**Exercise 06**

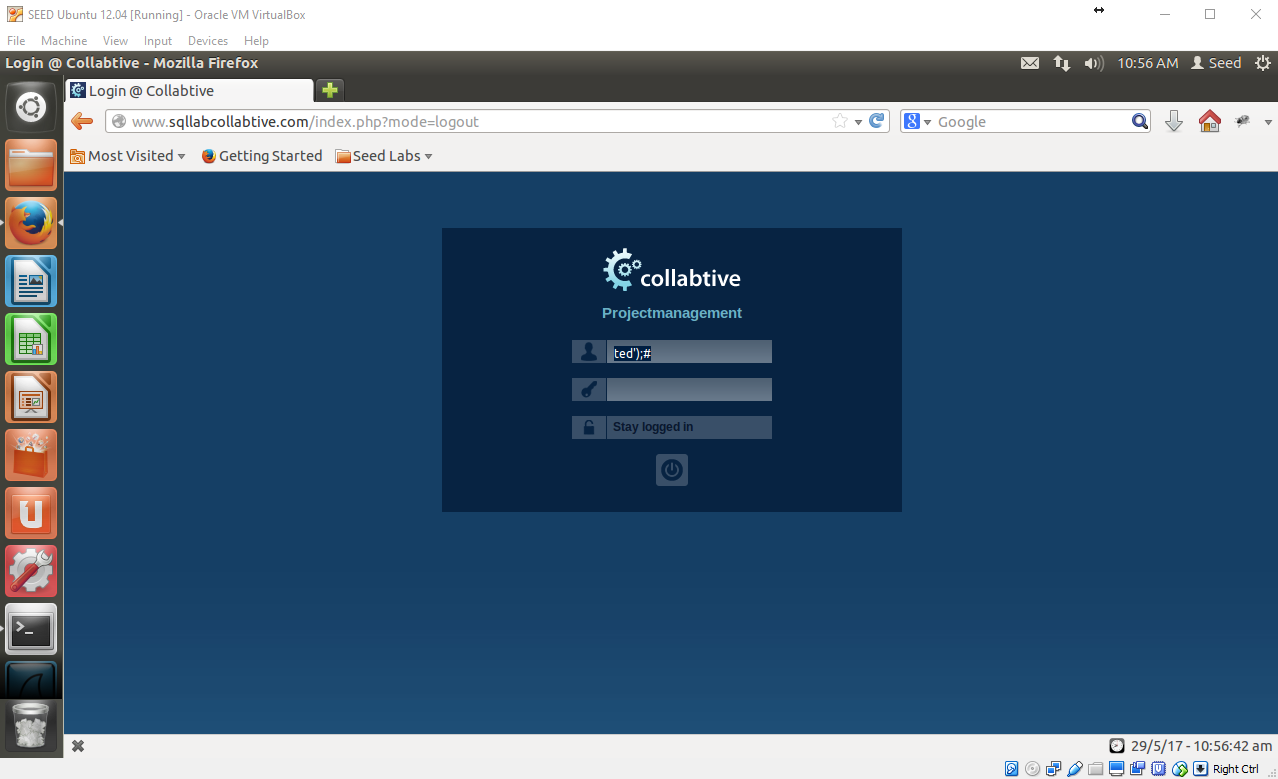
**Name:** Jose Juan Sandoval

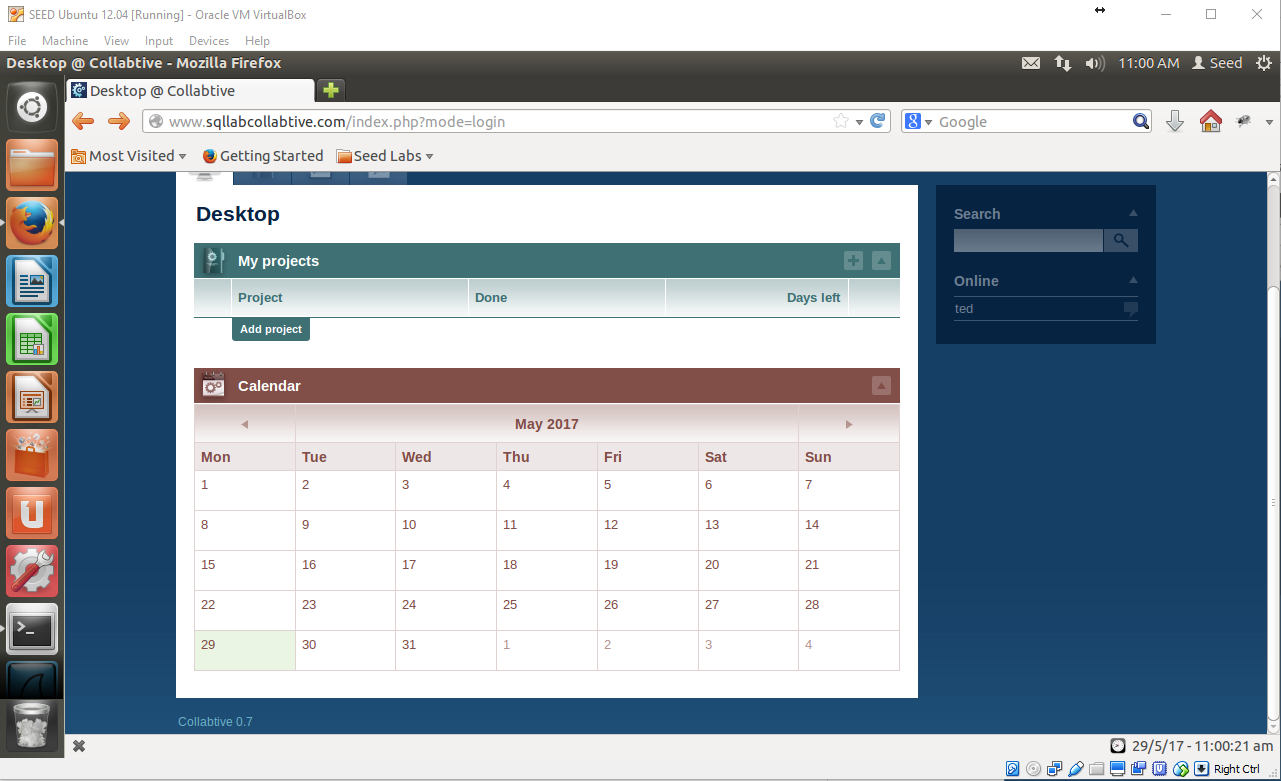
**Link to Project:** <https://github.com/Juanchiselo/CS380/tree/master/Exercises/Exercise%2006>

**Your first goal is to login as user ted without providing his password. Document your process for doing this and provide screenshots of the login screen with your input along with the web page after logging in.**

**Logged in as the User Ted:**

I was able to log in as the user Ted by entering his username followed by a quote to terminate the username string, followed by a parenthesis to close the WHERE clause, followed by a semicolon to end the SQL statement and finally a pound sign to comment out the rest of the original SQL statement.

