PPE 2 : Projet Appli-Frais

Rémy Petit, Rémy Lefebvre, Lilian Séon

12/05/2016

SOMMAIRE

-Responsabilité de l'équipe ……..

-Contexte ……

-Cahier des charges …….

Définition de l'objet …….

Forme de l'objet …….

Sécurité ……..

-Contrainte …….

-Structure de la base de données…….

-Mise en place de la base de données …….

-Vérification de la connexion Visiteur/Comptable …….

-Cryptage du mot de passe ……

-Saisie des frais forfaitaires des visiteurs ……

-Saisie des frais hors forfait des visiteurs ……

-Validation des fiches par les comptables ……..

-Fonctions supplémentaires……..

-Conclusion …….

Responsabilité de l’équipe

Notre équipe aura la responsabilité d'améliorer le processus actuel de remboursement des frais de déplacement.

Les visiteurs devront effectuer une saisie en ligne de leurs fiches de frais liées à leurs déplacements comme celui des repas aux restaurants, les nuits à l’hôtel, les frais kilométriques ainsi que le forfait « Etape » afin de pouvoir être remboursés par l’entreprise.

Notre équipe devra également créer une table comptable et modifier le code de connexion pour que les comptables puissent également se connecter aux sites afin d’effectuer les taches de clôturassions, validations et remboursements des fiches de frais correspondant aux différents visiteurs. Ce projet va se faire principalement en 5 étapes :

- Création de la base de données GSB_FRAIS

- Vérification que l’insertion et la consultation des fiches de frais est possible par le visiteur

- Modification de l’interface de connexion pour permettre la connexion du personnel comptable.

- Création de la page Validation des fiches de frais.

- Création d’un bouton permettant la clôturassions des fiches de frais

- Création d’un bouton permettant le remboursement des fiches de frais

Contexte GSB

Suite à la fusion des groupes dont est issu le laboratoire GSB, une recherche d'optimisation de l'activité du groupe est alors souhaitée. Etant donné que le nerf de la guerre d'un groupe pharmaceutique se trouve dans les démarches d'informations et de conseils effectuées auprès des prescripteurs et que les équipes sur le terrain ont été déstabilisées par le mouvement de fusions récentes, l'objectif est ici de pouvoir gagner une plus grande visibilité ainsi qu'une efficacité plus importante des activités effectuée auprès des métiers de la santé pouvant potentiellement prescrire les produits du laboratoire.

En effet, les habitudes de chacun n'ayant pas ou peu changées après les fusions, il n'y a pas eu d'harmonisation entre les différents acteurs sur le terrain et les responsables de secteurs.

Toujours dans l'optique d'une amélioration de l'activité des visiteurs médicaux, on aspire à valoriser et optimiser entre le personnel du terrain et les différents services du siège parisiens. Il faut une gestion du suivi des activités sur le terrain uniforme et disponible aisément pour tous les acteurs mobiles.

Cahier des charges

Définition de l’objet

L'application doit permettre d'enregistrer tout frais engagé, aussi bien pour l'activité directe (déplacement, restauration et hébergement) que pour les activités annexes (événementiel, conférences, autres) et de présenter un suivi daté des opérations menées par le service comptable (réception des pièces, validation de la demande de remboursement, mise en paiement, remboursement effectué).

Forme de l’objet

L’objet pour les visiteurs et les comptables aura la forme d’une application web disponible sur ordinateur seulement.

Sécurité

L’application est accessible uniquement après une authentification, ce qui signifie que seul le personnel de l’entreprise aura accès au site.

Le comptable et le visiteur médical ont des comptes distincts.

De plus les différents échanges avec la base de données sont cryptés et les utilisateurs ne peuvent pas accéder aux fichiers de fonctions du site.

Contraintes

L’architecture

L’application respectera l’architecture des scripts fournis ainsi que les noms des fichiers.

Ergonomie

L’ergonomie de la page web a été définie lors d’une consultation. Mais des modifications peuvent y être ajoutées.

Environnement

Le langage de script côté serveur doit être le même que celui utilisé dans les pages fournies.

