

# SQL-INJECTION STEPS:

## Step1:

Just inject the sql in Username (true ' or '1=1'#) in account.html  
.It will bypass the login

The image displays two screenshots of a web application interface, demonstrating a successful SQL injection attack on the login functionality.

**Top Screenshot (localhost/vulnerablesite/account.html):**

- The page features a green header with the "Donfort" logo and navigation links: HOME, PRODUCT, ABOUT, CONTACT, ACCOUNT.
- A central image shows various nuts and fruits in baskets and a sack.
- On the right, there is a login/register form. The "Login" tab is active.
- The Username field contains the injected payload: `true ' or '1=1'#`.
- The Password field is empty.
- A "Login" button is present.

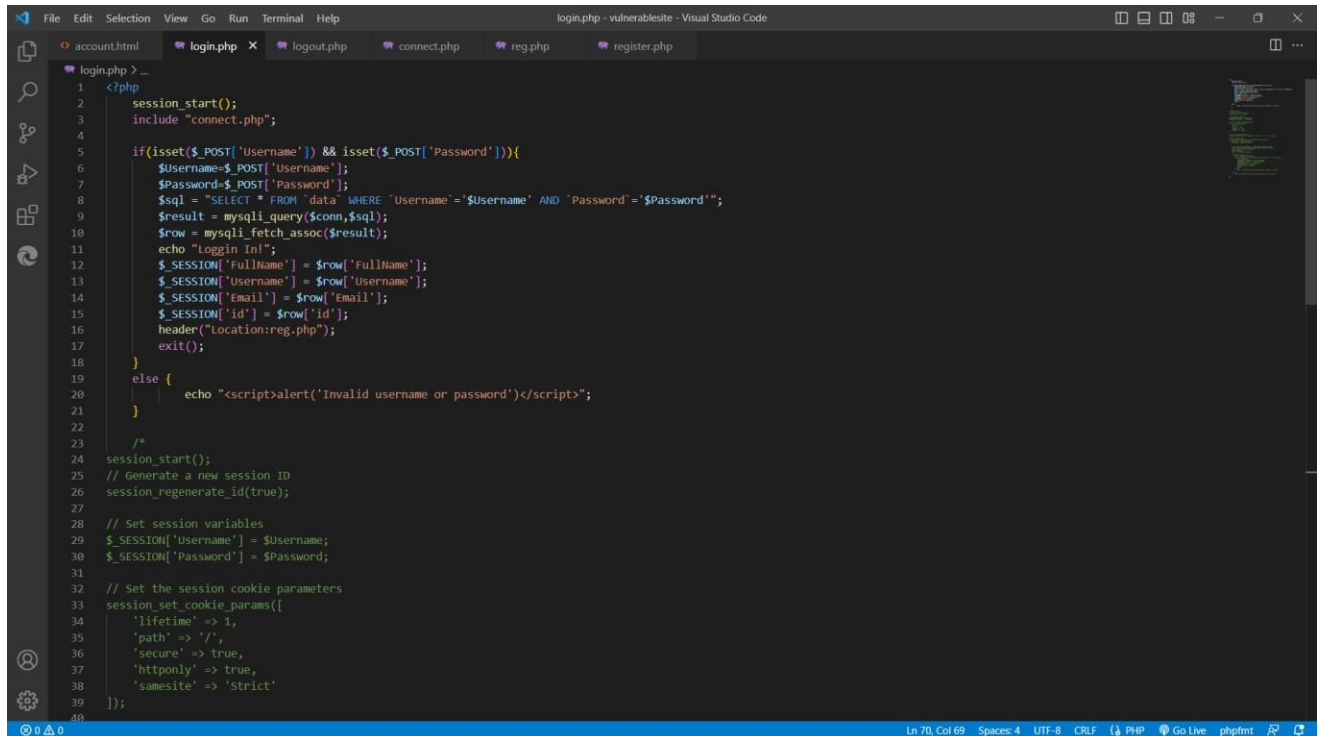
**Bottom Screenshot (localhost/vulnerablesite/reg.php):**

- The page layout is identical to the top screenshot.
- After the login attempt, the user is redirected to a page showing a successful login message: **HELLO, thulasi**.
- Below the message, user details are displayed:
  - NAME: thulasidpss
  - USERNAME: thulasi
  - ID: 13
  - EMAIL: thulasi@gmail.com
- A "LOGOUT" button is located below the user details.

