8c10703532b:/#
```

- Logged into the database

```
root@e8c10703532b:/# 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 (0.02 sec)
```

- Activated the database

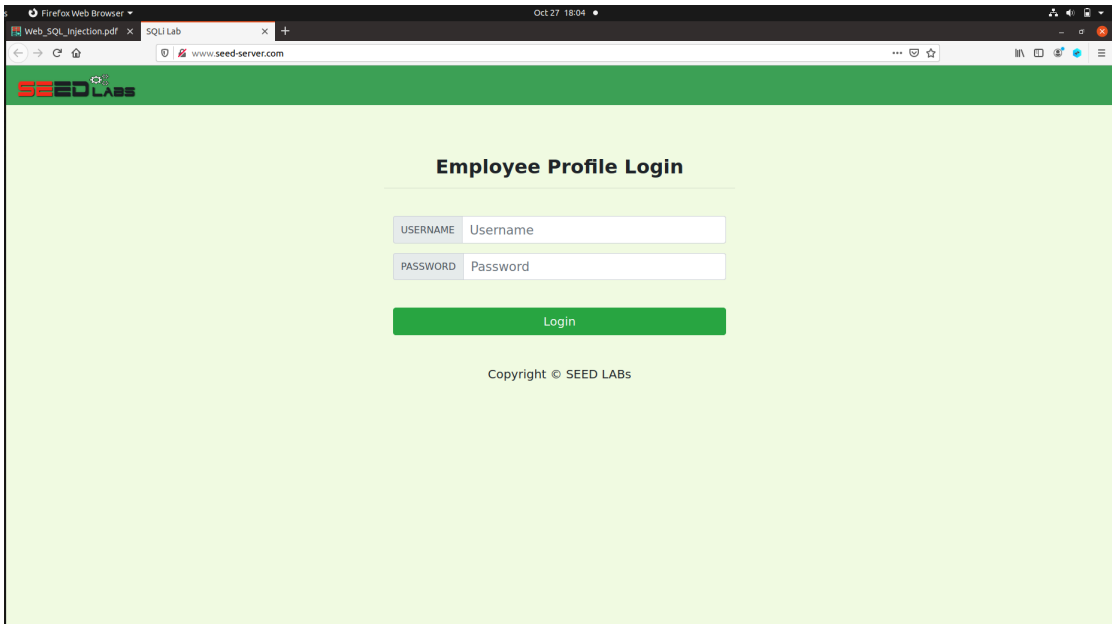
```
mysql> use sqllab_users;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

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

- Described the credential tables

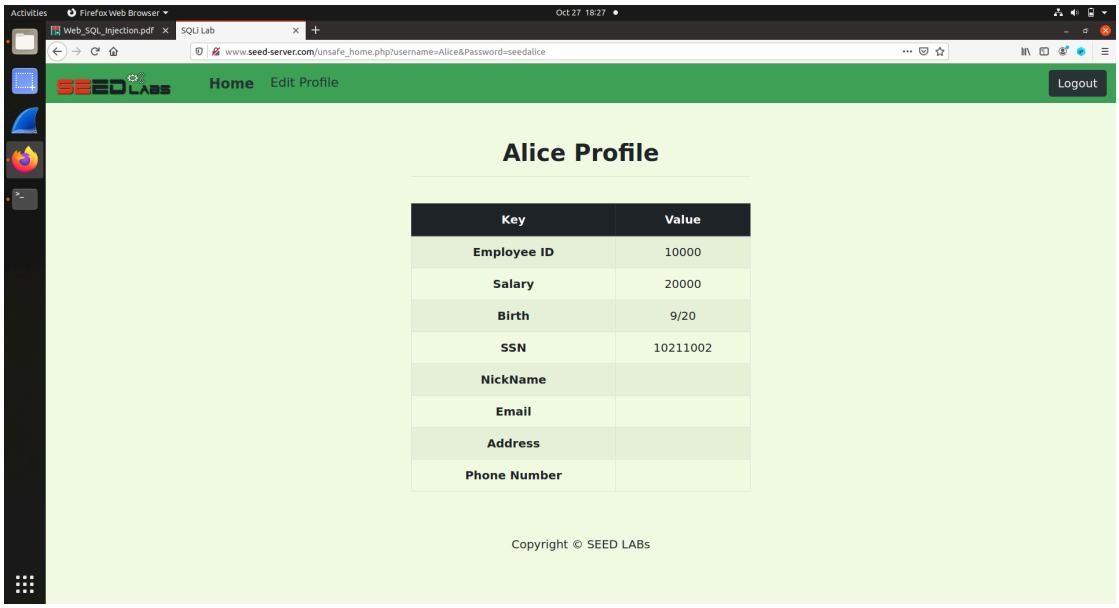
```
mysql> describe credential;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| ID          | int unsigned | NO   | PRI | NULL    | auto_increment |
| Name        | varchar(30)  | NO   |     | NULL    |                |
| EID         | varchar(20)  | YES  |     | NULL    |                |
| Salary      | int          | YES  |     | NULL    |                |
| birth       | varchar(20)  | YES  |     | NULL    |                |
| SSN         | varchar(20)  | YES  |     | NULL    |                |
| PhoneNumber | varchar(20)  | YES  |     | NULL    |                |
| Address     | varchar(300) | YES  |     | NULL    |                |
| Email       | varchar(300) | YES  |     | NULL    |                |
| NickName    | varchar(300) | YES  |     | NULL    |                |
| Password    | varchar(300) | YES  |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
11 rows in set (0.01 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>
```

```
[10/27/23]seed@VM:~/.../Labsetup$ echo -n 'seedalice' | sha1sum
fdbe918bdae83000aa54747fc95fe0470fff4976  -
[10/27/23]seed@VM:~/.../Labsetup$
```

