

Examen Final - TP Réseau Social Web en PHP et AJAX

Contexte

Pour ce projet final, vous devrez créer une application web de type réseau social en vous inspirant fortement du modèle Facebook. Votre solution devra intégrer toutes les fonctionnalités décrites ci-dessous, en respectant strictement les consignes fournies.

Technologies Autorisées

- Frontend : HTML, CSS, JavaScript (natif) ou framework JavaScript pour les requêtes HTTP (ex : Axios, Fetch)
- Backend : PHP natif exclusivement pour l'API
- Gestion temps réel : Node.js (facultatif) ou simulation par intervalle (JavaScript)
- Base de données : MySQL

Architecture Obligatoire (pour ceux utilisant technologies natives)

- assets
 - css
 - images
 - js
- vues
 - clients (pages du client)
 - back-office (pages d'administration)
- api (scripts PHP)
- Fichier index.html à la racine

Fonctionnalités à Implémenter

1. Module d'Authentification Complet

- **Inscription** (avec confirmation email via HTML template)
- **Connexion** (session gérée via sessionStorage en JavaScript)
- **Gestion du mot de passe oublié** (envoi d'email en HTML pour réinitialisation)

2. Page d'Accueil (Flux d'Articles)

- Affichage articles (image de profil/avatar, nom et prénom, description, image optionnelle)
- Système de likes/dislikes avec persistance côté utilisateur (icône change de couleur)
- Commentaires : bouton pour afficher les commentaires, champ de saisie fixé pour ajouter un commentaire sans recharger la page

3. Gestion des Amis

- Affichage des utilisateurs inscrits
- Envoi, réception et gestion (accepter/refuser) des invitations d'amitié
- Consultation des profils utilisateurs

4. Gestion du Profil Personnel

- Modification des informations personnelles, photo de profil, etc.
- Modification du mot de passe

5. Module Chat

- Sidebar affichant les utilisateurs avec conversations existantes
- Recherche d'amis pour démarrer une conversation
- Affichage des conversations sans recharge (utilisation de sockets Node.js ou intervalle JS toutes les 3s)
- Envoi de messages textuels et d'images

6. Back Office (Administration)

- Page de connexion distincte du client lambda
- Deux rôles : **Administrateur** et **Modérateur**
- Modérateur : gestion des articles et utilisateurs (consultation, suppression)
- Administrateur : gestion complète (ajout/suppression modérateurs et administrateurs, accès complet aux fonctionnalités du modérateur)
- Dashboard avec statistiques détaillées

Exigences Techniques

- Aucun recharge de page n'est autorisé après le chargement initial.

- Gestion des sessions utilisateur via JavaScript sessionStorage (équivalent PHP sessions)
- Tous les emails envoyés doivent être au format HTML et bien conçus.
- Code clair, structuré, bien commenté et sécurisé.

Dépôt du Projet

- Dépôt obligatoire sur GitHub ou GitLab
- Chaque membre doit effectuer des commits (évaluation participative)
- Fichier README.md détaillé :
 - Description du projet
 - Mode de fonctionnement
 - Identifiants de test (client et admin)
 - Liste des membres du groupe

Déploiement

- Pour les utilisateurs de frameworks frontend : déploiement frontend automatisé via Vercel ou Netlify
- Backend/API hébergé chez un hébergeur gratuit supportant PHP (ex: [InfinityFree](#))
- Utilisateurs de natif : application complète (frontend + backend) hébergée sur hébergeur gratuit supportant PHP (ex: InfinityFree)

Remise du Projet

- Envoyer par email à : marcosfaiwaz@gmail.com
- Contenu obligatoire de l'email :
 - Détails du projet
 - Lien vers le dépôt GitHub/GitLab
 - Lien vers la solution hébergée
 - Numéro du groupe

Date limite : 14 Juillet 2025 à 23h59

Tout retard entraînera une pénalité et risque d'être considéré comme non soumis.

Bon courage à tous !

