

## Documentation knowledge e-learning



# KNOWLEDGE

Lien repository GitHub : <https://github.com/katoudevb/knowledge>

Lien d'hébergement : <https://bricotteaux.alwaysdata.net/>

# Sommaire

1. Résumer du projet
2. Contexte et objectifs du projet
3. Architecture de l'application
  - 3.1 Architecture Back-End (Symfony 7)
  - 3.2 Architecture Front-End
4. Conception de la base de données
  - 4.1 Modèle conceptuel (UML / Merise)
  - 4.2 Modèle physique
5. Gestion des utilisateurs et sécurité
6. Fonctionnalités e-learning
  - 6.1 Création et activation de compte
  - 6.2 Achat de cursus et de leçons
  - 6.3 Validation des leçons et certifications
7. Composants e-commerce et sandbox
8. Tests unitaires et fonctionnels
9. Documentation du code
10. Conclusion et perspectives

## Résumé du projet

Le projet « Knowledge Learning » a pour objectif de créer une plateforme e-learning pour la société Knowledge, spécialisée dans l'édition de supports de formation. Cette plateforme permet aux utilisateurs de suivre des cours à distance, d'acheter des cursus complets ou des leçons spécifiques et d'obtenir des certifications après validation de tous les modules d'un cursus.

L'application est développée avec Symfony 7, en adoptant une architecture MVC 3-tiers, garantissant la maintenabilité et l'évolutivité du code. La sécurité des utilisateurs est assurée grâce à l'activation des comptes par email, à l'encodage des mots de passe et à la gestion des rôles (ROLE\_ADMIN et ROLE\_CLIENT).

La base de données relationnelle est conçue pour refléter les besoins fonctionnels et commerciaux, avec les entités principales suivantes : User, Theme, Course, Lesson, Purchase, LessonProgress et Certification. Chaque table contient des champs de traçabilité (created\_at, updated\_at, created\_by, updated\_by) pour assurer le suivi des actions et répondre aux exigences de sécurité.

Les utilisateurs peuvent créer un compte, acheter des contenus via un module e-commerce en mode sandbox, suivre les leçons et valider leur progression. Une fois toutes les leçons d'un cursus validées, la certification correspondante est automatiquement générée et consultable depuis une page récapitulative.

L'application inclut des tests unitaires couvrant les principales fonctionnalités : création de compte, connexion, achat de contenus et validation de cours. La documentation du code est réalisée en anglais, facilitant la compréhension et la maintenance future. Le front-end, bien que minimaliste, permet de naviguer entre les thèmes, cursus et leçons tout en respectant la charte graphique définie par l'entreprise.

En conclusion, « Knowledge Learning » fournit une solution complète, sécurisée et traçable pour transformer l'offre traditionnelle de formation de Knowledge en une plateforme e-learning moderne et accessible.

# Contexte et objectifs du projet

## Contexte

La société Knowledge est un éditeur reconnu dans le domaine de la formation, proposant des supports pédagogiques sous forme de livres ou de kits dans différents domaines tels que la musique, l'informatique ou le jardinage. Traditionnellement, ces formations étaient distribuées en librairie, limitant l'accès aux clients à des horaires et lieux spécifiques.

Avec la digitalisation croissante de l'éducation et la demande pour des solutions d'apprentissage flexibles, Knowledge souhaite étendre son offre en proposant un service de formation en ligne. L'objectif est de permettre aux clients d'apprendre à leur rythme, depuis n'importe quel endroit et en toute autonomie, tout en conservant la qualité pédagogique reconnue des supports Knowledge.

## Objectifs du projet

Le projet « Knowledge Learning » a pour objectif de concevoir et de développer une plateforme e-learning intégrant à la fois des fonctionnalités pédagogiques et commerciales. Les principaux objectifs sont les suivants :

- 1. Offrir une expérience utilisateur complète :**
  - a. Inscription et activation sécurisée des comptes par email.
  - b. Gestion des rôles : utilisateurs clients et administrateurs.
  - c. Navigation entre thèmes, cursus et leçons.
- 2. Permettre l'achat de contenus pédagogiques :**
  - a. Achat de cursus complets ou de leçons individuelles.
  - b. Intégration d'un module e-commerce en mode sandbox pour les tests.
- 3. Suivi pédagogique et certifications :**
  - a. Validation de chaque leçon par l'utilisateur.
  - b. Attribution automatique des certifications une fois tous les modules d'un cursus complétés.
  - c. Page récapitulative des certifications obtenues.
- 4. Garantir la sécurité et la traçabilité :**
  - a. Mots de passe cryptés et champs de traçabilité (created\_at, updated\_at, created\_by, updated\_by).
  - b. Protection contre les failles CSRF sur tous les formulaires.
- 5. Assurer la maintenabilité et la qualité du code :**

- a. Architecture orientée objet et MVC 3-tiers.
- b. Documentation complète du code et tests unitaires couvrant toutes les fonctionnalités critiques.