Normes

L’équipe devra respecter les normes de développement en PHP et HTML ainsi que les normes W3C.

Structure de la base de donnée

La base de donnée est constituée de 4 quatre tables. Seule la table utilisateur ne sert pas à contenir des frais saisi par un visiteur et leur état respectifs.

Serveur: Local Databases » Base de données: gsb_frais

Structure | SQL | Rechercher | Requête | Exporter | Importer | Opérations | Privilèges | plus

Table	Action	Lignes	Type	Interclassement	Taille	Perte
horsclassification	Afficher Structure Rechercher Insérer Vider Supprimer	6	InnoDB	utf8_general_ci	16 Kio	-
ligneforfait	Afficher Structure Rechercher Insérer Vider Supprimer	5	InnoDB	utf8_general_ci	16 Kio	-
lignehorsforfait	Afficher Structure Rechercher Insérer Vider Supprimer	12	InnoDB	utf8_general_ci	16 Kio	-
utilisateur	Afficher Structure Rechercher Insérer Vider Supprimer	5	InnoDB	utf8_general_ci	16 Kio	-
4 tables	Somme	28	MyISAM	utf8_general_ci	64 Kio	0 o

La table horsclassification contient les frais hors classification :

Ses élements porte le tag « hc ».

L'id, la situation et la DateOp (pour date opération) ne porte pas ce tag car ils sont communs avec les frais qui « vont avec » dans les autres tables.

De plus, les valeurs de DateOp et situation sont NULL par défaut, en effet il faut leur accorder une valeur le temps que le comptable s'occupe du frais en question.

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
1	**Id**	int(11)			Non	*Aucune*	AUTO_INCREMENT
2	**hcVisiteur**	varchar(50)	utf8_general_ci		Non	*Aucune*	
3	**hcMois**	varchar(20)	utf8_general_ci		Non	*Aucune*	
4	**hcNbJustificatif**	int(11)			Non	*Aucune*	
5	**hcMontantTotal**	int(11)			Non	*Aucune*	
6	**Situation**	tinyint(3)			Oui	*NULL*	
7	**DateOp**	datetime			Oui	*NULL*	

La table ligneforfait contient les frais au forfait et leur état respectifs.

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
1	**Id**	int(11)			Non	*Aucune*	AUTO_INCREMENT
2	**lfVisiteur**	varchar(30)	utf8_general_ci		Non	*Aucune*	
3	**lfMois**	varchar(20)	utf8_general_ci		Non	*Aucune*	
4	**lfRepas**	int(15)			Non	*Aucune*	
5	**lfNuitee**	int(15)			Non	*Aucune*	
6	**lfEtape**	int(15)			Non	*Aucune*	
7	**lfKm**	int(15)			Non	*Aucune*	
8	**Situation**	tinyint(3)			Oui	*NULL*	
9	**DateOp**	datetime			Oui	*NULL*	

Table horsforfait, contient les frais hors forfait saisi par le visiteur et leur état respectif entré par le comptable :

	#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
☐	1	**Id**	int(11)			Non	*Aucune*	AUTO_INCREMENT
☐	2	**lhVisiteur**	varchar(255)	utf8_general_ci		Non	*Aucune*	
☐	3	**lhMois**	varchar(20)	utf8_general_ci		Non	*Aucune*	
☐	4	**lhLibelle**	varchar(80)	utf8_general_ci		Non	*Aucune*	
☐	5	**lhDate**	date			Non	*Aucune*	
☐	6	**lhQte**	int(10)			Non	*Aucune*	
☐	7	**Situation**	tinyint(3)			Oui	*NULL*	
☐	8	**DateOp**	datetime			Oui	*NULL*	

