

# CLOSE ENCOUNTERS

with  
PHP



Level 3

# Validation & Security

Validation, Always



# Continuing With Our Application File

app/src/app.php

```
<?php
if($_SERVER['REQUEST_METHOD'] === 'POST') {
    $date = $_POST['date'];
    $email = $_POST['email'];
    $description = $_POST['desc'];

    echo "<p>Date: $date</p>";
    echo "<p>Email: $email</p>";
    echo "<p>$description</p>";
}
```



# Submitting the Form With No Validation

---

Let's look at some of the reasons we need to use validation.

- You are able to submit the form with no data
- If NULL values are stored to a database, they can cause issues when recalling the data

Date:

Email:



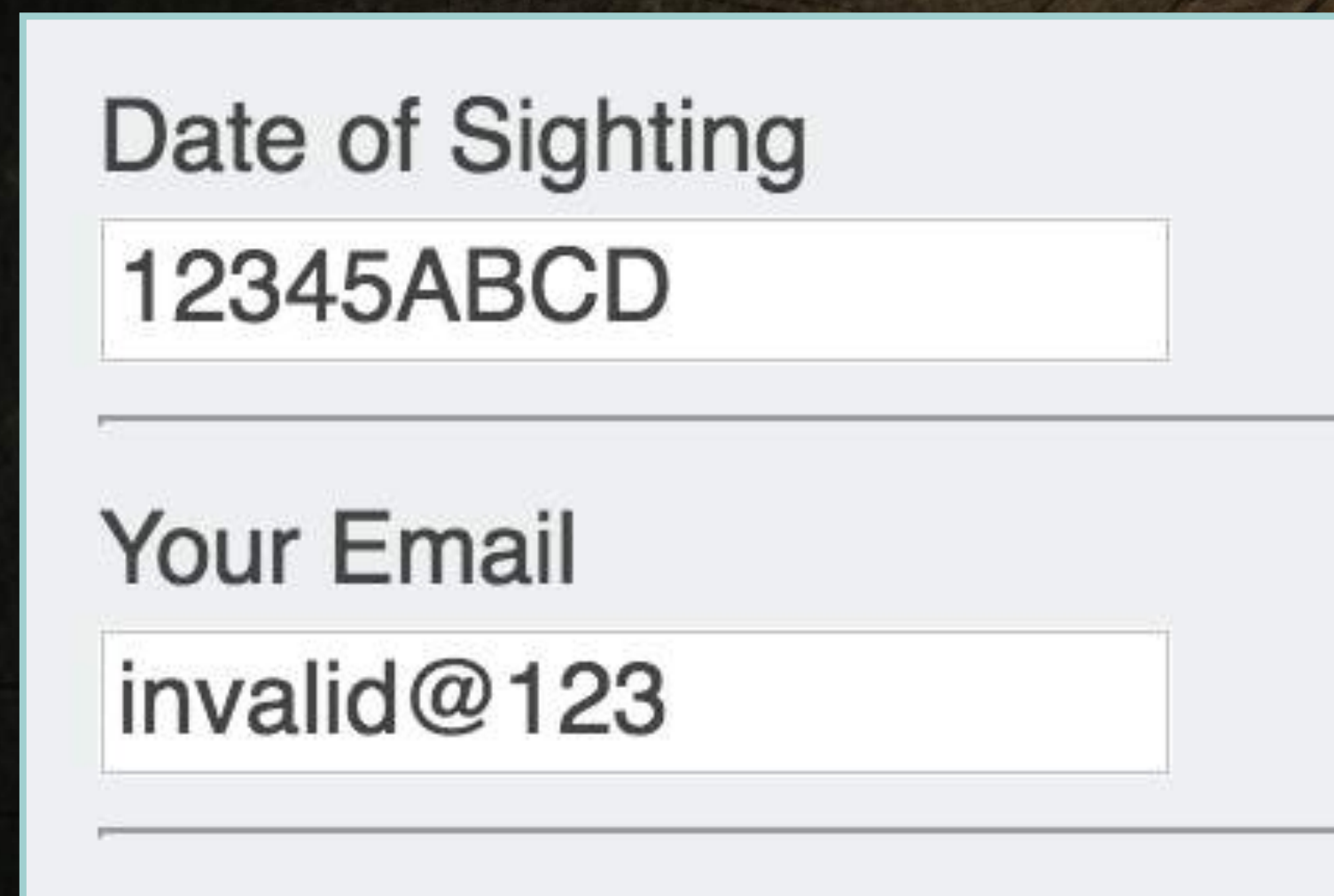
*Missing or NULL values can wreak havoc on a database!*



# Submitting the Form With No Validation

Let's look at some of the reasons we need to use validation.

