

# ADDIS ABABA INSTITUTE OF TECHNOLOGY CENTER OF INFORMATION TECHNOLOGY AND SCIENTIFIC COMPUTING

# SQL Injection

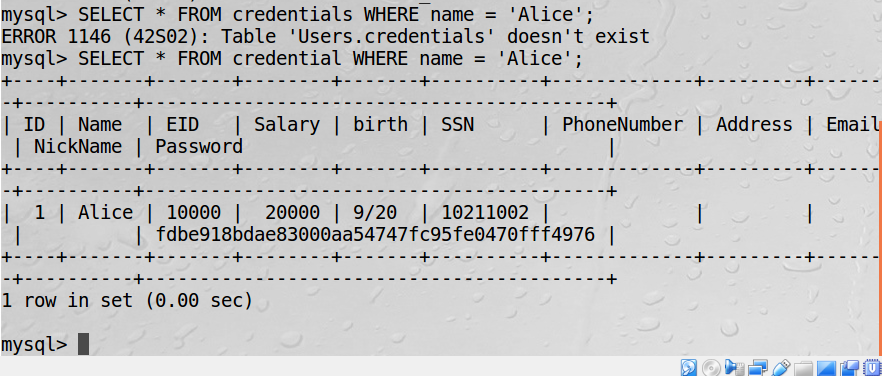
## Prepared by: - Amanuel Genene

ID: - ATE/5124/09

Submitted to: Mr. Abraham

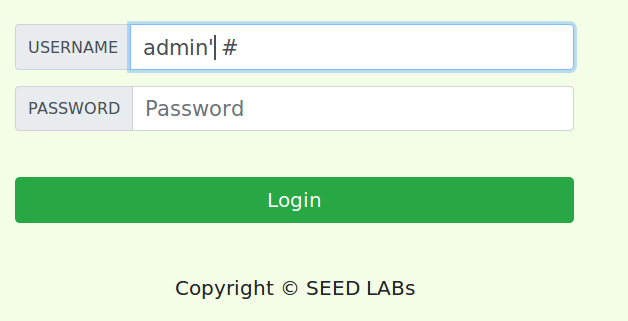
## Task 1

We print all the profile information about Alice. The SQL is << select \* from credential where Name = “Alice”; >>.