Task 2: SQL Injection Attack on SELECT Statement

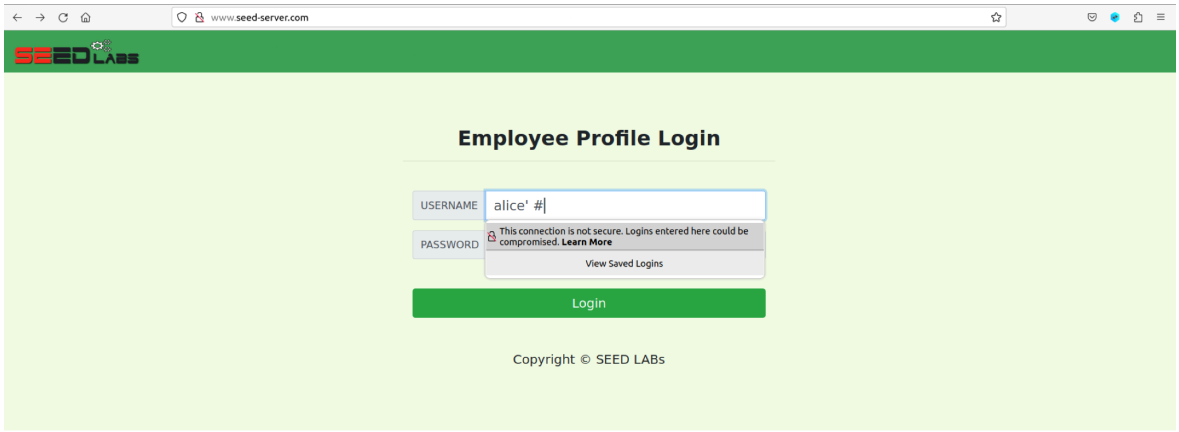


- Logged in using Alice's credential

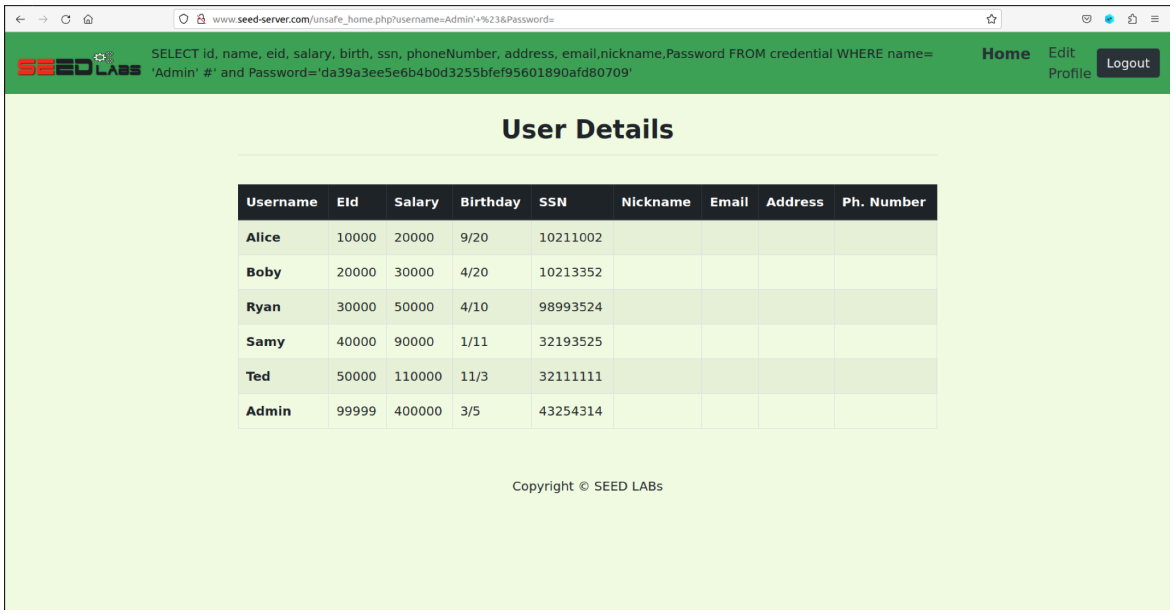


Task 2.1: SQL Injection Attack from webpage.

- USERNAME = **Admin' #**
- PASSWORD = **null**
- The statement after # will be regarded as comments, so we can log in as Admin.

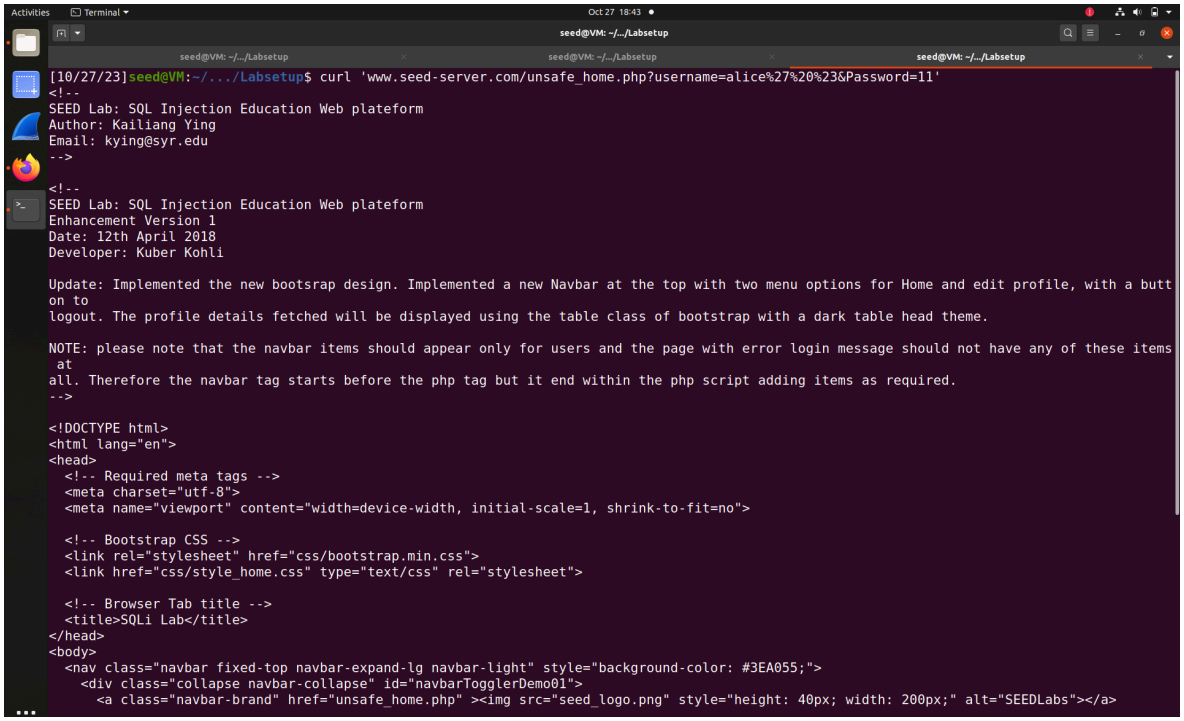


- Logged into **alice** profile
- Was able to see the user details



Task 2.2: SQL Injection Attack from command line.

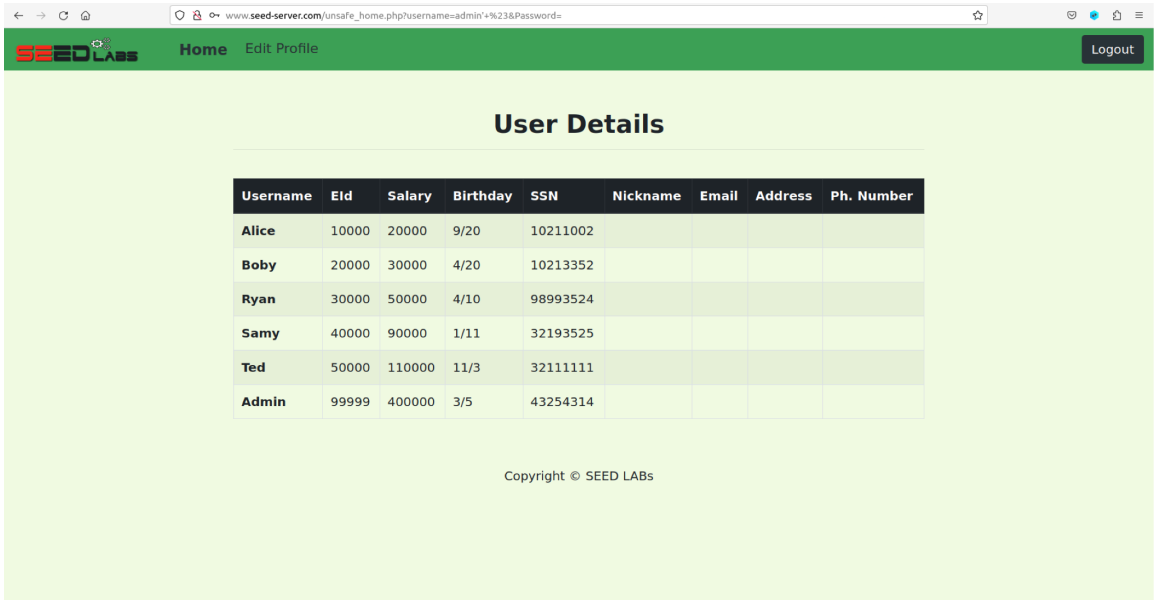
- curl ‘www.seed-server.com/unsafe_home.php?username=alice&Password=11’
- Modified the above link using below injection
- %27 (used for HASHTAG)
- %20 (single Quote)
- %23 (Spacebar)
- curl''www.seed-server.com/unsafe_home.php?username=alice%27%20%23&Password=11'