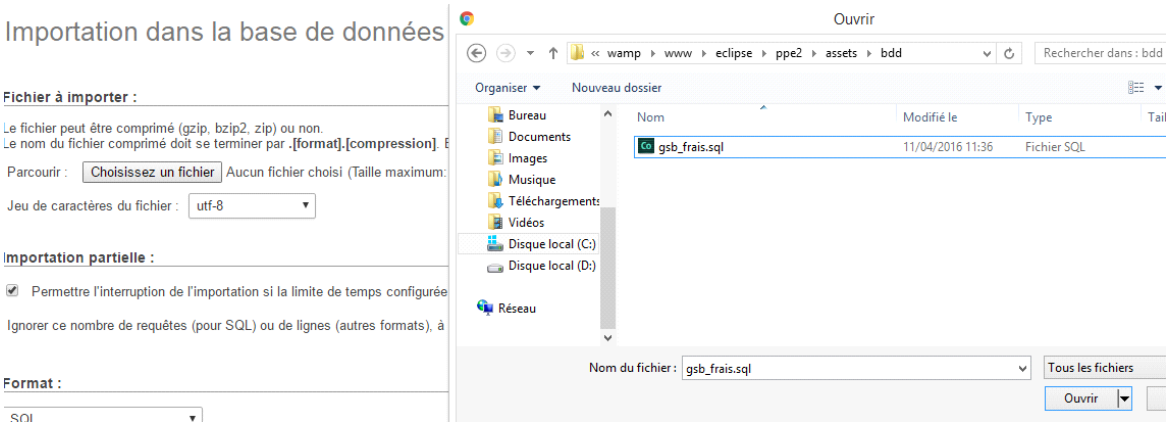
Enfin, la table utilisateur :

	#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
☐	1	**uId**	int(11)			Non	*Aucune*	AUTO_INCREMENT
☐	2	**uType**	int(4)			Non	*Aucune*	
☐	3	**uNom**	varchar(50)	utf8_general_ci		Non	*Aucune*	
☐	4	**uPrenom**	varchar(50)	utf8_general_ci		Non	*Aucune*	
☐	5	**uLogin**	varchar(50)	utf8_general_ci		Non	*Aucune*	
☐	6	**uMdp**	varchar(255)	utf8_general_ci		Non	*Aucune*	
☐	7	**uAdresse**	varchar(254)	utf8_general_ci		Non	*Aucune*	
☐	8	**uCp**	varchar(6)	utf8_general_ci		Non	*Aucune*	
☐	9	**uVille**	varchar(50)	utf8_general_ci		Non	*Aucune*	
☐	10	**uDateEmbauche**	date			Non	*Aucune*	

Mise en place de la base de données

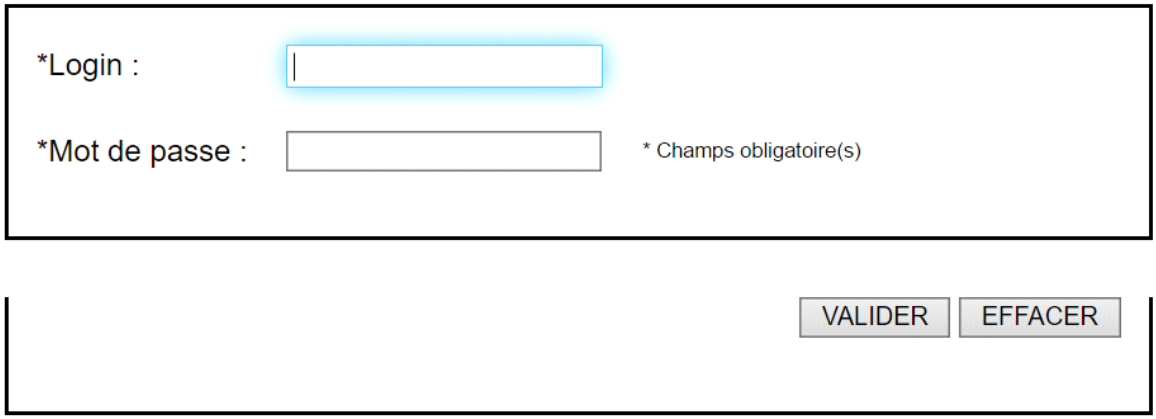
Premièrement il faut se connecter au server local comme Wamp / Lamp / Mamp (en fonction du système d'exploitation), puis ouvrir phpMyAdmin et créer la base de données « gsb_frais ».