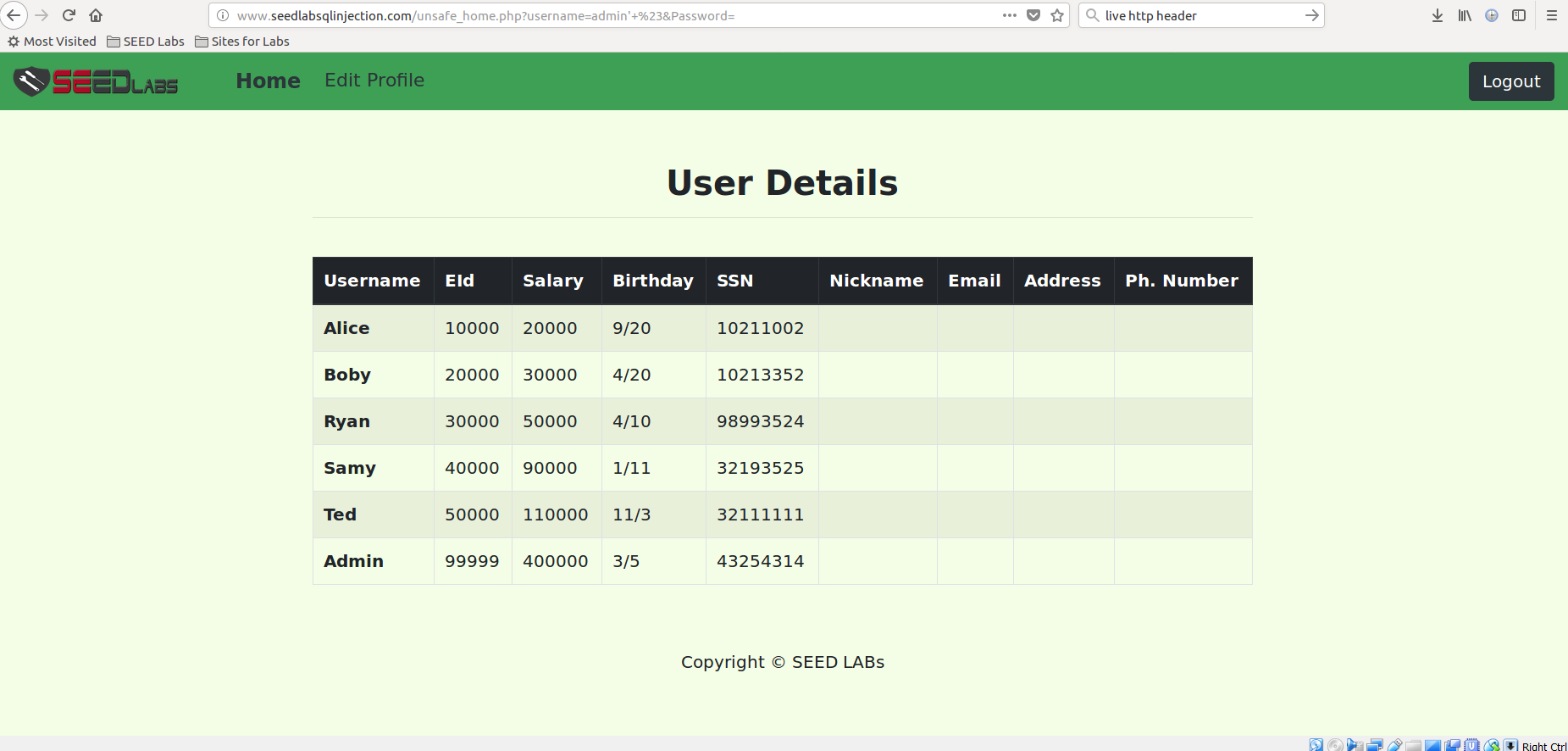


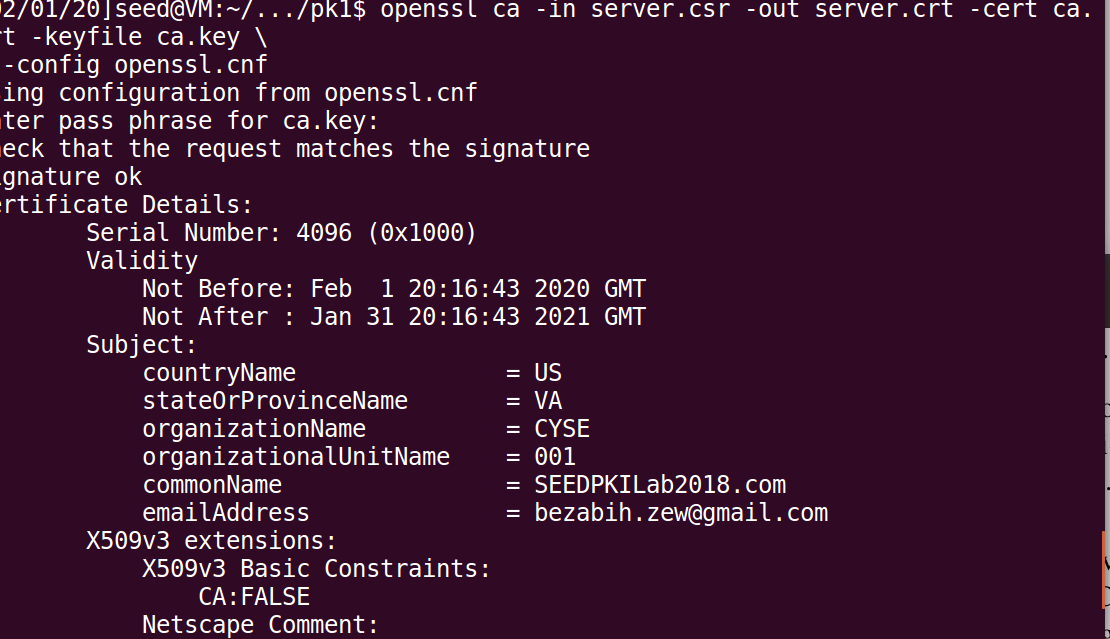
## Task 2.1

Log in to the web application as an administrator knowing only the username



The admin page looks like this.