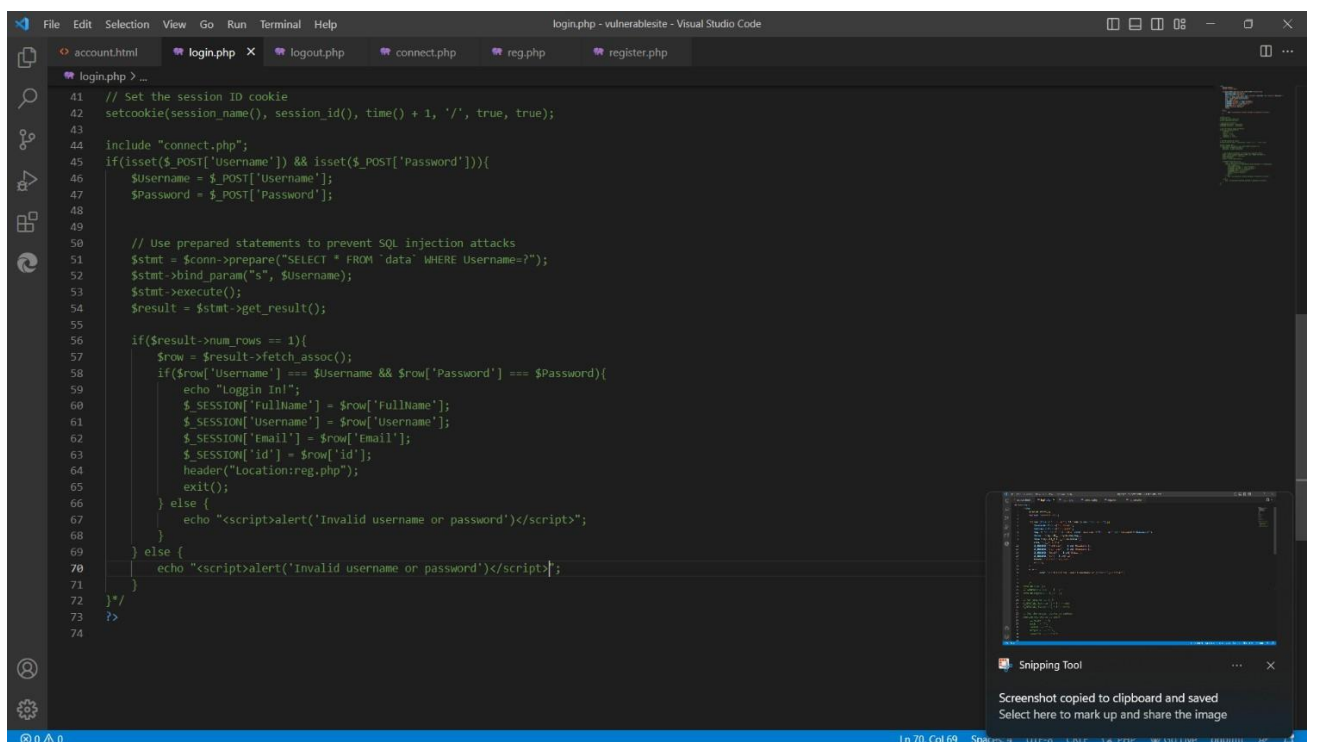
## Step2:

To prevent this just do commad the line from 2 to 21

And uncommand the line from 24 to 72 in login.php



```
1 <?php
2 session_start();
3 include "connect.php";
4
5 if(isset($_POST['Username']) && isset($_POST['Password'])){
6     $Username=$_POST['Username'];
7     $Password=$_POST['Password'];
8     $sql = "SELECT * FROM `data` WHERE `Username`='".$Username."' AND `Password`='".$Password.'";
9     $result = mysqli_query($conn,$sql);
10    $row = mysqli_fetch_assoc($result);
11    echo "Loggin In!";
12    $_SESSION['FullName'] = $row['FullName'];
13    $_SESSION['Username'] = $row['Username'];
14    $_SESSION['Email'] = $row['Email'];
15    $_SESSION['id'] = $row['id'];
16    header("Location:reg.php");
17    exit();
18 }
19 else {
20     echo "<script>alert('Invalid username or password')</script>";
21 }
22
23 /*
24 session_start();
25 // Generate a new session ID
26 session_regenerate_id(true);
27
28 // Set session variables
29 $_SESSION['Username'] = $Username;
30 $_SESSION['Password'] = $Password;
31
32 // Set the session cookie parameters
33 session_set_cookie_params([
34     'lifetime' => 1,
35     'path' => '/',
36     'secure' => true,
37     'httponly' => true,
38     'samesite' => 'Strict'
39 ]);
40
```



```
41 // Set the session ID cookie
42 setcookie(session_name(), session_id(), time() + 1, '/', true, true);
43
44 include "connect.php";
45 if(isset($_POST['Username']) && isset($_POST['Password'])){
46     $Username = $_POST['Username'];
47     $Password = $_POST['Password'];
48
49
50 // Use prepared statements to prevent SQL injection attacks
51 $stmt = $conn->prepare("SELECT * FROM `data` WHERE Username=?");
52 $stmt->bind_param("s", $Username);
53 $stmt->execute();
54 $result = $stmt->get_result();
55
56 if($result->num_rows == 1){
57     $row = $result->fetch_assoc();
58     if($row['Username'] === $Username && $row['Password'] === $Password){
59         echo "Loggin In!";
60         $_SESSION['FullName'] = $row['FullName'];
61         $_SESSION['Username'] = $row['Username'];
62         $_SESSION['Email'] = $row['Email'];
63         $_SESSION['id'] = $row['id'];
64         header("Location:reg.php");
65         exit();
66     } else {
67         echo "<script>alert('Invalid username or password')</script>";
68     }
69 } else {
70     echo "<script>alert('Invalid username or password')</script>";
71 }
72 }
73 ?>
74
```

**Step3:** now again inject the sql in Username (true ' or '1=1'#) in account.html .It will bypass the login