Suite à la création de la base de données, il faut importer le ficher SQL contenant la structure de la base de données.



Identification au site par le visiteur et le comptable

Les comptables et les visiteurs ont des login et des mots de passe dédié pour se connecter au site. La vérification du mot de passe, du login et du type de l'utilisateur va être faite par le script *assets/fonctions_php/login.lib.php*



Annexe : annexes\php\login.lib.php.odt

Ce script permet de fermé la session en cours.

Chaque page utilisateur possède un bouton de déconnection qui active ce script.

```
<?php

    $_SESSION = array ();
    session_destroy ();
    header ( "Location:../../index.php" );

?>
```

Cryptage du mot de passe

Le cryptage du mot de passe ce fait grâce à « sha1 » qui veux dire *US Secure Hash Algorithm 1.* Le sha1 est une fonction de hachage cryptographique conçue par la NSA. Cette fonction produit un résultat de 160 bits.

```
if ($data['uMdp'] != sha1 ( $_POST['uMdp'] )) {
```

La saisie des frais forfaitaires des visiteurs

Pour la saisie des frais forfaitaire, les visiteurs doivent simplement remplir le montant des repas, nuitée forfait étape et frais kilométrique. La partie période d'engagement se rempli automatiquement en fonction de la date du jour où il remplit le formulaire.

Les champs Mois et Année sont rempli automatiquement et impossible à changer par le visiteur grace à l'attribut readonly.

```
<input type="text" id="inputannee" value="<?php echo date_annee(); ?>" name="annee4c" readonly class="mini_radius" />
<input class="mini_radius" type="text" id="inputmois" value="<?php echo date_mois(); ?>" name="mois2c" readonly />
```

Les fonctions appelées ici :

assets/fonctions_php/date.lib.php

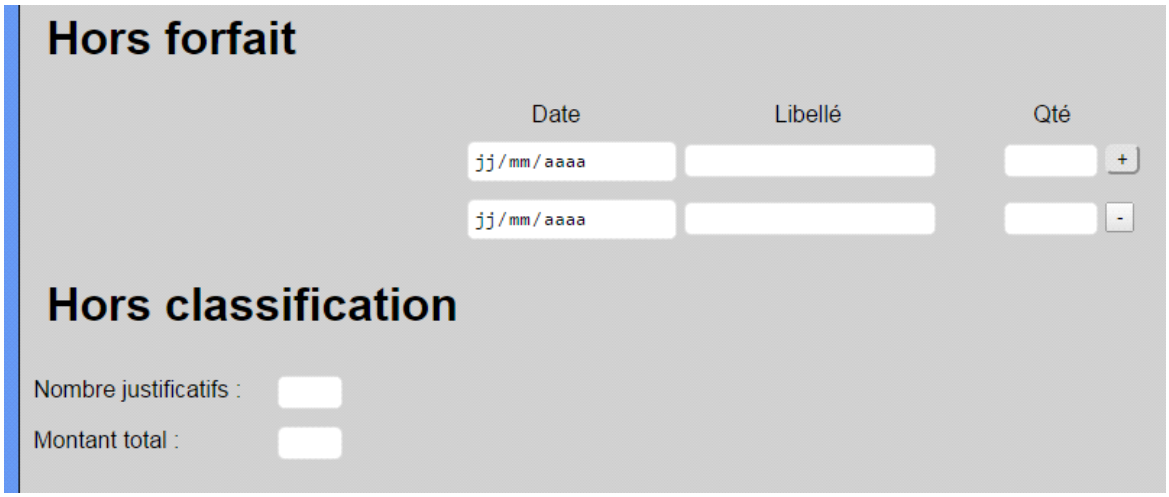
Annexe : annexes\php\date.lib.php.odt

Ces deux inputs sont donc les valeurs envoyé **$_POST['annee4c']** et **$_POST['mois2c']**

La saisie des frais hors forfait des visiteurs

Pour indiquer les frais hors forfait, les visiteurs doivent saisir la date, le libellé et la quantité(prix).

