

Roll No: 23CSM2R20
Assignment 8 - SQL Injection

Name: Sadam Hussain Ganie
Subject: WDS Lab

The screenshot shows a web browser window with the URL `localhost/index.php`. The page has a navigation bar with links: Home, Sign up, Login, Insecure Login, and About us. Below the navigation bar, there is a registration form with fields for Full name, Email, Address, City, State, Zip, and Password. The 'Login' and 'Insecure Login' links are highlighted with red arrows. The 'Secure login' label is placed over the 'Login' link, and the 'Insecure login' label is placed over the 'Insecure Login' link.

Now user with email "lab8@nitw.in" exists in database as shown below:

```
mysql> select * from users;
+----+-----+-----+-----+-----+-----+
| name | email | password | address | city | state | zip |
+----+-----+-----+-----+-----+-----+
| fgggg | egerge@ghrth.hg | $2y$10$/LXhtTnKxNAHRBs8BLIudeFr0vxxekQguB8uoJYnspUjeFB5VmsWG | j | j | Karnataka | 67567 |
| Lab 8 | lab8@nitw.ac.in | $2y$10$Ht90PLD52ZZ5kToVzLdkueT8Su8AdZkVgkC1PwInbViev08Ca3oYK | Nothing. | Test | Telengana | 123454 |
+----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> insert into users values('Sadam', 'lab8@nitw.in', '123456', 'nothing', 'nothing', 'Karnataka', 8768);
Query OK, 1 row affected (0.08 sec)

mysql> select * from users;
+----+-----+-----+-----+-----+-----+
| name | email | password | address | city | state | zip |
+----+-----+-----+-----+-----+-----+
