

Rapport de projet développement web « PinIt »

C. DEFRETIERE et J. PAYEN

9 mai 2019

Résumé

Dans le cadre de l'unité d'enseignement « Développement Web », ce rapport sera un pseudo carnet de bord du développement du site web PinIt, il s'agit d'explicitier ce projet afin de rendre accessible le code source et de commenter la résolution des divers problèmes rencontrés.

Bonne lecture.

Table des matières

1	Présentation	3
1.1	Introduction	3
1.2	Exploration	3
1.2.1	Connexion	3
1.2.2	Interface globale	3
1.2.3	Panel gauche : gestion des tableaux	3
1.2.4	Panel droit : gestion des post-its	4
2	Installation	5
3	Développement	6
3.1	Philosophie	6
3.2	Plus en détail	7
3.2.1	Le système de connexion	7
3.2.2	Sauvegarde dynamique	7
3.2.3	Sécurité	7

1 Présentation

1.1 Introduction

Pinit vient de l'anglais "pin it" qui peut se traduire par "épingler", le but est de pouvoir dématérialiser ces petites étiquettes communément appelées "post-its". Le but de cette numérisation est de pouvoir accéder aux post-its quelle que soit l'endroit où l'on se trouve, de partager entre utilisateurs et d'obtenir une meilleure organisation.

1.2 Exploration

La page d'accueil est une page de connexion, on peut s'y connecter en entrant son pseudonyme et son mot de passe. Si c'est notre première visite on peut simplement cliquer sur "register" pour créer un compte. L'interface du site se veut facile à prendre en main et agréable d'utilisation. Tout le système de connexion est donc disponible et surtout adapté pour mobiles, malheureusement par manque de temps, le reste du site ne l'est pas.

1.2.1 Connexion

Pour une meilleure explication de l'interface, créons un compte et explorons le site.

Sur la page de connexion on clique donc sur register, ici on nous demande un nom d'utilisateur, un mot de passe et la confirmation de ce mot de passe, on s'assure ici que le nom d'utilisateur n'est pas déjà utilisé, et que les mots de passe correspondent.

1.2.2 Interface globale

Une fois connecté on peut observer deux parties distinctes, un espace de gestion sur la gauche et un espace d'affichage sur la droite.

En haut à gauche notre profil, constitué d'une photo d'utilisateur par défaut (nous n'avons pas eu le temps de faire en sorte que la photo soit modifiable), de notre pseudonyme et d'un lien pour se déconnecter.

1.2.3 Panel gauche : gestion des tableaux

Il n'apparaît pas d'autre choix que de cliquer sur le bouton "plus" vert, en bas de la page pour continuer l'exploration. En cliquant une case se crée dans l'espace de gauche, c'est un nouveau tableau, on lui donne un nom et on confirme en pressant entrée ou en cliquant ailleurs sur la page.

Quand on clique sur ce nouveau tableau des boutons de contrôle de ce dernier apparaissent. Si l'on clique sur la croix rouge cela le supprime. Une fois un tableau sélectionné, l'interface de droite lui correspond.

Pour naviguer entre les différents tableau il suffit de cliquer sur le nom ainsi créé dans la barre de gauche.

Intéressons nous maintenant aux paramètres. Lorsqu'un tableau est sélectionné on a accès au bouton paramètre situé à droite de notre profil.

On peut changer ici le nom du tableau et partager le tableau à d'autres utilisateurs. Pour cela, rien de plus simple, il nous suffit de cliquer sur le bouton "add" et de rentrer le pseudonyme de la personne à qui nous voulons le partager.

Si nous ne cochons pas la case "write" la personne ne pourra que consulter le tableau, elle ne pourra pas non plus accéder aux paramètres. Mais il s'affichera tout de même sur sa liste. Lorsque nous sommes propriétaire le titre s'affiche en gras et est vert sinon il s'affiche en noir.

Récapitulons, si un utilisateur apparaît dans le tableau des droits c'est qu'il a le droit de lecture, on peut lui rajouter le droit d'écriture et il devient ainsi une sorte d'administrateur du tableau. Le propriétaire n'apparaît pas sur le tableau des droits. Pour supprimer tous les droits de quelqu'un il suffit de cliquer sur la croix.

Pour sauvegarder les modifications des droits il suffit de fermer les paramètres avec la croix rouge en haut à droite.