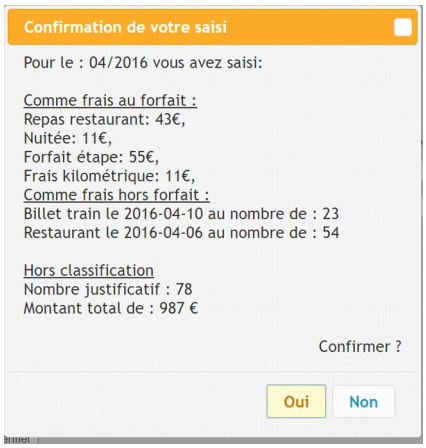
Il est possible d'envoyer plusieurs lignes en même temps grâce au bouton qui permet d'ajouter (ou retirer) des lignes.



Le fichier *assets/js/ProgLigne.js* s'occupe de gérer les créations et suppression de ligne, ainsi que la gestion de leur name respectif.

Annexe pour ProgLigne.js : annexes\javascript\ProgLigne.js.odt

Une fois le remplissage du formulaire terminé, le visiteur a un bouton de vérification en bas de page, cliquer dessus ouvrira une popup via jQuery-ui affichant les données saisis, lui permettant des vérifier, exemple :



Le fichier *assets/js/ProgConfirm.js* crée et rempli cette popup,

Annexe : annexes\javascript\ProgConfirm.js.odt

Dans le cas où le visiteur clique sur le bouton « Oui » alors le formulaire est submit de par son attribut action :

```
<form name="formu_frais" action="fonctions_php/envoi_saisi.lib.php" method="POST">
```

Ici submit le formulaire exécutera la fonction assets/fonctions_php/envoi_saisi.lib.php décrite dans l'**annexe annexes\php\envoi_saisi.lib.php.odt** qui insérera les données saisi par le visiteur dans la base de données.

La validation des fiches par les comptables

Le comptable peut choisir de quel visiteur il veut s'occuper.

Ensuite il peut visionner les différents frais que le visiteur a remplis au préalable qui s'affiche automatiquement dans les tableaux.

Puis il n'a plus qu'à sélectionner la situation et les frais seront soit enregistré, validé ou remboursé. Avec une simple sélection la base de donnée est mise à jour dès le moment où le comptable clique.

Les scripts utilisés ici sont :

assets/fonctions_php/liste_visiteur.php pour le select affichant la liste des utilisateurs de type visiteur médical, **annexe :**

annexes\php\liste_visiteur.php.odt

assets/js/validationAjax.js

Annexe : annexes\javascript\validationAjax.js.odt

assets/fonctions_php/validationAjax.lib.php

Annexe : annexes\php\validationAjax.php.odt

Ces deux fichiers fonctionnent ensemble pour effectuer une requête AJAX qui va permettre d'afficher les frais du visiteurs sélectionnés, dépendant du mois courant, instantanément et sans recharger la page, ainsi le comptable a très rapidement accès aux frais entrés préalablement par le visiteur choisi et peut en choisir la situation.

Une fois les frais affichés, le comptable doit faire un choix pour dire qu'il a enregistré, validé ou remboursé le frais, encore une fois c'est une requête AJAX qui va mettre à jour la base de donnée DÈS que le comptable va cliquer sur l'option, ce qui justifie l'absence d'un bouton valider ici.

Les fichiers utilisés pour ces requêtes ajax de mise à jour sont :

/assets/js/EnvoiValidAjax.js

Annexe : annexes\javascript\EnvoiValidAjax.js.odt

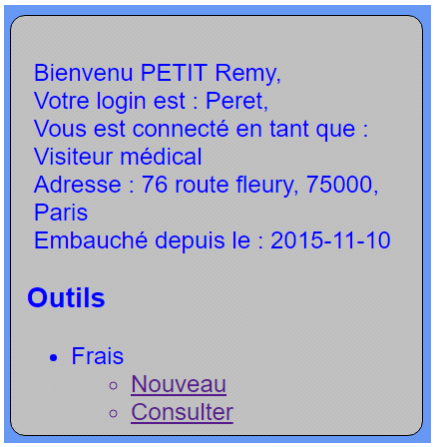
Et */assets/fonctions_php/EnvoiValidation.lib.php*