En résumé, l'objectif de ce projet est de transformer l'offre traditionnelle de Knowledge en une plateforme numérique moderne, permettant un apprentissage autonome, sécurisé et certifié, tout en respectant les standards techniques et les bonnes pratiques de développement.

## Architecture de l'application

La plateforme « Knowledge Learning » est conçue selon une architecture **MVC (Model-View-Controller) 3-tiers**, garantissant une séparation claire entre les données, la logique métier et la présentation. Cette organisation assure la maintenabilité, la réutilisabilité et la sécurité de l'application, tout en facilitant son évolution future.

### 1. Architecture Back-End

Le back-end est développé avec **Symfony 7**, framework PHP robuste et reconnu pour la gestion des applications web complexes. L'architecture suit les principes suivants :

- **Entities** : représentent les objets métiers (User, Theme, Course, Lesson, Purchase, Certification). Ces entités sont gérées via **Doctrine ORM**, assurant la persistance des données et les relations entre les tables.
- **Repositories** : contiennent les méthodes d'accès et de requêtage des données. Chaque entité dispose d'un repository dédié pour centraliser la logique liée à la récupération et la manipulation des données.
- **Controllers** : gèrent les routes HTTP et orchestrent les interactions entre l'utilisateur, le service métier et la base de données.
- **Services** : encapsulent la logique métier, notamment pour la gestion des achats, la validation des leçons et l'attribution des certifications.
- **Security** : gère l'authentification, l'autorisation et la protection contre les failles CSRF. Les rôles utilisateurs (ROLE\_ADMIN, ROLE\_CLIENT) définissent l'accès aux différentes fonctionnalités.

## 2. Architecture Front-End

Le front-end est minimaliste mais fonctionnel, suffisant pour tester et démontrer toutes les fonctionnalités de la plateforme :

- Pages de connexion et d'inscription avec validation par email.
- Navigation entre thèmes, cursus et leçons.
- Boutons d'achat simulant la transaction en mode sandbox.
- Bouton "Valider la leçon" et pages récapitulatives des certifications.
- Mise en conformité avec la charte graphique fournie (police, couleurs et logo).

## 3. Interaction entre les couches

- **Utilisateur → Controller** : L'utilisateur interagit avec l'application via le navigateur. Les requêtes HTTP sont reçues par les controllers Symfony.
- **Controller → Service** : Les controllers délèguent la logique métier aux services (ex. achat, validation d'une leçon, génération de certification).
- **Service → Repository** : Les services utilisent les repositories pour interagir avec la base de données.
- **Repository → Base de données** : Les repositories effectuent les opérations CRUD sur les entités Doctrine, garantissant l'intégrité des données.

Cette architecture modulaire permet une **extensibilité facile**, par exemple l'ajout futur de nouveaux thèmes, cursus ou types de certification, sans impacter les autres parties de l'application.

## Conception de la base de données

La base de données de la plateforme « Knowledge Learning » est conçue selon un **modèle relationnel**, structuré pour répondre à la fois aux besoins pédagogiques et e-commerce de l'application. Chaque entité reflète un concept métier, et les relations entre les tables permettent de gérer efficacement la navigation, les achats et le suivi pédagogique.

### 1. Modèle conceptuel (UML )

Les principales entités sont les suivantes :

- **User** : représente un utilisateur de la plateforme.
  - Attributs principaux : id, email, password, is\_verified, roles, created\_at, updated\_at, created\_by, updated\_by.
  - Un utilisateur peut avoir plusieurs achats, leçons validées et certifications.
- **Theme** : regroupe plusieurs cursus autour d'un même domaine (ex. Musique, Informatique).
  - Attributs : id, name.
  - Relation : 1 Theme → N Course.
- **Course (Cursus)** : ensemble de leçons liées à un thème.
  - Attributs : id, title, price, theme\_id.
  - Relation : 1 Course → N Lesson.