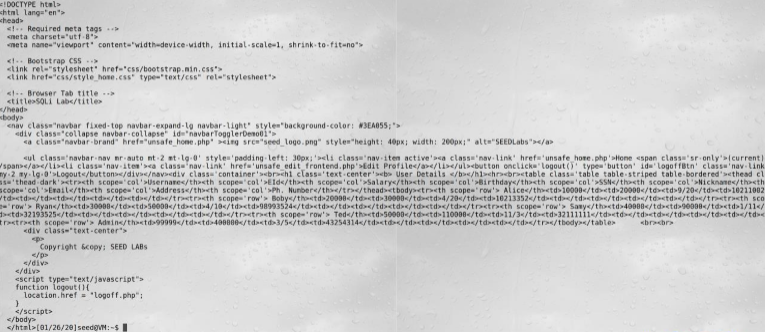


## Task 2.2

Repeat the above task using command line tools like curl. We can send HTTP request using the command line using curl.

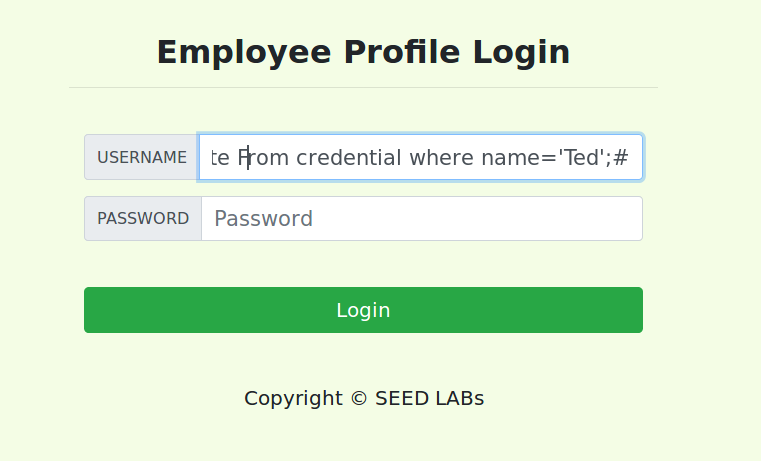


We get the output

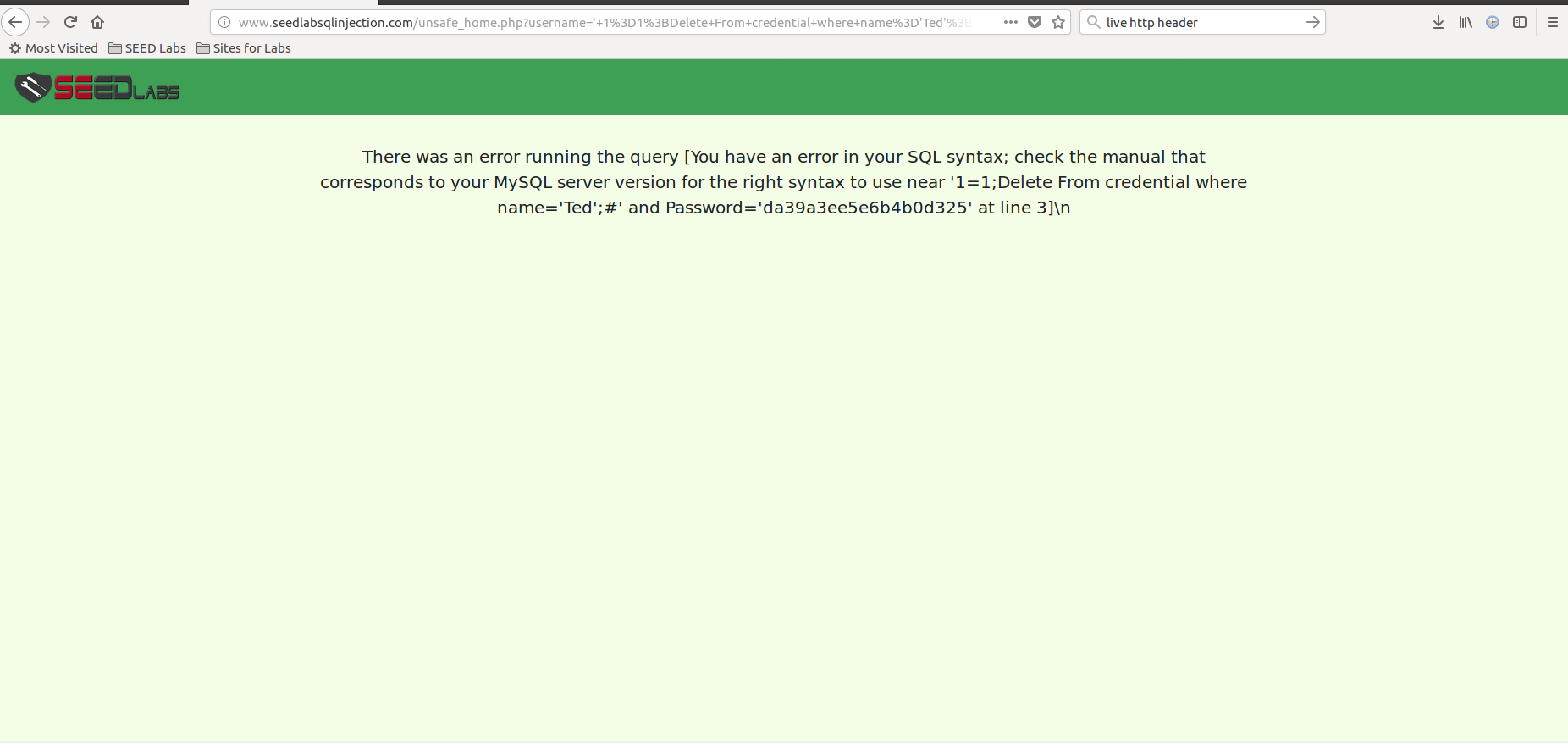


## Task 2.3

Append a new SQL statement and make the application execute two queries.

