

 3iLINGENIEURS </les sciences informatiques>		Année : 2022/2023
NOM :		
Prénom :	Web – J. Grasset	15/03/2023
Groupe :		

Partie pratique

Contexte et éléments fournis

Un ex-informaticien spécialiste du développement de chat-bots (ou robots conversationnels, comme ChatGPT) décide de changer de carrière et de s'éloigner du numérique. Il se reconvertit dans l'élevage et le dressage de chats (les petits animaux...), poussant très loin leur éducation pour qu'ils puissent accomplir des tâches complexes et assister les personnes qui voudront les adopter.

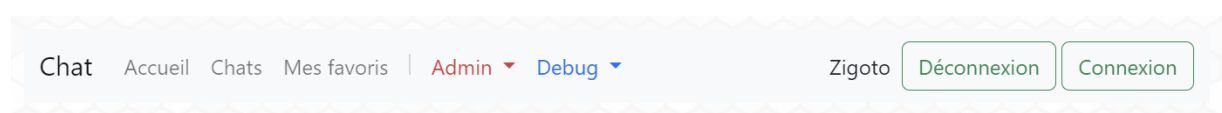
Le prototype d'application web dont vous disposez permet de présenter ces chats, avec la possibilité pour des utilisateurs enregistrés de mémoriser des favoris.

Il y a 2 rôles avec privilège :

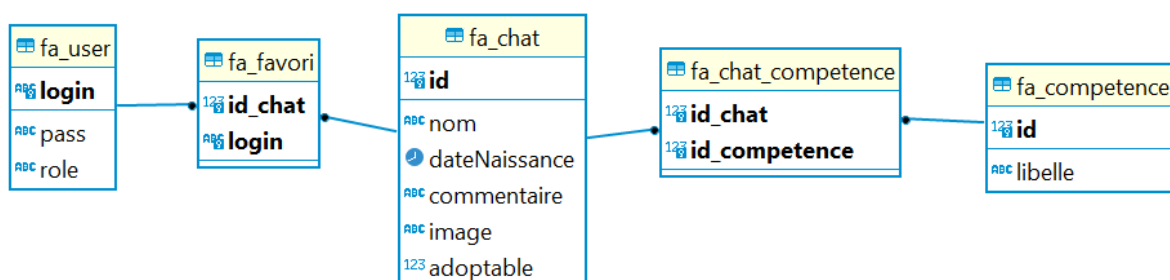
- admin, pour un administrateur
- inscrit, pour un visiteur enregistré

Dans le menu :

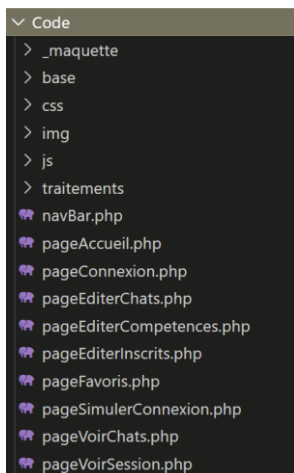
- Tous les liens existent, mais les fonctionnalités ne sont pas toutes implémentées
- La rubrique Admin (en rouge) donne accès à des pages qui seront à terme réservées aux administrateurs. Ces pages s'affichent avec un fond rouge pour vous aider à vous repérer.
- La rubrique Debug (en bleu) donne accès à des pages destinées aux développeurs. Elles s'affichent sur fond bleu.



Le schéma relationnel de la base de données fournie (script à la racine du zip):



Structure de l'arborescence du code :



- A la **racine** : la navBar (menu) et les pages visibles pour les utilisateurs
- Dossier **base** : l'instanciation d'un objet PDO, avec informations de connexion à modifier
- Dossiers **css, js, img** : comme le nom l'indique
- Dossier **traitements** : des scripts destinés à réaliser une action puis à rediriger ou retourner une valeur.
- Les codes javascript utilisés par les pages sont définis dans des balises script en fin de chaque fichier et non dans un fichier externe (en supposant que cela pouvait faciliter votre travail). Voir [pageEditerChats.php](#) par exemple.