- **Lesson** : unité pédagogique d'un cursus, contenant texte et vidéo.
  - Attributs : id, title, price, content, video\_url, course\_id.
- **Purchase** : enregistre les achats des utilisateurs.
  - Attributs : id, user\_id, course\_id (nullable), lesson\_id (nullable), amount, created\_at.
- **UserLesson** : suit la validation des leçons par un utilisateur.
  - Attributs : id, user\_id, lesson\_id, validated, validate\_at.
- **Certification** : atteste qu'un utilisateur a complété un cursus.
  - Attributs : id, user\_id, course\_id, obtained\_at.

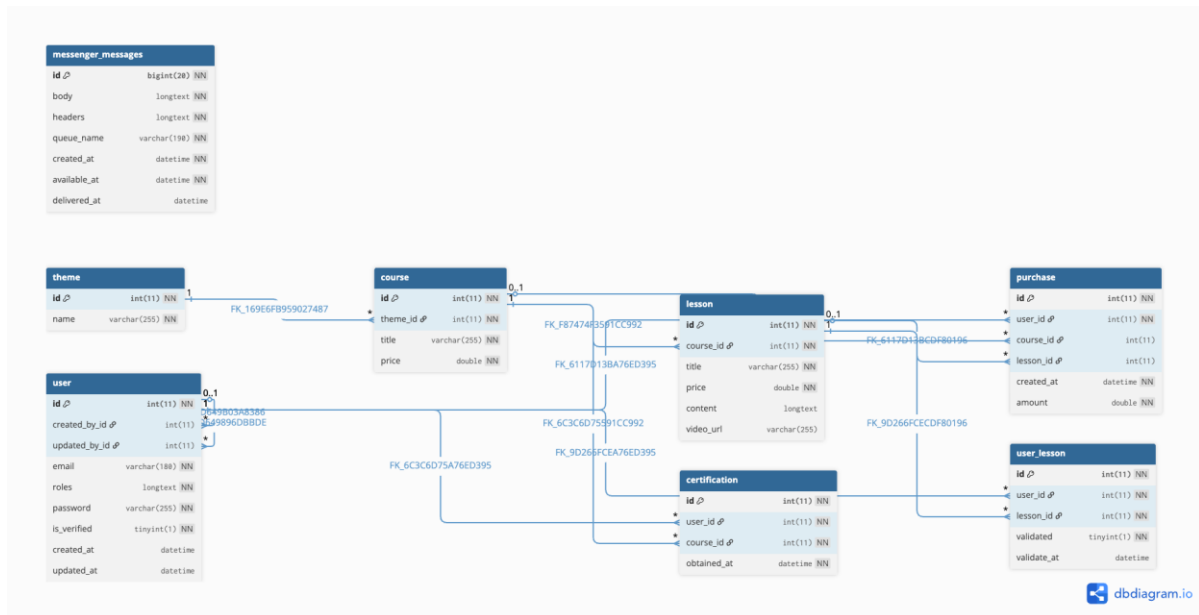
## 2. Modèle physique

- Les relations clés :
  - Theme 1 → N Course
  - Course 1 → N Lesson
  - User 1 → N Purchase
  - User 1 → N LessonProgress
  - User 1 → N Certification
- Chaque table inclut des champs de traçabilité (created\_at, updated\_at, created\_by, updated\_by) pour suivre l'historique des modifications.
- Les mots de passe des utilisateurs sont stockés de manière sécurisée avec un **hash cryptographique**.
- Les entités et leurs relations permettent de :
  - Gérer l'accès aux leçons et aux cursus achetés.
  - Assurer la validation automatique des cursus pour les certifications.
  - Faciliter les tests unitaires et la maintenance.

Cette conception garantit la **cohérence et l'intégrité des données**, tout en offrant la flexibilité nécessaire pour étendre la plateforme (ajout de nouveaux thèmes, cursus ou fonctionnalités e-commerce) sans perturber les données existantes.



## Diagrammes MPD



## Gestion des utilisateurs et sécurité

La gestion des utilisateurs et la sécurité sont des aspects centraux de la plateforme « Knowledge Learning ». Elles garantissent à la fois l'intégrité des données, l'accès sécurisé aux contenus et la protection des informations personnelles des utilisateurs.

