

TP PHP (/HTML/CSS/PostgreSQL)

ENSIIE - 2017/2018

Le but de ce TP est de compléter les fichiers fournis pour réaliser une micro-application interactive qui permet, suivant le choix de l'utilisateur, de :

- visualiser le nom et le débit d'un client dont il saisit le numéro ;
- modifier le nom et le débit du client dont il saisit le numéro ;
- ajouter un achat pour un client donné ;
- créer un nouveau client en saisissant son numéro, son nom et son débit initial.

Remarque : Dans la suite, *login* est à remplacer par votre login de salle TP.

Partie 1 - Mise en place du TP

Une base de données `tpphp` a été créée, qui contient les tables suivantes :

```
client(num_client, nom_client, debit_client
achat(num_achat, montant_achat, date_achat, client)
```

Pour vous connecter, vous pouvez utiliser la commande suivante (mais ce n'est pas nécessaire) :

```
psql -h pgsq12 -d tpphp
avec le login tpphp et le mot de passe tpphp.
```

1 - Si vous ne l'avez pas encore fait, créez un répertoire `~/public_html`, et créez-y un sous-répertoire `tp2`.

- Enregistrez dans votre home l'archive contenant les fichiers du TP à partir de ce lien.
- Déplacez l'archive dans le répertoire `~/public_html/tp2` :
`mv tpPHP.tar tp2/`
- Déplacez-vous dans le répertoire `tp2`, puis extrayez de l'archive les fichiers nécessaires au TP :
`tar xvf tpPHP.tar`
- Modifiez si nécessaire les droits sur les répertoires `~/public_html`, `tp2` et `~` pour les rendre traversables (`chmod o+x`), et sur les fichiers de `tp2` pour qu'ils soient lisibles par tous (`chmod o+r`).

Le code contient :

- des fichiers correspondant à une partie contrôleur :
 - un fichier `index.php` qui est la page principale de l'application et comprend le menu de choix ;
 - des fichiers `tpGestionClient.php`, `tpCreationClient.php`, `tpModifClient.php` et `tpNvelAchat.php`, qui sont les contrôleurs correspondant aux différentes parties de l'application.
- des fichiers correspondant à une partie modèle :
 - un module gérant le lien avec la base de données `db.php` ;
 - un fichier `tpModele.php` comprenant le code d'accès aux données ;
 - un fichier de configuration `config.php` ;
 - des pages gérant la connexion et déconnexion de l'utilisateur : `tpConnexion`, `tpVerifMDP.php`, et `tpQuitter.php`.
- des fichiers correspondant à une partie vue :
 - un fichier `tpVue.php` comprenant les affichages ;
 - une feuille de style `tpStyle.css`.

Partie 2 - Affichage des pages

Le travail consiste à compléter, suivant les indications données ci-après, les différents fichiers PHP fournis pour réaliser la micro application de gestion client. Dans le texte des pages PHP, les parties à ajouter ou modifier sont en commentaire. Après chaque modification, il faut tester si la page se comporte comme prévu.

Pour tester :

2 - Lancez un navigateur (firefox ou autre) sur votre machine.

3 - Allez à l'URL : `http://pgsql/~login/tp2`
ou `http://pgsql.ensiie.fr/~login/tp2`

ATTENTION : respectez les noms des variables utilisées dans les programmes fournis que vous devez compléter

Partie 3 - Modification du contenu statique et de la présentation de la page d'accueil

4 - Complétez la fonction `pied()` dans le fichier `tpVue.php` afin qu'elle génère les balises fermantes pour le corps des pages (`</body>`) et pour le document HTML (`</html>`).

5 - Complétez la fonction `affiche_menu()`, toujours dans le fichier `tpVue.php`, afin qu'elle affiche le menu complet permettant de choisir entre Visualisation, Modification Création et Achat (voir figure 1).

