**Lab Session 08**

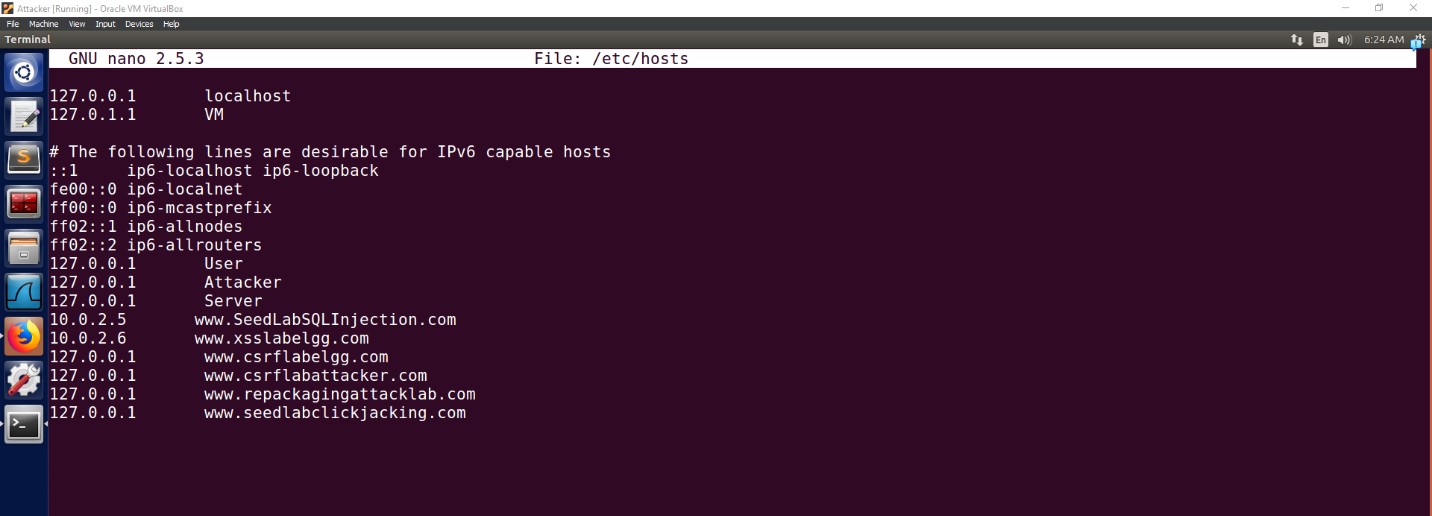
**SQL Injection Attack**

**Environment Setup:**

**Prepare the environment using the pre-built SEED Ubuntu 16.04 virtual machine, configuring two VMs: VM1 designated as the Attacker and VM2 as the Victim. Proceed to edit the /etc/hosts file, updating the IP address linked with http://www.SEEDLabSQLInjection.com to match the IP address of the Victim machine.**

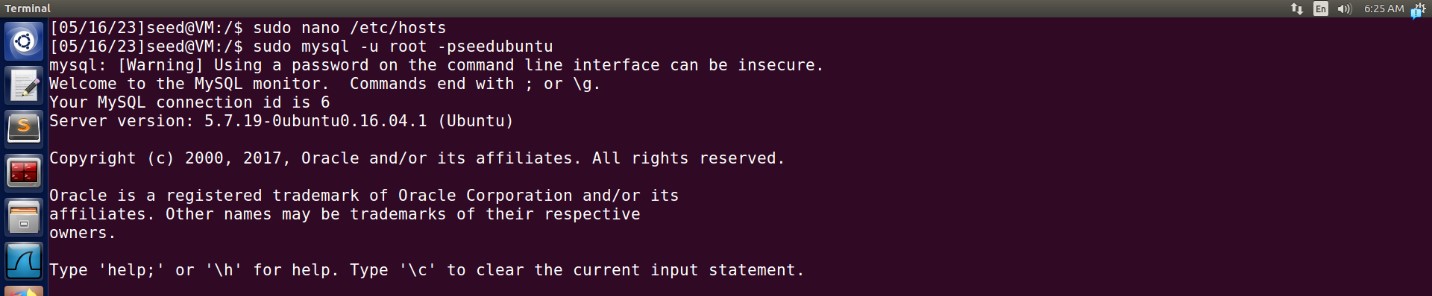
Utilize the SEED Ubuntu 16.04 pre-configured virtual machine (VM) to establish a setup comprising two VMs: VM1 acting as the Attacker and VM2 serving as the Victim.