- You are able to submit the form with no data
- If NULL values are stored to a database, they can cause issues when recalling the data
- Invalid dates and invalid email formats will cause issues as well when being recalled from the database



Date of Sighting  
12345ABCD

Your Email  
invalid@123

*An invalid date or email can break functionality of an application*



# Submitting the Form With No Validation

Let's look at some of the reasons we need to use validation.

Describe the Sighting

`<h1>ALIENS!</h1>`

- You are able to submit the form with no data
- If NULL values are stored to a database, they can cause issues when recalling the data
- Invalid dates and invalid email formats will cause issues as well, when being recalled from the database
- We will need to strip out any HTML or other code for security and formatting
- Otherwise, we will need to redirect back to the form!

*HTML and other code must be removed for security!*



## app/src/app.php

```
<?php
if($_SERVER['REQUEST_METHOD'] === 'POST') {
    $date = $_POST['date'];
    $email = $_POST['email'];
    $description = $_POST['desc'];

    echo "<p>Date: $date</p>";
    echo "<p>Email: $email</p>";
    echo "<p>$description</p>";
}
```

### Validation to Do:

\$date exists

\$email exists

\$description exists

remove whitespace

sanitize output

validate email

validate date

# Validation of Existence

## Validation to Do:

\$date exists ✓  
\$email exists  
\$description exists  
remove whitespace  
sanitize output  
validate email  
validate date

app/src/app.php

```
<?php
if($_SERVER['REQUEST_METHOD'] === 'POST') {
    $date = $_POST['date'];
    $email = $_POST['email'];
    $description = $_POST['desc'];

    if (!empty($date)) {
        echo "<p>Date: $date</p>";
    }

    echo "<p>Email: $email</p>";
    echo "<p>$description</p>";
}
```

*Validate that \$date exists and is not empty*

*Run code ONLY when if evaluates to true*



# Validation of Existence

app/src/app.php

```
<?php
if($_SERVER['REQUEST_METHOD'] === 'POST') {
    $date = $_POST['date'];
    $email = $_POST['email'];
    $description = $_POST['desc'];
```

```
    if (!empty($date) && !empty($email) && !empty($description)) {
        echo "<p>Date: $date</p>";
        echo "<p>Email: $email</p>";
        echo "<p>$description</p>";
    }
}
```

*Validate that all three exist and are not empty*

*&& represents the logical operator for "and"*

Validation to Do:

\$date exists ✓

\$email exists ✓

\$description exists ✓

remove whitespace

sanitize output

validate email

validate date

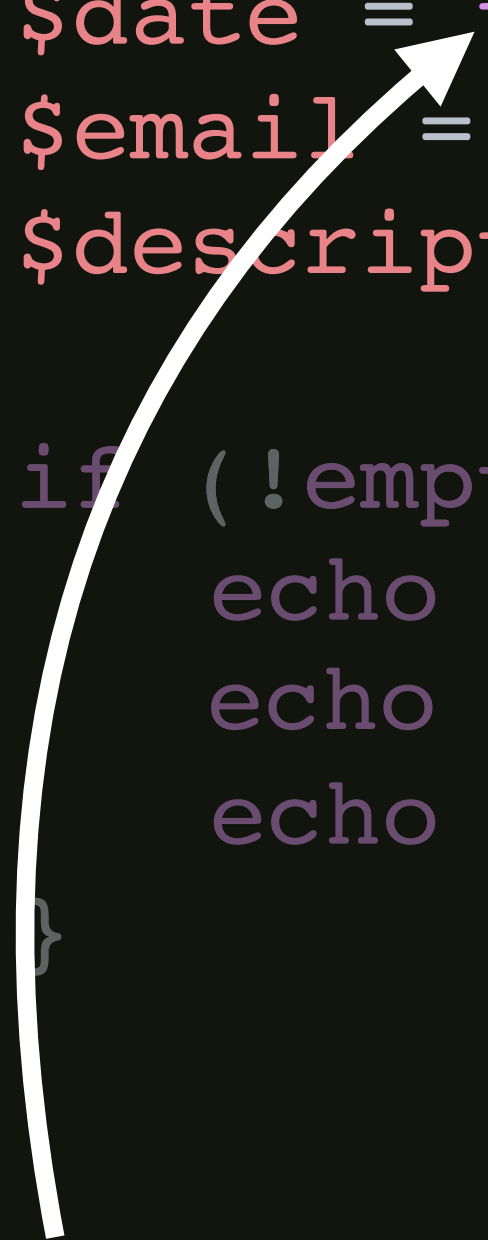


# Validation of Content

app/src/app.php

```
<?php
if($_SERVER['REQUEST_METHOD'] === 'POST') {
    $date = trim($_POST['date']);
    $email = trim($_POST['email']);
    $description = trim($_POST['desc']);

    if (!empty($date) && !empty($email) && !empty($description)) {
        echo "<p>Date: $date</p>";
        echo "<p>Email: $email</p>";
        echo "<p>$description</p>";
    }
}
```



*trim will remove any leading or trailing whitespace*

## Validation to Do:

- \$date exists ✓
- \$email exists ✓
- \$description exists ✓
- remove whitespace ✓
- sanitize output
- validate email
- validate date



# Filter Input, Sanitize Output

app/src/app.php

```
<?php
if($_SERVER['REQUEST_METHOD'] === 'POST') {
    $date = trim($_POST['date']);
    $email = trim($_POST['email']);
    $description = trim($_POST['desc']);

    if (!empty($date) && !empty($email) && !empty($description)) {
        echo "<p>Date: $date</p>";
        echo "<p>Email: $email</p>";
        echo '<p>' . htmlspecialchars($description) . '</p>';
    }
}
```

*htmlspecialchars encodes a string to HTML entities*

## Validation to Do:

- \$date exists ✓
- \$email exists ✓
- \$description exists ✓
- remove whitespace ✓
- sanitize output ✓
- validate email
- validate date



# Filter Input, Sanitize Output

app/src/app.php

```
<?php
if($_REQUEST['method'] === 'POST') {
    $date = $_POST['date'];
    $email = $_POST['email'];
    $description = trim($_POST['desc']);
```

```
    if(!empty($email) && !empty($description)) {
        echo "<p>Date: $date</p>";
        echo "<p>Email: $email</p>";
        echo "<p>Describe the Sighting: " . htmlspecialchars($description) . "</p>";
    }
}
```

Date: tomorrow  
Email: invalid@123  
<h1>ALIENS!</h1>

tomorrow  
Your Email  
invalid@123  
Describe the Sighting  
<h1>ALIENS!</h1>

*If the user submits HTML, now they are encoded*

Validation to Do:

\$date exists ✓

\$email exists ✓

\$description exists ✓

remove whitespace ✓

sanitize output ✓

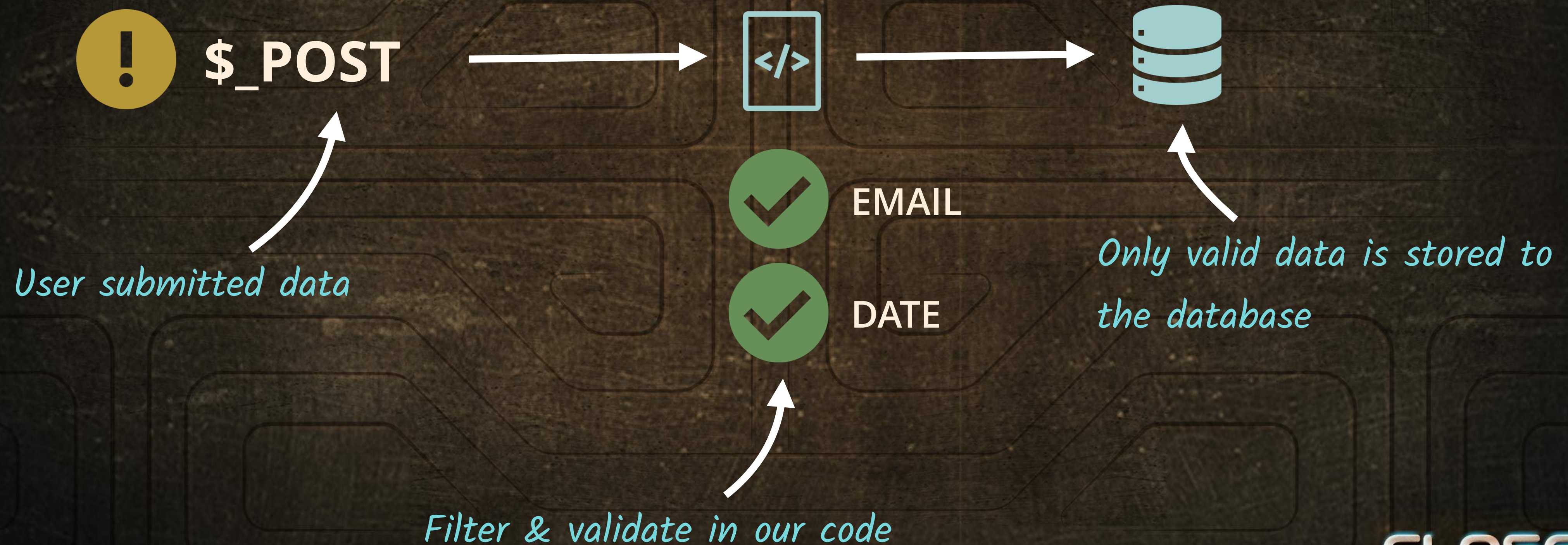
validate email

validate date



# Filtering & Sanitizing in Review

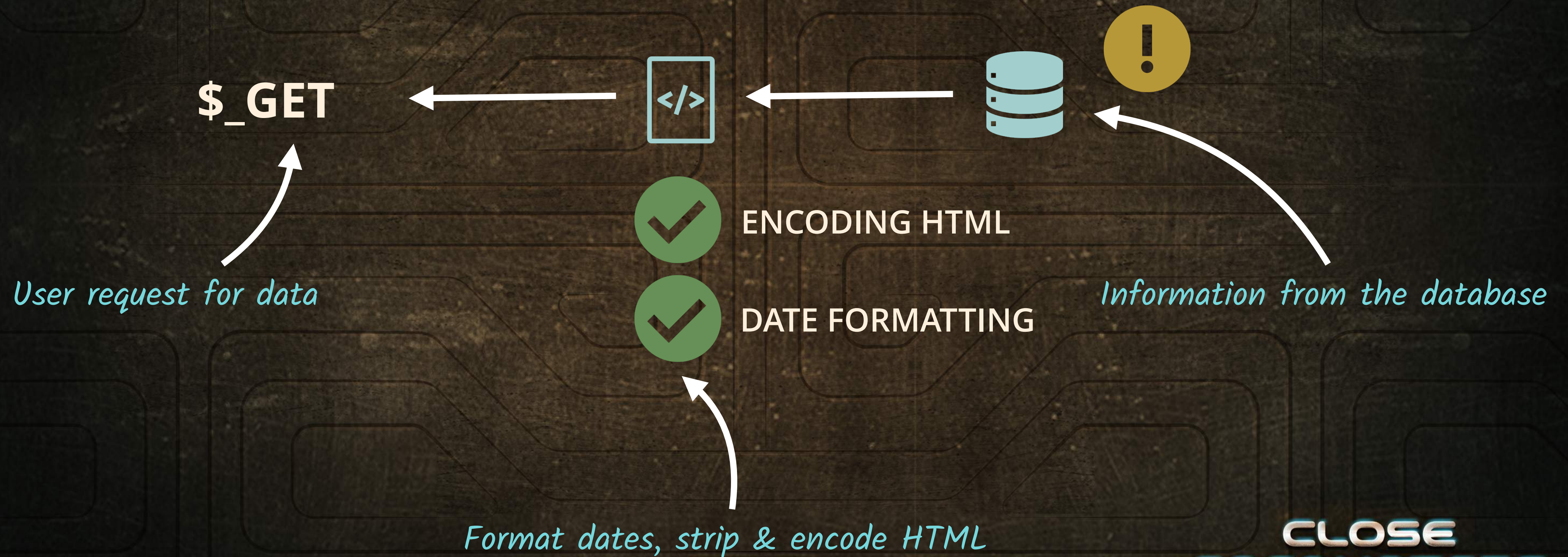
Why do we filter input and sanitize our output?





# Filtering & Sanitizing in Review

Why do we filter input and sanitize our output?





# CLOSE ENCOUNTERS

with  
PHP