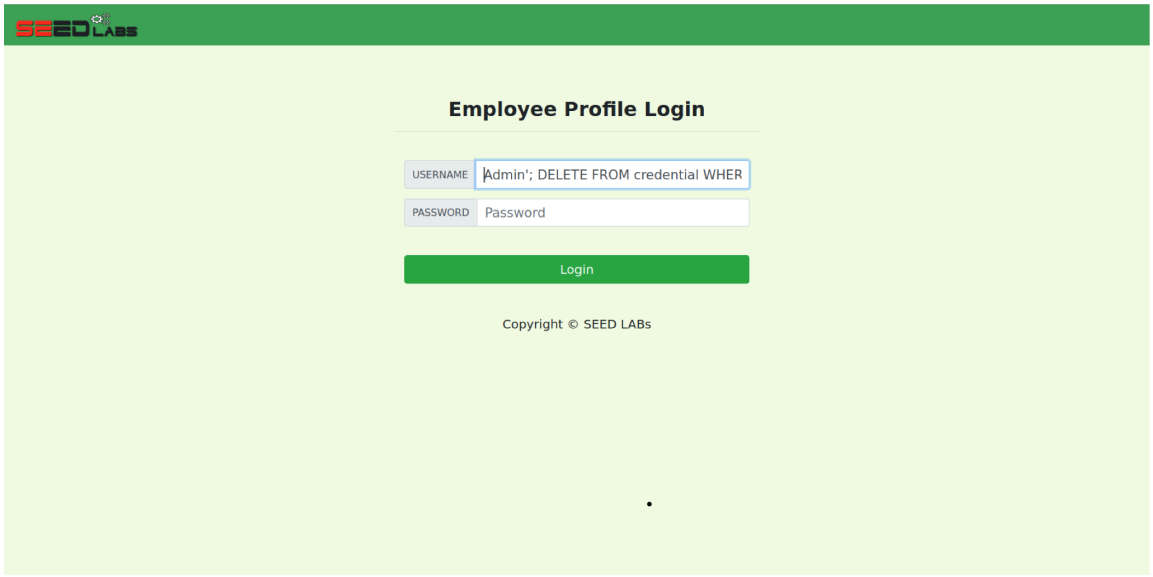
- Attack was done on the webpage, by using the ‘**curl**’ I was able to make request to the webpage, targeting user **Alice**.

Task 2.3: Append a new SQL statement.

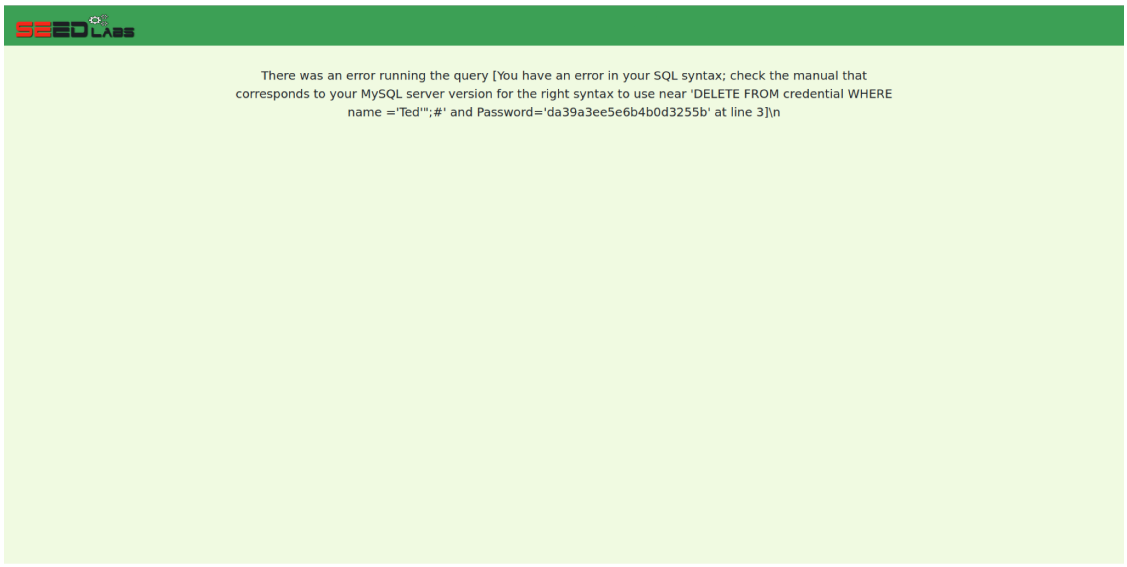
- Displayed the current members in the database.



- I am targeting to delete the user, **Ted** from the database.
- Admin'; DELETE FROM credential WHERE name ='Ted'";#



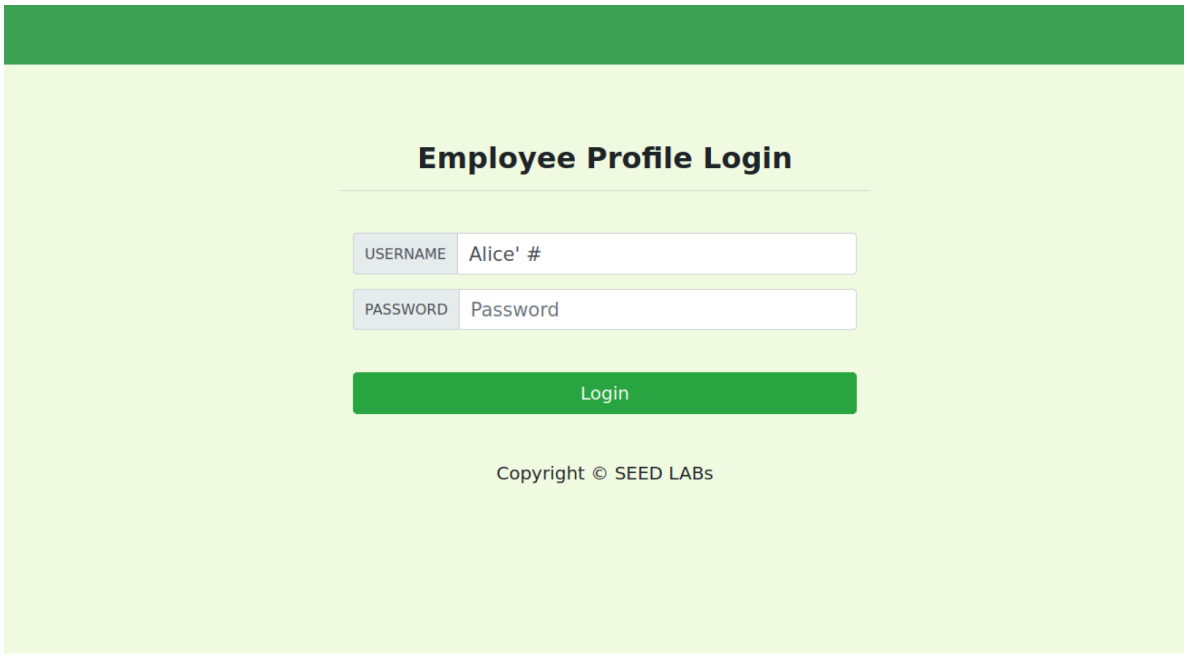
- I tried executing the command to delete the user **Ted** from the database, but it kept on giving me error.
- After researching on internet I got the explanation for the above error.
- An attack cannot be rendered on MYSQL because of PHP's MySQL extension, i.e. **mysqli::query()**
- API; does not allow multiple queries to run in the database server, as a way of combating SQL injection