| fgggg | egerge@ghrth.hg | $2y$10$/LXhtTnKxNAHRBs8BLIudeFr0vxxekQguB8uoJYnspUjeFB5VmsWG | j | j | Karnataka | 67567 |
| Lab 8 | lab8@nitw.ac.in | $2y$10$Ht90PLD52ZZ5kToVzLdkueT8Su8AdZkVgkC1PwInbViev08Ca3oYK | Nothing. | Test | Telengana | 123454 |
| Sadam | lab8@nitw.in | 123456 | nothing | nothing | Karnataka | 8768 |
+----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

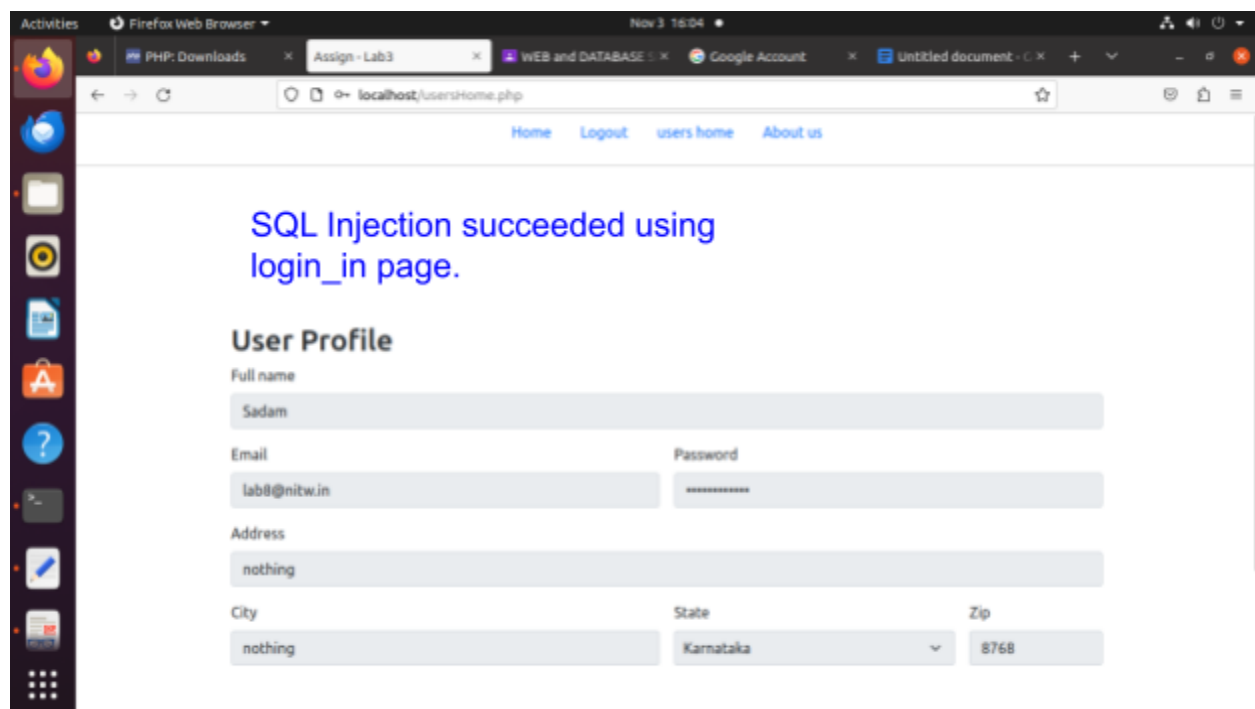
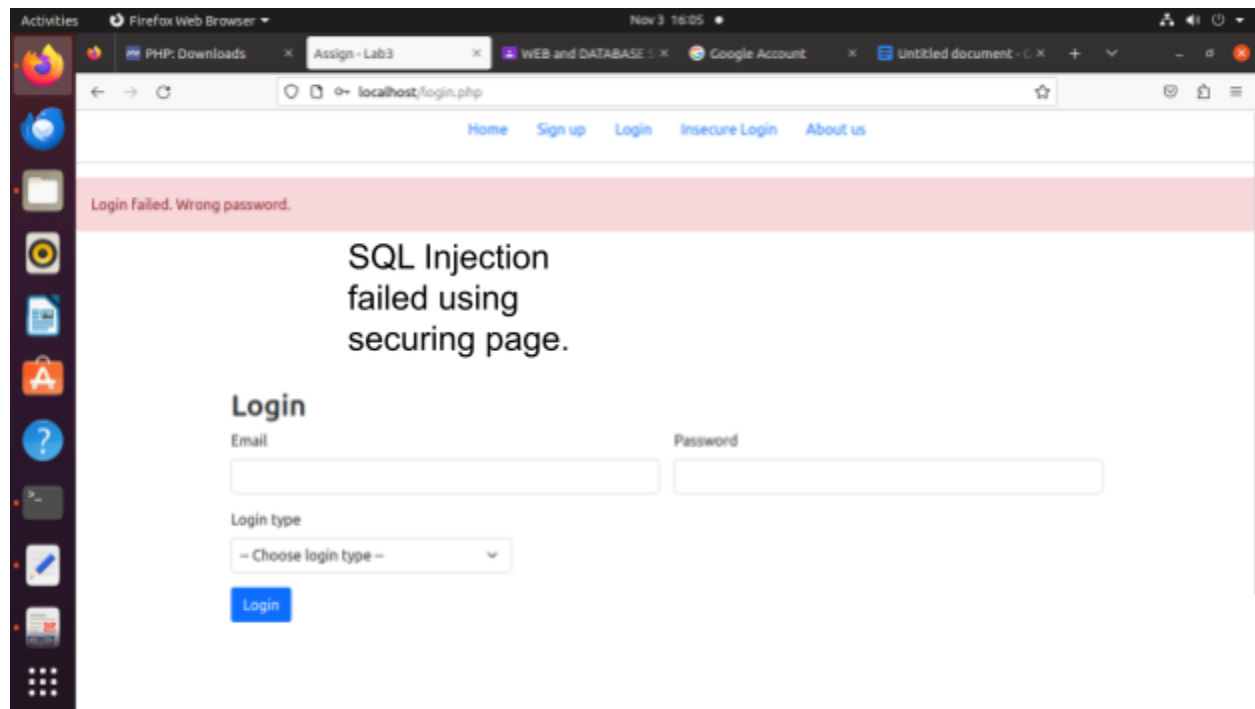
A red arrow points to the new user entry in the second query result. A text box with the text "User lab8@nitw.in" is placed next to the arrow.

Now let's try to login using the “**login.php**” page and “**login_in.php**”. (Here login.php is a secure page where SQL injection won't work, whereas login_in.php is an insecure page which can be easily attacked by sql injection).

Let us say we know the user email that is “lab8@nitw.in” but we don't know the password. Let us prepare a password(Injection query).

Let us try with email = lab8@nitw.ac.in and password = **abcd' or '1'='1**

Following is the output:



Why?

Insecure login page does not validate user input and executes sql query directly. In above example **abcd' or '1'='1** means: **'Select * from users where email=lab8@nitw.in && password=abcd' or '1'='1'** which will always be true irrespective of password supplied.

Incase of a secure login page, I have used prepared queries which verify the user input. This can be prevented from frontend as well using Javascript(input validation).