Access the /etc/hosts file and modify the IP address associated with http://www.SEEDLabSQLInjection.com to match the IP address of the Victim machine.

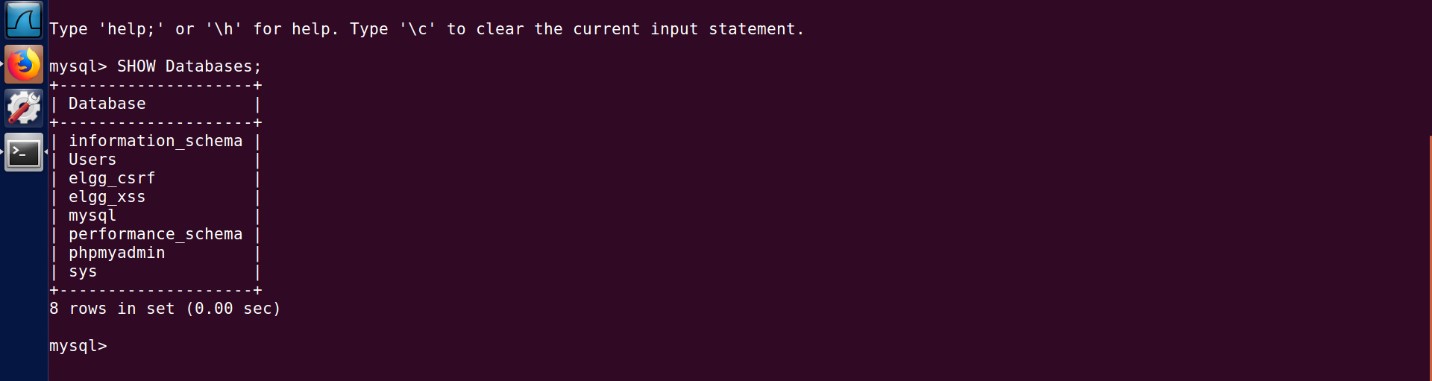


# Task 1: Getting Familiar with SQL

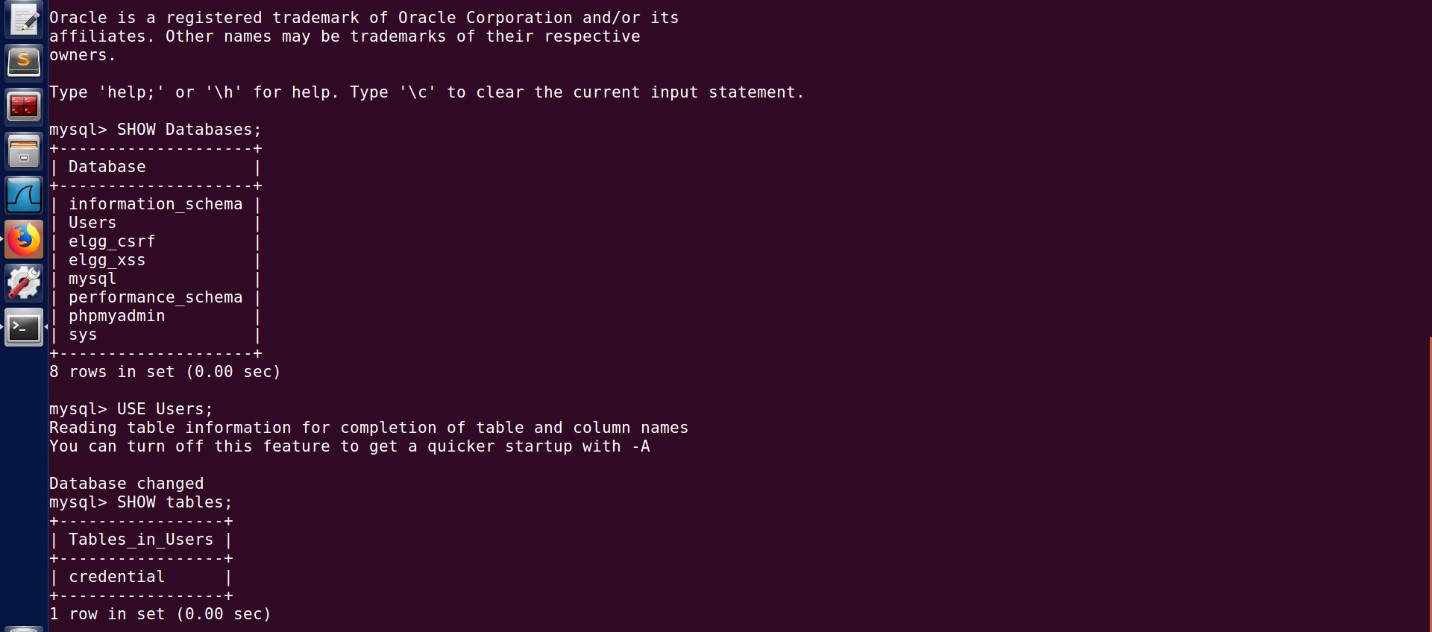
1. login to MySQL console using this command 'mysql -u root -pseedubuntu'



