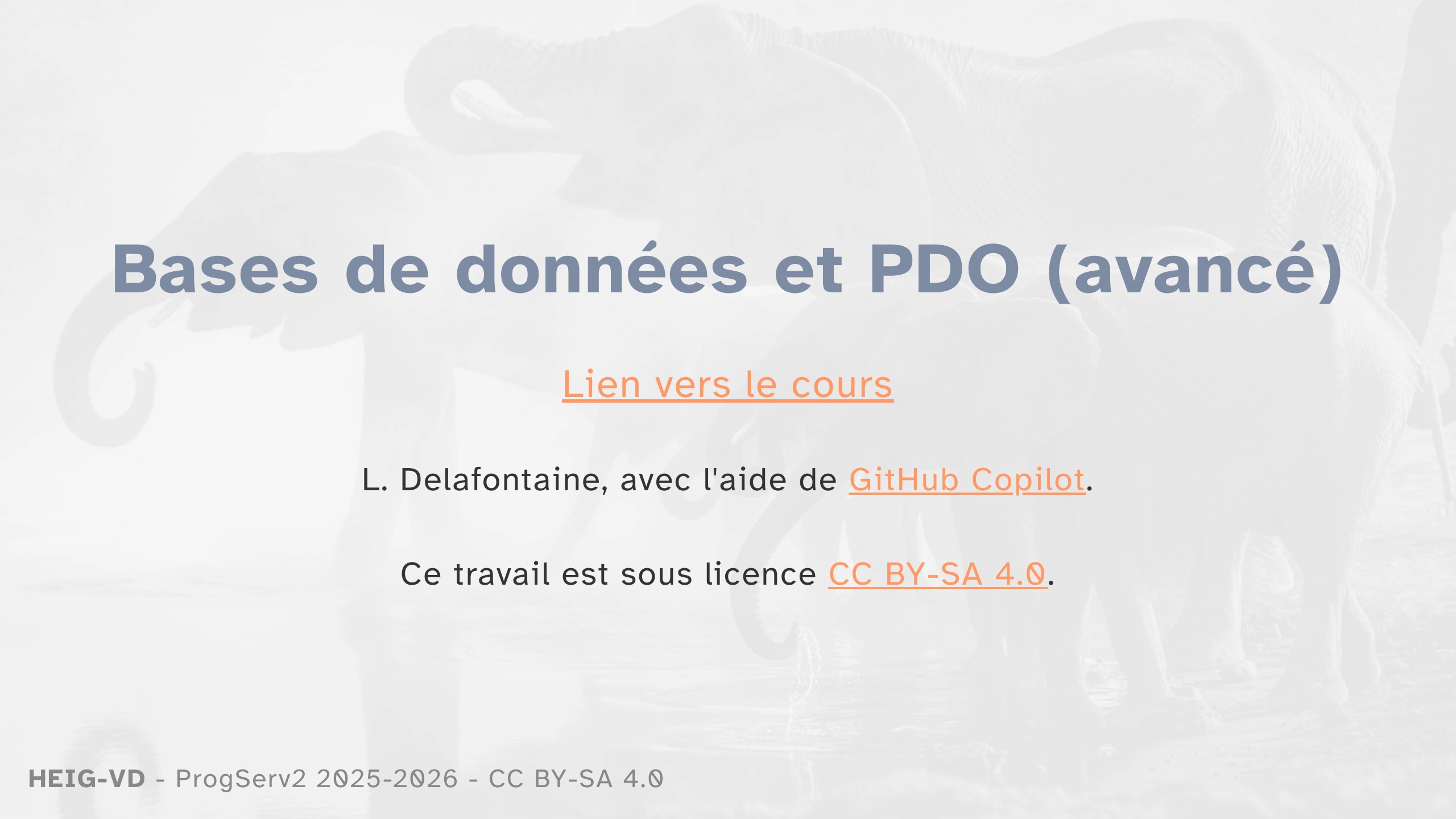


A faint, grayscale background image of a herd of elephants in a savanna setting, with one elephant in the foreground and others in the background.

Bases de données et PDO (avancé)

[Lien vers le cours](#)

L. Delafontaine, avec l'aide de [GitHub Copilot](#).

Ce travail est sous licence [CC BY-SA 4.0](#).