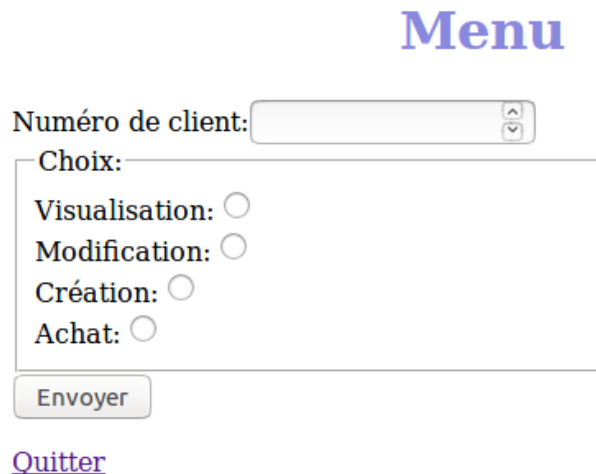


FIGURE 1 – Page index.php

Vérifiez que le code source de la page `index.php` est bien en HTML valide.

Remarque : Si vous voulez tester la validité du code HTML de vos pages, vous pouvez copier/coller le code source de vos pages et demander la validation directe de ce code au validateur du W3C.

6 - Modifiez la feuille de style fournie (`tpStyle.css`) afin d'afficher les titres de page dans la couleur de votre choix (vous pouvez consulter la page de w3schools concernant les couleurs pour vous aider).

Partie 4 - Réalisation de l'application

7 - Dans le fichier `tpModele.php`, complétez la fonction `get_client()`. Il faut faire exécuter une requête `SELECT` au SGBD, en utilisant le module `db.php`. Cette requête retourne soit une erreur, soit un seul tuple. Transmettez le résultat sous le format indiqué dans le commentaire.

8 - Dans `tpGestionClient.php`, dans le cas « v » (visualisation), affichez le résultat de l'exécution de la fonction `get_client()` (voir figure 2), en veillant bien à prendre en compte tous les cas d'erreur.

9 - Dans le fichier `tpVue.php`, complétez la fonction `retour_menu()`.
Si tout fonctionne, la partie consultation d'un client est complètement réalisée.

Gestion des clients

Consultation du numéro client : 2

Nom: Kent, Débit: -0,60€

[Menu](#)

[Quitter](#)

FIGURE 2 – Consultation des informations sur un client

10 - Dans `tpVue.php`, complétez la fonction `affiche_form_modif()`, qui crée un formulaire envoyant vers la page `tpModifClient.php`, transmet le numéro de client en champ caché, et demande les nouveaux nom et débit souhaités (schéma - écran 4).

11 - Dans `tpGestionClient.php`, complétez ensuite le cas « m », modification du débit du client, pour faire appel à la fonction `get_client` puis à `affiche_form_modif()` ou afficher une erreur.

12 - Complétez `tpModifClient.php`. (schéma - écran 5). La partie « m », modification du débit du client est alors complètement réalisée.

13 - Dans `tpGestionClient.php`, complétez ensuite le cas « a », achat par un client, pour faire appel à la fonction `get_client` puis à `affiche_form_achat()` ou afficher une erreur.

14 - Dans `tpVue.php`, complétez la fonction `affiche_form_achat()`, qui crée un formulaire envoyant vers la page `tpNvelAchat.php`, transmet le numéro de client en champ caché, et indiquant le montant de l'achat.

15 - Dans `tpModele.php`, complétez la fonction `insert_achat()`. Attention, il y a deux requêtes à faire : mise à jour d'un client, et insertion d'un nouvel achat, et ces requêtes doivent être faites toutes les deux, ou aucune des deux. On utilisera donc une transaction afin de s'assurer que la base de données reste cohérente.

16 - Modifiez la valeur de la variable `AUTHENT` dans `config.php` en la passant à 1 pour que l'authentification des utilisateurs soit requise. Complétez la fonction `verif_authent` dans `tpModele.php`.

17 - Complétez le fichier `tpVerifMDP.php`.

18 - Réalisez la partie « c », création du client : saisie du numéro de client, nom et débit du client. Créez le client dans la base de données en vérifiant que le numéro n'existe pas déjà, affichez un message suivant le résultat de la création et affichez le menu.

19 - Si cela n'a pas été fait au fur et à mesure du TP, reprendre le code de ce TP pour le rendre aussi modulaire et générique que possible.

Si vous pensez avoir réalisé une amélioration majeure du code de ce TP, n'hésitez pas à nous envoyer votre code par mail en expliquant les améliorations effectuées et à gagner jusqu'à deux points sur votre note de programmation web.