

 <b>3iLINGENIEURS</b> </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.

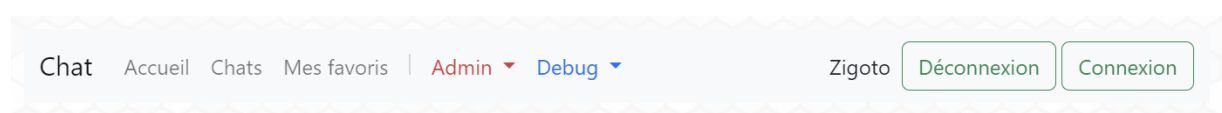
Le prototype d'application web dont vous disposez permet de présenter ces chats, avec la possibilité pour des utilisateurs enregistrés de mémoriser des favoris.

Il y a 2 rôles avec privilège :

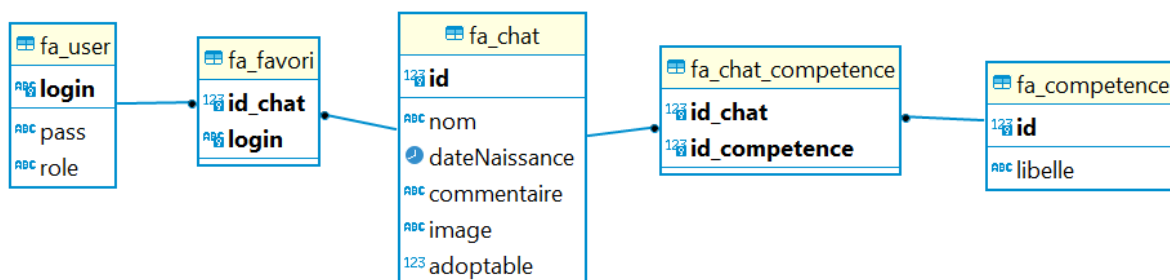
- admin, pour un administrateur
- inscrit, pour un visiteur enregistré

Dans le menu :

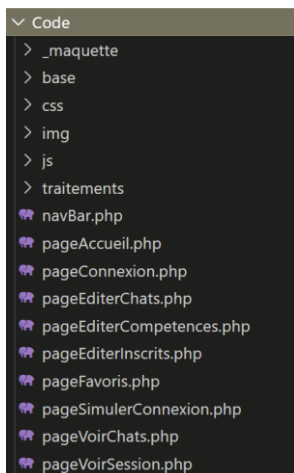
- Tous les liens existent, mais les fonctionnalités ne sont pas toutes implémentées
- La rubrique Admin (en rouge) donne accès à des pages qui seront à terme réservées aux administrateurs. Ces pages s'affichent avec un fond rouge pour vous aider à vous repérer.
- La rubrique Debug (en bleu) donne accès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

- le texte non adoptable ou adoptable selon le cas (champ `adoptable` dans la table `fa_chat`)
- la liste de ses compétences (sous la forme d'une liste à puces). Une requête est fournie et déjà exécutée pour chaque chat, les compétences du chat se trouvent dans le tableau associatif `$tabComp`. \*\*\*

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

## 2. Connexion

### Question 1

Modifiez la page de **simulation de connexion** (`pageSimulerConnexion.php`) :

- l'utilisateur ne choisit pas librement un login, il doit disposer d'une liste déroulante des logins présents dans la base, avec indication du rôle de chacun, et doit choisir une de ces valeurs.
- Il ne doit pas être possible de choisir un login qui n'existe pas.

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

### Question 2

Faites en sorte que la page `pageSimulerConnexion.php` ne soit accessible qu'aux visiteurs connectés en tant qu'admin.

## 3. Gestion des favoris

Il est prévu que l'application permette de placer un chat dans ses favoris selon le fonctionnement suivant :

- Si le visiteur n'est pas connecté, les favoris fonctionnent avec un simple stockage dans la session php.
- Si le visiteur est connecté, ses favoris sont stockés en session mais aussi dans la base de données.

### Question 1

Le script `traitements/miseAJourFavori.php` qui est fourni permet uniquement la mise en session. **Modifiez-le** pour que lorsqu'un utilisateur est connecté en tant qu'*admin* ou *inscrit* le stockage se fasse également dans la base de données (ajout ou suppression dans la table `fa_favori`).

### Question 2

Assurez-vous qu'au moment de la connexion (réelle ou simulée) d'un utilisateur, ses **favoris soient lus dans la base et placés en session**.

*Remarque : si l'utilisateur avait placé des chats en favoris avant de se connecter ils sont oubliés et remplacés par ceux de la base.*

### Question 3 - Bonus

Votre page actuelle ne permet pas de visualiser si un chat est déjà dans les favoris ou non. Faites afficher :

- 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

## 4. Un peu d'asynchrone

Dans la page [pageEditerChats.php](#) (menu admin/chats), le bouton éditer présent sur chaque ligne permet de visualiser les compétences d'un chat et devrait permettre de les modifier.

Faites fonctionner la modification, en mode asynchrone avec les 2 étapes ci-dessous.

### Question 1

Le script [traitements/miseAJourComp.php](#) est fourni : il permet d'ajouter une compétence à un chat, pas d'en supprimer une. **Complétez ce script pour gérer également la suppression.**

### Question 2

Lorsqu'on clique sur «éditer », la modale qui s'affiche contient l'id du chat concerné (image) dans un élément html ayant pour id idChat.

Si on clique sur une case à cocher, la fonction `toggleCompetence(idComp)` est appelée, avec l'idComp correspond à l'identifiant de la compétence associée à cette case.

**Ecrivez le code de cette fonction : elle doit récupérer l'idChat et l'état de la case à cocher, puis appeler de façon asynchrone `miseAJourComp` en lui transmettant les paramètres nécessaires. Prévoyez de traiter la réponse du serveur en affichant un message dans la console.**

### Élément technique :

Une façon de **vérifier si une case est cochée** , en jQuery :

```
if ($("#monSelecteur").is(":checked")) {...}
```