Retrouvez plus de détails dans le support de cours

Cette présentation est un résumé du support de cours. Pour plus de détails, consultez le [support de cours](#).

Objectifs (1/2)

- Rappeler les concepts de base des formulaires HTML, validation et sécurité.
- Utiliser PDO pour interagir avec une base de données MySQL/MariaDB.
- Utiliser les exceptions pour la gestion des erreurs en PHP.



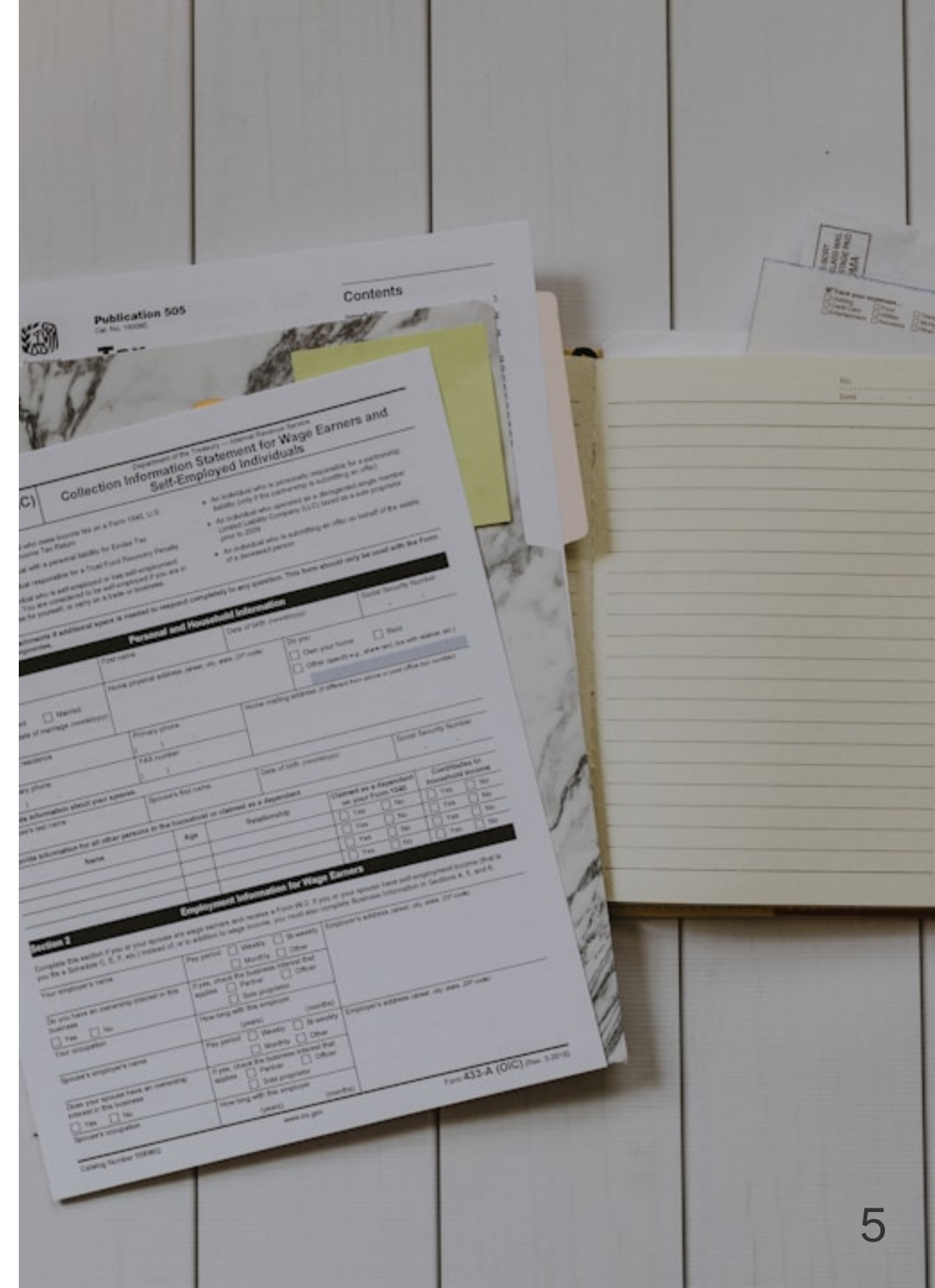
Objectifs (2/2)

- Utiliser les fichiers de configuration pour stocker les paramètres de connexion à la base de données.



