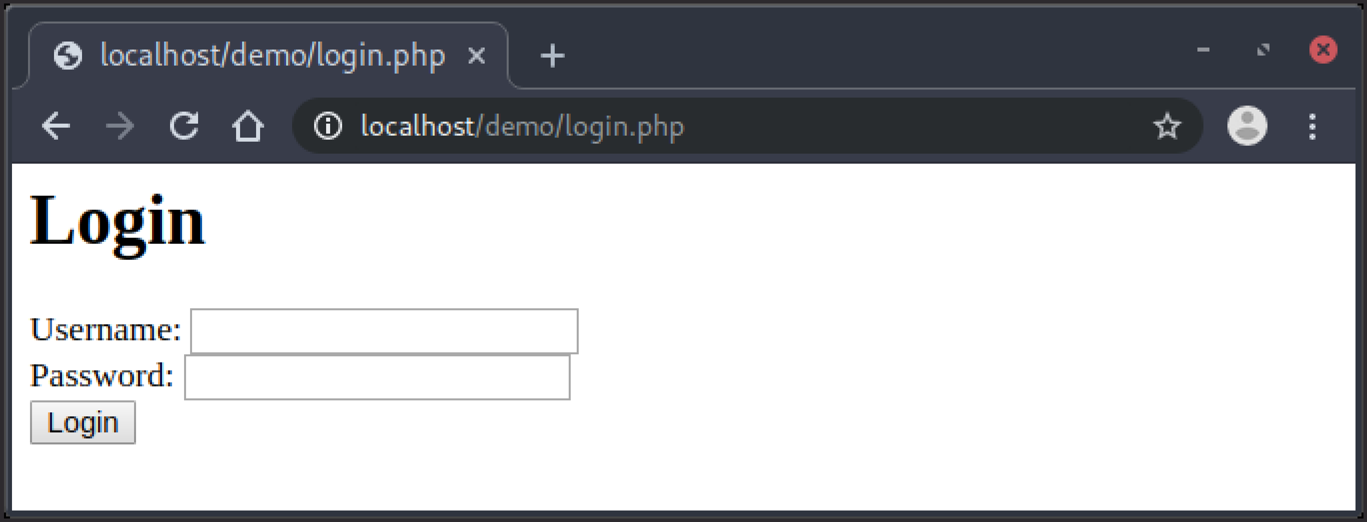
Abusing SQL “Features”

```
select * from slides where 1=0 and id='{$_GET['id']}';
```

```
1337' or 1=1; -- #
```

Ngo Wei Lin
(@Creastery)

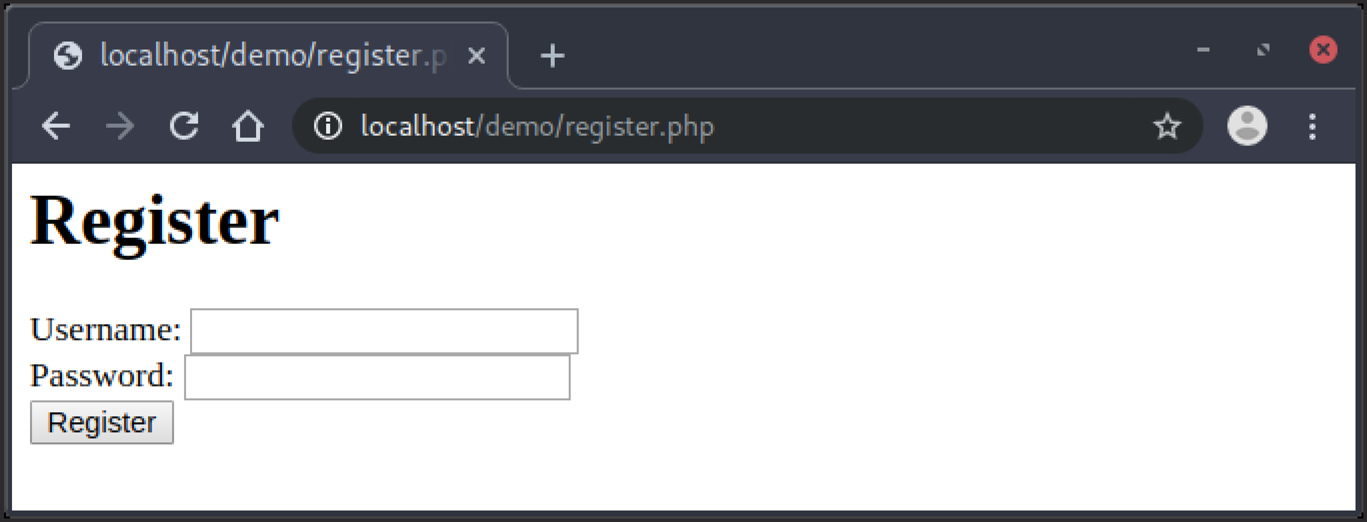
Let’s explore how to exploit this webapp.



Login

Username:

Password:

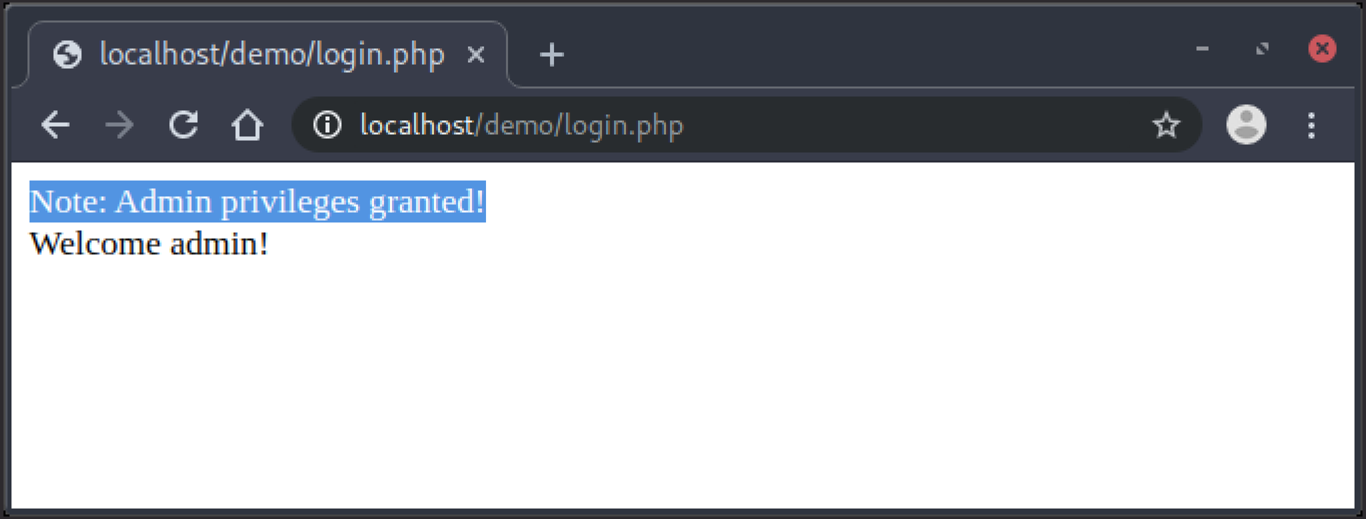


Register

Username:

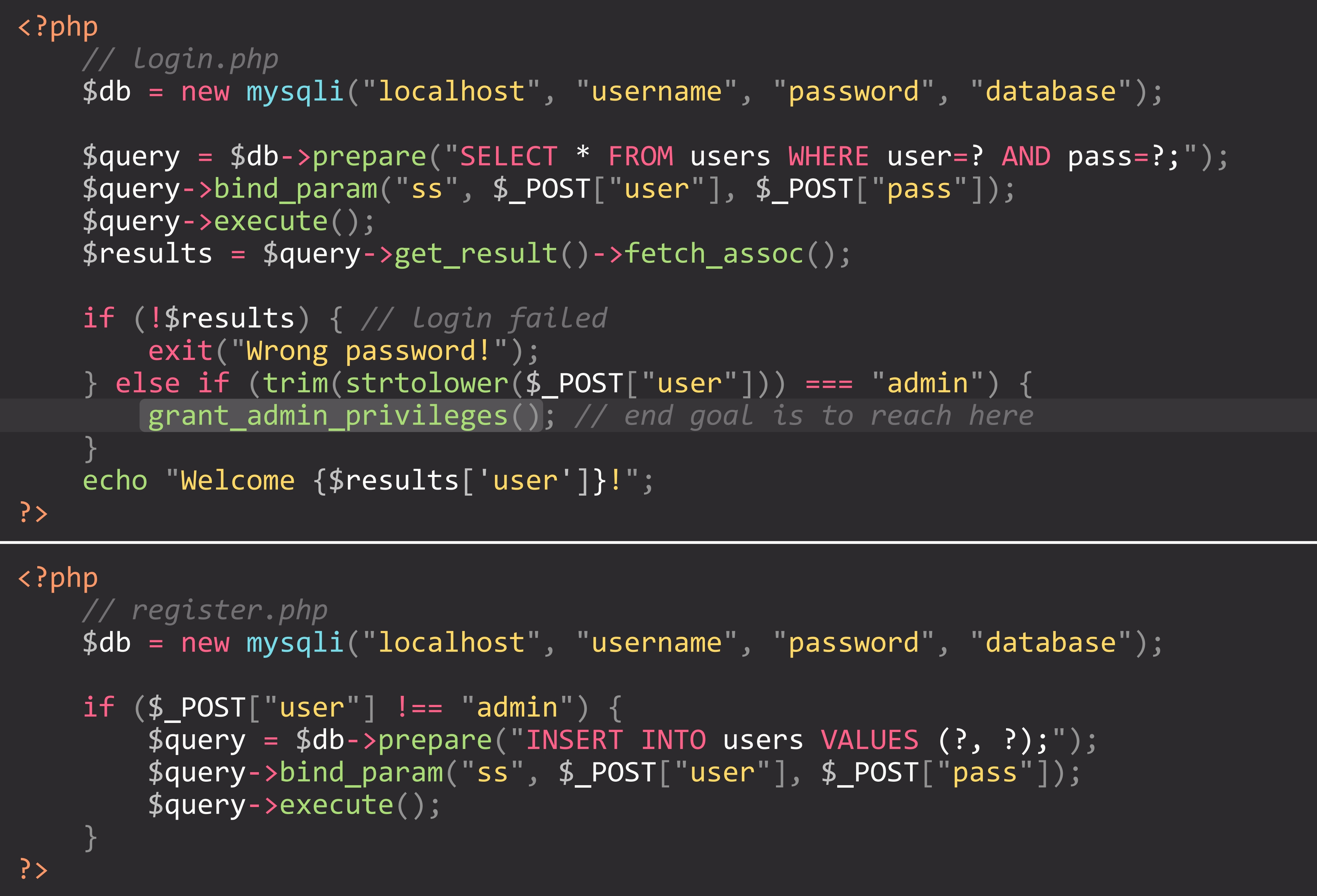
Password:

Goal: Get admin privileges



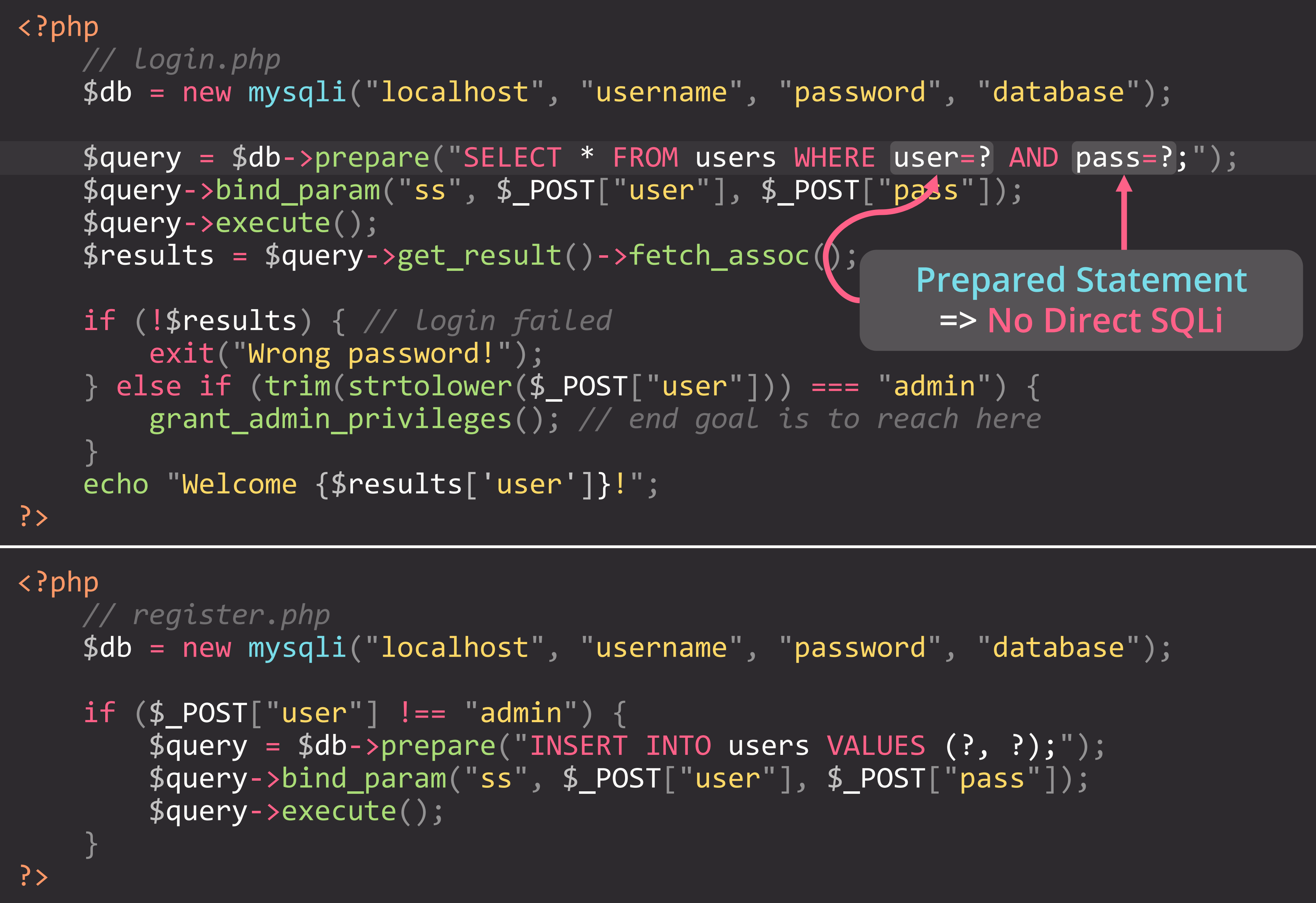
Note: Admin privileges granted!