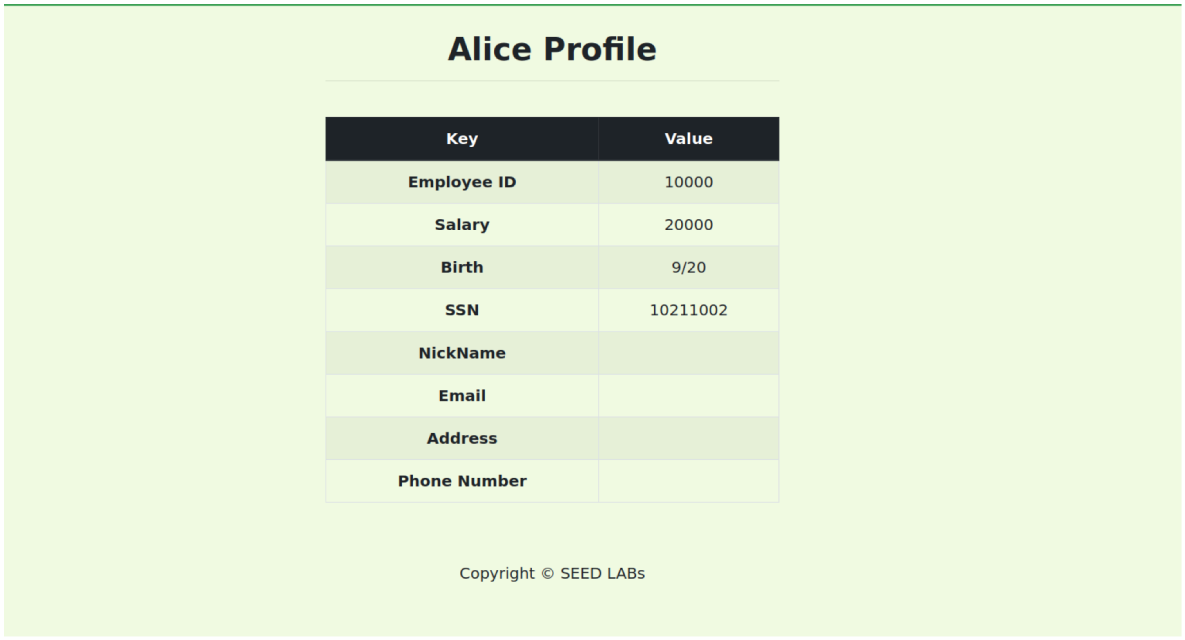
Task 3.1: Modify your own salary.

SQL Injection attack on UPDATE Statement

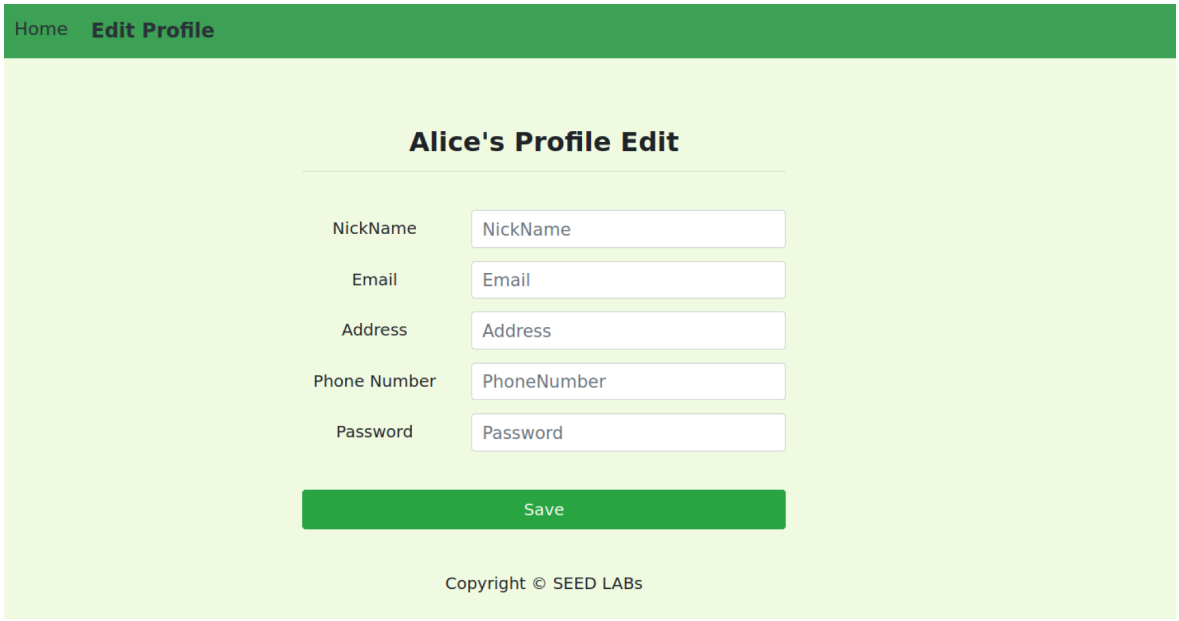
- In order to update Alice's profile I'll need to access the details of Alice from the database using Injection Attack and by logging in Alice's profile .



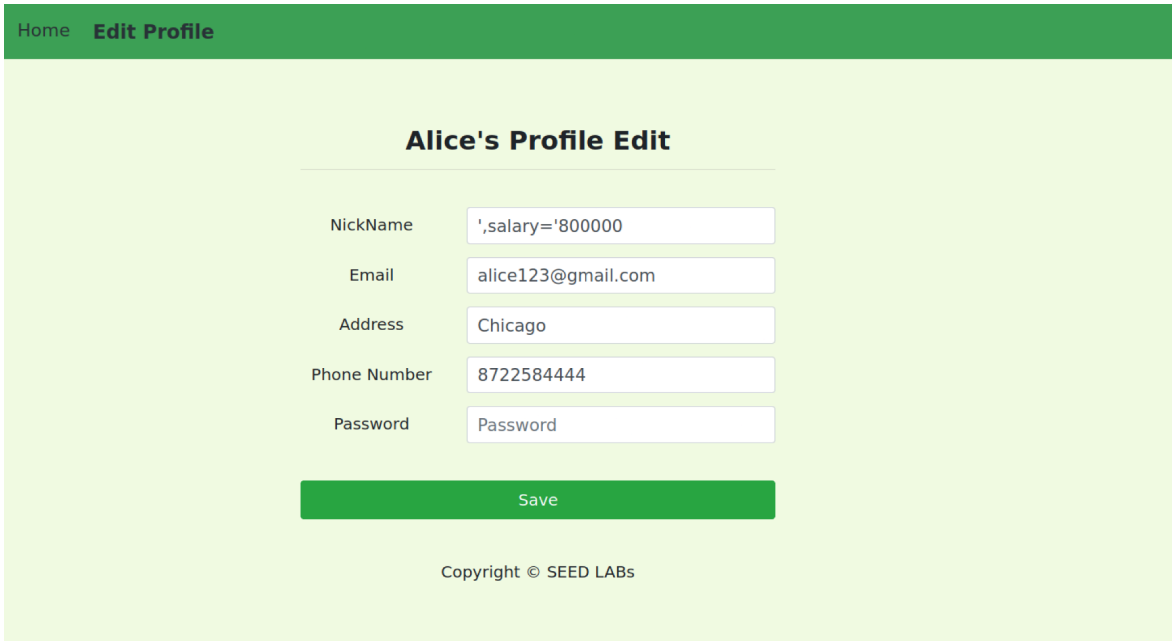
- Logged into **Alice's** profile
- Inorder to update the profile of **Alice** we need to click on **Edit Profile** tab of the page



