IUT - département informatique septembre 2021
LPIOT M5 – J-M. Ilié
projet PWEB.

Un bureau virtuel
de gestion des locations de voitures

Projet à faire par équipe de 3 étudiants maximum.

BUT du projet

On considère une application web écrite en **PHP-MVC** avec un fichier frontal `index.php`. Cette application permet à une entreprise cliente de louer une flotte de véhicules.

Elle est fondée sur l'existence d'une **base de données**. Les photos des voitures seront stockées dans un sous répertoire du site web de l'application.

Synopsys du projet

Les utilisateurs de l'application sera le loueur et les entreprises clientes, il y a aussi l'administrateur du site web. Une voiture pourra être disponible à la location, ou indisponible du fait de ses nécessités de révision, ou encore loué et donc affecté à la flotte des véhicules loués par une entreprise.

Services dédiés aux entreprises non abonnées :
- afficher le site et avoir d'emblée une description des voitures du loueur, avec photos et caractéristiques principales pour chaque voiture (type, energie, BoiteAuto,).
- s'inscrire via un formulaire d'inscription. Une inscription réussie vaudra contextuellement une connexion. On notera que l'inscription donnera lieu à récupérer un numéro d'abonnement qui sera stocké sur le poste client sous la forme d'un cookie.

Services dédiés aux entreprises abonnées :
- se connecter au site en tant qu'abonné et se déconnecter.
- afficher sa flotte de véhicules en cours de location avec les dates de début et fin de location
- sélection/deselection de véhicules disponibles dans cette liste pour la location,
- spécifier des dates de début et fin pour la location d'une voiture de sa flotte. Sans date de fin, la location est supposée mensualisée (règlement pour le mois courant).

Services dédiés au loueur :
- se connecter au site en tant que loueur
- affichage des voitures du stock
- affichage des locations en cours, à savoir les flottes de voitures des entreprises clientes.
- entrée de nouvelles voitures dans le stock de voitures à louer, via un formulaire (dont upload d'une photo)
- ajout ou retrait de voitures du stock
- calcul des factures des entreprises pour le mois courant
- afficher la facture de la flotte de véhicules loués par une entreprise, avec une ligne de facturation par véhicule. Considérer une tarification différente si la durée de la location est importante (à évaluer en nb de jours ou de mois, par exemple). Il y a une réduction de 10% supplémentaire si le nombre de véhicules

de la flotte est >10. Si la durée restante de la location dépasse le mois, la facturation est mensualisée et à payer pour le mois courant.

Ce qu'il vous faudra principalement étudier :

- **La gestion du temps** nécessite de mémoriser des données. On pourra classiquement les mémoriser sous forme de chaine de caractères formatées, type `'YYYY/MM/DD'`, cependant, pour faire des opérations sur les dates, il est judicieux d'utiliser des estampilles de temps, données sous la forme d'un entier mesurant des minutes à partir d'une date de rérérence invariante. Par ce biais, vous ferez simplement des comparaisons entre 2 estampilles par exemple la différence de 2 estampilles évalue la durée les séparant. En PHP, voir les fonctions suivantes :

```
mktime() pour produire une estampille de temps à partir d'une date
date() pour produire une date (heure et mn comprise) à partir d'une estampille.
time() pour le temps courant notamment.
Une classe DateTime existe pour un plus haut niveau de gestion.
```

- les données complexes comme les tableaux peuvent être stockées sous forme de chaîne de caractères au **format `json`** qui possède 2 formes élémentaires qui peuvent se combiner: le dictionnaire d'un ensemble {'clé':valeur, …} et la liste d'un ensemble de valeurs [v1,v2,…]. En PHP, ces 2 types de structures sont vues respectivement comme un tableau associatif indexé sur les clefs et un tableau simple de valeurs, indéxé par des entiers. Voir le manuel en ligne de PHP (français) :

```
json_encode()
json_decode().
```

Description de la base de données :

La base de données contient les tables

appli [id, nomApp, codeAb]
client [id, nom, prenom, pseudo, mdp, email, jeton, datemaxjeton],
vehicule [id, type, nb, caract, photo, etatL]
facture [id, ide, idv, dateD, dateF, valeur, etatR]

Toutes les tables ont un identifiant comme clé primaire. Une clé secondaire composite est ajoutée dans facture pour des questions d'unicité.

table entreprise `//decription des entreprises clientes`
`nom` de l'entreprise, son `email` de l'entreprise, et `mdp` son mot de passe crypté (fonction sha).

table vehicule `//decription des véhicules`
type, nb : type de la voiture (exemple 206) et quantité en stock
`caract` : chaîne de caractères représentant les caractéristiques saillantes de la voiture (type d'energie, automatique, nbplaces). La chaîne representera untableau sérialisé au format JSON. Par exemple, la chaîne "[‘moteur’ :'hybride', ‘vitesse’ :'automatique']" représente un véhicule à moteur hybride avec une boite de vitesse automatique.
`photo` : nom de la photo de la voiture
etatL : chaîne de caractères représentant l'état de location de la voiture : "disponible", "en_revision" ou l'identifiant de l'entreprise qui loue le véhicule en cas de location en cours.

table facture `//historique des facturations`
type, nb : type de la voiture (exemple 206) et quantité en stock
idv et ide : identifiant du véhicule et de l'entreprise
dateD et dateF : estampille qualifiant la date de début et la date de fin d'une location
valeur : valeur numéraire à règler
etatR : indication booleenne du reglement, fait ou non.