Welcome admin!



```php
<?php
    // login.php
    $db = new mysqli("localhost", "username", "password", "database");

    $query = $db->prepare("SELECT * FROM users WHERE user=? AND pass=?;");
    $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
    $query->execute();
    $results = $query->get_result()->fetch_assoc();

    if (!$results) { // login failed
        exit("Wrong password!");
    } else if (trim(strtolower($_POST["user"])) === "admin") {
        grant_admin_privileges(); // end goal is to reach here
    }
    echo "Welcome {$results['user']}!";
?>
```

```php
<?php
    // register.php
    $db = new mysqli("localhost", "username", "password", "database");

    if ($_POST["user"] !== "admin") {
        $query = $db->prepare("INSERT INTO users VALUES (?, ?);");
        $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
        $query->execute();
    }
?>
```



```
<?php
    // login.php
    $db = new mysqli("localhost", "username", "password", "database");

    $query = $db->prepare("SELECT * FROM users WHERE user=? AND pass=?;");
    $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
    $query->execute();
    $results = $query->get_result()->fetch_assoc();

    if (!$results) { // login failed
        exit("Wrong password!");
    } else if (trim(strtolower($_POST["user"])) === "admin") {
        grant_admin_privileges(); // end goal is to reach here
    }
    echo "Welcome {$results['user']}!";
?>
```

Prepared Statement
=> No Direct SQLi

```
<?php
    // register.php
    $db = new mysqli("localhost", "username", "password", "database");

    if ($_POST["user"] !== "admin") {
        $query = $db->prepare("INSERT INTO users VALUES (?, ?);");
        $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
        $query->execute();
    }
?>
```

```php
<?php
    // login.php
    $db = new mysqli("localhost", "username", "password", "database");

    $query = $db->prepare("SELECT * FROM users WHERE user=? AND pass=?;");
    $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
    $query->execute();
    $results = $query->get_result()->fetch_assoc();

    if (!$results) { // login failed
        exit("Wrong password!");
    } else if (trim(strtolower($_POST["user"])) === "admin") {
        grant_admin_privileges(); // end goal is to reach here
    }
    echo "Welcome {$results['user']}!";
?>
```

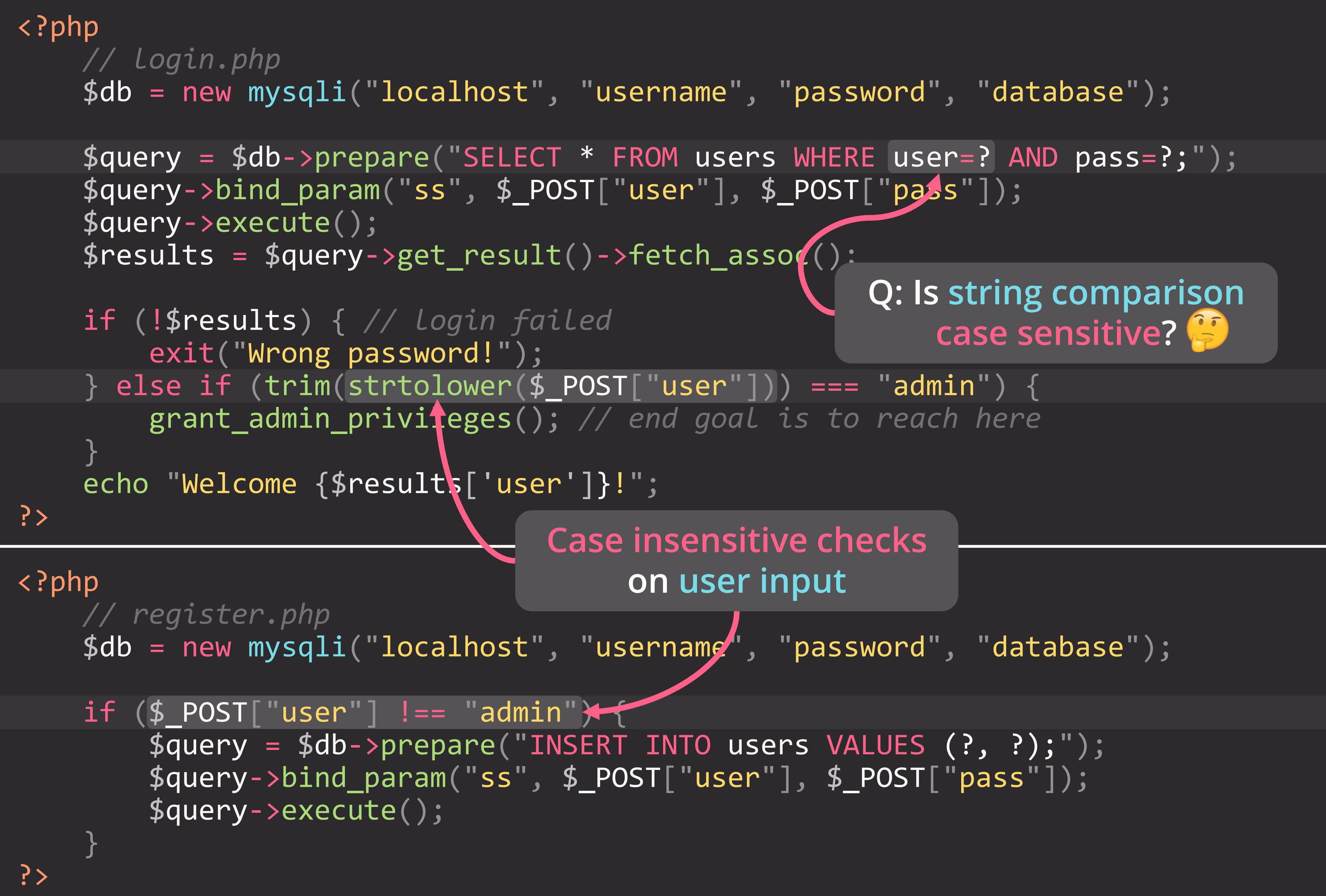
Case insensitive checks
on user input

```php
<?php
    // register.php
    $db = new mysqli("localhost", "username", "password", "database");

    if ($_POST["user"] !== "admin") {
        $query = $db->prepare("INSERT INTO users VALUES (?, ?);");
        $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
        $query->execute();
    }
?>
```

```php
<?php
    // login.php
    $db = new mysqli("localhost", "username", "password", "database");

    $query = $db->prepare("SELECT * FROM users WHERE user=? AND pass=?;");
    $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
    $query->execute();
    $results = $query->get_result()->fetch_assoc();

    if (!$results) { // login failed
        exit("Wrong password!");
    } else if (trim(strtolower($_POST["user"])) === "admin") {
        grant_admin_privileges(); // end goal is to reach here
    }
    echo "Welcome {$results['user']}!";
?>
```

```php
<?php
    // register.php
    $db = new mysqli("localhost", "username", "password", "database");

    if ($_POST["user"] !== "admin") {
        $query = $db->prepare("INSERT INTO users VALUES (?, ?);");
        $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
        $query->execute();
    }
?>
```

Case insensitive checks
on user input



```php
<?php
    // login.php
    $db = new mysqli("localhost", "username", "password", "database");

    $query = $db->prepare("SELECT * FROM users WHERE user=? AND pass=?;");
    $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
    $query->execute();
    $results = $query->get_result()->fetch_assoc();

    if (!$results) { // login failed
        exit("Wrong password!");
    } else if (trim(strtolower($_POST["user"])) === "admin") {
        grant_admin_privileges(); // end goal is to reach here
    }
    echo "Welcome {$results['user']}!";
?>
```

Q: Is string comparison case sensitive? 🤔

Case insensitive checks on user input

```php
<?php
    // register.php
    $db = new mysqli("localhost", "username", "password", "database");

    if ($_POST["user"] !== "admin") {
        $query = $db->prepare("INSERT INTO users VALUES (?, ?);");
        $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
        $query->execute();
    }
?>
```

String Comparisons

How does DBMS handles string comparisons?

