

TP noté

2024-2025

Licence Informatique 3ème année

Alexandre Niveau — Jean-Marc Lecarpentier

TP noté

2024-2025

Durée : 2h


Consignes générales

- Le TP noté aura lieu de 9h15 à 11h15.
- Vous devez utiliser les machines des salles TP.
- Les téléphones portables doivent être éteints et au fond des sacs.
- Vous avez le droit d'accéder aux fichiers sur votre compte, à d'éventuels documents papier, et à toute ressource sur le web (y compris stackoverflow), mais **vous ne devez pas communiquer**, ni entre vous, ni avec des tierces personnes, ni avec une IA.
- Le code sera à déposer uniquement sur ecampus (rien ne sera évalué en-dehors de l'archive déposée).
- Vous pouvez travailler jusqu'à l'heure de fin (11h15), mais à cette heure-là il faut passer à la création de l'archive et au dépôt sur ecampus. La zone de dépôt reste ouverte cinq minutes après 11h15 pour vous laisser le temps de déposer, mais n'utilisez pas ce temps pour travailler.

Mise en place de l'examen

Téléchargez **[l'archive du sujet](#)**, et décompressez-la. Elle contient un répertoire tp-note-tw4b-2024-VOTRELOGIN dans lequel vous effectuerez votre travail. **Renommez-le en remplaçant VOTRELOGIN par votre login** (ça donnera donc par exemple tp-note-tw4b-2024-dupont223 ou tp-note-tw4b-2024-21912345).

Vous pouvez placer ce répertoire où vous voulez, pas forcément sur votre espace web (on ne corrigera que ce que vous déposerez sur ecampus).

 En particulier, il est recommandé de plutôt laisser le répertoire sur votre

compte normal et utiliser le *serveur de développement inclus avec PHP*, de la façon suivante.

1. Ouvrez un terminal dans le répertoire `tp-note-tw4b-2024-LOGIN` que vous avez extrait et renommé
2. Lancez la commande `php -S localhost:20202 -d display_errors=true` (remplacez 20202 par un autre port si celui-ci ne marche pas...).
3. Allez à l'URL `http://localhost:20202/script.php` : vous devriez voir le résultat de l'exécution du fichier `script.php`. (Il y a des erreurs PHP, c'est normal, c'est pour que vous voyiez ce que ça donne.)

Attention, les erreurs affichées dans les pages sautent moins aux yeux que sur le serveur, et la stack trace est moins lisible. Cependant, toutes les erreurs PHP sont aussi affichées dans le terminal où vous avez lancé le serveur, et la stack trace y est un peu mieux. Notez aussi que ce serveur de développement, très basique, ne sait pas afficher d'index des répertoires, comme le fait le serveur Apache utilisé pour votre `www-dev`.

Important

Vous devez rendre à l'issue de l'examen une archive zip contenant (à la racine) votre répertoire `tp-note-tw4b-2024-LOGIN`, déposée sur Ecampus dans la zone prévue à cet effet. Le nom de l'archive n'a pas d'importance, mais il est important que l'archive contienne le répertoire demandé (et pas son contenu par exemple).

Faites bien attention à ne pas déposer l'archive initiale, mais bien celle avec votre travail ! Normalement, si vous avez bien renommé le répertoire, il ne devrait pas y avoir de confusion possible.

Exercice

Le répertoire `tp-note-tw4b-2024-LOGIN` contient deux fichiers. Le fichier `script.php` sera le script principal dont il est question dans l'exercice, et que vous allez compléter en suivant l'énoncé (dans la suite, je dis souvent simplement « le script »). L'autre, `data.php`, contient une liste (= un tableau), `$IMAGE_URLS` avec des URL d'images (qui sont hébergées sur ensweb, mais c'est un détail). Les images sont utilisables librement, elles proviennent du site [PublicDomainPictures.net](https://publicdomainpictures.net).

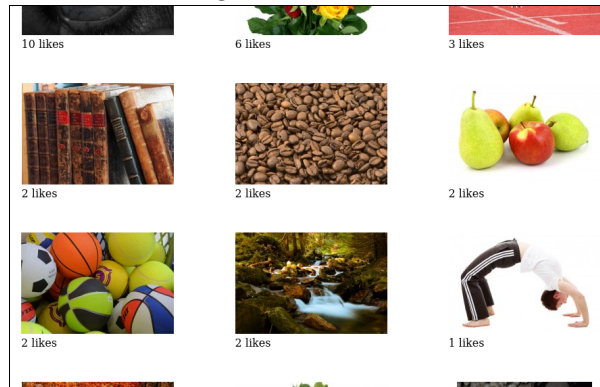
L'exercice consiste à implémenter un mini-site qui propose des images aléatoires. Rendre un code propre et bien organisé est évidemment important, mais **la priorité est que ça fonctionne**. NB : ce n'est pas un exercice d'application de l'architecture MVC. Noter aussi que la validation des données est facultative (mais des efforts en ce sens seront appréciés).