Formulaires HTML et PDO, un rappel

- Éléments de base du monde web.
- Permettent de collecter des données utilisateur.
- Nécessitent une attention particulière en matière de validation et de sécurité.



Structure d'un formulaire HTML

- Utilisation de la balise `<form>` .
- L'action (`action`) définit où les données sont envoyées.
- La méthode (`method`) définit comment les données sont envoyées (`GET` ou `POST`).
- Analysons l'exemple `01-simple-form.php` .



Récupération des données côté serveur

- Les données des formulaires sont accessibles via les superglobales `$_GET` et `$_POST`.
- Analysons l'exemple `02-get-data-server-side.php`.



Validation côté serveur

- Valider les données permet de garantir leur intégrité et sécurité.
- Utiliser des filtres et des expressions régulières pour valider les données.
- Analysons l'exemple
`03-validate-data-server-side.php`.



Conservation des données en cas d'erreurs

- Par défaut, les données saisies sont perdues une fois soumises.
- Il est possible de conserver les données saisies en cas d'erreurs de validation.
- Analysons l'exemple

`04-keep-data-on-errors.php` .



Connexion à une base de données SQLite avec PDO

- PDO (PHP Data Objects) est une extension PHP pour accéder aux bases de données (SQLite, MySQL, Postgres, etc.).
- Analysons l'exemple `05-pdo-and-sqlite.php`.



Nettoyage des données et persistance avec les requêtes préparées

- Nettoyer les données évite les vulnérabilités par injections SQL.
- PDO met à disposition les requêtes préparées.
- Analysons l'exemple `05-pdo-and-sqlite.php`.



Affichage sécurisé des données

- Pour éviter les attaques XSS, il est important d'échapper les caractères spéciaux lors de l'affichage des données utilisateur.
- La fonction `htmlspecialchars()` permet d'échapper ces caractères.
- Analysons l'exemple `06-escape-special-characters.php`.



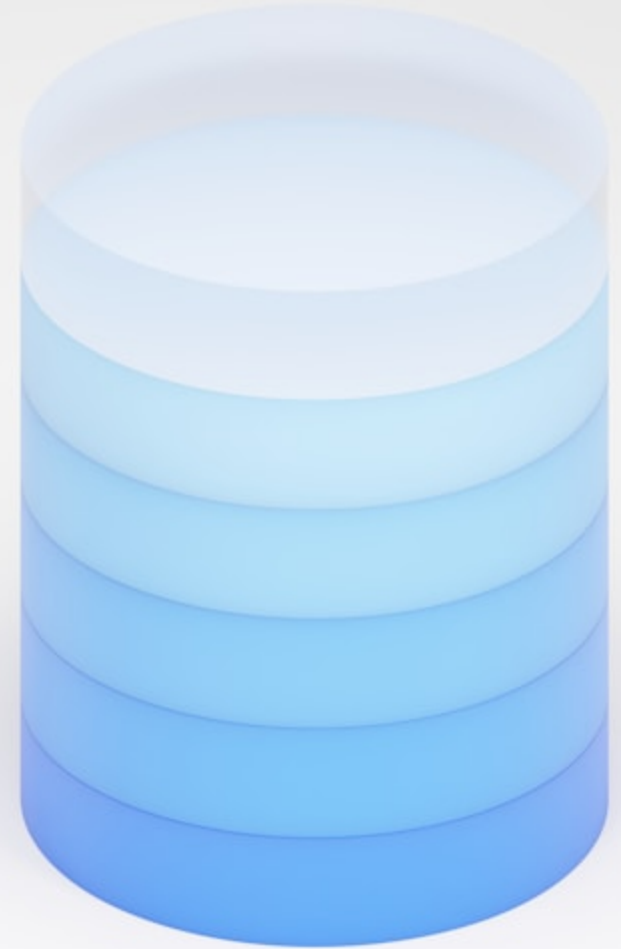
Validation côté client

- Améliore l'expérience utilisateur.
- **Ne remplace pas la validation côté serveur.**
- Repose sur les attributs HTML (`required`, `type`, `min`, etc.).
- Analysons l'exemple `07-validate-data-client-side.php`.