Case Sensitivity

> "By default, string comparisons are not case-sensitive..."
>
> — *MySQL Reference Manual*
>
> *(https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html)*

Case Sensitivity

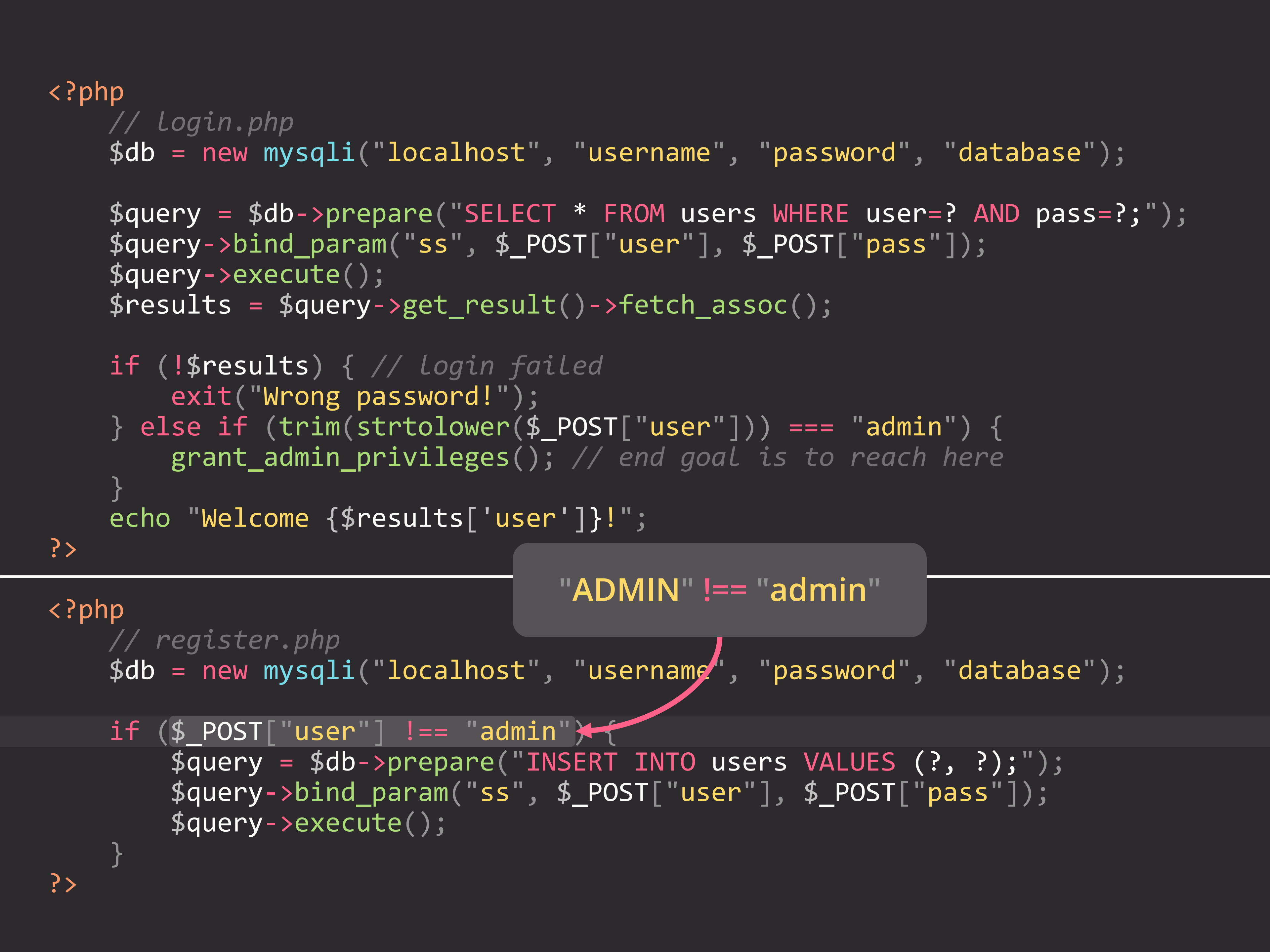
Likewise for MSSQL and SQLite...
*(not true for all DBMS)

Case Sensitivity

How do we use this “feature”?

Case Sensitivity

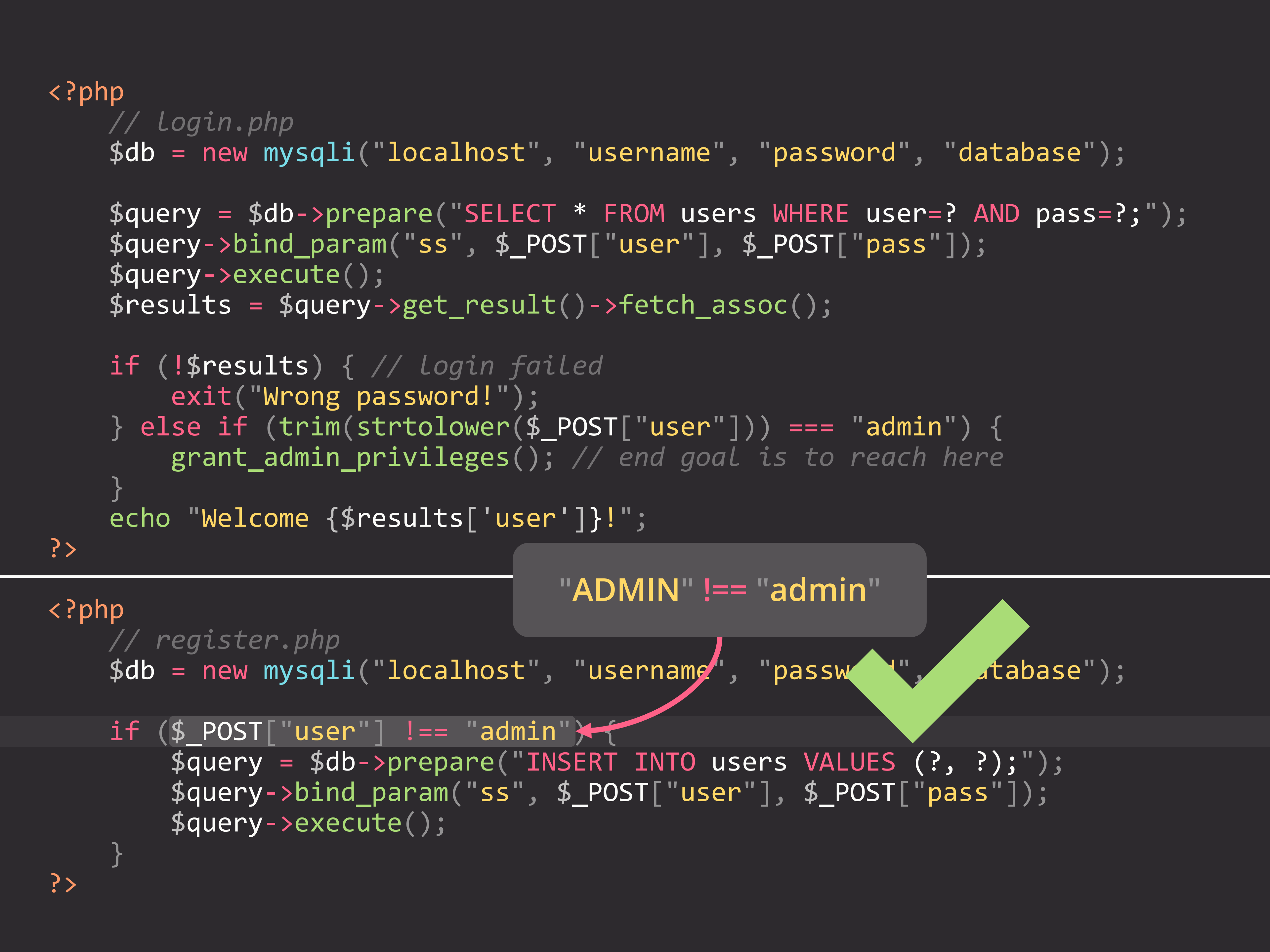
Step 1: Register as "ADMIN"



```php
<?php
    // login.php
    $db = new mysqli("localhost", "username", "password", "database");

    $query = $db->prepare("SELECT * FROM users WHERE user=? AND pass=?;");
    $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
    $query->execute();
    $results = $query->get_result()->fetch_assoc();

    if (!$results) { // login failed
        exit("Wrong password!");
    } else if (trim(strtolower($_POST["user"])) === "admin") {
        grant_admin_privileges(); // end goal is to reach here
    }
    echo "Welcome {$results['user']}!";
?>
```

```php
<?php
    // register.php
    $db = new mysqli("localhost", "username", "password", "database");

    if ($_POST["user"] !== "admin") {
        $query = $db->prepare("INSERT INTO users VALUES (?, ?);");
        $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
        $query->execute();
    }
?>
```

