

# TaskFlow - Gestionnaire de Tâches Collaboratif

Une application web moderne développée avec Symfony pour gérer vos projets et tâches en équipe.

## À propos du projet

TaskFlow est une application de gestion de projets et de tâches conçue pour faciliter le travail collaboratif. Elle permet aux utilisateurs de créer des projets, d'assigner des tâches, de suivre les progrès et de collaborer efficacement en équipe.

## Fonctionnalités principales

- **Gestion de projets** : Créez, modifiez et organisez vos projets
- **Gestion de tâches** : Ajoutez des tâches avec priorités, dates limites et assignation
- **Collaboration** : Invitez des utilisateurs à collaborer sur vos projets
- **Notifications** : Recevez des notifications pour les nouvelles activités
- **Tableau de bord** : Vue d'ensemble de vos projets et tâches
- **Interface responsive** : Fonctionne parfaitement sur mobile et desktop

## Technologies utilisées

- **Backend** : PHP 8.2+ avec Symfony 7.3
- **Base de données** : MySQL 9.1+
- **Frontend** : Bootstrap 5.3, Font Awesome 6.5
- **Template** : Twig
- **Email** : Symfony Mailer
- **Sécurité** : Symfony Security avec authentification par email

## Prérequis

Avant de commencer, assurez-vous d'avoir installé :

- PHP 8.2 ou version supérieure
- Composer (gestionnaire de dépendances PHP)
- MySQL 5.7+ ou MariaDB 10.2+
- Un serveur web (Apache, Nginx, ou serveur de développement Symfony)
- Node.js et npm (optionnel, pour la gestion des assets)

# Installation

## 1. Cloner le projet

```
bash

git clone https://github.com/votre-username/taskflow.git
cd taskflow
```

## 2. Installer les dépendances

```
bash

composer install
```

## 3. Configuration de l'environnement

Copiez le fichier `.env` et configurez vos paramètres :

```
bash

cp .env .env.local
```

Éditez le fichier `.env.local` et configurez :

```
env

# Configuration de la base de données
DATABASE_URL="mysql://username:password@127.0.0.1:3306/taskflow"

# Configuration du mailer (pour les notifications)
MAILER_DSN=smtp://localhost:1025

# Environnement de développement
APP_ENV=dev
```

## 4. Créer la base de données

```
bash

# Créer la base de données
php bin/console doctrine:database:create

# Appliquer les migrations
php bin/console doctrine:migrations:migrate
```

## 5. (Optionnel) Charger des données de test

```
bash  
php bin/console doctrine:fixtures:load
```

## 6. Lancer le serveur de développement

```
bash  
symfony server:start
```

Ou avec PHP :

```
bash  
php -S localhost:8000 -t public/
```

L'application sera accessible à l'adresse : `http://localhost:8000`

## Utilisation

### Premier pas

1. **Inscription** : Créez votre compte sur la page d'inscription
2. **Confirmation email** : Confirmez votre email (en développement, les emails sont visibles dans le profiler Symfony)
3. **Connexion** : Connectez-vous à votre compte
4. **Créer un projet** : Commencez par créer votre premier projet
5. **Ajouter des tâches** : Ajoutez des tâches à votre projet
6. **Inviter des collaborateurs** : Invitez d'autres utilisateurs à rejoindre vos projets

## Fonctionnalités détaillées

### Gestion des projets

- Créer, modifier et supprimer des projets
- Voir la progression globale de chaque projet
- Gérer les collaborateurs

### Gestion des tâches

- Créer des tâches avec titre, description, priorité
- Assigner des tâches aux collaborateurs

- Définir des dates limites
- Suivre le statut : À faire, En cours, Terminé

## Collaboration

- Inviter des utilisateurs par email
- Gérer les demandes de collaboration
- Système de notifications en temps réel

## Structure du projet

```
taskflow/  
├── src/           # Code source Symfony  
│   ├── Controller/  # Contrôleurs  
│   ├── Entity/      # Entités Doctrine  
│   ├── Form/        # Formulaires Symfony  
│   ├── Repository/  # Repositories Doctrine  
│   └── Security/    # Configurations de sécurité  
├── templates/      # Templates Twig  
│   ├── base.html.twig # Template de base  
│   ├── dashboard/    # Tableau de bord  
│   ├── project/      # Gestion des projets  
│   ├── task/         # Gestion des tâches  
│   └── collaboration/ # Fonctionnalités collaboratives  
├── public/         # Fichiers publics  
│   ├── css/        # Styles CSS  
│   └── js/          # Scripts JavaScript  
├── migrations/     # Migrations de base de données  
└── config/         # Configuration Symfony
```

## Résolution des problèmes courants

### Problème de permissions

```
bash  
  
chmod -R 755 var/cache var/log
```

### Cache Symfony

```
bash  
  
php bin/console cache:clear
```

### Problèmes de base de données

```
bash
```

```
# Réinitialiser la base de données
```

```
php bin/console doctrine:database:drop --force
```

```
php bin/console doctrine:database:create
```

```
php bin/console doctrine:migrations:migrate
```

## Problème d'emails en développement

En développement, utilisez Mailhog ou configurez un serveur SMTP de test.

## Développement

### Commandes utiles

```
bash
```

```
# Créer une nouvelle migration après modification d'entité
```

```
php bin/console make:migration
```

```
# Créer une nouvelle entité
```

```
php bin/console make:entity
```

```
# Créer un nouveau contrôleur
```

```
php bin/console make:controller
```

```
# Vider le cache
```

```
php bin/console cache:clear
```

```
# Voir les routes disponibles
```

```
php bin/console debug:router
```

## Tests

```
bash
```

```
# Lancer les tests
```

```
php bin/phpunit
```

## Déploiement en production

1. Configurez les variables d'environnement pour la production
2. Installez les dépendances sans les dépendances de développement :

```
bash
```

3. Configurez votre serveur web (Apache/Nginx)
4. Appliquez les migrations sur votre base de données de production
5. Configurez les permissions appropriées

## Contribution

Ce projet a été développé dans un cadre pédagogique pour apprendre Symfony et les bonnes pratiques de développement web.

## Objectifs pédagogiques atteints

- Développement d'une application Symfony multi-utilisateurs
- Gestion des relations entre entités (projets ↔ tâches ↔ utilisateurs)
- Mise en place d'un CRUD complet avec interface intuitive
- Découverte des notions d'organisation et de suivi (statuts, priorités)
- Système d'authentification et de sécurité
- Interface responsive avec Bootstrap

## Licence

Ce projet est développé à des fins éducatives.

## Auteur

Développé par christian ROUPIOZ dans le cadre d'un apprentissage de Symfony.

---

**Note :** Cette application a été développée avec Symfony 7.3 et PHP 8.2. Assurez-vous d'avoir les bonnes versions pour éviter les problèmes de compatibilité.