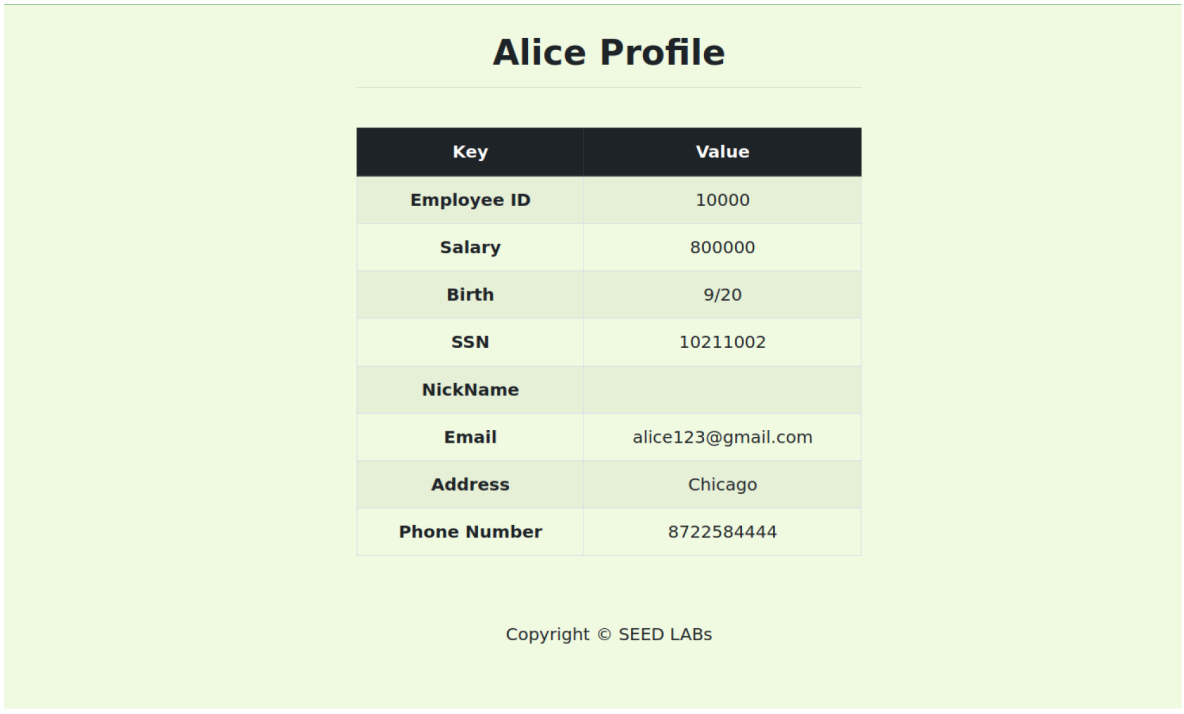
- View of **Edit Profile**



- Tried to inject salary column into the database, and used that to update salary of Alice



- Alice's salary was updated successfully
- The salary previously mentioned was 20000 after updating it was 800000

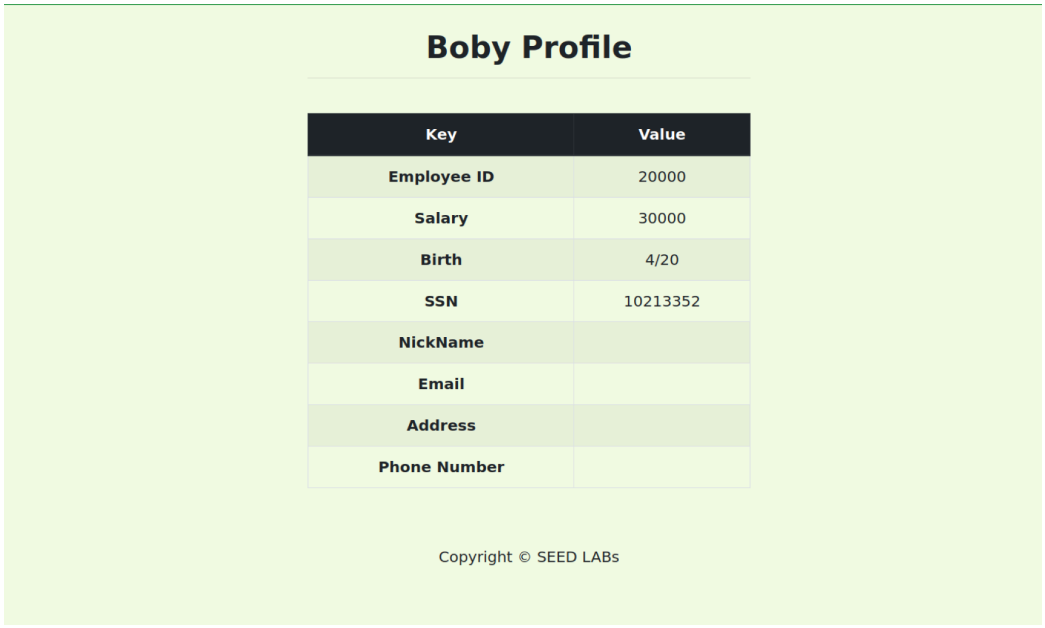


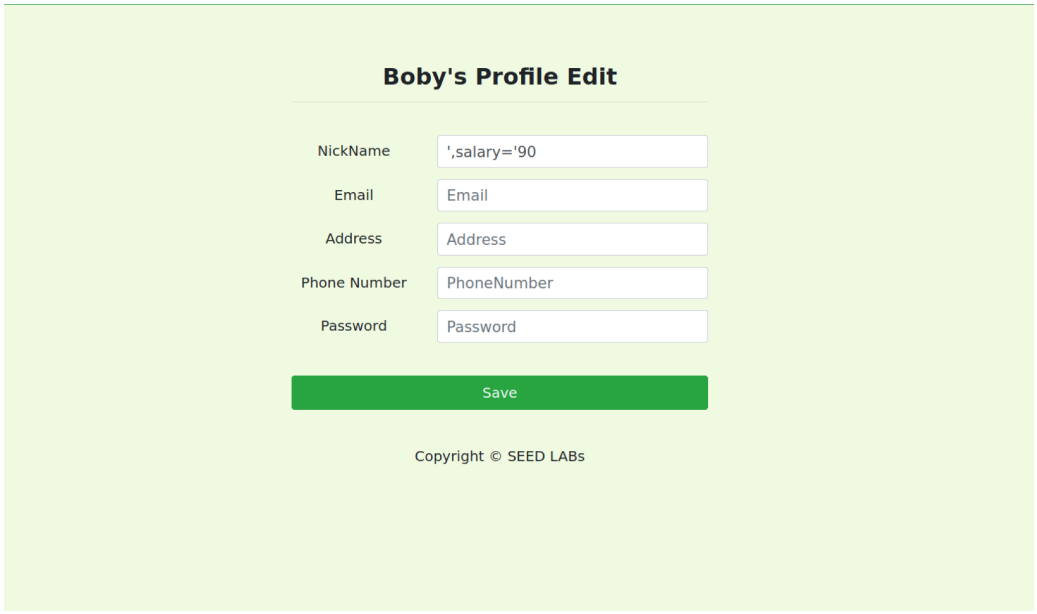
Task 3.2: Modify other people' salary.