Bases de données et PDO (avancé)

- Dans un environnement de production, SQLite n'est pas toujours adapté.
- MySQL/MariaDB est un système de gestion de base de données (SGBD) plus robuste.
- Nous allons voir comment utiliser MySQL/MariaDB avec PDO.



MySQL/MariaDB

- Bases de données relationnelles.
- MariaDB est un fork (= un clone) de MySQL.
- Les deux offrent les mêmes fonctionnalités de base.
- PDO permet d'interagir avec MySQL/MariaDB.
- Analysons l'exemple `08-mysql-with-constants.php`.



Accéder à MySQL/MariaDB avec MAMP et Visual Studio Code (1/2)

- Une base de données MySQL est accessible via MAMP.
- L'utilisateur par défaut est `root`, sans mot de passe.



Accéder à MySQL/MariaDB avec MAMP et Visual Studio Code (2/2)

- Pour accéder à la base de données, utilisez:
 - [MySQL Workbench](#)
 - Visual Studio Code avec [Database Client](#).



Gestion des erreurs avec les exceptions

(1/3)

- Lors de la connexion à la base de données ou de l'exécution de requêtes, des erreurs peuvent survenir.
- Lorsqu'une erreur survient, PDO lance une exception.
- Une exception est un objet qui représente une erreur.
- Une exception peut être "jetée" (throw) lorsqu'une erreur se produit, et elle peut être "attrapée" (catch) dans un bloc `try-catch` pour gérer l'erreur de manière appropriée.
- Une exception non gérée provoque l'arrêt de l'application.

Gestion des erreurs avec les exceptions

(2/3)

- Ce qui est dans le bloc `try` peut lever une exception. Si une exception est levée, le code dans le bloc `catch` est exécuté.

```
try {  
    // Code qui peut générer une exception  
    throw new Exception("Une erreur s'est produite.");  
} catch (Exception $e) {  
    // Gestion de l'exception  
    echo "Une exception a été capturée : " . $e->getMessage();  
}
```