"ADMIN" !== "admin"



```php
<?php
    // login.php
    $db = new mysqli("localhost", "username", "password", "database");

    $query = $db->prepare("SELECT * FROM users WHERE user=? AND pass=?;");
    $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
    $query->execute();
    $results = $query->get_result()->fetch_assoc();

    if (!$results) { // login failed
        exit("Wrong password!");
    } else if (trim(strtolower($_POST["user"])) === "admin") {
        grant_admin_privileges(); // end goal is to reach here
    }
    echo "Welcome {$results['user']}!";
?>
```

"ADMIN" !== "admin"

```php
<?php
    // register.php
    $db = new mysqli("localhost", "username", "password", "database");

    if ($_POST["user"] !== "admin") {
        $query = $db->prepare("INSERT INTO users VALUES (?, ?);");
        $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
        $query->execute();
    }
?>
```

Case Sensitivity

Step 1: Register as "ADMIN"
Step 2: Log in as "ADMIN"

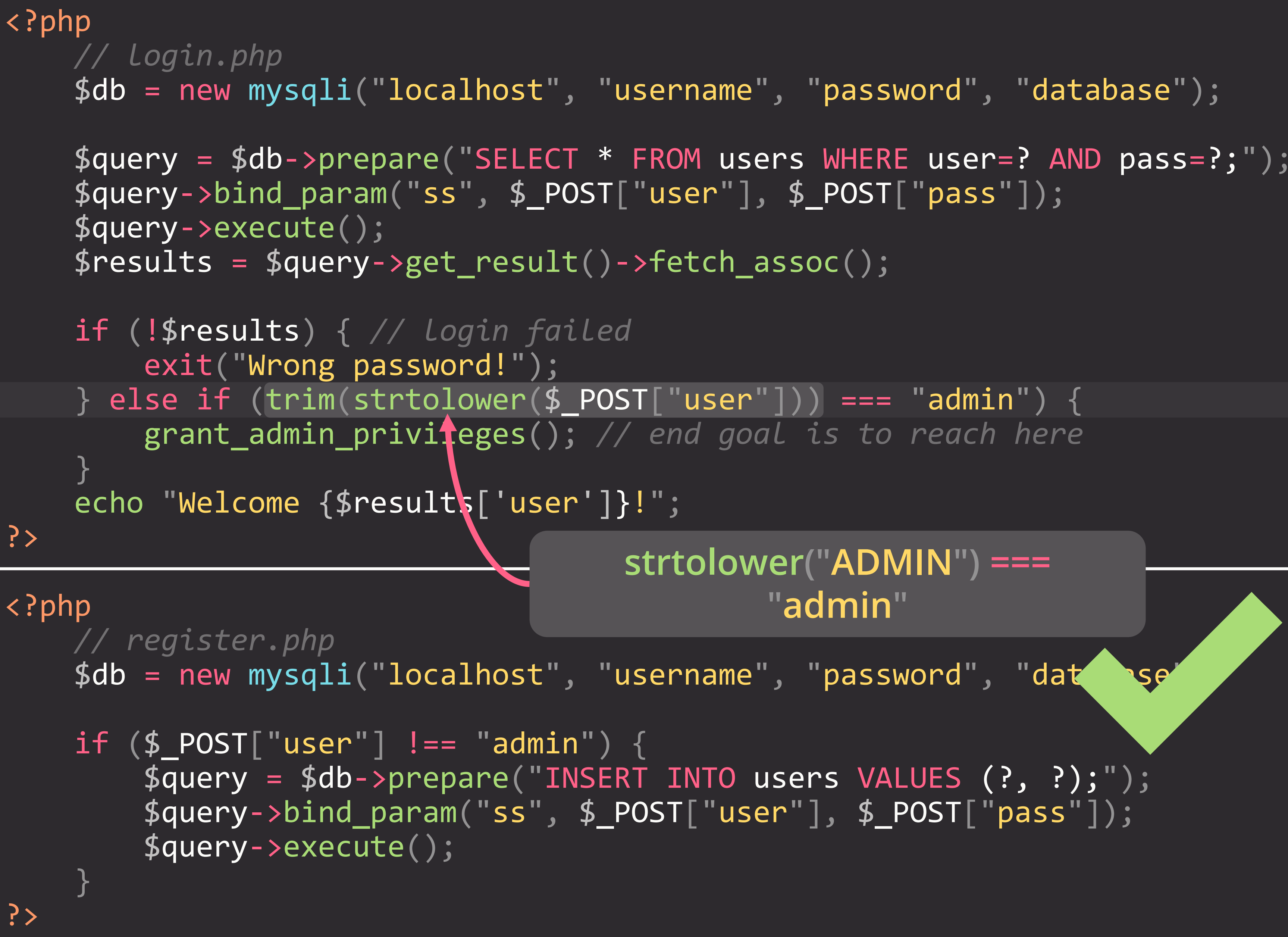
```php
<?php
    // login.php
    $db = new mysqli("localhost", "username", "password", "database");

    $query = $db->prepare("SELECT * FROM users WHERE user=? AND pass=?;");
    $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
    $query->execute();
    $results = $query->get_result()->fetch_assoc();

    if (!$results) { // login failed
        exit("Wrong password!");
    } else if (trim(strtolower($_POST["user"])) === "admin") {
        grant_admin_privileges(); // end goal is to reach here
    }
    echo "Welcome {$results['user']}!";
?>

<?php
    // register.php
    $db = new mysqli("localhost", "username", "password", "database");

    if ($_POST["user"] !== "admin") {
        $query = $db->prepare("INSERT INTO users VALUES (?, ?);");
        $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
        $query->execute();
    }
?>
```

Select "ADMIN" user

```php
<?php
    // login.php
    $db = new mysqli("localhost", "username", "password", "database");

    $query = $db->prepare("SELECT * FROM users WHERE user=? AND pass=?;");
    $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
    $query->execute();
    $results = $query->get_result()->fetch_assoc();

    if (!$results) { // login failed
        exit("Wrong password!");
    } else if (trim(strtolower($_POST["user"])) === "admin") {
        grant_admin_privileges(); // end goal is to reach here
    }
    echo "Welcome {$results['user']}!";
?>
```

strtolower("ADMIN") === "admin"

```php
<?php
    // register.php
    $db = new mysqli("localhost", "username", "password", "database");

    if ($_POST["user"] !== "admin") {
        $query = $db->prepare("INSERT INTO users VALUES (?, ?);");
        $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
        $query->execute();
    }
?>
```



```php
<?php
    // login.php
    $db = new mysqli("localhost", "username", "password", "database");

    $query = $db->prepare("SELECT * FROM users WHERE user=? AND pass=?;");
    $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
    $query->execute();
    $results = $query->get_result()->fetch_assoc();

    if (!$results) { // login failed
        exit("Wrong password!");
    } else if (trim(strtolower($_POST["user"])) === "admin") {
        grant_admin_privileges(); // end goal is to reach here
    }
    echo "Welcome {$results['user']}!";
?>
```

strtolower("ADMIN") === "admin"

```php
<?php
    // register.php
    $db = new mysqli("localhost", "username", "password", "dat    se"

    if ($_POST["user"] !== "admin") {
        $query = $db->prepare("INSERT INTO users VALUES (?, ?);");
        $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
        $query->execute();
    }
?>
```

Case Sensitivity

Step 1: Register as "ADMIN"
Step 2: Log in as "ADMIN"
Step 3: ???