## 1. Gestion des utilisateurs

La plateforme distingue deux types d'utilisateurs principaux :

- **Administrateur (ROLE\_ADMIN) :**
  - Accède au back-office pour gérer les comptes utilisateurs, les contenus pédagogiques et les achats.
  - Peut créer, modifier ou supprimer des thèmes, cursus et leçons.
- **Client (ROLE\_CLIENT) :**
  - Peut s'inscrire, activer son compte et se connecter.
  - Achète des cursus ou des leçons individuelles et accède aux contenus associés.
  - Valide les leçons et obtient des certifications une fois tous les modules d'un cursus complétés.

Lors de l'inscription, chaque utilisateur se voit attribuer automatiquement le rôle `ROLE_CLIENT`. Les informations personnelles et les mots de passe sont stockés de manière sécurisée dans la base de données.

## 2. Inscription et activation

- L'utilisateur remplit un formulaire d'inscription.
- Un email contenant un **token d'activation** est envoyé pour confirmer l'adresse email.
- L'utilisateur doit activer son compte avant de pouvoir acheter des contenus ou valider des leçons.

## 3. Connexion et gestion des sessions

- L'authentification est gérée via Symfony Security, avec vérification du rôle et de l'état `isActive`.
- Les sessions sont sécurisées et protégées contre les accès non autorisés.

## 4. Sécurité des données

- Les mots de passe sont cryptés avec un algorithme sécurisé (hashing via `PasswordHasherInterface`).
- Les formulaires utilisent un **middleware CSRF**, protégeant l'application contre les attaques de type cross-site request forgery.
- Les données sensibles sont tracées via les champs `created_at`, `updated_at`, `created_by` et `updated_by`, garantissant un suivi complet des actions effectuées par les utilisateurs.
- Les rôles et permissions permettent de restreindre l'accès aux fonctionnalités critiques et au back-office.

## 5. Avantages de cette approche

- Assure une sécurité robuste pour les utilisateurs et les données.
- Permet un contrôle fin des accès selon les rôles.
- Garantit la traçabilité et l'audit des actions.
- Facilite la maintenance et l'évolution future de la plateforme.

En combinant **gestion des rôles**, **activation par email**, **cryptage des mots de passe** et **protection CSRF**, la plateforme offre un environnement sécurisé et fiable pour les utilisateurs et les administrateurs.

### Code Connexion :

ADMIN :

admin@example.com

Admin123

CLIENT :

client@example.com

Client123@

UTILISATEUR :

testuser1@example.com

User1234@

## Fonctionnalités e-learning

La plateforme « Knowledge Learning » propose un ensemble de fonctionnalités permettant aux utilisateurs de suivre des cours à distance, d'acheter des contenus pédagogiques et de valider leur progression pour obtenir des certifications. Ces fonctionnalités sont conçues pour offrir une expérience utilisateur fluide et sécurisée.

### 1. Création et activation de compte

- Les utilisateurs peuvent créer un compte via un formulaire d'inscription.
- Après inscription, un **email d'activation** est envoyé avec un lien unique.
- L'utilisateur doit activer son compte avant de pouvoir accéder aux achats et aux contenus pédagogiques.
- L'inscription attribue automatiquement le rôle `ROLE_CLIENT` à l'utilisateur.

### 2. Connexion et gestion des sessions

- Les utilisateurs activés peuvent se connecter avec leur email et mot de passe.
- La session utilisateur est sécurisée et gérée par Symfony Security.
- Les utilisateurs non activés sont empêchés d'accéder aux fonctionnalités d'achat ou aux leçons.

### 3. Achat de cursus et de leçons

- L'utilisateur peut acheter :
  - Un **cursus complet**, comprenant toutes les leçons d'un thème.
  - Une **leçon individuelle**, si l'utilisateur souhaite un apprentissage ciblé.
- Les achats sont enregistrés dans la table Purchase et gérés via un module e-commerce en **mode sandbox** pour la phase de prototype.
- Chaque achat confère à l'utilisateur le droit d'accéder aux pages des leçons correspondantes.

#### 4. Validation des leçons et suivi

- Sur chaque page de leçon, un bouton “Valider la leçon” permet à l'utilisateur de confirmer l'apprentissage.
- La validation est enregistrée dans la table LessonProgress.
- Lorsque toutes les leçons d'un cursus sont validées, le cursus est automatiquement marqué comme **complété**.

#### 5. Obtention des certifications

- Une fois toutes les leçons d'un cursus validées, la plateforme génère automatiquement une **certification** associée à l'utilisateur.
- Les certifications sont stockées dans la table Certification.
- Une page récapitulative permet à l'utilisateur de consulter toutes les certifications obtenues.