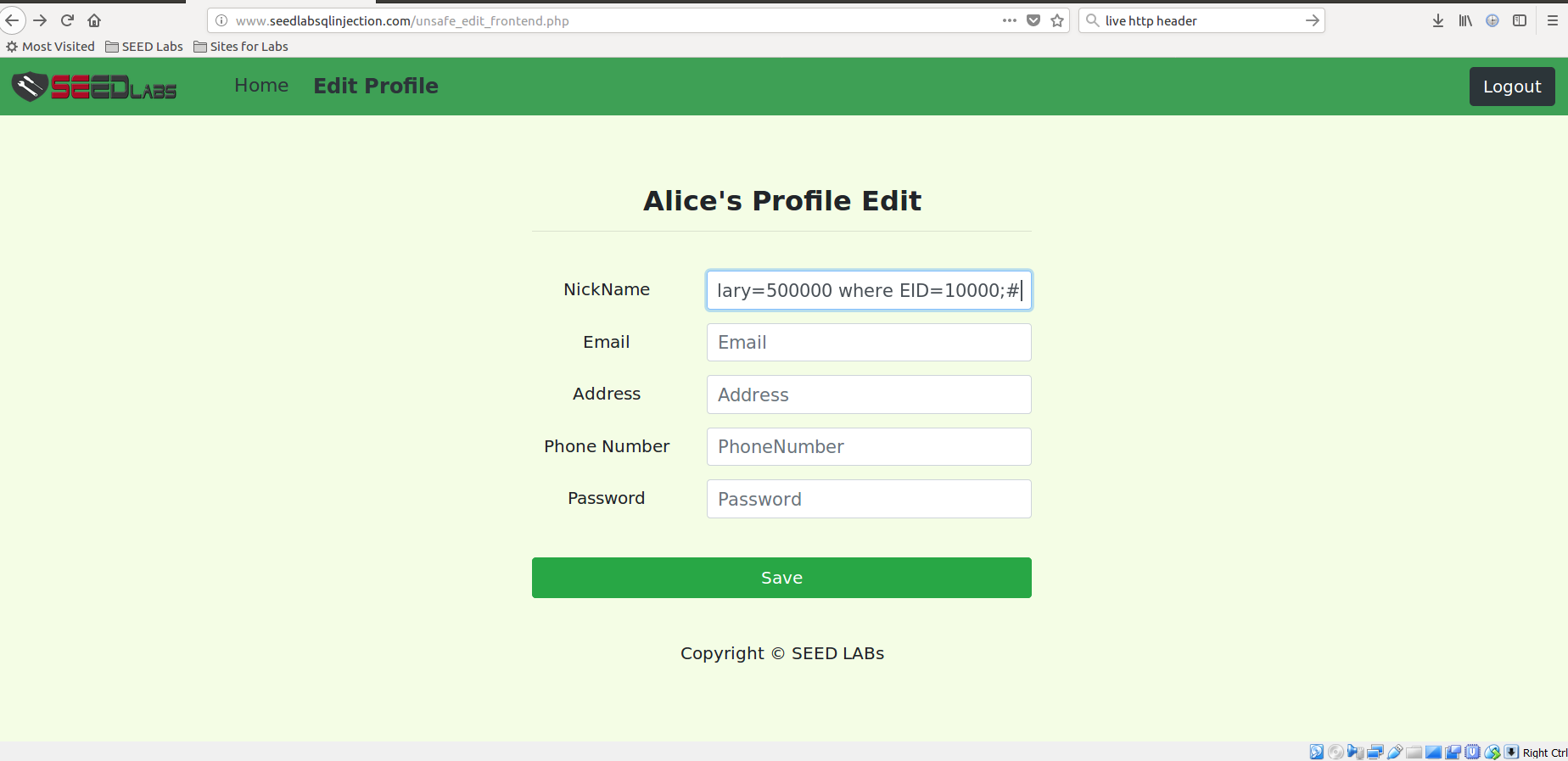


But it did not work because only execute one query as a security measurement.