Case Sensitivity

Step 1: Register as "ADMIN"
Step 2: Log in as "ADMIN"
Step 3: ???
Step 4: Profit!

Case Sensitivity

In general, most string comparisons are
not case sensitive!

Now things become a little harder…

```php
<?php
    // login.php
    $db = new mysqli("localhost", "username", "password", "database");

    $query = $db->prepare("SELECT * FROM users WHERE user=? AND pass=?;");
    $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
    $query->execute();
    $results = $query->get_result()->fetch_assoc();

    if (!$results) { // login failed
        exit("Wrong password!");
    } else if (trim(strtolower($_POST["user"])) === "admin") {
        grant_admin_privileges(); // end goal is to reach here
    }
    echo "Welcome {$results['user']}!";
?>
```

Case sensitivity problem
is FIXED!

```php
<?php
    // register.php
    $db = new mysqli("localhost", "username", "password", "database");

    if (strtolower($_POST["user"]) !== "admin") {
        $query = $db->prepare("INSERT INTO users VALUES (?, ?);");
        $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
        $query->execute();
    }
?>
```

```php
<?php
    // login.php
    $db = new mysqli("localhost", "username", "password", "database");

    $query = $db->prepare("SELECT * FROM users WHERE user=? AND pass=?;");
    $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
    $query->execute();
    $results = $query->get_result()->fetch_assoc();

    if (!$results) { // login failed
        exit("Wrong password!");
    } else if (trim(strtolower($_POST["user"])) === "admin") {
        grant_admin_privileges(); // end goal is to reach here
    }
    echo "Welcome {$results['user']}!";
?>
```

Something is different...

```php
<?php
    // register.php
    $db = new mysqli("localhost", "username", "password", "database");

    if (strtolower($_POST["user"]) !== "admin") {
        $query = $db->prepare("INSERT INTO users VALUES (?, ?);");
        $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
        $query->execute();
    }
?>
```

Let’s see how DBMS does string comparisons…

```
select * from users
where user='admin  ';
```

```
select * from users
where user='admin  ';
```

DBMS: Guess I’ll compare user in every row ¯_(ツ)_/¯

```
select * from users
where user='admin  ';
```

```
select * from users
where 'guest'='admin  ';
```

DBMS: 1st row’s user column is 'guest'

```
select * from users
where user='admin  ';
```

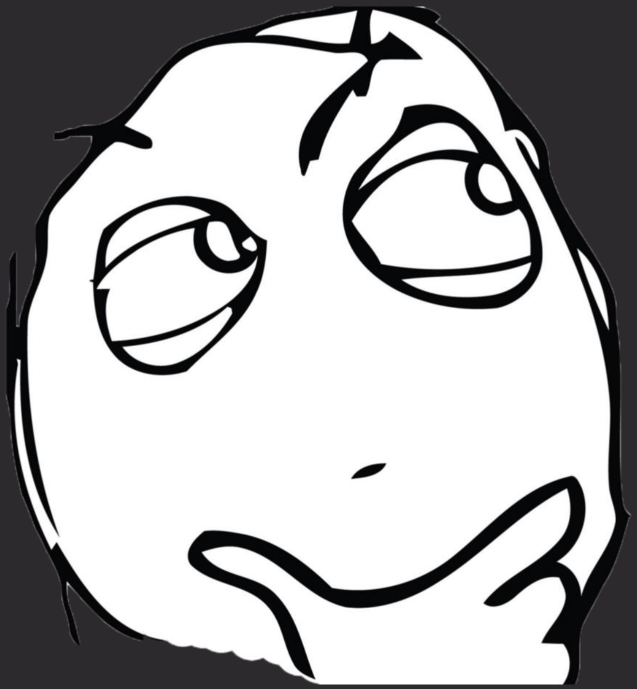
```
select * from users
where 'test'='admin  ';
```

DBMS: 2nd row’s user column is 'test'

```
select * from users
where user='admin  ';
```

```
select * from users
where 'admin'='admin  ';
```

DBMS: nth row's user
column is 'admin'



Surely,
'admin'='admin '
is false, right?

Trailing Whitespaces

"Most MySQL collations have a pad attribute of PAD SPACE… For PAD SPACE collations, trailing spaces are insignificant in comparisons; strings are compared without regard to any trailing spaces."

— *MySQL Reference Manual*

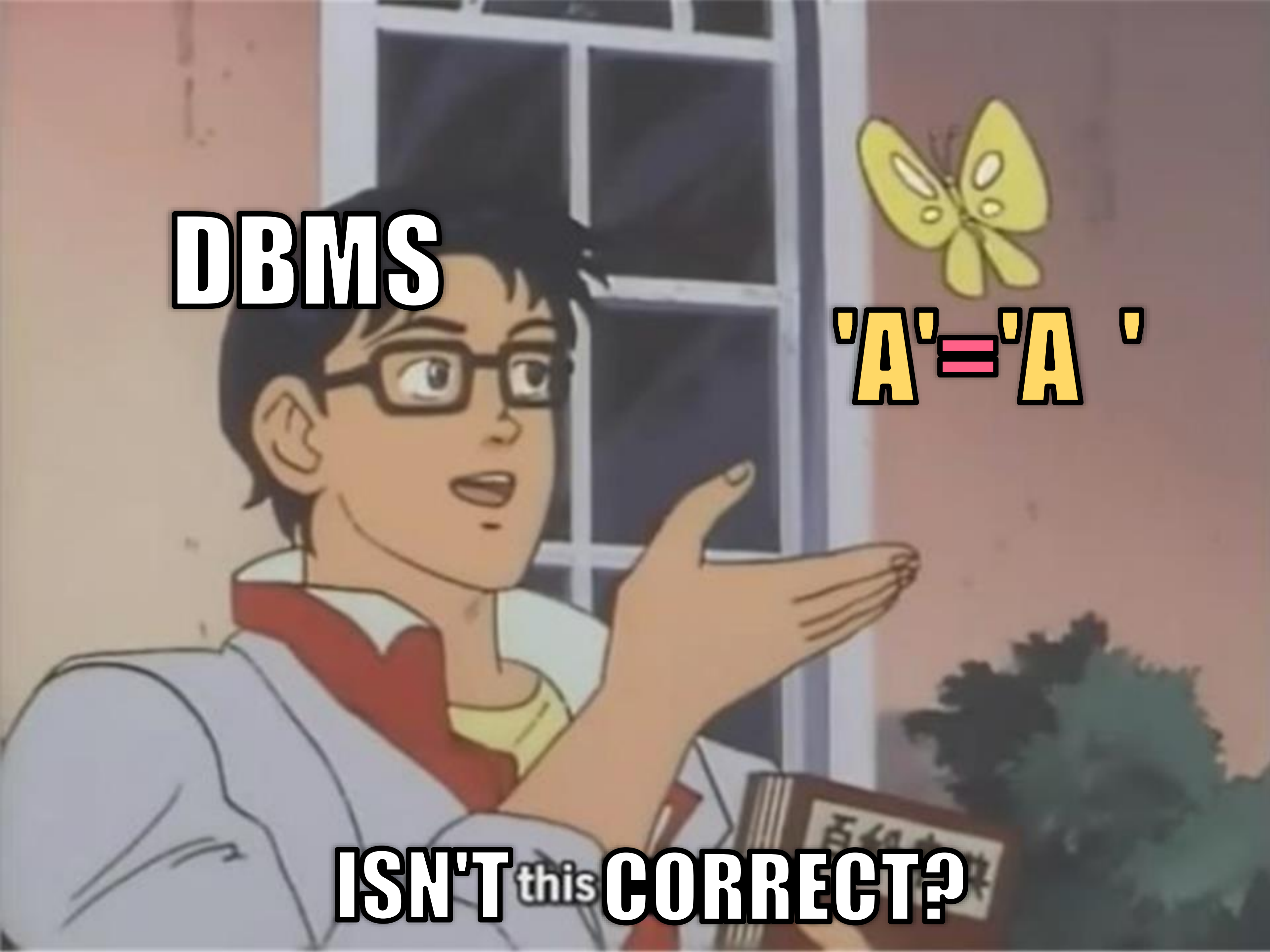
(https://dev.mysql.com/doc/refman/8.0/en/char.html)