- There are multiple ways to modify other people's salary

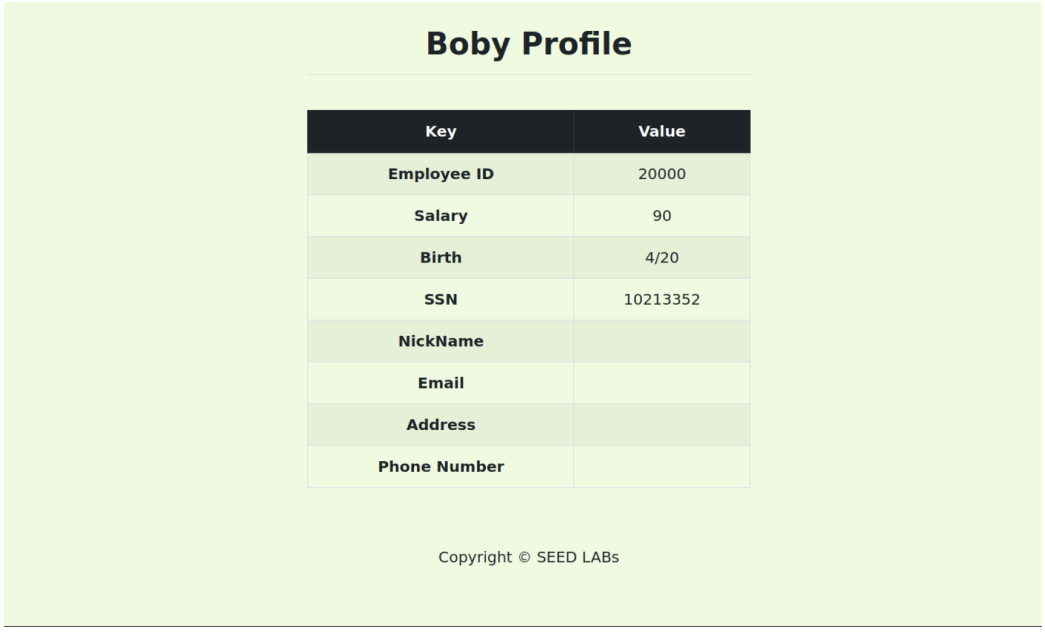
1. Attacking the user's credential and then accessing their details, an updating them .
2. In Alice's profile, we can write an update statement to update the details.

- Boby's salary before updating





- Boby's salary was changed successfully



Boby Profile

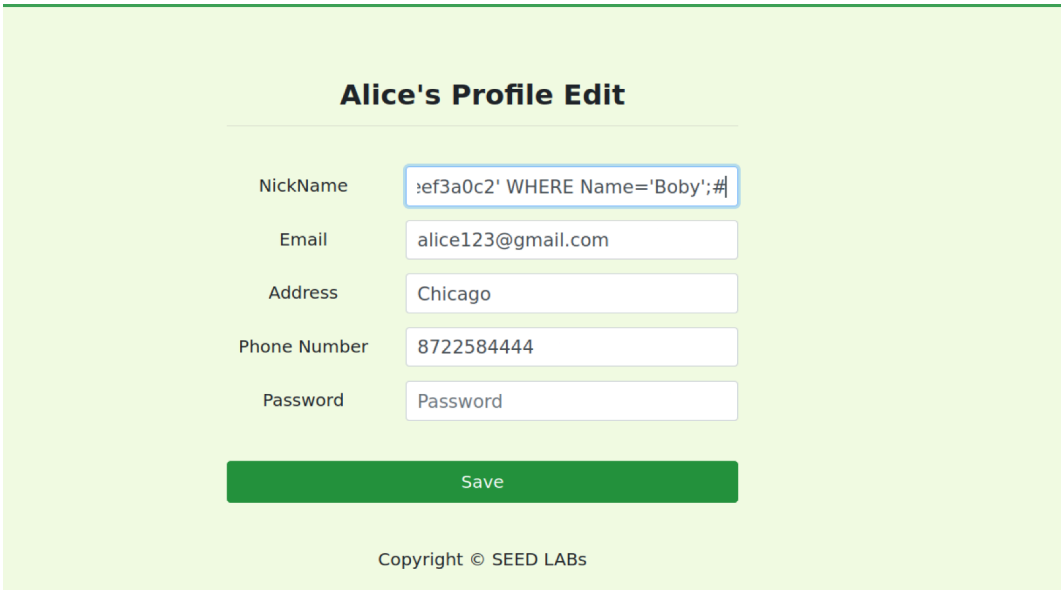
Key	Value
Employee ID	20000
Salary	90
Birth	4/20
SSN	10213352
NickName	
Email	
Address	
Phone Number	

Task 3.3: Modify other people' password.

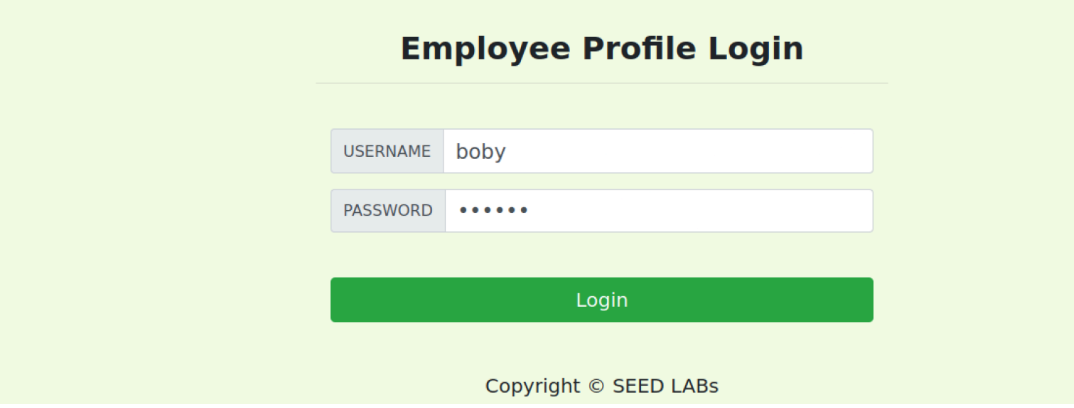
- In php code we can see that all the user passwords that are entered are hashed with SHA1 hash algorithm, so the password that I am creating for boby will also be hashed, so it would be easy for database to relate
- Created a new password and saved it in **password.txt** file
- Hashed the new password using SHA1 algorithm to generate the hash code .

```
[11/08/23]seed@VM:~/.../Labsetup$ echo -n boby23 > password.txt
[11/08/23]seed@VM:~/.../Labsetup$ echo -n 'boby23' | sha1sum
ee9176a05386f1e4c86b115e69c6ca9eeef3a0c2  -
[11/08/23]seed@VM:~/.../Labsetup$
```

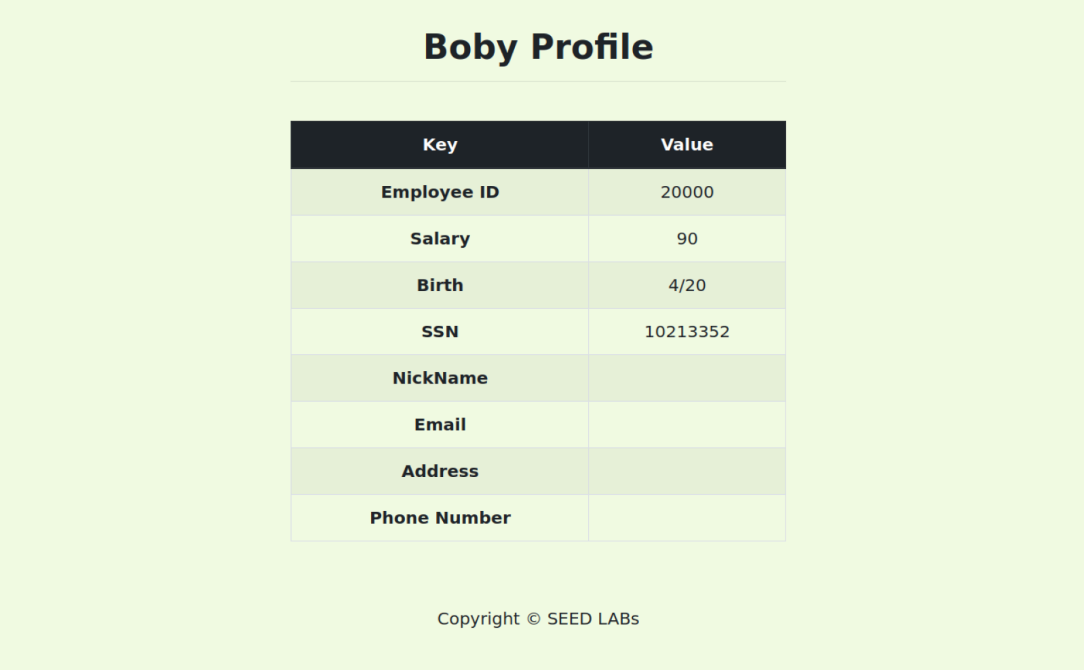
- Inorder to change the password for boby, I need to update the password through Alice's profile as done previously for changing the salary of bob .