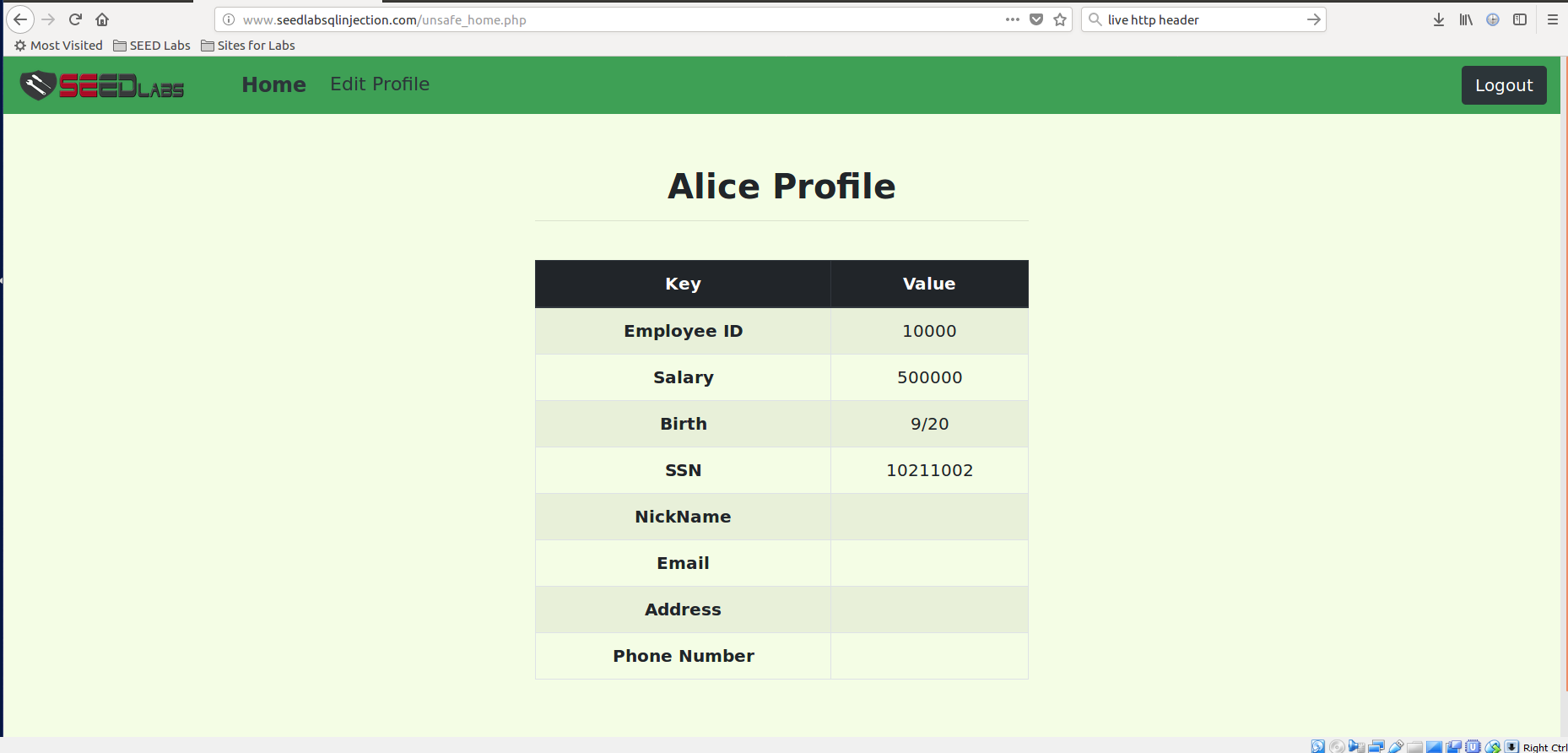


## Task 3.1

Modify the salary of Alice.