Trailing Whitespaces

"The ANSI standard requires padding for the character strings used in comparisons so that their lengths match before comparing them.
For example, Transact-SQL considers the strings 'abc' and 'abc ' to be equivalent for most comparison operations."

— Microsoft KB316626

(https://support.microsoft.com/en-gb/help/316626/inf-how-sql-server-compares-strings-with-trailing-spaces)

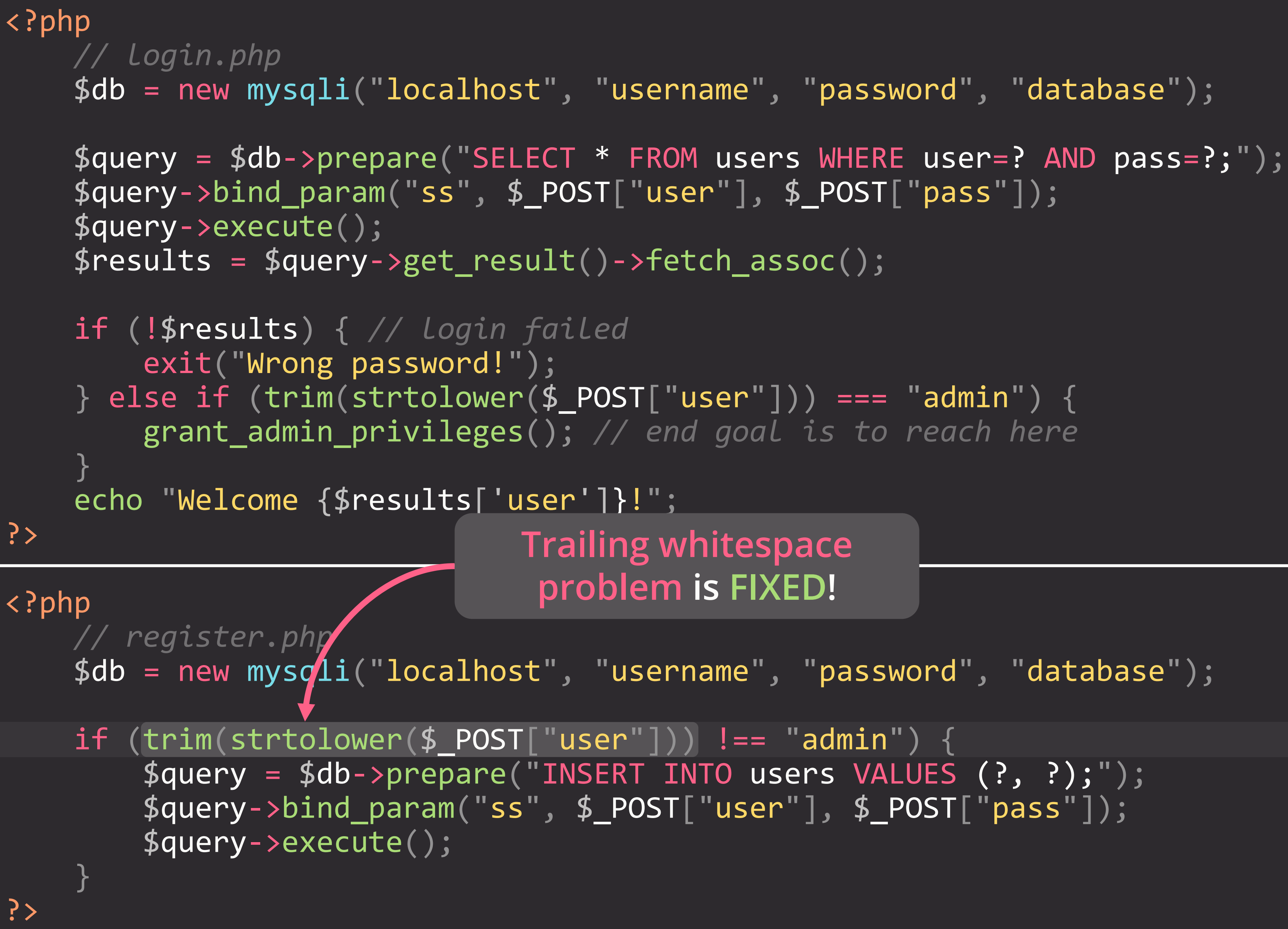

DBMS
'A'='A '
ISN'T this CORRECT?

DEVELOPERS

Trailing Whitespaces

This characteristic is often neglected,
but it’s so useful!

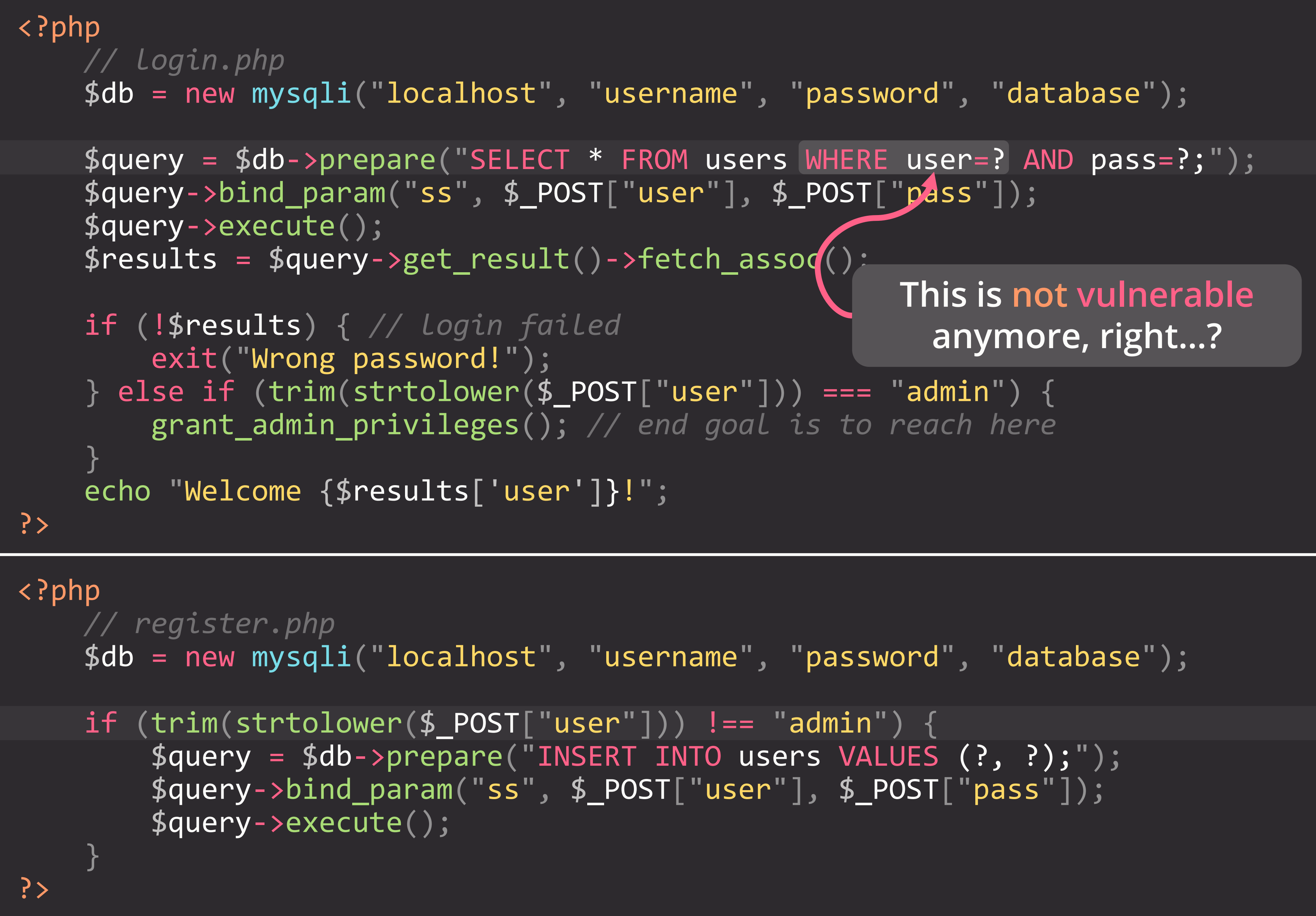
Okay, let’s fix this again…



```php
<?php
    // login.php
    $db = new mysqli("localhost", "username", "password", "database");

    $query = $db->prepare("SELECT * FROM users WHERE user=? AND pass=?;");
    $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
    $query->execute();
    $results = $query->get_result()->fetch_assoc();

    if (!$results) { // login failed
        exit("Wrong password!");
    } else if (trim(strtolower($_POST["user"])) === "admin") {
        grant_admin_privileges(); // end goal is to reach here
    }
    echo "Welcome {$results['user']}!";
?>
```