#### 6. Avantages pédagogiques

- Permet un apprentissage autonome et structuré.
- Offre un suivi précis de la progression des utilisateurs.
- Facilite l'obtention de certifications reconnues par la plateforme.
- Encourage la motivation des utilisateurs grâce à un suivi clair et des récompenses certifiées.

Ces fonctionnalités forment le cœur pédagogique de la plateforme et garantissent une expérience d'apprentissage complète, accessible et traçable, tout en intégrant les aspects e-commerce et sécurité requis par le projet.

## Composants e-commerce et sandbox

La plateforme « Knowledge Learning » intègre un module e-commerce permettant aux utilisateurs d'acheter des cursus complets ou des leçons individuelles. Pour la phase de prototype, les transactions sont simulées en **mode sandbox**, garantissant la démonstration des fonctionnalités sans passer par un paiement réel.

## 1. Gestion des achats

- Les achats sont enregistrés dans la table Purchase, qui contient :
  - user\_id : identifiant de l'utilisateur.
  - course\_id : identifiant du cursus (nullable).
  - lesson\_id : identifiant de la leçon (nullable).
  - amount : montant payé.
  - created\_at : date de l'achat.
- Chaque achat donne à l'utilisateur le droit d'accéder aux contenus correspondants.

## 2. Simulation de paiement (sandbox)

- Le module e-commerce en mode sandbox permet de tester :
  - L'achat d'un cursus complet.
  - L'achat d'une leçon individuelle.
- Aucun paiement réel n'est effectué. Les tests simulent le processus de transaction pour valider le flux fonctionnel.
- Ce mode permet également de vérifier la création automatique des droits d'accès aux contenus pédagogiques et des entrées correspondantes dans Purchase.

## 3. Intégration avec la logique métier

- Lorsqu'un achat est validé dans le sandbox :
  - L'utilisateur obtient les droits d'accès aux leçons achetées.
  - Les contrôles de progression (LessonProgress) peuvent être activés pour les leçons concernées.
- Cette intégration garantit que la logique pédagogique et la gestion commerciale sont cohérentes et synchronisées.

## 4. Avantages du mode sandbox

- Permet de livrer un prototype fonctionnel sans risque financier.
- Facilite le test des différents scénarios d'achat.
- Assure la traçabilité et la fiabilité des flux avant intégration d'un module de paiement réel.

En combinant **gestion des achats**, **simulation sandbox** et **intégration avec le suivi pédagogique**, la plateforme assure une expérience utilisateur complète et sécurisée, tout en respectant les contraintes du projet et les bonnes pratiques de développement e-commerce.

## Tests unitaires et fonctionnels

La qualité et la fiabilité de la plateforme « Knowledge Learning » sont assurées par la mise en place de **tests unitaires et fonctionnels**, couvrant l'ensemble des fonctionnalités critiques. Ces tests permettent de détecter rapidement les anomalies et garantissent le bon fonctionnement de l'application avant toute mise en production.

### 1. Tests unitaires

Les tests unitaires portent sur les composants individuels de l'application, notamment :

- **Inscription et activation :**
  - Vérifie que la création d'un utilisateur fonctionne correctement.
  - Vérifie l'envoi du mail d'activation et la validation du token.
- **Connexion :**
  - Teste l'authentification des utilisateurs activés.

- Vérifie que les utilisateurs non activés ne peuvent pas se connecter.
- **Achats :**
  - Teste l'enregistrement des achats dans la table Purchase.
  - Vérifie la génération des droits d'accès aux leçons et cursus achetés.
- **Validation des leçons et certifications :**
  - Vérifie que la progression des leçons est correctement enregistrée dans LessonProgress.
  - Contrôle la génération automatique des certifications une fois toutes les leçons d'un cursus validées.
- **Repositories et services :**
  - Testent les méthodes d'accès aux données et la logique métier associée.
  - Vérifient la cohérence des données et l'intégrité des relations entre entités.

## 2. Tests fonctionnels

Les tests fonctionnels simulent le parcours complet d'un utilisateur sur la plateforme :

- Création et activation du compte.
- Connexion et accès aux fonctionnalités selon le rôle.
- Achat d'un cursus ou d'une leçon individuelle.
- Accès aux leçons achetées et validation du contenu.
- Génération et consultation des certifications.