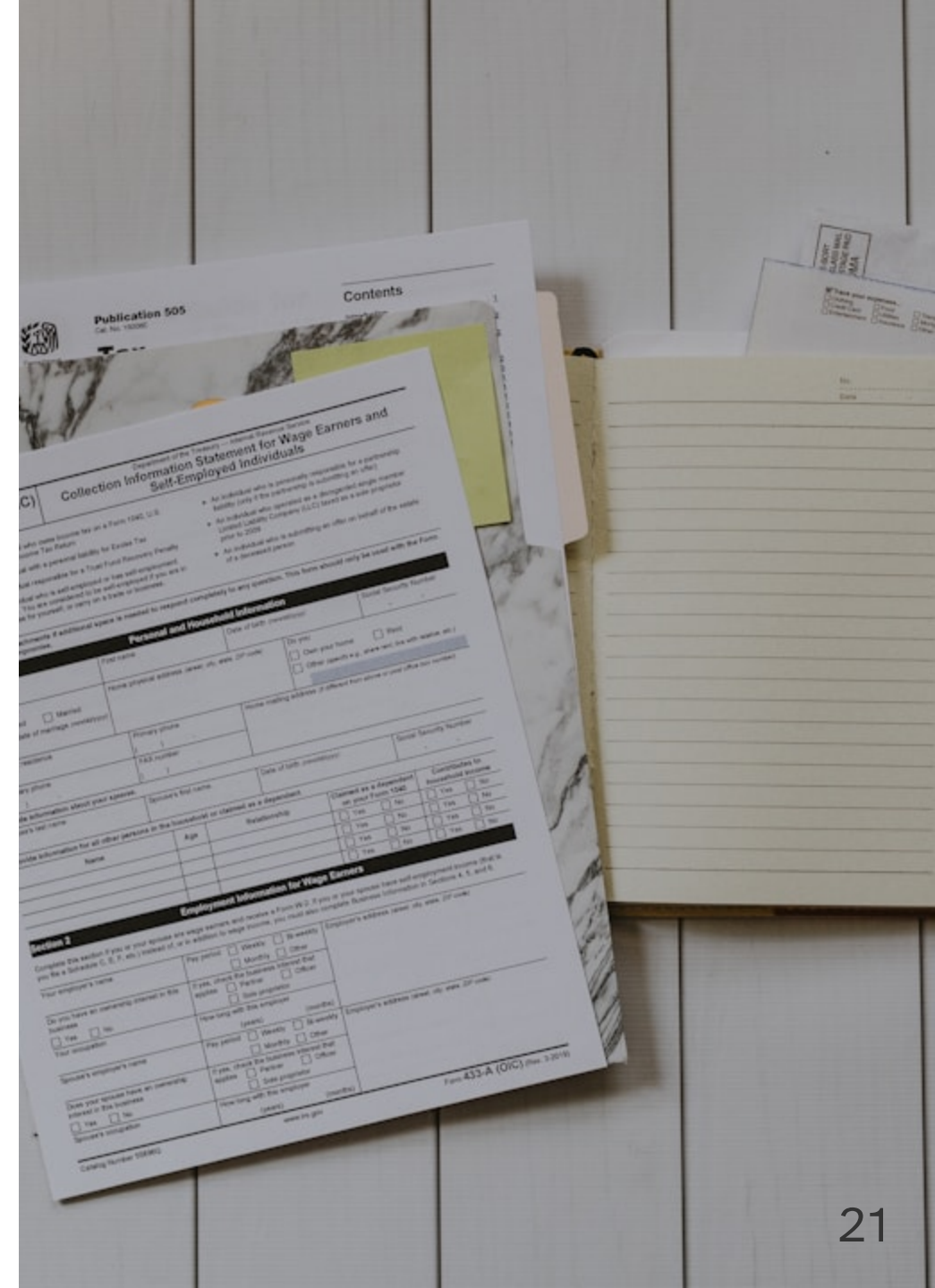
Gestion des erreurs avec les exceptions (3/3)

- Analysons l'exemple
`09-handle-exceptions.php`.



Fichiers de configuration (1/2)