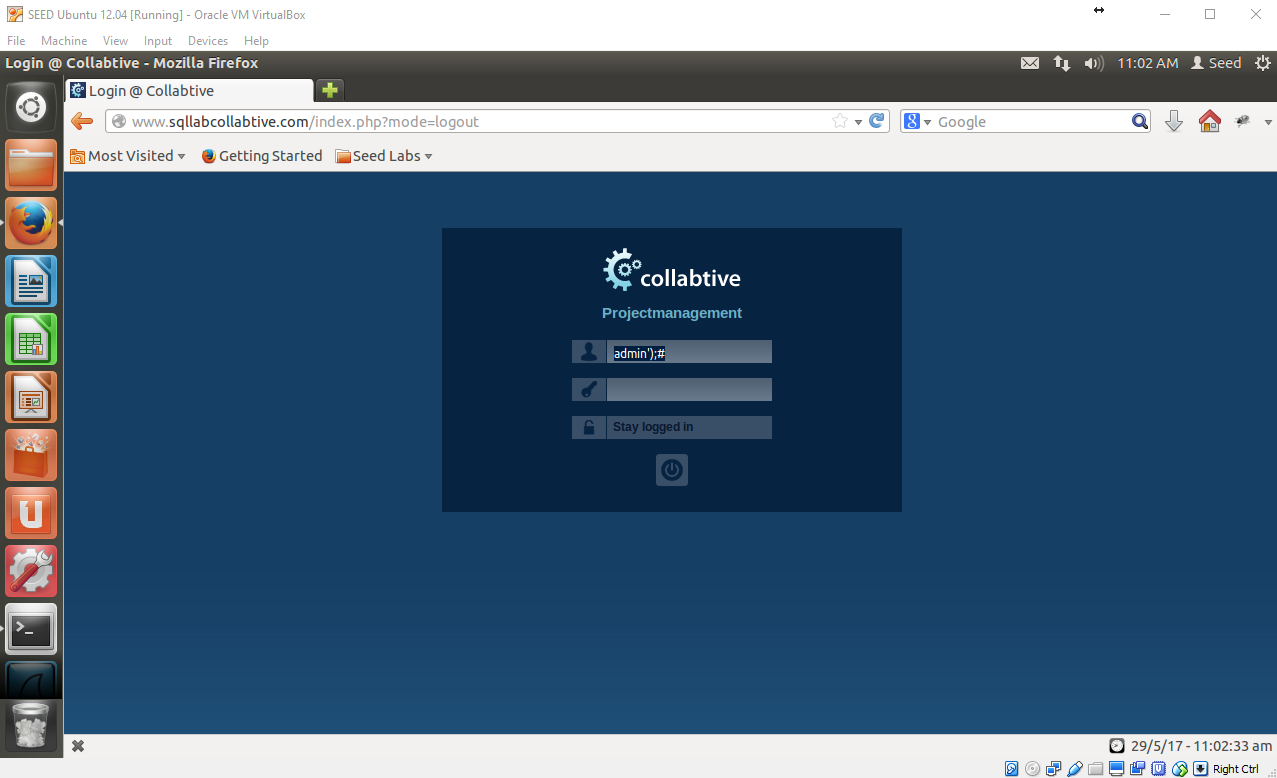


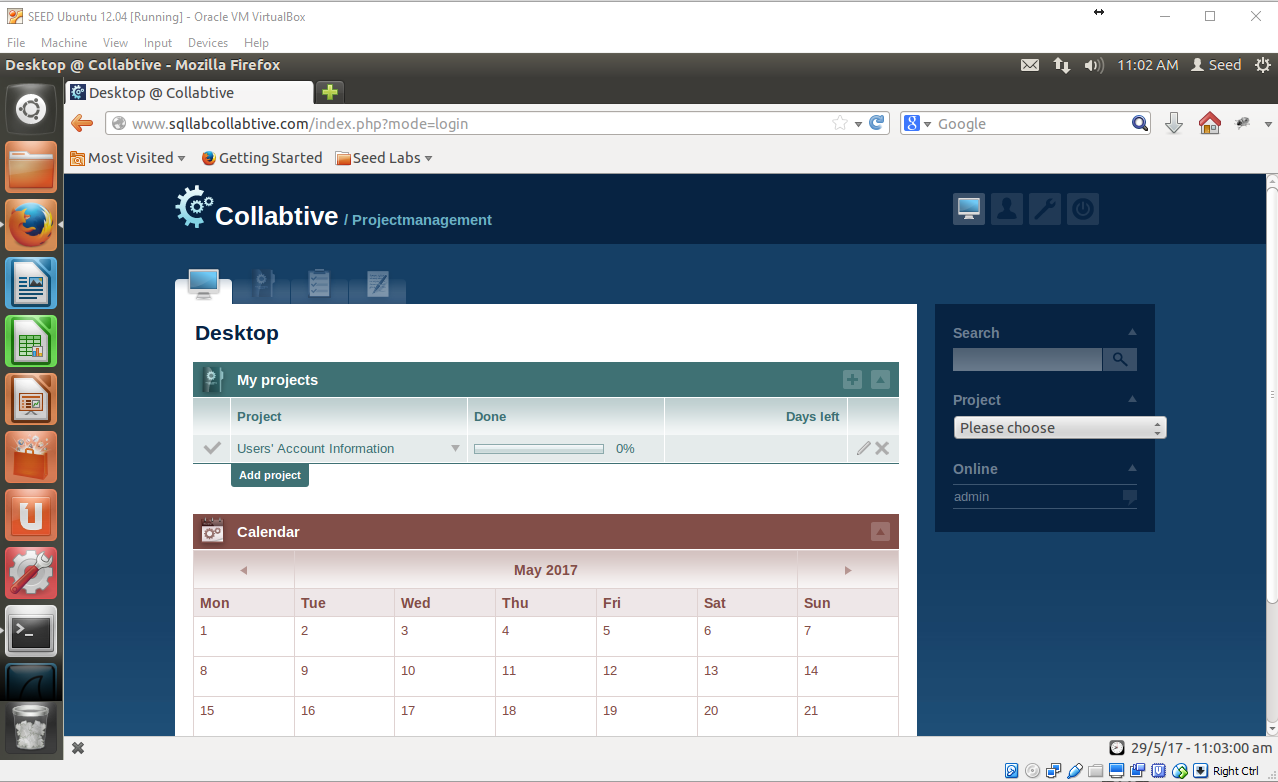


**Once you can login as ted, try the same process for the admin user. Can you login as the administrator**

**without providing their password? Document the results and provide screenshots.**

Yes, I was able to login as the administrator without providing his password. I used the same syntax for the SQL injection as I used to log in with Ted’s account.