Annexe : annexes\php\EnvoiValidation.lib.php.odt

Consultation des frais par le visiteur

Le visiteur possède à gauche de sa page des « outils » :

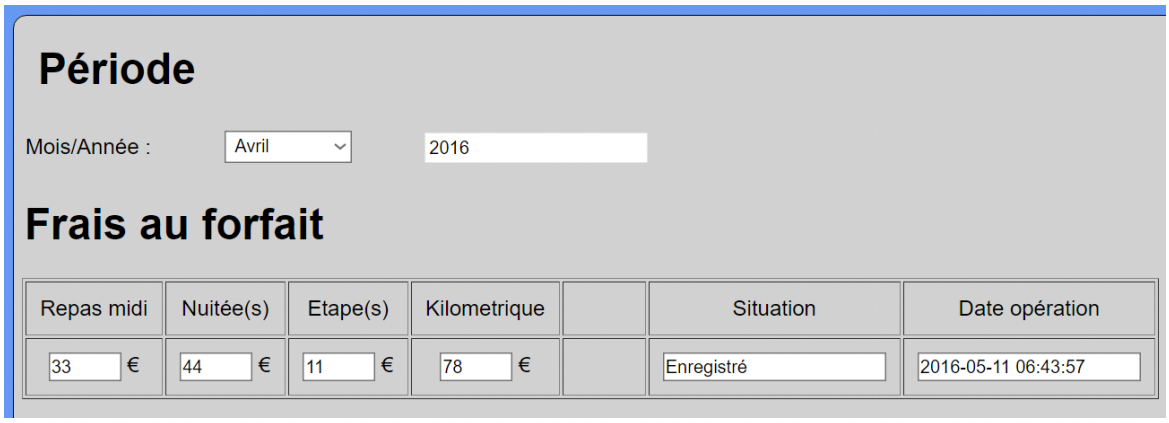
Et tout comme le comptable, une description de son profil utilisateur est affichée mais nous y reviendront plus tard.



En cliquant sur Consulter, le visiteur est dirigé vers la page consultation_frais.php

Cette page lui permet d'afficher les frais qu'il a déjà rentré et surtout, de consulter leurs traitement par le comptable.

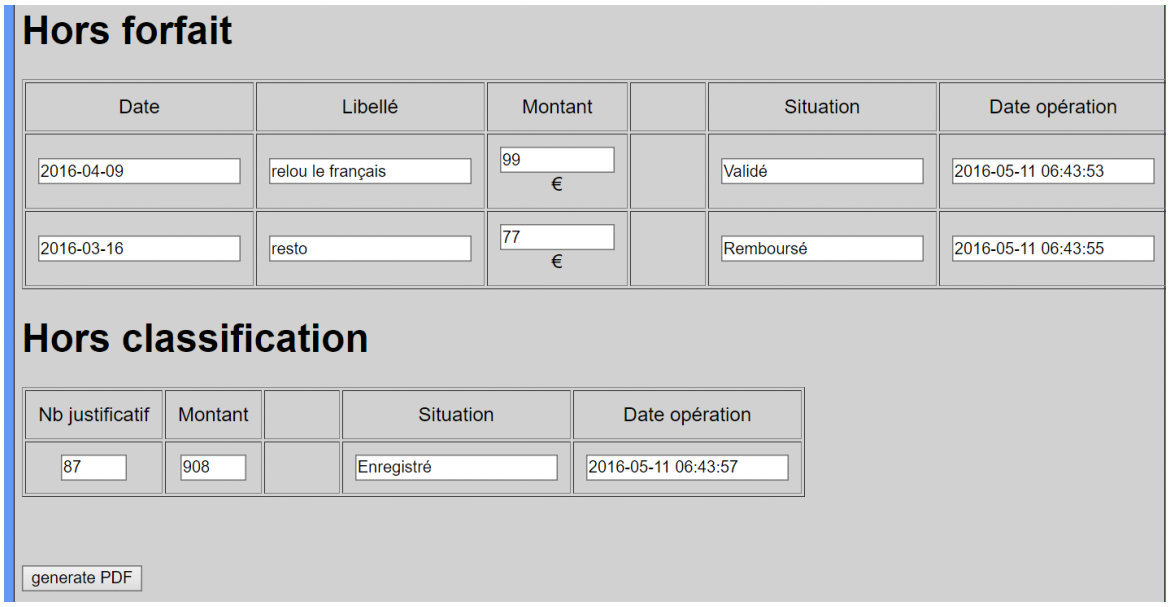
Front :