Remarque : pour faciliter une lecture et compréhension rapides et vous faire gagner un peu de temps en EI il n'y a pratiquement aucun des contrôles basiques indispensables pour apporter un peu de robustesse (isset par exemple). Ne prenez pas ces codes en exemple sur ce plan pour vos TP/projets, par exemple...

Attention : une erreur de ma part, j'ai donné le même nom, connexion.php à 2 fichiers. L'un dans le dossier base qui contient l'instanciation de PDO, l'autre dans le dossier traitements qui est vide. Ne les confondez pas.

0. Mise en place

Récupérez les codes source fournis et les scripts pour la base de données. Installez le tout sur votre serveur.

N'oubliez pas de modifier le fichier base/connexion.php pour utiliser vos identifiants et votre base.

1. Affichage des chats



La page [pageVoirChats.php](#) affiche la liste des chats pour le visiteur.

Question

Ajoutez sur la carte de chaque chat les informations suivantes :

- son commentaire

- la liste de ses compétences (sous la forme d'une liste numérotée). Une requête est fournie et déjà exécutée pour chaque chat, les compétences du chat se trouve dans le tableau associatif \$tabComp.

Respectez le modèle html fourni dans le dossier maquette : uneCarte.html.

2. Connexion

Question 1

Rendez fonctionnel le formulaire d'authentification ([pageConnexion.php](#)), avec le principe suivant :

- Si l'identification (login/pass) est valide et que le rôle est *admin*, l'utilisateur est redirigé vers la page [pageEditerChats.php](#)
- Si l'identification (login/pass) est valide et que le rôle est inscrit, l'utilisateur est redirigé vers la page [pageVoirChats.php](#)
- Si l'identification n'est pas valide, l'utilisateur est redirigé vers le formulaire de connexion (il n'est pas demandé d'afficher un message).

Remarque : dans le script SQL fourni, tous les mots de passe sont 123456.

Question 2

Bonus : affichez dans la navbar le login de la personne connectée (à la place de « zigoto »).

Question 3

Faites en sorte que la page [pageEditerChats.php](#) ne soit accessible qu'aux visiteurs connectés en tant qu'admin.

3. Gestion des favoris

NB : vous n'avez pas à travailler avec la table de favoris stockée dans la base de données.

Dans le code fourni pour [pageVoirChats.php](#), l'ajout des chats en favoris (dans la session) fonctionne partiellement. Vous devez procéder à quelques améliorations.

Question 1

On veut que l'ajout et la suppression d'un favori soit effectués en **mode asynchrone** plutôt que de recharger la page comme actuellement. Apportez les modifications nécessaires.

Question 2

Il serait utile de pouvoir visualiser si un chat est déjà en favori ou non.

Pour cela, faites :

- Uniquement le bouton bleu (d'ajout) lorsque le chat n'est pas en favori
- Uniquement le bouton rouge (suppression) lorsque le chat est déjà en favori

Cette règle est à appliquer au **chargement de la page** et à chaque **mise à jour asynchrone**.

Éléments techniques qui peuvent être utiles :

- La **classe d-none de Bootstrap** permet de cacher un élément HTML (elle applique un `display:none`).
- Pour ajouter une classe, avec jQuery : `$("monSelecteur").addClass("d-none");`
- Pour retirer une classe, avec jQuery : `$("monSelecteur").removeClass("d-none");`
- Dans le code fourni, les **boutons** de chaque carte ont un **identifiant** unique construit en fonction de **l'identifiant du chat**.
- Le script de mise à jour des favoris doit retourner l'identifiant du chat qu'il vient de modifier, pour permettre la mise à jour de la bonne carte.

4. Affichage des compétences - Bonus

Question

Complétez la page `pageEditorCompetences.php` : affichez un tableau HTML qui contient les compétences (id et libellé) ainsi que pour chacune le nombre de chats qui la possède.