1.2.4 Panel droit : gestion des post-its

Dans l'interface de droite, le petit "plus" blanc tout en bas de l'écran permet de créer ce que l'on appelle un post-it. En cliquant dessus on peut écrire du texte qui sera sauvegardé automatiquement dans le tableau si l'on rafraîchit la page ou si l'on clique ailleurs. Sur le même principe que les tableaux on remarque une croix qui apparaît lorsqu'on sélectionne un post-it, elle est destinée à la suppression.

2 Installation

Technologies testées :

- Apache HTTP Server 2.4
- PHP 7.0
- MySQL (MariaDB 10.1)

Il n'est pas obligatoire d'avoir la même configuration, mais si des problèmes surviennent, c'est peut-être de là qu'ils viennent.

Pour installer ce site web, il suffit de mettre les fichiers dans un dossier disponible sur le serveur, de configurer les variables dans le fichier `/vars.php`, et de lancer le script `/setup.php`.

Pour le lancer il suffit de le lancer via la commande `$ php setup.php` ou de naviguer sur `http://monurlperso.com/setup.php`.

Ce "setup" crée la base de données, les tables et le dossier "postits", cependant il risque d'y avoir des problèmes de droits sur ce dossier, il faut que le serveur, pour nous apache, ait les droits d'**écriture** et de **lecture**.

En cas de doutes : `$ mkdir -p postits && chmod a+rwX postits` (pas sécurisé).

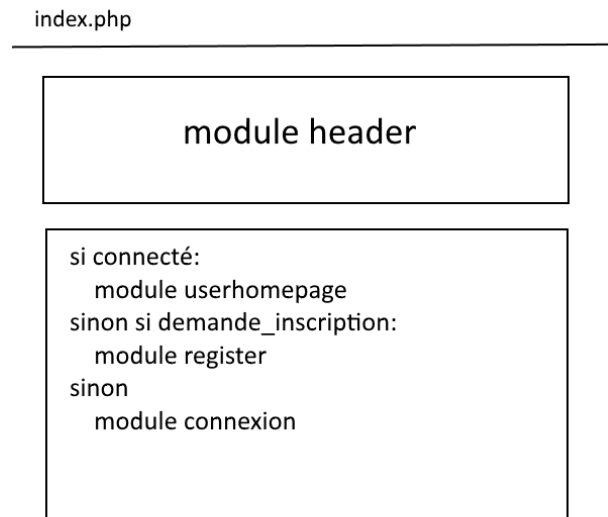
3 Développement

3.1 Philosophie

Ce projet est plus une application web qu'un site web, c'est à dire qu'il se comporte presque comme un logiciel. C'est pour cela que c'est un "single page website", c'est à dire que l'utilisateur reste sur `index.php` tout le long de son utilisation.

En fonction de la demande, php charge des "modules" dans le dossier `modules`, un module est une partie du fichier index final.

Notre fichier `index.php` se comporte donc comme un HUB :



Chaque module reçoit le code javascript et css appelé dans le module head, il est commun. Ils peuvent aussi avoir leurs propre code qui sera stocké dans `modules/js/` et `modules/css/`. Par soucis d'organisation, les scripts propres au module doivent porter le nom de ce dernier.

Exemple :

Le module `modules/foo.php` fait appel à `modules/css/foo.css` et `modules/js/foo.js`

Le HTML c'est bien, mais la plupart des balises n'avaient pas vraiment d'utilité, étant donné que l'on modifiait leurs comportement avec CSS et Javascript quasiment sans arrêt, nous le rappelons, il s'agit là d'une application web. Ainsi, nous n'avons pratiquement utilisé que des `DIV`, connaissant leurs propriétés par défaut, il est plus facile de les manipuler par la suite.

Le système de connexion est classique et est le seul à rafraichir la page via des formulaires, toute l'interface du module `userhomepage.php` est basé sur des XMLHttpRequests.

C'est donc javascript qui se charge de faire des demandes au serveur, toutes les demandes javascript sont réceptionnées par des scripts php situés dans `requests/`.

Ces scripts réalisent des tâches telles que supprimer un tableau de la base, ajouter un post-it, etc...

Tous les fichiers php peuvent se servir dans `scripts/` où des scripts php implémentent des fonctions récurrentes.

3.2 Plus en détail

3.2.1 Le système de connexion

3.2.2 Sauvegarde dynamique

3.2.3 Sécurité