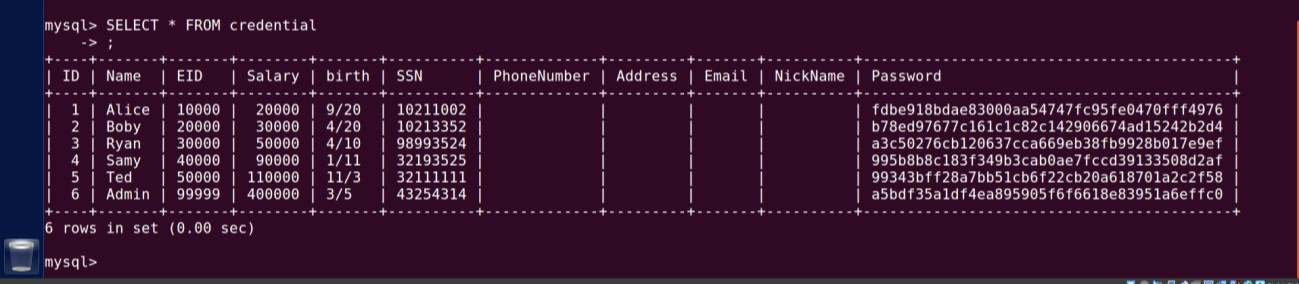
1. Run SHOW Databases command to view all the present databases on the server. There are total 8 databases present.



1. Switch to Users database by 'USE Users' command and view the tables by 'SHOW tables' command. There is only one table present in the Users database by name credentials.