1. Effacer le contenu de `script.php` après vous être assuré·e que le serveur fonctionne, puis inclure le fichier `data.php` depuis `script.php` afin de pouvoir utiliser le tableau `$IMAGE_URLS` (**sans faire de copier-coller du tableau !**)

2. Faire en sorte que le script affiche une des URL de la liste, choisie aléatoirement à chaque appel. Rappel, on peut utiliser le code `rand(0, $n)` pour tirer un entier entre 0 et `$n` inclus.
3. Par défaut, le serveur indique dans les en-têtes HTTP que le contenu est du HTML, mais ce n'est pas le cas ici. Faire en sorte que le contenu renvoyé soit de type `text/plain`.
4. Modifier le script pour qu'il ne fasse tout cela que lorsque le paramètre d'URL `action` a la valeur `url` ; s'il n'y a pas de paramètre `action`, ou s'il a une autre valeur, le script doit afficher une page HTML (pas *plain text* !) indiquant une erreur.
5. Modifier le script pour qu'il reconnaisse une autre valeur du paramètre `action`, la valeur `redirect`. Pour cette action, le script doit choisir une URL aléatoire dans la liste, puis effectuer une redirection « 303 See Other » vers l'URL en question.
6. Modifier le script pour qu'il reconnaisse une autre valeur du paramètre `action`, la valeur `page`. Pour cette action, le script doit choisir une URL aléatoire dans la liste, puis afficher une page HTML contenant l'image correspondant à cette URL.
7. Modifier le script pour qu'il reconnaisse une autre valeur du paramètre `action`, la valeur `gallery`. En réponse, le script doit générer une page HTML contenant une galerie de 10 images, dont les URL sont tirées aléatoirement par le serveur puis écrites dans le HTML. (Ce n'est pas grave s'il y a plusieurs fois la même URL/image.)
8. Modifier le script pour qu'il reconnaisse une autre valeur du paramètre `action`, la valeur `history`. En réponse, le script doit générer une page HTML contenant une galerie des dernières images choisies par le tirage aléatoire (maximum 20), de la plus récente à la plus ancienne. **NB: cela nécessite de modifier du code déjà écrit, puisqu'il faut en particulier se souvenir de toutes les URL tirées !** On utilisera une session PHP pour la persistance. (Encore une fois, ce n'est pas grave s'il y a plusieurs fois la même URL/image.)
9. Modifier la réponse HTML de l'action `gallery` : il doit maintenant y avoir un bouton «Ajouter une image» qui fait une requête XHR vers l'URL `script.php?action=url` pour récupérer une nouvelle URL aléatoire, puis qui ajoute dans la galerie l'image correspondant à l'URL reçue.
10. Modifier le script pour qu'il reconnaisse une autre valeur du paramètre `action`, la valeur `like`. Cette action ne doit être exécutée que si la méthode de la requête est POST (rappel : `$_SERVER['REQUEST_METHOD']` contient la méthode HTTP utilisée dans la requête), et si l'URL contient aussi le nouveau paramètre `url`, qui comme son nom l'indique est censé contenir une URL. Quand il reçoit une telle requête POST, le script doit incrémenter le compteur de « likes » (local à la session uniquement, ce qui est totalement irréaliste mais répond à un souci de simplification du TP) de l'URL correspondante, et afficher « Like bien pris en compte. ». Si l'URL n'est pas dans la liste, une erreur doit s'afficher.
11. Modifier l'action `page` pour que la page renvoyée (avec une image aléatoire

seule) contienne un bouton « Like ». Cliquer sur ce bouton doit effectuer une requête POST vers l'URL `script.php?action=like&url=BLABLA` (avec BLABLA remplacé par l'URL de l'image présente sur la page).

12. Modifier le script pour qu'il reconnaisse une autre valeur du paramètre action, la valeur favorites. En réponse, le script doit générer une page HTML contenant une galerie des images les plus likées (maximum 20), avec leur nombre de likes, comme sur l'image.



Attention à bien mettre les plus likées en premier !

13. **Optionnel:** Modifier les pages HTML générées pour les actions page, gallery et history, afin qu'un clic sur une image génère un like sur cette image via une requête XHR. (Remarque : on ne demande pas ce comportement sur la page favorites, car il faudrait la maintenir à jour dynamiquement et ce serait moins simple.) Attention : cela doit fonctionner aussi pour les images ajoutées dynamiquement à la galerie via le bouton « Ajouter une image ».