Ici on peut voir ce que notre utilisateur, pour Avril 2016 a rempli, mais aussi que le comptable a « Enregistré » ces frais, à la date et l'heure inscrite à côté.

Il en va de même pour les frais hors forfaits et hors classification, dans le cas où le frais n'a pas encore été traité, la valeur dans la base de donnée sera null pour situation et date opération, alors

le script inscrira « En cours de traitement » dans le champs.



Hors forfait

Date	Libellé	Montant		Situation	Date opération
2016-04-09	relou le français	99 €		Validé	2016-05-11 06:43:53
2016-03-16	resto	77 €		Remboursé	2016-05-11 06:43:55

Hors classification

Nb justificatif	Montant		Situation	Date opération
87	908		Enregistré	2016-05-11 06:43:57

generate PDF

De plus, il est impossible pour le visiteur de changer ces valeurs grâce à l'attribut readonly vu plus haut.

Cet affichage est également réalisé à l'aide de AJAX qui permet d'afficher les valeurs et créer les tableaux de données sans recharger la page.

Les scripts : */assets/js/ConsultAjax.js*

Annexe : annexes\javascript\ConsultAjax.js.odt

Et le script php : */assets/fonctions_php/ConsultAjax.lib.php*

Annexe : annexes\php\ConsultAjax.lib.php.odt

Le bouton generate PDF permet de générer un PDF à partir des informations affichées sur la page, permettant ainsi aux visiteurs d'imprimer ses fiches de frais.

Annexe : annexes\javascript\jspdf.odt

Fonctions supplémentaires :

Dans l'objectif de nous simplifier la tâche et de rendre plus ergonomique l'application, nous avons créé une fonction permettant d'afficher en permanence la description de l'utilisateur connecté, ainsi qu'une fonction qui permet « d'inscrire » un utilisateur dans la base de donnée.

Premièrement, la fonction pour afficher les informations sur la session en cours :

/assets/fonctions_php/infoconnect.lib.php

Appel de la fonction :

```
<?php
    echo '<br/>'.infoconnect();
?>
```

Annexe : annexes\php\infoconnect.php.odt

Deuxièmement, la fonction insert_bdd, il s'agit d'une fonction d'inscription que nous avons créée pour pouvoir effectuer nos tests plus efficacement.

Page : *assets/fausse_inscription.php*

Script : *assets/fonctions_php/insert_bdd.lib.php*

Annexe : annexes\php\insert_bdd.lib.php.odt

Conclusion

Nous avons réalisé ce projet en groupe de 3, chacun avec des tâches différentes.

Rémi était chargé de la partie graphique de l'application,

Lilian d'une partie du codage des pages html et de la documentation,

Rémy de la totalité du codage back et la gestion de la base de donnée.

Durant la réalisation de ce projet nous avons subi quelques difficultés, la première étant nos faibles connaissances pour le langage JavaScript et la découverte d'Ajax ensuite, nous ne savions pas traiter plusieurs tables en même temps dans une base de données.
Persévérance et nuit de sommeil loupé nous ont permis ensemble de mener à bien ce projet, de le rendre aussi fonctionnel, rapide, optimisé et ergonomique possible, autant pour les utilisateurs que les administrateurs.

Nous avons réalisé ensemble une application web permettant de gérer des frais comptables, et plusieurs types d'utilisateurs. Nous avons tous trois muri grâce à cette expérience et sommes fier du résultat, surtout au niveau des requêtes Ajax qui étaient tout sauf gagnées.