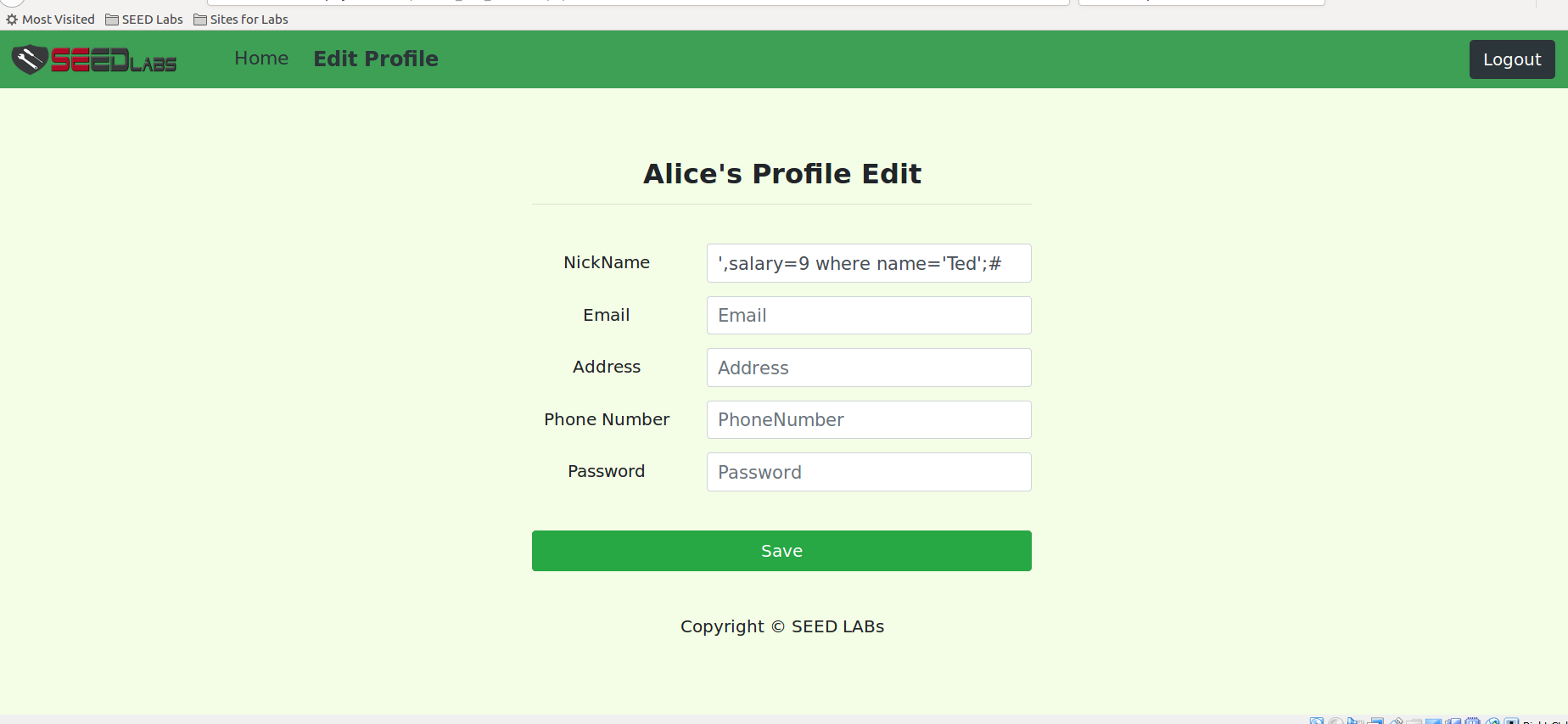


We can see the change.

