

# Gestion d'une agence location de voiture

## Rapport du Projet Symfony - Système d'Authentification

### Introduction:

Ce projet est un système d'authentification développé avec Symfony 6, permettant la gestion des utilisateurs avec différents rôles (ADMIN et USER). Le projet implémente un système de connexion sécurisé avec une interface utilisateur intuitive.

### Objectifs du projet:

- ❖ Création d'un système d'authentification sécurisé
- ❖ Gestion des rôles utilisateurs
- ❖ Interface de connexion responsive
- ❖ Gestion des sessions utilisateurs
- ❖ Protection des données sensibles
- ❖ Les Erreurs Rencontrées et Solutions

### 1. Erreurs de Configuration

# Erreur initiale dans security.yaml

security:

providers:

users\_in\_memory: { memory: null } # Configuration incorrecte

# Solution appliquée

security:

providers:

app\_user\_provider:

entity:

class: App\Entity\User

property: email.

## 2. Erreurs de Base de Données

Problème de migration avec les champs uniques

Solution : Nettoyage et recréation des migrations

## 3. Erreurs de Sécurité

Protection CSRF manquante initialement

Correction avec l'ajout de tokens CSRF

## Partie Code:

### 1. Structure du Projet

project/

```
├─ src/
|   ├─ Controller/
|   |   └─ SecurityController.php
|   |   └─ HomeController.php
|   └─ Entity/
|       └─ User.php
|       └─ Security/
|           └─ LoginFormAuthenticator.php
└─ templates/
    ├─ base.html.twig
    └─ security/
        └─ login.html.twig
```

## 2- Points Clés du Code

### Contrôleur de Sécurité

SecurityController.php

```
#[Route('/login', name: 'app_login')]
public function login(AuthenticationUtils $authenticationUtils): Response
{
    if ($this->getUser()) {
        return $this->redirectToRoute('home');
    }

    return $this->render('security/login.html.twig', [
        'last_username' => $authenticationUtils->getLastUsername(),
        'error' => $authenticationUtils->getLastAuthenticationError()
    ]);
}
```

### Authentificateur

LoginFormAuthenticator.php

```
class LoginFormAuthenticator extends AbstractLoginFormAuthenticator
{
    public function authenticate(Request $request): Passport
    {
        $email = $request->request->get('email');
        $password = $request->request->get('password');

        return new Passport(
            new UserBadge($email),
            new PasswordCredentials($password),
            [new CsrfTokenBadge('authenticate', $request->request->get('_csrf_token'))]
        );
    }
}
```

## Aperçu du Site-Pages Principales:

### 1. Page de Connexion

- ❖ Formulaire de connexion sécurisé
- ❖ Gestion des erreurs d'authentification
- ❖ Design responsive
- ❖ Dashboard
- ❖ Interface adaptée selon le rôle
- ❖ Menu de navigation personnalisé
- ❖ Affichage des informations utilisateur
- ❖ Fonctionnalités
- ❖ Authentification sécurisée
- ❖ Gestion des sessions
- ❖ Protection contre les attaques CSRF
- ❖ Messages d'erreur personnalisés
- ❖ Redirection intelligente

- ❖ Sécurité Implémentée
- ❖ Hachage des mots de passe
- ❖ Protection contre les injections SQL
- ❖ Validation des formulaires
- ❖ Gestion des tokens CSRF
- ❖ Conclusion

## Points Forts:

### 1. Sécurité

- ❖ Système d'authentification robuste
- ❖ Protection des données utilisateurs
- ❖ Gestion efficace des sessions
- ❖ Maintenabilité
- ❖ Code bien structuré
- ❖ Documentation claire
- ❖ Respect des standards Symfony
- ❖ Performance
- ❖ Optimisation des requêtes
- ❖ Mise en cache efficace
- ❖ Temps de chargement optimisé
- ❖ Axes d'Amélioration

### 2. Fonctionnalités

- ❖ Ajout de la réinitialisation de mot de passe
- ❖ Implémentation de l'authentification 2FA
- ❖ Gestion plus fine des permissions
- ❖ Interface
- ❖ Amélioration du design responsive
- ❖ Ajout d'animations
- ❖ Optimisation de l'expérience utilisateur
- ❖ Tests
- ❖ Augmentation de la couverture de tests
- ❖ Ajout de tests d'intégration
- ❖ Tests de performance
- ❖ Recommandations

### 3. Court terme

- ❖ Implémenter la récupération de mot de passe
- ❖ Ajouter des logs d'activité
- ❖ Améliorer la documentation utilisateur

- ❖ Long terme
- ❖ Mise en place d'une API REST
- ❖ Intégration avec des services tiers
- ❖ Système de notifications

## Conclusion:

Ce projet constitue une base solide pour un système d'authentification, avec un potentiel d'évolution important pour répondre à des besoins plus spécifiques.