

LP 2021, Web dynamique

Projet V 1.2

Travail en groupe à rendre avant le **10 janvier 2022**, 0h00 (donc le dimanche 9 au plus tard).

HotHotHot est une application dont le rôle est d'être une interface d'information et de gestion d'un système domotique.

Cette application reçoit des données en provenance de différents capteurs (ex, température, humidité, état d'un périphérique etc.)

Elle affiche ces données, leur historique si nécessaire et propose des suggestions, émet des alertes à l'attention des utilisateurs.

l'écran d'accueil comprend a minima :

Un menu avec :

- des actions liées à l'utilisateur (déconnexion , info compte)
- accès à une page de documentation

L'affichage de l'état de deux capteurs thermiques (température donc...), l'un situé en extérieur, l'autre à l'intérieur d'une pièce.

L'affichage d'un graphe synthèse des capteurs avec historique

Un système d'alertes et d'information comprenant :

- Synthèse journalière avec temp min et max de chacun
- Capteur ext > 35 degrés : alerte : Hot Hot Hot !
- Capteur ext < 0 degrés : alerte : Banquise en vue !
- Capteur int > 22 degrés : alerte : Baissez le chauffage !
- Capteur int > 50 degrés : alerte : Appelez les pompiers ou arrêtez votre barbecue !
- Capteur int < 12 : alerte : montez le chauffage ou mettez un gros pull !
- Capteur int < 0 : alerte : canalisations gelées, appelez SOS plombier - et mettez un bonnet !

Le système d'alerte permet, au clic, de faire apparaître le détail de l'alerte

Travail attendu :

Après avoir mis en commun vos travaux précédents (cours langage pour le web et création d'un framework MVC), vous réaliserez l'application HotHotHot, **en prenant soin de bien distinguer ce qui est de l'ordre du Framework et ce qui tient de la partie métier de l'application.**

Vous devrez réaliser :

1 HotHotHot comprend une rubrique « Documentation »

Il s'agit d'un ensemble de pages html liées entre elles (arborescence, parent / enfant modulable) et organisée de la manière suivante :

Application métier HotHotHot

Vous y décrirez le fonctionnement de l'application, tout ce qu'un utilisateur doit connaître (comment créer son compte, le supprimer etc). Cette partie est donc destinée au grand public.

Framework

Cette partie comprendra tout ce dont un développeur a besoin pour comprendre votre outil MVC et l'utiliser. Cette partie est donc destinée à d'autres développeurs voulant utiliser pour leurs projets votre framework.

Les pages doivent être modifiable via l'interface d'administration de l'application (CRUD).

Vous devrez également créer un système de cache permettant d'accélérer l'affichage de ces pages.

Ce système de cache est à penser en tant qu'élément du framework (nous pourrions avoir besoin de mettre en cache d'autres parties de l'application...)

Note : Un système de pagination, propre au framework devra être mis en place.

Bonus : créer votre propre éditeur wisiwig, le plus respectueux possible d'un Html accessible et donc écrit dans les règles de l'art.

Cet éditeur est évidemment à penser dans le cadre de votre framework, et pas seulement pour HotHotHot.

2 Une gestion des utilisateurs

Vous envisagerez cette partie comme un module utilisable ou non au sein du framework (donc activables ou non).

Fonctionnalités :

- Création d'un compte par inscription directe (y compris via API externe : compte google, compte Facebook, etc.)
- Connexion/ déconnexion
- récupération de mot de passe (usage d'un token unique expirant à très court terme)
- comptabilisation du nombre de connexions
- blocage du compte après N connexions avortées

Seront également stockées les informations suivantes :

- Date et heure de la dernière connexion
- Date et heure de la connexion courante

Note : chiffrement via md5, sha1 ou tout autre méthode obsolète et non sécurisée est strictement interdit !! Pour une première expérience, recherchez dans la documentation officielle de PHP ;)

3 Données Hot Hot Hot

Vous intégrerez votre travail du cours *Langages pour le web* à votre outil, le rendant ainsi pleinement fonctionnel :

- Usage des web socket en lieu et place de l'Ajx pour récupérer les données (adresse du service à venir)

Vous devrez également stocker un fichier JSON comportant un nombre assez important de valeurs, ce fichier vous servira pour le cours suivant (BigData et Nosql)...

De même, vous intégrerez à votre framework un connecteur MongoDB

<https://www.php.net/manual/fr/class.mongodb.php>

Enfin, vous mettrez en place quelques éléments d'une PWA, parmi lesquels :

- consultation en mode déconnecté de la documentation
- Installation via un bouton de l'application
- mode déconnecté pour les données Hothothot (historique et dernières données collectées).