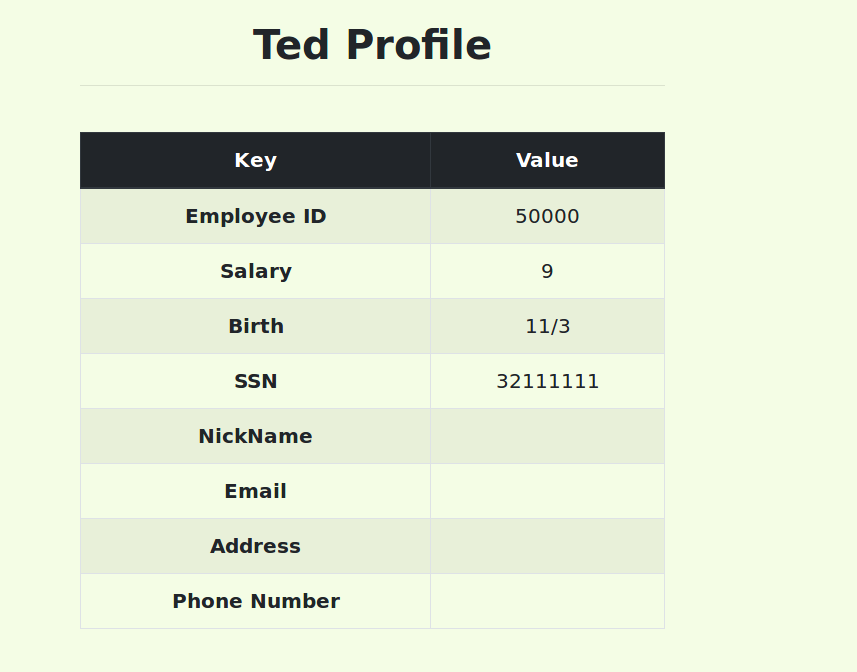


## Task 3.2 Modify Ted’s salary

Here we try changing Ted’s salary from allice’s account.



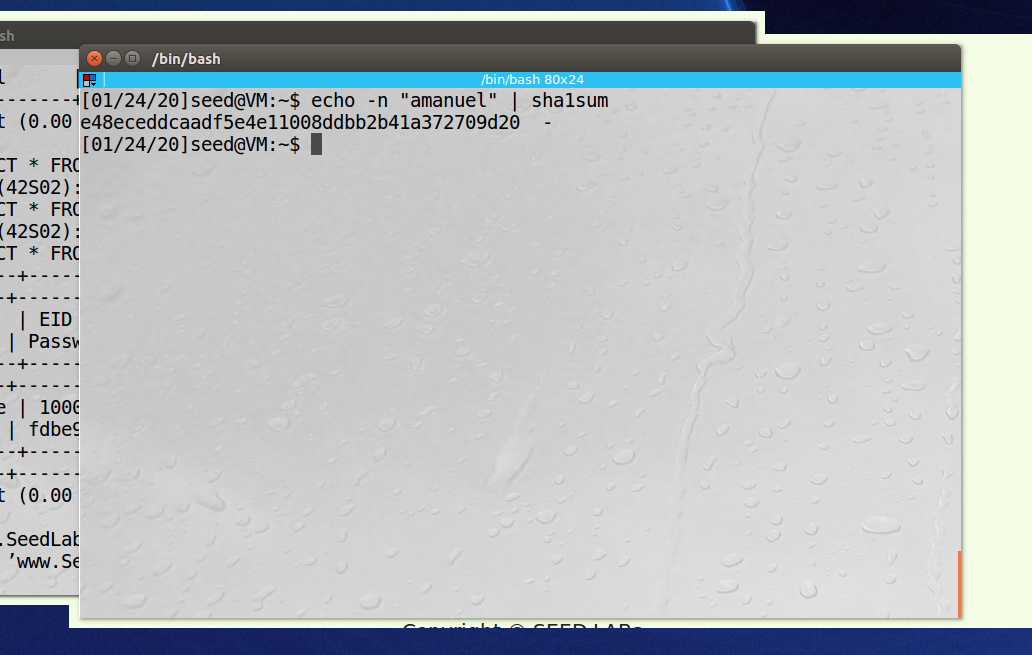
We can see from that ted’s salary is changed into 9.

