

TP : Gestion d'utilisateurs et de tickets avec contrôle d'accès

1. Création des tables dans la base de données

1. Créer un schéma de base de données comprenant au moins deux tables :
 - Table **users** :
 - **id** : identifiant unique
 - **username** : nom d'utilisateur.
 - **password** : mot de passe (haché).
 - **role** : rôle de l'utilisateur
 - **status** : statut de l'utilisateur
 - Table **tickets** (ou **tasks**) :
 - **id** : identifiant unique
 - **user_id** : clé étrangère pointant vers **users.id**.
 - **title** : titre du ticket.
 - **description** : description du ticket
 - **status** : statut du ticket .
 2. Créer les tables avec les contraintes nécessaires dans un fichier SQL ou via un outil d'administration de base de données.
-

2. Insertion de données fictives

1. Insérer dans la table **users** :
 - Un utilisateur administrateur (avec le rôle **admin**).
 - Plusieurs utilisateurs standard (avec le rôle **user**).
 - Gérer le champ **status** pour certains (actif ou inactif).
 2. Insérer des tickets liés aux utilisateurs :
 - Pour l'utilisateur administrateur et pour quelques utilisateurs standards.
 - Données de test (titre, description, statut varié).
-

3. Gestion de l'authentification

1. Mise en place d'un mécanisme de connexion (**login**) :
 - Formulaire HTML (ou autre) demandant **username** et **password**.
 - Vérification dans la base de données :
 - Comparer le mot de passe stocké (haché de préférence) avec celui entré.
 - En cas de succès : stocker l'identifiant de l'utilisateur et son rôle dans la session (par ex. `session['user_id']`, `session['role']`).
 - En cas d'échec : afficher un message d'erreur ou rediriger.
 2. Mécanisme de déconnexion (**logout**) :
 - Route ou bouton "Se déconnecter".
 - Effacer les informations de la session (ex. `session.clear()`).
 - Rediriger l'utilisateur (page d'accueil ou page de connexion).
-

4. Protection des routes grâce au rôle administrateur

1. Créer une fonction ou un décorateur pour vérifier le rôle administrateur :
 - Vérifier `session['role'] == 'admin'`.
 - En cas d'absence de ce rôle, renvoyer un message d'erreur ou rediriger.
 2. Appliquer ce mécanisme de contrôle à toutes les routes d'administration :
-

5. Page d'administration

1. Créer une vue (tableau de bord) réservée aux administrateurs :
 - Afficher la liste de tous les utilisateurs (`SELECT * FROM users`).
 - Afficher les informations principales : **id**, **username**, **role**, **status**.
 - Proposer des actions éventuelles : voir leurs tickets, modifier leur statut, etc.
 2. Navigation vers la page admin :
 - Ajouter un bouton ou un lien visible uniquement si le rôle est admin (vérifier la session).
 - L'accès direct à l'URL doit être protégé par le contrôle d'accès (décorateur `admin_required`).
-

6. Affichage des données propres à chaque utilisateur

1. Dans la page d'administration, permettre de voir les tickets d'un utilisateur spécifique :
 - Récupérer l'ID de l'utilisateur.
 - Afficher la liste de ses tickets (via la table `tickets`).
 2. Protéger également cette vue pour qu'elle ne soit accessible qu'à l'administrateur :
 - Utiliser le même décorateur ou mécanisme `admin_required`.
-

7. Création d'un tableau de bord (dashboard) pour l'utilisateur

1. Définir une route ou page (par ex. `/dashboard`) affichée après la connexion :
 - Récupérer `session['user_id']`.
 - Interroger la base pour récupérer tous les tickets dont `user_id = session['user_id']`.
 - Afficher ces tickets : titre, description, statut, etc.
 2. Interface : lister les tickets de l'utilisateur connecté dans un tableau ou une liste.
-

8. Ajout d'un bouton ou d'un lien "Créer un nouveau ticket"

1. Dans le tableau de bord, ajouter un bouton "Nouveau ticket" :
 - Lien vers une route (ex. `/tickets/new`) affichant un formulaire :
 - Champs : `title`, `description`, `status`.
 - Bouton "Valider" pour envoyer les données.
 2. À la soumission du formulaire :
 - Vérifier que l'utilisateur est connecté.
 - Insérer un nouvel enregistrement dans `tickets` avec `user_id = session['user_id']`.
 - Rediriger vers le tableau de bord pour afficher la liste mise à jour.
-

9. Modification et suppression de tickets

9.1. Modification

- Afficher un bouton “Modifier” pour chaque ticket dans la liste.
- Lien vers une route (ex. [/tickets/<id>/edit](#)) :
 - Afficher un formulaire pré-rempli (titre, description, statut).
 - Bouton “Enregistrer” pour mettre à jour.
- À la soumission :
 - Vérifier que le ticket appartient à l'utilisateur connecté (en comparant `ticket.user_id` et `session['user_id']`).
 - Mettre à jour la base de données.
 - Rediriger vers le tableau de bord.

9.2. Suppression

- Bouton “Supprimer” à côté de chaque ticket.
 - Route (ex. [/tickets/<id>/delete](#)) :
 - Vérifier la propriété du ticket (même logique que pour la modification).
 - Supprimer le ticket.
 - Rediriger vers le tableau de bord.
 - (Optionnel) Ajouter une boîte de dialogue de confirmation avant suppression.
-

10. (Facultatif) Détails d'un ticket

1. Ajouter un lien sur le titre ou un bouton “Détails”.
 2. Route (ex. [/tickets/<id>](#)) :
 - Affiche une page détaillée du ticket : titre, description, statut, date de création, etc.
 - Possibilité de réutiliser ici les boutons “Modifier” et “Supprimer”.
-

11. Vérification des permissions et sécurité

- Contrôle d'accès systématique : à chaque action (création, modification, suppression), vérifier que l'utilisateur connecté est propriétaire du ticket.
 - Gérer les erreurs : si un utilisateur tente d'accéder à un ticket qui ne lui appartient pas, refuser ou rediriger.
-

12. Finitions et ergonomie

1. Améliorer l'interface :
 - Utiliser des boutons ou liens clairs (Créer, Modifier, Supprimer...).
 - Mettre en valeur le statut du ticket ("Open", "Closed" ...).
 - Ajouter la date de création ou de mise à jour si besoin.
 2. Retour utilisateur :
 - Après chaque opération (création, modification, suppression), afficher un message de confirmation.
 3. Faciliter la navigation :
 - Ajouter un bouton "Annuler" dans les formulaires pour revenir au tableau de bord.
 - Ajouter un bouton "Se déconnecter" pour permettre le logout rapide.
-

13. Tests et validation

1. Créer plusieurs tickets depuis le tableau de bord d'un utilisateur :
 - Vérifier qu'ils apparaissent correctement.
 - Vérifier la modification et la suppression.
2. Vérifier les droits d'accès :
 - Tenter de modifier ou supprimer un ticket depuis un autre compte.
 - Confirmer que l'opération est refusée (erreur ou redirection).
3. Expérience utilisateur :
 - Vérifier que la navigation est fluide.
 - S'assurer que les rôles ([user](#), [admin](#)) fonctionnent comme prévu.