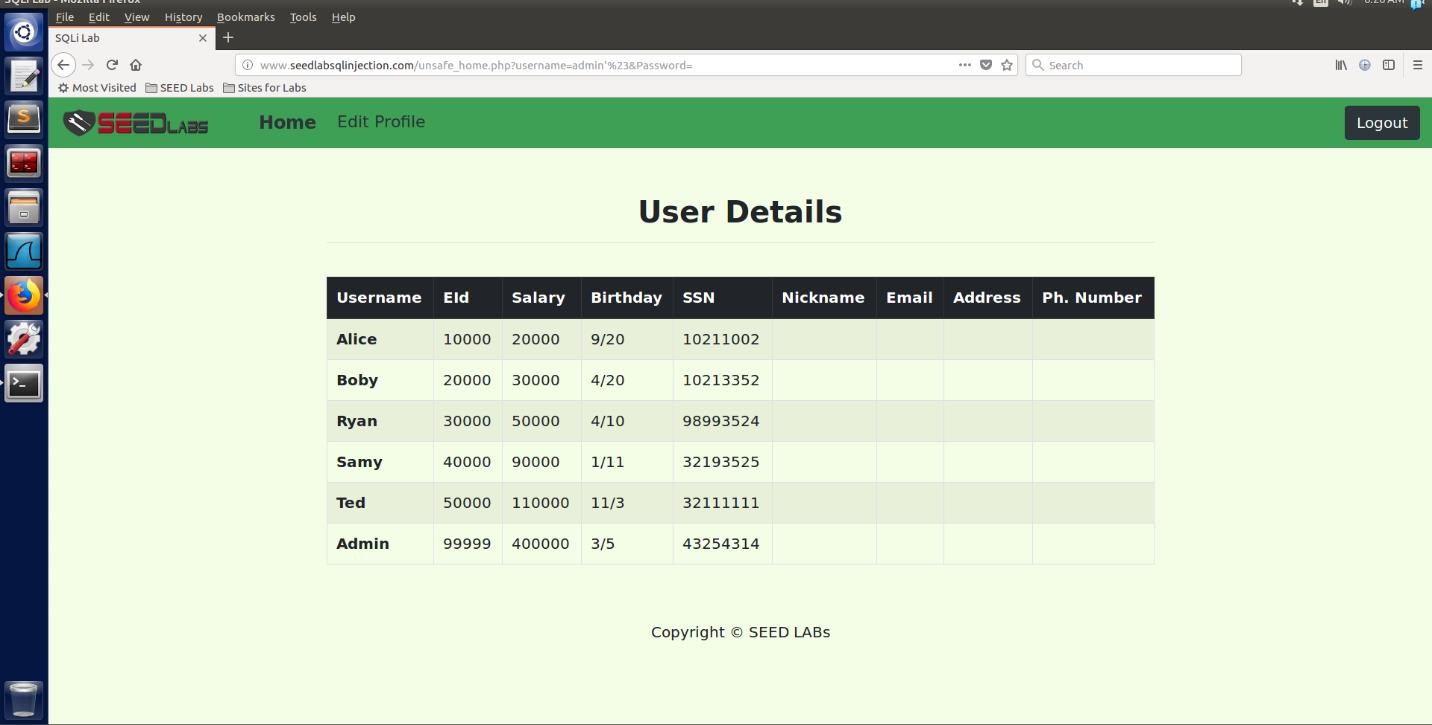
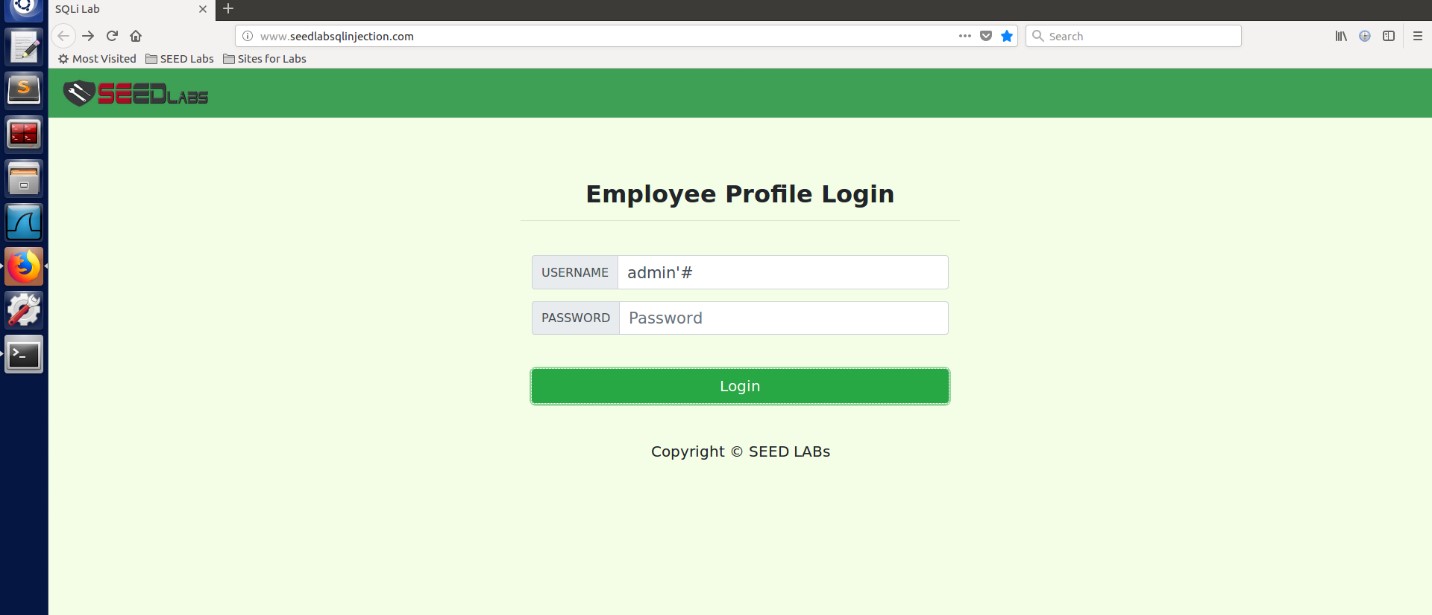