- Boby's password was changed successfully.



- New password granted the access to boby's account

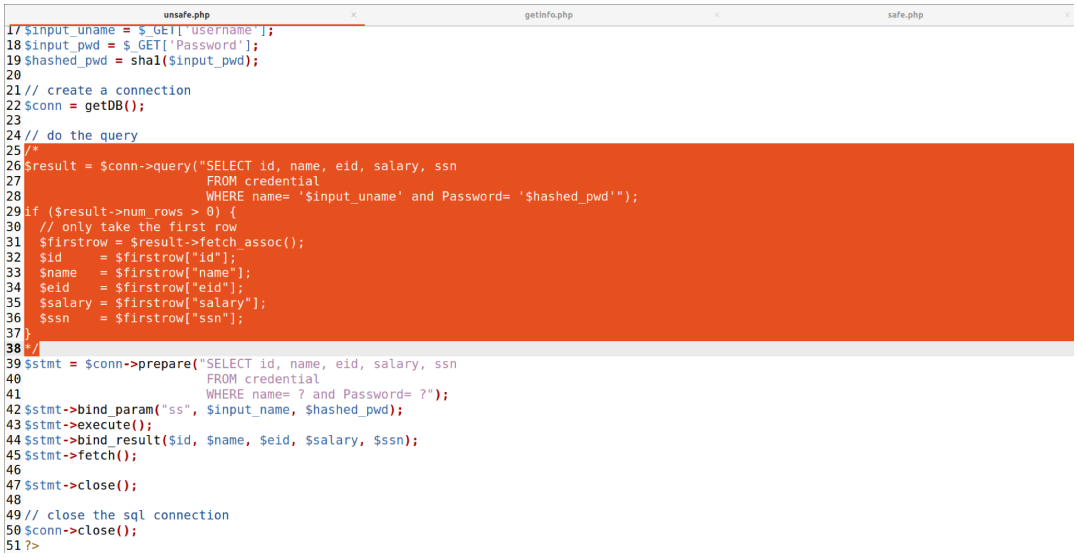


Boby Profile

Key	Value
Employee ID	20000
Salary	90
Birth	4/20
SSN	10213352
NickName	
Email	
Address	
Phone Number	

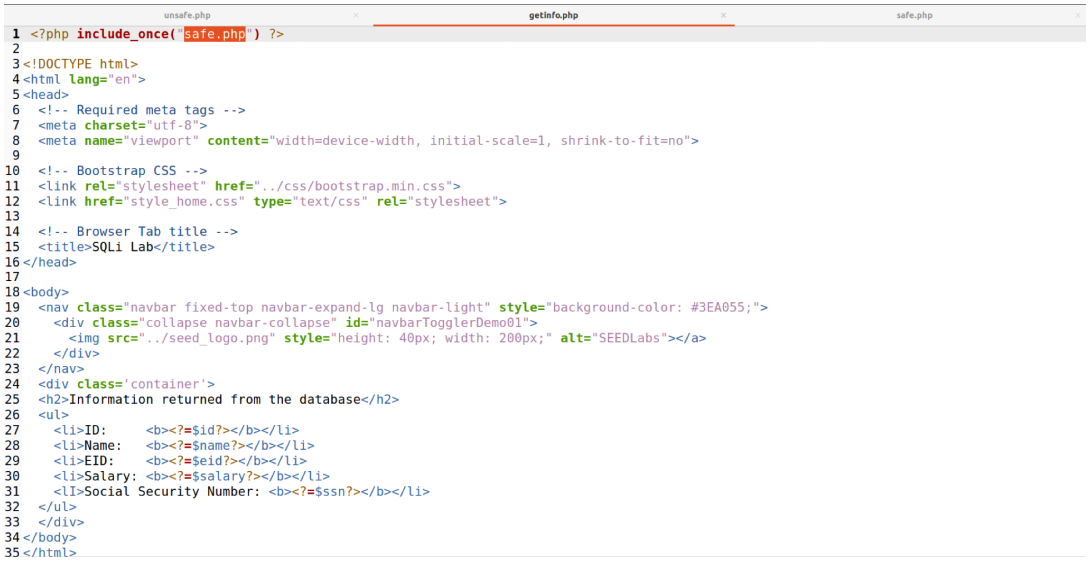
Task 4: Countermeasure — Prepared Statement

- Edited the **unsafe.php** as shown below
- Commented out the **highlighted part**
- Injected the prepared statement from **line 39 - line 47**



```
unsafe.php | getinfo.php | safe.php

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

$stmt->close();

// close the sql connection
$conn->close();
?>
```

- Inside the **getinfo.php** file made the changes as highlighted below **safe.php**
- Saved the file



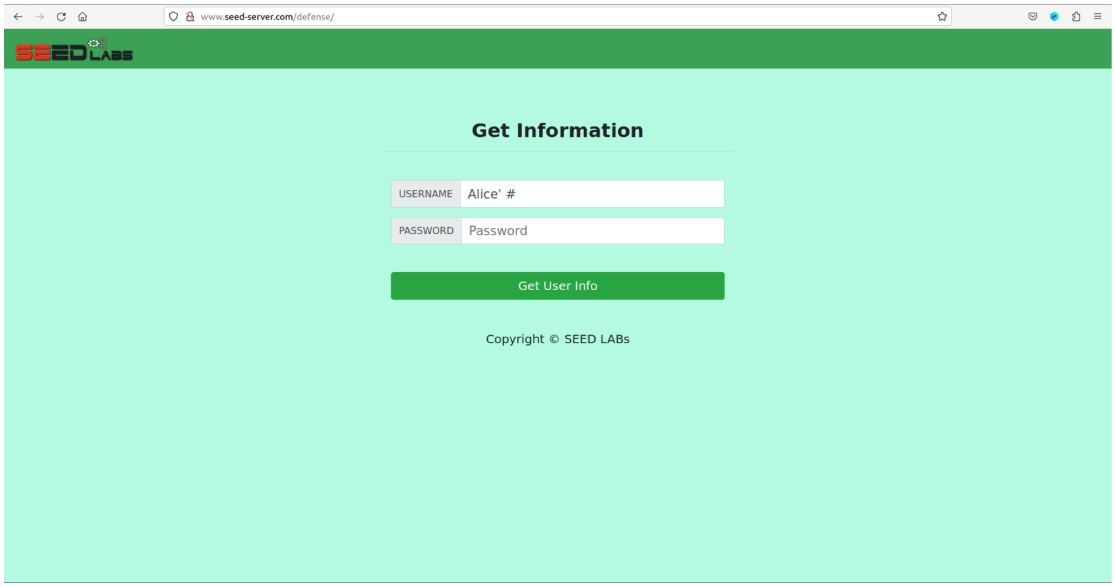
```
unsafe.php | getinfo.php | safe.php

<?php include_once("safe.php") ?>

<!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="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">
      <img src="../seed_logo.png" style="height: 40px; width: 200px;" alt="SEEDLabs"></a>
    </div>
  </nav>
  <div class='container'>
  <h2>Information returned from the database</h2>
  <ul>
    <li>ID:     <b><?=$id?></b></li>
    <li>Name:   <b><?=$name?></b></li>
    <li>EID:    <b><?=$eid?></b></li>
    <li>Salary: <b><?=$salary?></b></li>
    <lI>Social Security Number: <b><?=$ssn?></b></li>
  </ul>
  </div>
</body>
</html>
```