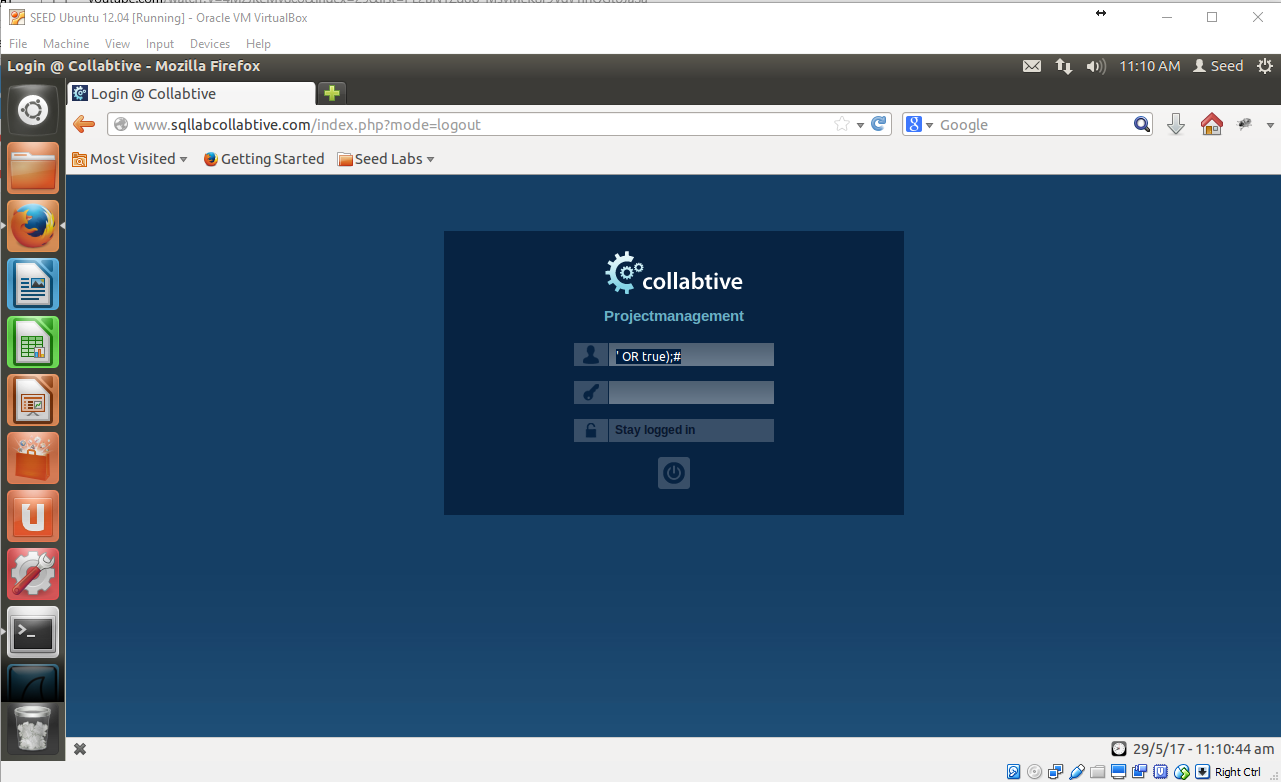


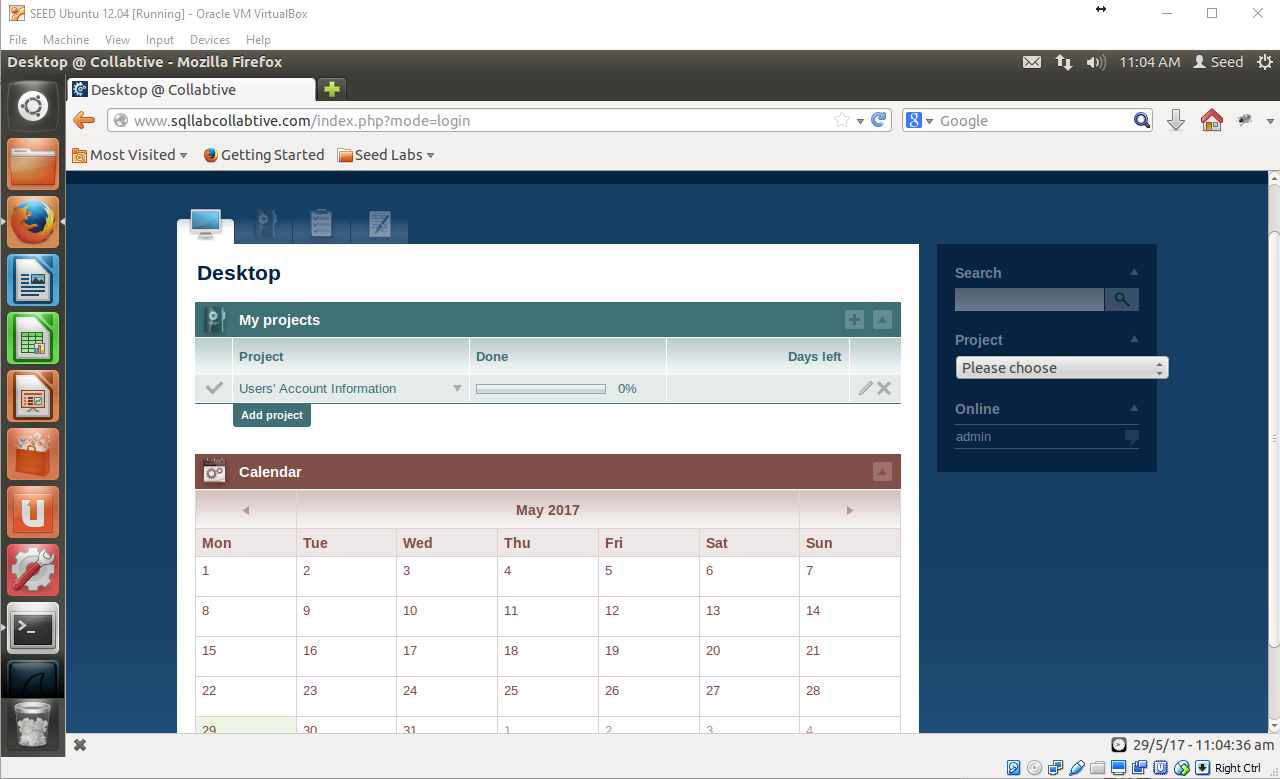


**Finally, can you find a way to login with a blank username (i.e., the user name in the query is considered blank but the user field in the login form is not necessarily blank)? Document how you are able to do this. What happens when you login this way? Which user does the website think you are? Document the results and provide screenshots.**