1. Access the credentials table using 'SELECT \* FROM credential;' command.



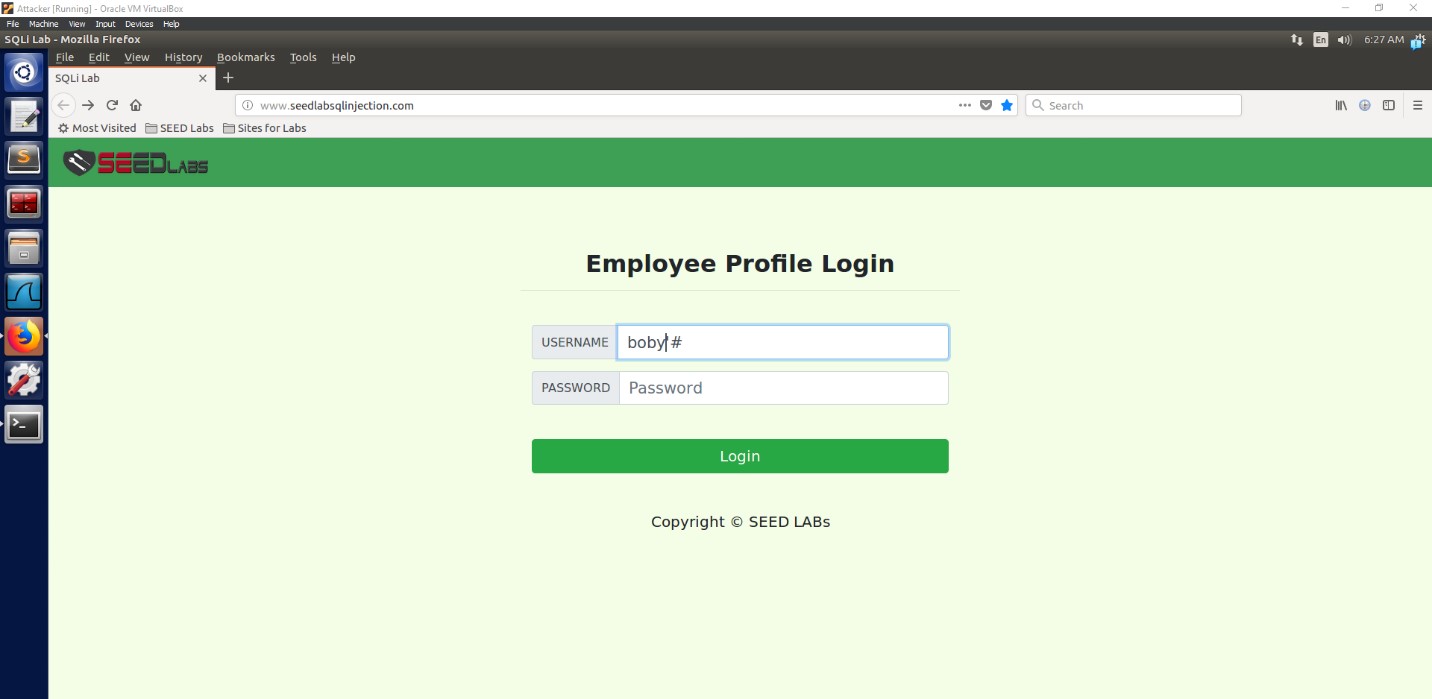
# Task 2: SQL Injection Attack from Webpage and Command Line

