STS SN IR 1 **TP Environnement Internet**

Objectifs :
- Créer une table avec l'interface PhpMyAdmin
- Réaliser des classes Php
- Lire, écrire et se connecter à une base de données Sql avec l'API mysqli
- Créer un formulaire Html avec vérification en javascript
- Rédiger une synthèse permettant une utilisation ultérieure des méthodes , de réaliser une connexion et requête SQL en gérant notamment les dates.

But : Réaliser un livre d'or .

Cahier des charges : les visiteurs doivent pouvoir laisser un message et consulter les messages des visiteurs précédents.

Travail demandé :

1) PHPMyAdmin

- A partir du fichier *bavardage.sql* créer la table *bavardage* dans votre BDD sur le serveur **172.16.254.8**.

2) Dans le fichier *index.php* :

2.1) Après la lecture et modification du fichier *login.php*, transmettre les variables correctes au constructeur de la classe **`livreDorDAO`** (ligne 15)

2.2) Visualiser les paramètres contenus dans le tableau REQUEST, notamment "recherche" et "monbouton", suite à des clics sur les boutons envoi, Ok et radioboutons;
Ceci afin de comprendre le fonctionnement des tests sur *recherche* et *monbouton*.

3) classe `livreDorDAO`

3.1) Modifier la méthode **`ajoute`,** afin d'enregistrer le nom, le message et l'adresse en plus de la date.
3.2) Cette méthode doit retourner **`true`** si un des arguments reçus est vide.
3.3) Compléter et tester la méthode **`cherche`**(ligne 55) afin d'ajouter une restriction: `$champ='$valeur'`.On ne doit afficher que les messages de la personne recherchée.

classe Ihm

4 HTML CSS

4.1 méthode `afficheInviteFormulaire`

- Implémenter cette méthode afin de définir une ancre "**`haut`**" qui servira pour un lien *haut de page* placé en bas de la page
- Créer un lien avec le mot formulaire . Ce lien doit diriger vers le formulaire.
- Ce lien sera inclus dans une balise `label` de classe CSS " `invite ` " déjà définie dans le fichier de style.

4.2 Déterminer comment un clic dans le champ message du formulaire efface le texte prédéfini.

4.3) méthode `afficheRecherche`

- 4.3.1 Le nom ou l'adresse email affichés au-dessus du tableau doivent être inclus dans une balise `label` de classe CSS " `bleu` "
- 4.3.2 Modifier le label précédent afin qu'il soit inclus dans un label " `bleuGras` ".
  Créer au préalable la classe CSS " `bleuGras` " qui reprend la classe CSS " `bleu` " et ajoute le texte en gras(`font-weight: bold`)
- 4.3.3 Donner le style " `cadre` " au tableau affichant le résultat de la recherche.

5 PHP SQL

méthode `afficheRecherche`

.-5.1 Le nombre de messages doit être affiché au dessus du tableau affichant le résultat de la recherche.

-5.2 Modifier cette méthode `afficheRecherche` afin qu'elle n'affiche pas que le premier message mais tous les messages envoyés par l'usager recherché.

6 Améliorations

6.1) Modifier ou créer le formulaire afin de proposer un affichage ordonné par *date*(cas par défaut) ou par *nom*. .

6.2) Proposer un affichage par dates croissantes ou décroissantes

6.3) Modifier le code afin que la vérification du formulaire s'effectue localement sur la machine du visiteur donc en javascript.

6) En PHP ou javascript, vérifier la présence de '@' dans l'email

Fichiers joints: livredor.zip

Démo sur http://btsirisinfo.free.fr/aide/livredor (début)

Aide Php sur http://php.net/manual/fr/indexes.functions.php

Critères d'évaluation :

- degré d'autonomie + niveau d'avancement du travail
- rédaction du compte-rendu