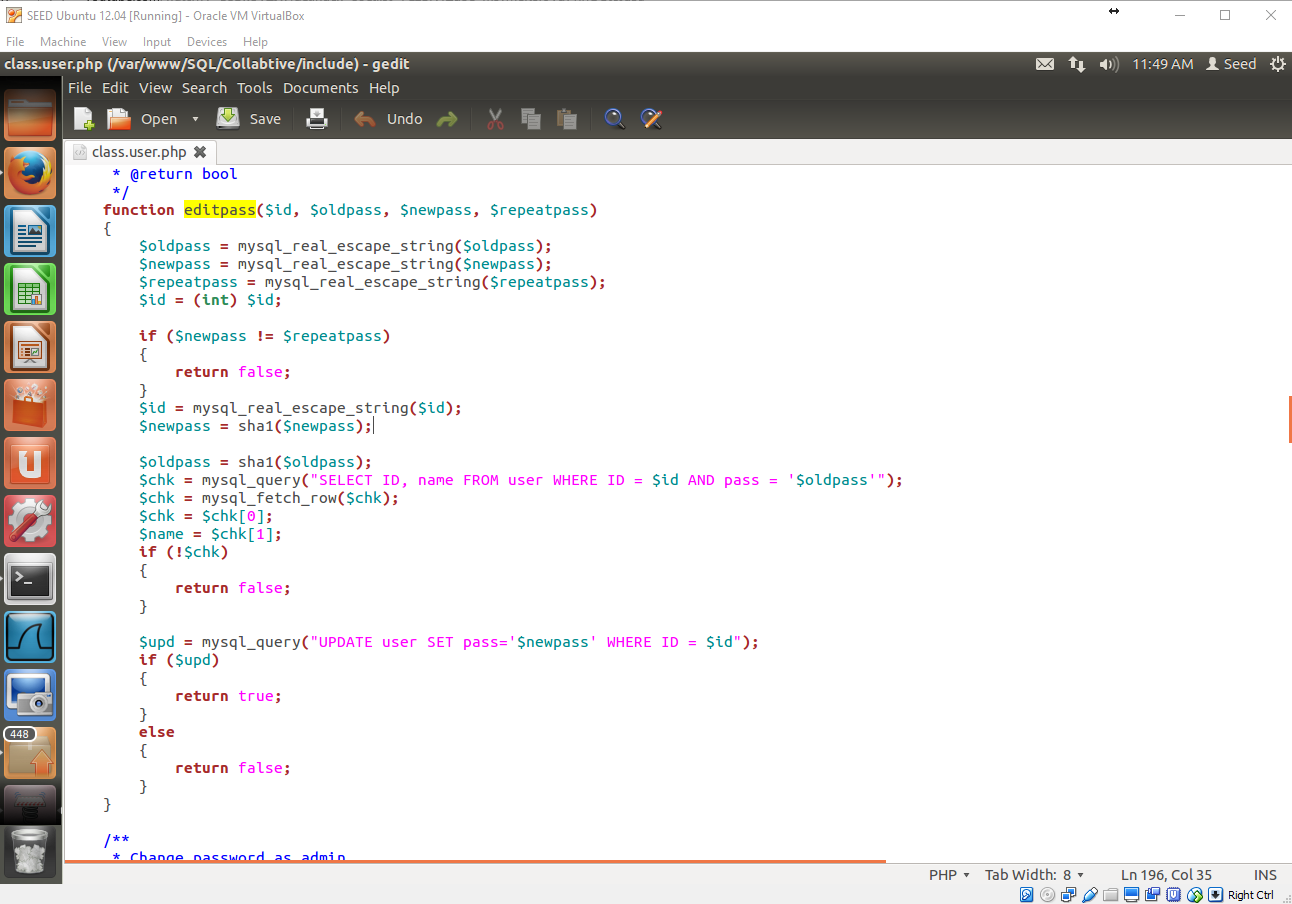
Yes, I was able to login with a blank username by starting with a quote to terminate the username string leaving it empty and because this will make the WHERE clause fail, I also appended an OR operator followed by a true Boolean so the WHERE clause always evaluates to true. Then I closed the WHERE clause, terminated the SQL statement and commented out the rest of the original SQL statement.

When I login this way, the website thinks I am the administrator.





**In the web application a user can modify their profile by logging in then clicking My Account, then Edit. This page allows the user to fill in the information then some PHP code in include/class.user.php will put the information in to an SQL UPDATE statement. This UPDATE statement also has an SQL injection vulnerability. Look in the PHP code for how it constructs the UPDATE statement to determine how to perform the SQL injection.**



**Your job here is to log in as ted and use this page to change the password of alice. Document how you do this and provide screenshots. Show that you have logged in as alice with the new password.**

I could not get this to work but the line below is the one that I thought would work. The problem is that $newpass gets hashed so I can’t put the line on the New Password field of the form because then the whole line will get hashed and won’t execute on the UPDATE statement.

**newpass’ WHERE ID = (SELECT ID FROM user WHERE name=’alice’;);#**

I’ll try to crack it before the quarter ends but since I’m already late with this exercise I’ll just turn it in without the extra credit.