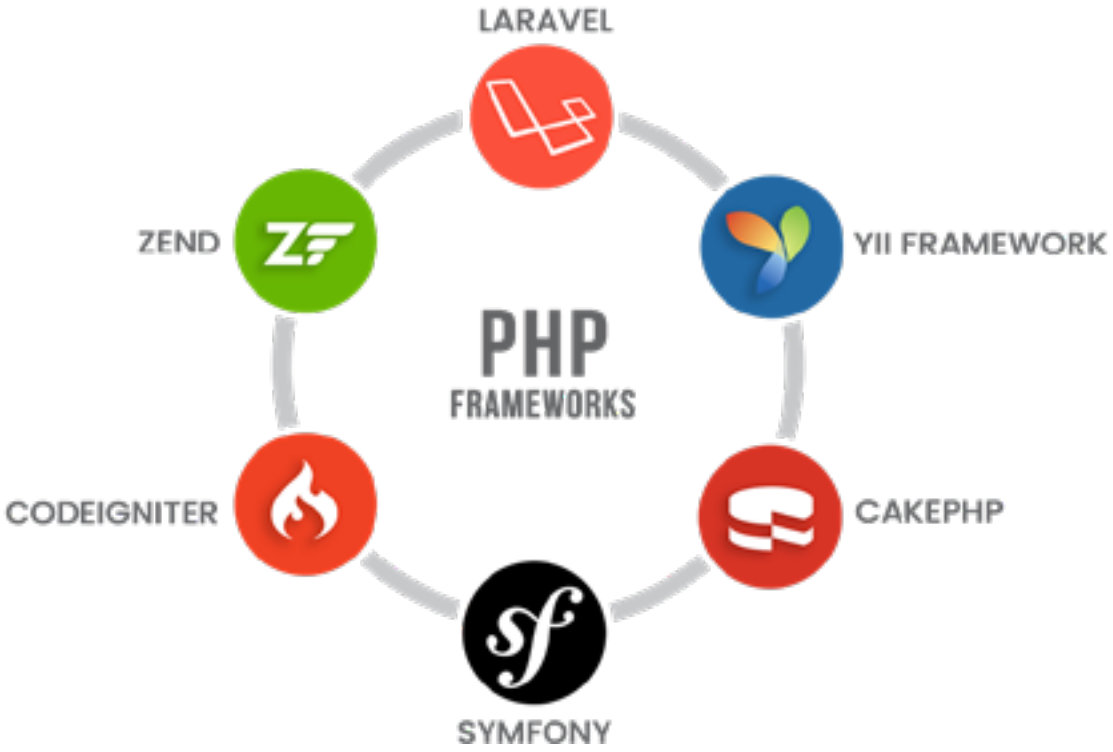
IUT Lyon 1
l'excellence technologique

Projet Web



Objectifs et organisation

- Objectifs
  - Programmation avancée avec PHP
  - Développement applications Web solides
- Organisation
  - 15 TPs (2H) => 30H
  - DS (1H) + TP noté (2H)
- Requis
  - Maîtrise de PHP et de la POO

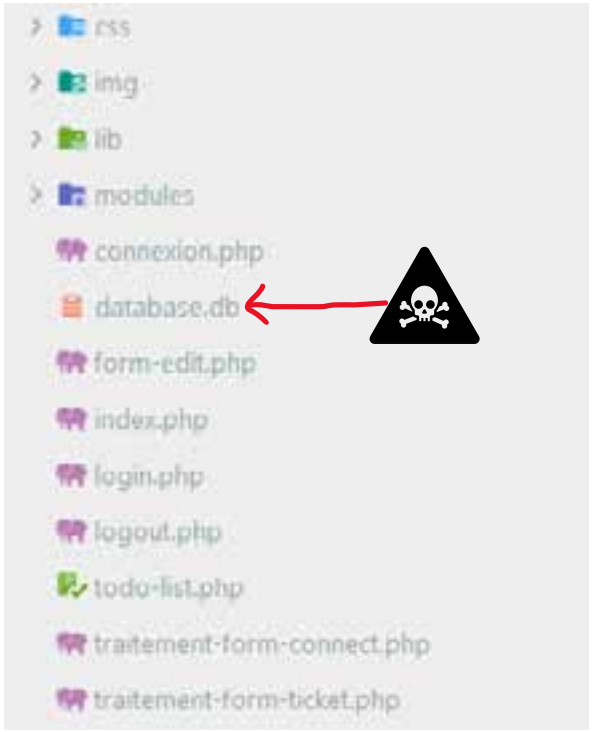
Evaluation

- Bonus / malus TP
- DS écrit
- TP noté
- Projet final (en binôme)

Sécurisons les fichiers

La méthode procédurale vu en S3 présente des soucis d'organisation et de sécurité majeurs

WEBROOT



Tous les fichiers sont accessibles via une url

Ex: localhost/modules/footer.php

Ce qui nécessite :

- de sécuriser chaque accès à tous les fichiers
- vérifier que les dépendances aux script d'un fichier soit bien déclarées
- ...

Sécurisons les fichiers

Une solution est de déplacer les fichiers script et sensible dans un dossier qui n'est plus accessible par une url :

Le serveur de PHP sera exécuté dans le dossier webroot.

Plus aucun fichier php ne sera accessible via une url

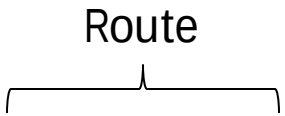
sauf index.php qui devient le **point d'entrée** de notre application

C'est-à-dire que toutes les url (sauf les fichiers statiques) appelleront le fichier index.php

Routing

Le routing est le principe de définir un routage vers un partie de notre application en fonction de l'url appelé ou, plus précisément, en fonction de la requête au serveur.

Le serveur de PHP intègre un système d'« url rewriting »



Si on appelle l'url : http://localhost/blog/article-1
(sans extension de fichier
Et qu'aucun fichier n'existe à ce chemin)

Alors : le fichier index.php présent à la racine du webroot sera le script exécuté

On se servira de la variable superglobale $_SERVER à l'index ``PATH_INFO``
pour récupérer la **route**

Routing

On différenciera également la méthode appelant cette page :

- GET
- POST

Web

- DELETE
- PUT
- PATCH
- ...

API REST

On utilisera la superglobale $_SERVER à l'index ``REQUEST_METHOD`` afin d'établir des conditions.

Séparation de code

Le principe est de faire toute la logique avant affichage, afin de construire le rendu et au final faire un seul « echo »

La fonction **ob_start()** permet de démarrer un temporisation de sortie :

```php
ob_start();
?>
<!DOCTYPE html>
<html lang="fr">
```

On peut générer de l'affichage sans qu'il soit envoyé au navigateur, ainsi on peut utiliser les fonctions **header(), setcookie()** et **session_start()** sans alerte.

```php
</body>
</html>
<?php
$content = ob_get_clean();
echo $content;
```

Seul retour d'affichage envoyé au navigateur