Document de cadrage de projet Symfony

1. Introduction

Le sujet du projet est libre et peut être choisi en fonction des intérêts de chacun (gestion d'événements, réseau social, plateforme de e-commerce, etc.), mais il doit impérativement respecter l'ensemble des contraintes techniques définies ci-après.

2. Contraintes techniques obligatoires

Pour assurer une homogénéité dans les projets et développer des compétences spécifiques, les étudiants devront intégrer les éléments suivants :

- **Utilisation du framework Symfony :** Le projet doit être développé sur Symfony, en exploitant ses fonctionnalités pour assurer une architecture MVC robuste et modulable.
- **Intégration de Twig pour les vues :** Toutes les pages de l'application devront être générées avec le moteur de templates Twig, permettant ainsi une séparation claire entre la logique applicative et la présentation.
- **Utilisation de Webpack Encore pour la gestion des assets (CSS/JS) :** La gestion des fichiers CSS et JavaScript se fera via Webpack Encore afin d'optimiser le chargement et la maintenance des ressources front-end.
- **Création et gestion de formulaires Symfony :** L'application devra inclure des formulaires pour les interactions avec l'utilisateur, en tirant profit des composants et des bonnes pratiques de Symfony pour la validation et la sécurisation des données.
- **Utilisation de Doctrine pour la base de données :** La persistance des données se fera avec Doctrine ORM, permettant la création, la gestion et la migration des schémas de base de données.
- **Implémentation de services Symfony :** Les fonctionnalités métier devront être encapsulées dans des services Symfony, favorisant la réutilisation du code et le respect du principe de séparation des responsabilités.
- **Gestion de la sécurité et de l'authentification :** Le projet doit inclure un système d'authentification (inscription, connexion, gestion des rôles) et mettre en œuvre des

mesures de sécurité pour protéger l’accès aux ressources sensibles.

---

3. Compétences attendues (obligatoires)

À l’issue du projet, les étudiants doivent être capables de :

- **Configurer un projet Symfony :** Créer et initialiser un nouveau projet, configurer l’environnement de développement et gérer les dépendances.
- **Configurer et gérer le routage :** Définir et organiser les routes de l’application afin d’assurer une navigation fluide entre les différentes pages et fonctionnalités.
- **Créer et gérer les formulaires :** Utiliser les composants Symfony pour créer des formulaires interactifs, avec validation et gestion des erreurs.
- **Intégrer une base de données avec Doctrine :** Concevoir le schéma de la base de données, gérer les entités et mettre en place des migrations pour garantir la cohérence des données.
- **Créer et utiliser les services :** Développer des services pour encapsuler la logique métier, favoriser la réutilisation et simplifier les tests.
- **Créer des templates dynamiques avec Twig :** Concevoir des vues dynamiques et modulaires en exploitant les capacités du moteur de templates Twig.
- **Gérer la sécurité et l'authentification :** Mettre en place un système d’authentification robuste et gérer les permissions d’accès aux différentes parties de l’application.

---

4. Compétences complémentaires (bonus)

Pour aller au-delà des exigences de base, les étudiants pourront, à titre optionnel, intégrer les compétences suivantes :

- **Développer et consommer des API :** Concevoir des endpoints REST ou GraphQL pour permettre l’interaction avec des applications tierces ou des applications mobiles.
- **Mettre en place des tests unitaires et fonctionnels :** Écrire et exécuter des tests pour valider le bon fonctionnement des composants de l’application, assurant ainsi la qualité et la robustesse du code.

- **Déployer une application Symfony :** Mettre en œuvre un processus de déploiement continu et configurer l'application pour une mise en production sur un serveur ou une plateforme cloud.

---

5. Livrables attendus

Les étudiants devront fournir les éléments suivants à la fin du projet :

- **Code source :** Un code propre, structuré et documenté, hébergé sur une plateforme de gestion de version (ex. GitHub, GitLab).
- **README :** Un fichier README détaillé décrivant le projet, les choix techniques, les instructions d'installation et d'exécution.
- **Présentation orale ou vidéo de démonstration :** Une présentation permettant de découvrir les principales fonctionnalités de l'application, le processus de développement et les défis rencontrés.
- **Documentation technique :** Une documentation comprenant la structure du projet, le modèle de base de données, les schémas d'architecture et les étapes de déploiement, afin de faciliter la maintenance et l'évolution future du projet.

---

6. Grille d'évaluation

L'évaluation du projet se fera selon les critères suivants :

- **Atteinte des compétences obligatoires :** La maîtrise et l'implémentation correcte des éléments techniques obligatoires (Symfony, Twig, Webpack Encore, formulaires, Doctrine, services, sécurité).
- **Qualité du code et de la documentation :** La propreté, la lisibilité, la modularité du code ainsi que la qualité de la documentation associée.
- **Respect des délais et des livrables :** Le respect du planning et la complétude des livrables attendus (code source, README, présentation, documentation technique).
- **Intégration des compétences complémentaires (bonus) :** La prise en compte et l'implémentation des fonctionnalités optionnelles telles que les API, les tests et le déploiement, qui viendront valoriser le projet.

---

7. Ressources recommandées

Pour aider les étudiants dans la réalisation de ce projet, les ressources suivantes sont vivement recommandées :

- **Symfony Documentation :** Symfony
- **Twig Documentation :** Twig
- **Webpack Encore Documentation :** Webpack Encore
- **Formulaires Symfony :** Formulaires Symfony
- **Doctrine ORM :** Doctrine
- **Authentification et Sécurité :** Sécurité Symfony