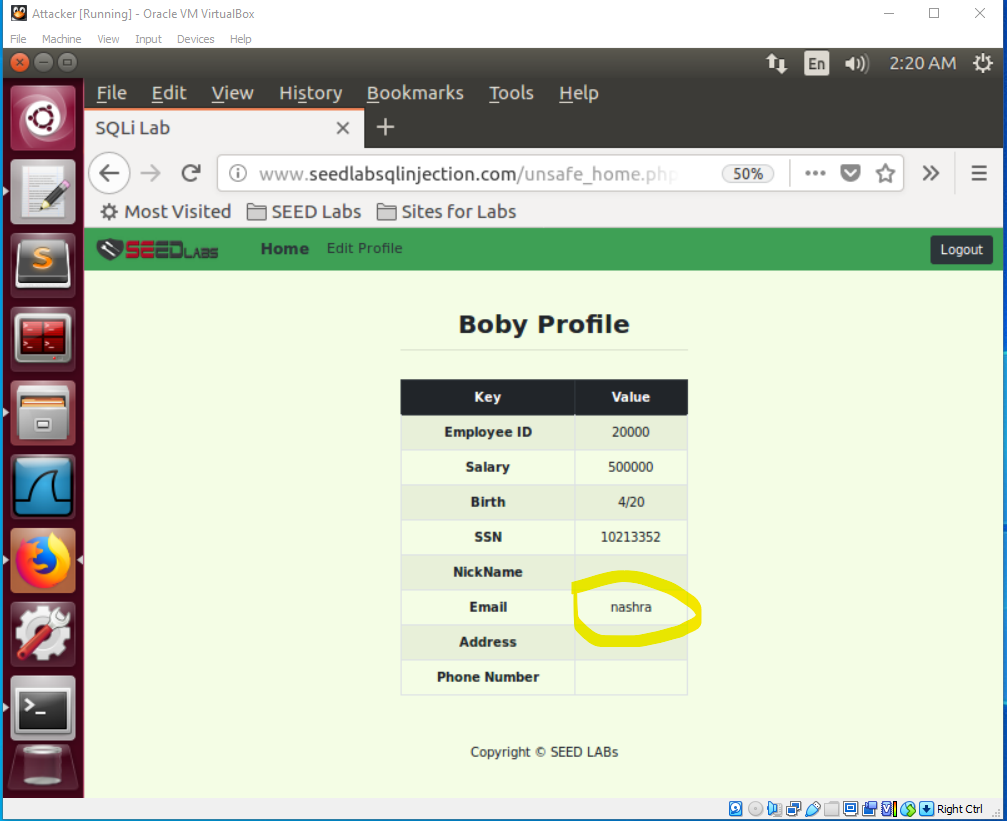
At the login page write admin'#. The following command comment out the remaining query after the name field in the WHERE clause of the sql written in the backend which allows the attacker to login easily into the account of admin.



# Task 3: Modify Data using SQL Injection Attack on Update Statement

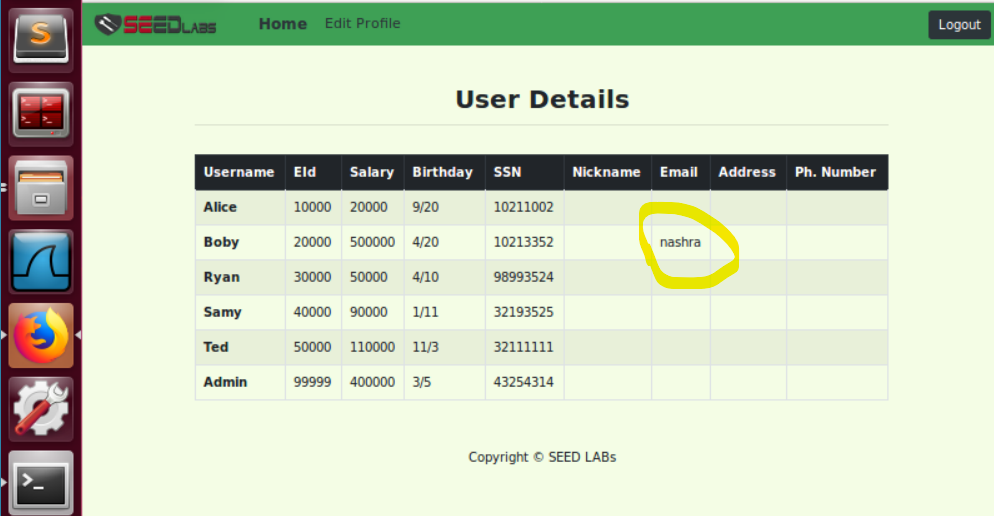
1. Login as Boby and go to the Edit profile page. Write ',email=nashra and click on save, you will see changes in the email column of Boby.





To update the employee’s email, we have written the following script in the email field of Boby’s profile: ‘,email=’nashra

1. Login as Admin and you will see the Boby's email as nashra.



# Task 3: SQL Injection Countermeasure

The following queries written below has vulnerability in them because we can pass any script in it and it will get executed due to no checking parameter to avoid this attack, we write a parameterize query in order to convert any entered script into its desired format whether it be string or integer to avoid execution.

*$result = mysql\_query("select \* from users where (name='$user' and pass='$pass');");*

**OR**

*$result = mysql\_query("select \* from users where (name='$--' and pass='$pass');");*

**OR**

*$result = mysql\_query("select \* from users where (name='\--' and pass='$pass');");*

**STEPS To parameterize:**

*$statement = $db ->prepare*

**CREATE a DB Handle**

*$db = new mysql("localhost","users","passw","DB");*

**CREATE a PREPARED statement**

*$statement = $db->prepare("SELECT \* FROM users WHERE name=? and pass=?);");*

**Bind & Decouple**

*$statement->bind\_param("ss",$user,$pass);*

*$statement->execute();*