Trailing whitespace problem is FIXED!

```php
<?php
    // register.php
    $db = new mysqli("localhost", "username", "password", "database");

    if (trim(strtolower($_POST["user"])) !== "admin") {
        $query = $db->prepare("INSERT INTO users VALUES (?, ?);");
        $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
        $query->execute();
    }
?>
```



```php
<?php
    // login.php
    $db = new mysqli("localhost", "username", "password", "database");

    $query = $db->prepare("SELECT * FROM users WHERE user=? AND pass=?;");
    $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
    $query->execute();
    $results = $query->get_result()->fetch_assoc();

    if (!$results) { // login failed
        exit("Wrong password!");
    } else if (trim(strtolower($_POST["user"])) === "admin") {
        grant_admin_privileges(); // end goal is to reach here
    }
    echo "Welcome {$results['user']}!";
?>
```

This is not vulnerable anymore, right...?

```php
<?php
    // register.php
    $db = new mysqli("localhost", "username", "password", "database");

    if (trim(strtolower($_POST["user"])) !== "admin") {
        $query = $db->prepare("INSERT INTO users VALUES (?, ?);");
        $query->bind_param("ss", $_POST["user"], $_POST["pass"]);
        $query->execute();
    }
?>
```

SQL Column Truncation

> “If strict SQL mode is not enabled and you assign a value to a CHAR or VARCHAR column that exceeds the column's maximum length, the *value is truncated to fit...*”

— *MySQL Reference Manual*

(https://dev.mysql.com/doc/refman/8.0/en/char.html)

SQL Column Truncation

```
create table users (
    user    varchar(20),
    pass    varchar(100)
);
```

SQL Column Truncation

```
create table users (
    user    varchar(20),
    pass    varchar(100)
);
```

```
insert into users
(user, pass)
values
('admin<15 whitespace>excess',
'pass');
```

SQL Column Truncation

```
create table users (
    user    varchar(20),
    pass    varchar(100)
);
```

```
insert into users
(user, pass)
values
('admin<15 whitespace>excess',
'pass');
```

DBMS: Oh no!
Let me take care of it!!!

SQL Column Truncation

```
create table users (
    user    varchar(20),
    pass    varchar(100)
);
```

```
insert into users
(user, pass)
values
('admin<15 whitespace>[redacted],
'pass');
```

DBMS: Oh no!
Let me take care of it!!!

SQL Column Truncation

```
create table users (
    user    varchar(20),
    pass    varchar(100)
);
```

```
insert into users
(user, pass)
values
('admin<15 whitespace>’,
'pass');
```

DBMS: All is well now ☺

SQL Column Truncation

```
create table users (
    user    varchar(20),
    pass    varchar(100)
);
```

```
insert into users
(user, pass)
values
('admin<15 whitespace>’,
'pass');
```

✓

DBMS: Query OK, 1 row(s) affected.

```
select user from users
where user='admin';
```

```sql
select user from users
where user='admin';
```

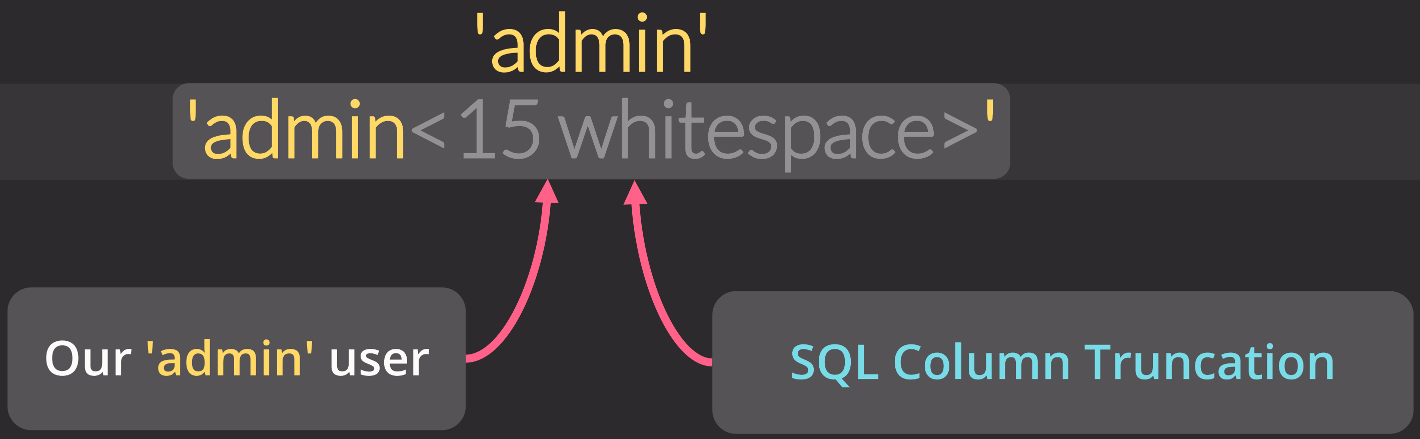
```
'admin'
'admin<15 whitespace>'
```

```
select user from users
where user='admin';
```

```
'admin'
'admin<15 whitespace>'
```

Our 'admin' user

```
select user from users
where user='admin';
```



'admin'

'admin<15 whitespace>'

Our 'admin' user

SQL Column Truncation

```sql
select user from users
where user='admin';
```

```
'admin'='admin<15 whitespace>'
```

Trailing whitespaces ignored

```
select user from users
where user='admin';
```

```
'admin'='admin<15 whitespace>'
```

Trailing whitespaces
ignored

✓

IT'S NOT A BUG
IT'S A "FEATURE"

SQL Column Truncation

This is dependent on the DBMS settings,
but there’s no harm trying!

Thank You!

Ngo Wei Lin
(@Creastery)