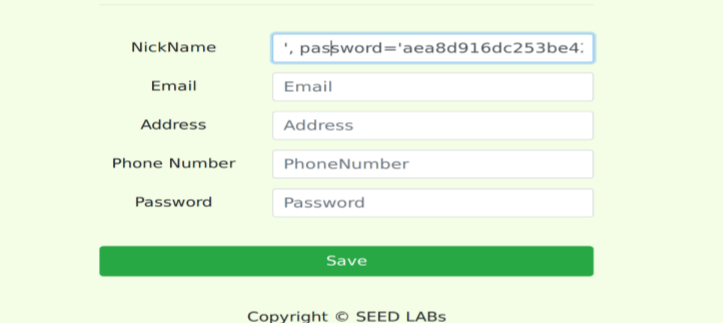


## Task 3.3 Change other members password

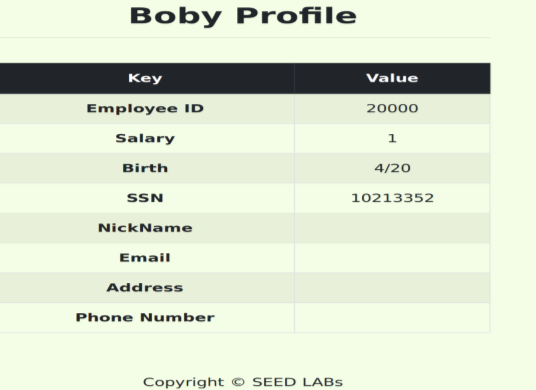
We change the password of bobby.

First generate a hash from password selected.





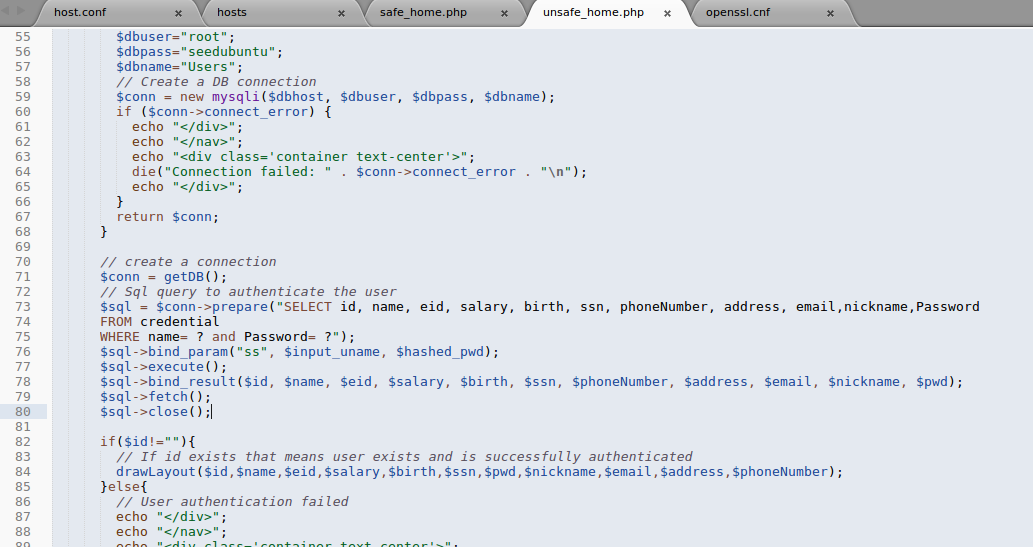
We can now login using our password



## Task 4: countermeasure

Change the code that is used to login users into the application by the following code.

Using the prepared statement mechanism, we divide the process of sending a SQL statement to the database into two steps.



Now we can’t use SQL attack.