- Made copy of **unsafe.php** file and saved it as **safe.php**

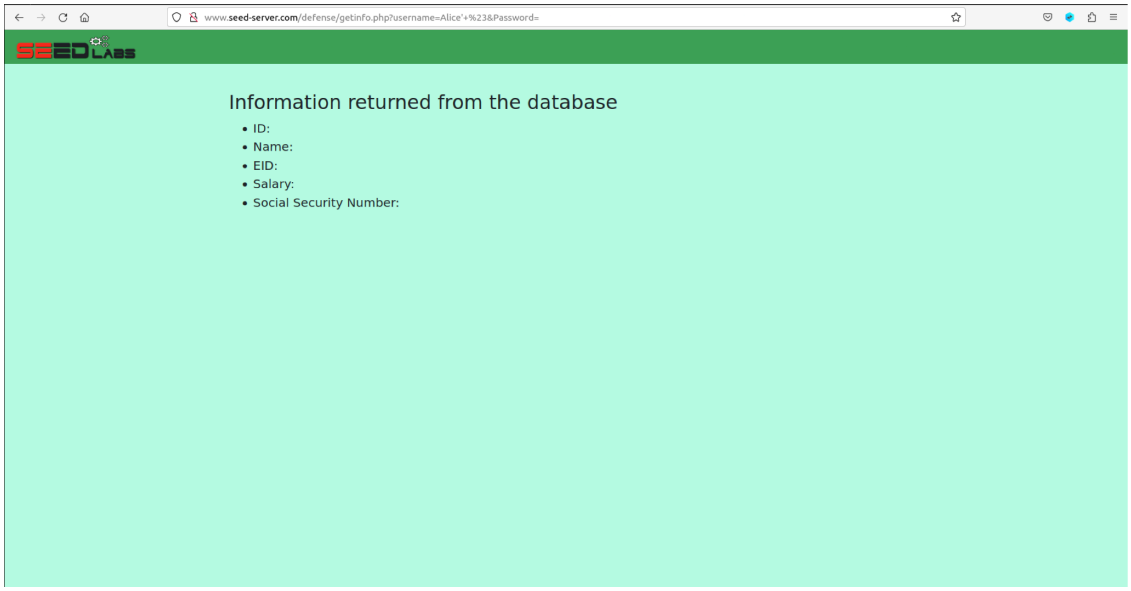
unsafe.php | getinfo.php | safe.php

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

$stmt->close();

// close the sql connection
$conn->close();
?>
```

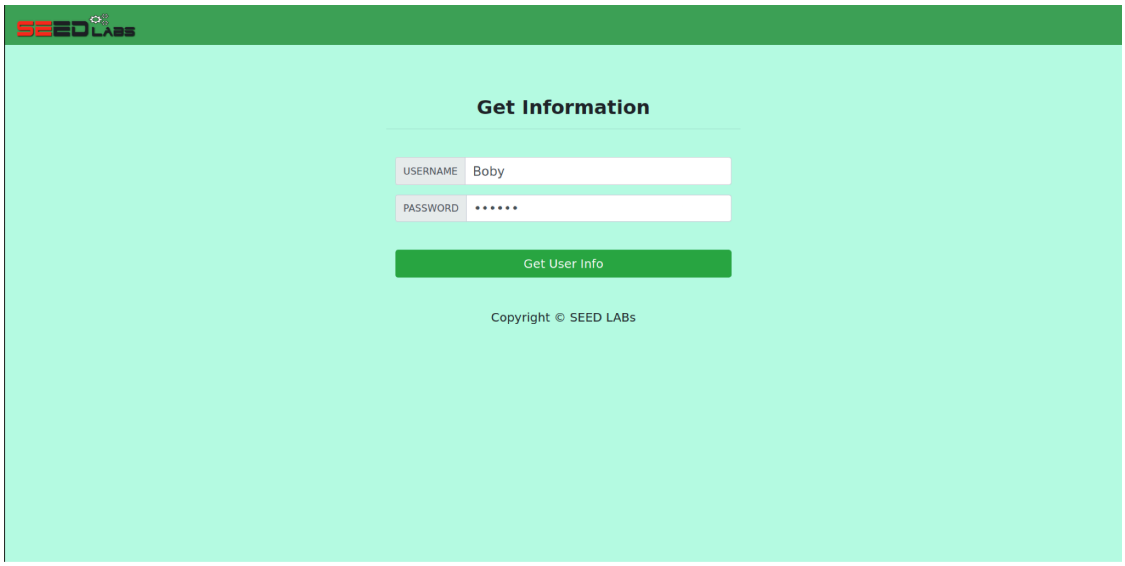
- After implementing the prepared statement, I tried the attack again as shown below
- I tried to login into **Alice's** profile but it didn't showed any info from the database



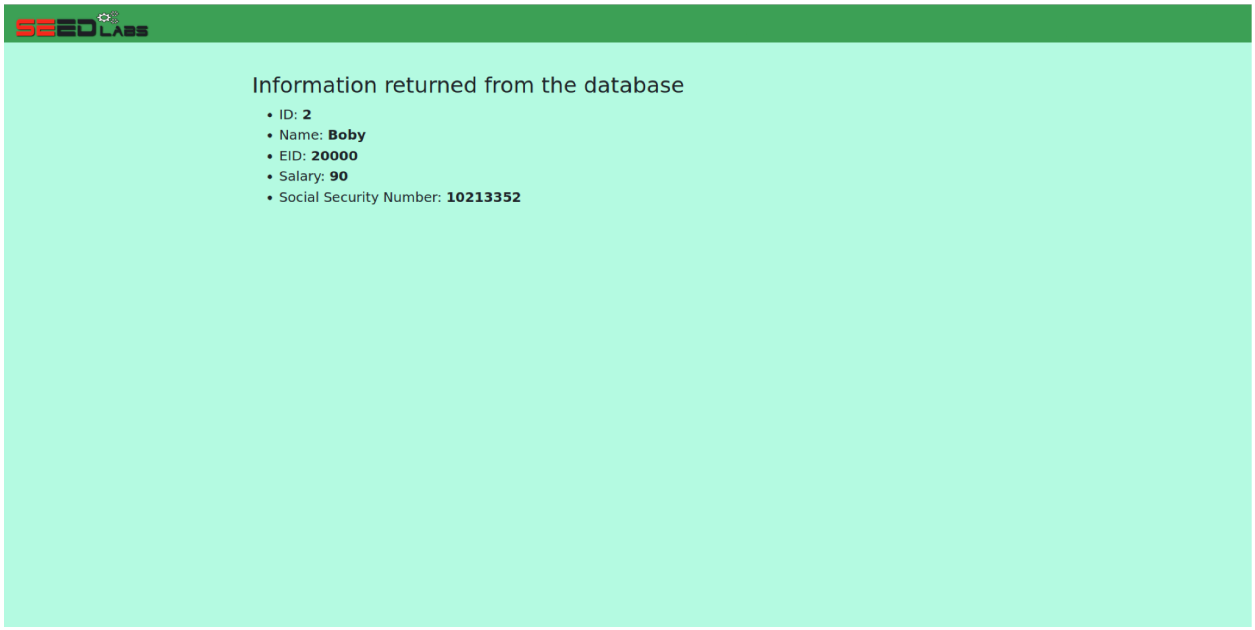
- The following output was obtained from the database and the login attempt failed



- I tried login with **Bobby's** credential



- Got the information from the database



- Was able to successfully implement the countermeasures