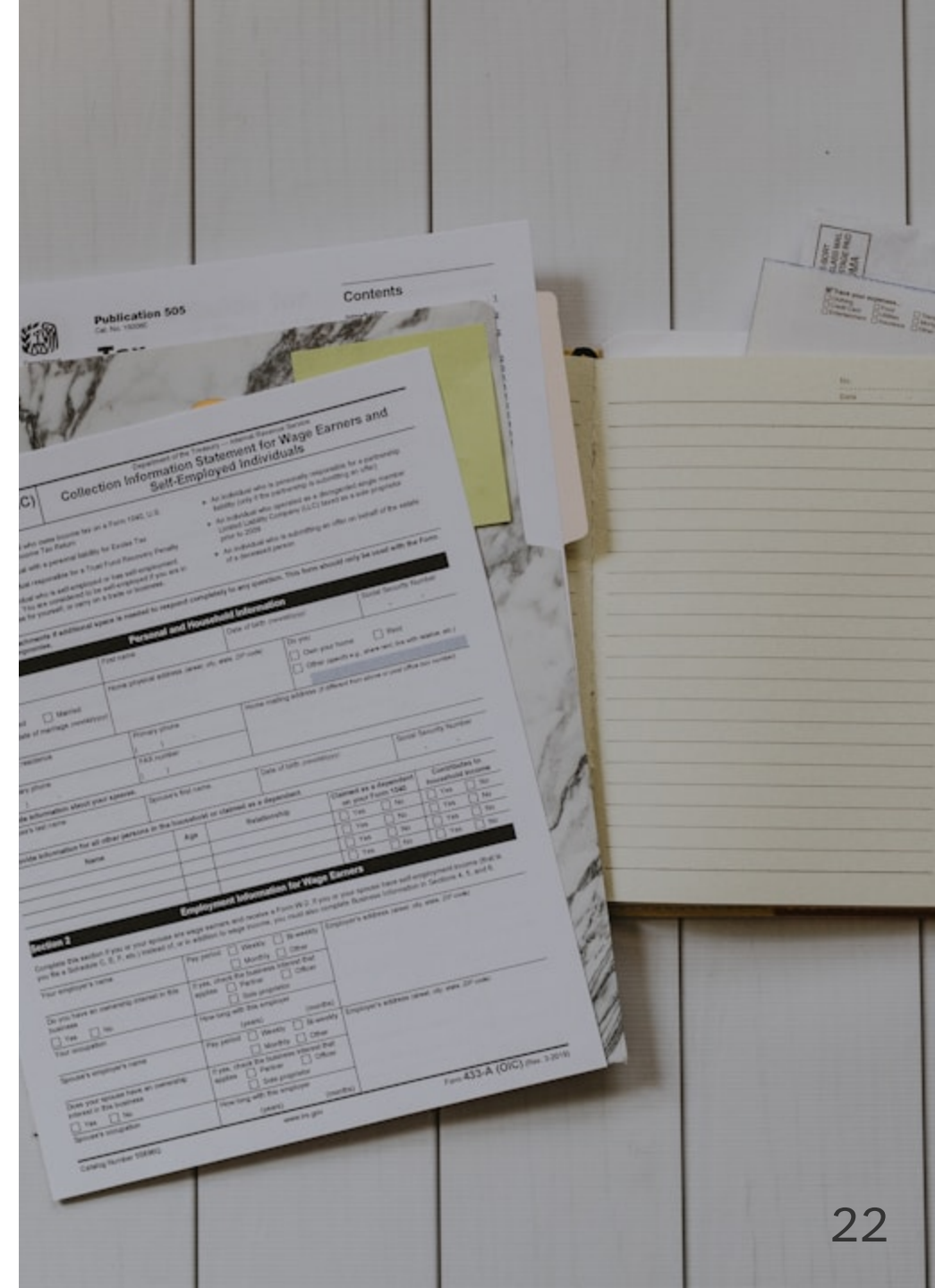
- Stocker les paramètres de connexion dans le code source n'est pas recommandé.
- Utiliser un fichier de configuration pour les paramètres de connexion.
- Attention à la sécurité : le fichier de configuration ne doit pas être accessible publiquement.



Fichiers de configuration (2/2)

- Analysons l'exemple
10-database-configuration-file.php

•



Conclusion

- PDO permet d'interagir avec différentes bases de données.
- MySQL/MariaDB est un SGBD robuste pour les applications en production.
- La gestion des erreurs avec les exceptions améliore la robustesse du code.
- Utiliser des fichiers de configuration pour une meilleure gestion des paramètres.
- Toujours valider et nettoyer les données utilisateur issues des formulaires pour des raisons de sécurité.

Questions

Est-ce que vous avez des questions ?

À vous de jouer !

- (Re)lire le support de cours.
- Explorer les exemples de code.
- Faire les exercices.
- Poser des questions si nécessaire.

➡ [Lien vers le cours](#)

N'hésitez pas à vous entraidez si vous avez des difficultés !



Sources (1/2)

- [Illustration principale](#) par [Richard Jacobs](#) sur [Unsplash](#)
- [Illustration](#) par [Aline de Nadai](#) sur [Unsplash](#)
- [Illustration](#) par [Kelly Sikkema](#) sur [Unsplash](#)
- [Illustration](#) par [Anastasiia Nelen](#) sur [Unsplash](#)
- [Illustration](#) par [Shavr IK](#) sur [Unsplash](#)
- [Illustration](#) par [Taylor Vick](#) sur [Unsplash](#)
- [Illustration](#) par [Kelly Sikkema](#) sur [Unsplash](#)

Sources (2/2)

- [Illustration](#) par [Jan Antonin Kolar](#) sur [Unsplash](#)
- [Illustration](#) par [Towfiqu barbhuiya](#) sur [Unsplash](#)
- [Illustration](#) par [Blake Wisz](#) sur [Unsplash](#)
- [Illustration](#) par [Sunder Muthukumaran](#) sur [Unsplash](#)
- [Illustration](#) par [Sigmund](#) sur [Unsplash](#)
- [Illustration](#) par [Nikita Kachanovsky](#) sur [Unsplash](#)