

# 1. Présentation du projet

**Nom du projet :** SkillSwap

**Objectif :** SkillSwap est une plateforme web permettant aux utilisateurs d'échanger des compétences entre eux. Les utilisateurs peuvent proposer leurs compétences, rechercher des offres et envoyer des demandes d'échange.

## 2. Acteurs du projet

- **Utilisateur (User) :** Peut s'inscrire, rechercher des offres, envoyer des demandes d'échange, noter et commenter des échanges.
- **Utilisateur banni (Banned User) :** Un utilisateur est banni s'il reçoit plusieurs signalements sur son offre.
- **Administrateur (Admin) :** Gère les utilisateurs, valide les compétences proposées, supervise les échanges.

## 3. Fonctionnalités principales

### Authentification et Gestion des Comptes

- Inscription avec email et mot de passe.
- Vérification de l'email après inscription.
- Connexion et déconnexion.
- Modification des informations personnelles.
- Gestion des rôles (user, banned\_user, admin).

### Gestion des compétences

- Chaque utilisateur peut renseigner ses compétences.
- Recherche de compétences existantes lors de l'inscription.
- Proposition de nouvelles compétences, soumises à validation par un administrateur.
- Validation des compétences par l'administrateur.

### Gestion des offres d'échange

- Création d'une offre d'échange avec description, compétence proposée et recherchée.
- Consultation des offres disponibles.
- Recherche et filtrage des offres par compétence, localisation, ou utilisateur.

### Système de demande d'échange

- Envoi de demandes d'échange via un bouton sur une offre.
- Possibilité d'ajouter un message personnalisé.
- Notification de l'utilisateur concerné.
- Acceptation ou refus de la demande.
- Suivi des demandes envoyées et reçues dans un tableau de bord.

### Gestion des avis et notations

- Après un échange, possibilité de noter l'expérience et de laisser un avis.
- Système de notation basé sur 5 étoiles.
- Affichage des avis sur le profil de l'utilisateur et sur les offres.

## Espace Administrateur

- Gestion des utilisateurs : suppression, modification des rôles.
- Validation des compétences proposées.
- Modération des avis et commentaires inappropriés.
- Supervision des échanges et gestion des litiges.

## 4. Architecture technique

### Technologies utilisées

- **Backend** : Symfony 7.2.1
- **Frontend** : Twig pour le rendu des templates
- **Base de données** : MySQL
- **Sécurité** : Authentification avec Symfony Security, gestion des rôles avec Voter
- **Tests** : PHPUnit pour les tests unitaires et fonctionnels

### Routes et API

- GET /offers : Afficher les offres disponibles
- POST /offer : Créer une nouvelle offre
- POST /request/{offer\_id} : Envoyer une demande d'échange
- PUT /request/{request\_id}/accept : Accepter une demande
- PUT /request/{request\_id}/decline : Refuser une demande
- POST /skill/suggest : Proposer une nouvelle compétence
- PUT /skill/{skill\_id}/validate : Valider une compétence (Admin)