## 3. Avantages des tests

- Garantissent le bon fonctionnement des fonctionnalités critiques.
- Assurent la sécurité et la fiabilité des processus d'inscription, d'achat et de validation pédagogique.
- Facilitent la maintenance et l'évolution de l'application.
- Permettent de détecter rapidement tout comportement anormal ou régression lors des mises à jour du code.

En combinant **tests unitaires** et **tests fonctionnels**, la plateforme offre un environnement robuste et fiable, essentiel pour un projet e-learning et e-commerce destiné à des utilisateurs réels.



## Documentation du code

La documentation du code de la plateforme « Knowledge Learning » est un élément clé pour assurer la **maintenabilité**, la **compréhension** et l'**évolutivité** de l'application. Elle a été réalisée en suivant les bonnes pratiques de développement Symfony et PHP, et est entièrement rédigée en anglais conformément aux consignes.

### 1. Commentaires dans le code

- Chaque classe, méthode et fonction est documentée avec **PHPDoc**, incluant :
  - La description de la fonctionnalité.
  - Les paramètres attendus et leur type.
  - La valeur de retour et son type.
  - Les exceptions éventuelles.
- Les commentaires expliquent également la logique métier critique, comme :
  - La validation des leçons et l'attribution des certifications.
  - La gestion des achats et la simulation du paiement en mode sandbox.
  - Les contrôles d'accès en fonction des rôles utilisateurs.

### 2. Documentation des tests

- Les tests unitaires et fonctionnels sont documentés pour chaque scénario, précisant :
  - L'objectif du test.
  - Les conditions initiales (préparation de la base de données ou des entités).
  - Les résultats attendus.

- Les éventuelles contraintes ou prérequis.
- Cette approche permet de comprendre rapidement **quoi est testé et pourquoi**, facilitant la maintenance et l'ajout de nouvelles fonctionnalités.

### 3. Outils utilisés

- **PHPUnit** : pour les tests unitaires et fonctionnels.
- **PHPDoc** : pour générer automatiquement la documentation technique à partir des commentaires du code.
- **Symfony Profiler** : pour suivre et documenter les performances et le comportement des routes et services.

### 4. Avantages

- Facilite la lecture et la compréhension du code par d'autres développeurs.
- Permet une évolution rapide de la plateforme sans risque d'introduire des bugs.
- Assure une traçabilité complète des fonctionnalités et de la logique métier.
- Renforce la qualité globale du projet et la conformité aux bonnes pratiques professionnelles.

En résumé, la documentation complète et structurée du code garantit que la plateforme « Knowledge Learning » reste **robuste, compréhensible et facilement maintenable**, tout en répondant aux exigences académiques et professionnelles du projet.

## Conclusion et perspectives

Le projet « Knowledge Learning » a permis de concevoir et de développer une plateforme e-learning complète, sécurisée et modulable, répondant aux besoins pédagogiques et commerciaux de la société Knowledge. L'application offre aux utilisateurs la possibilité de s'inscrire, d'activer leur compte, d'acheter des cursus ou des leçons individuelles, de suivre leur progression et d'obtenir des certifications reconnues.

L'architecture MVC 3-tiers, combinée à Symfony 7, a assuré une séparation claire entre la présentation, la logique métier et la gestion des données. La conception orientée objet, associée à une base de données relationnelle structurée, a permis de gérer efficacement les relations entre utilisateurs, thèmes, cursus, leçons, achats et certifications. La sécurité des utilisateurs et des données a été garantie par l'encodage des mots de passe, la gestion des rôles et la protection CSRF.

La mise en place de tests unitaires et fonctionnels a permis de valider toutes les fonctionnalités critiques, assurant la fiabilité et la robustesse de l'application. La documentation complète du code, rédigée en anglais, facilite la compréhension, la maintenance et l'évolution future de la plateforme.

### Perspectives

- **Intégration d'un module de paiement réel** pour remplacer le mode sandbox et permettre des transactions en conditions réelles.
- **Ajout de nouveaux thèmes et cursus** pour enrichir l'offre pédagogique de la plateforme.
- **Fonctionnalités avancées de suivi pédagogique**, telles que statistiques de progression, quiz interactifs et recommandations personnalisées.
- **Optimisation du front-end** pour améliorer l'expérience utilisateur et l'accessibilité sur différents appareils.
- **Internationalisation** afin de rendre la plateforme accessible à des utilisateurs de différentes langues et zones géographiques.

En conclusion, « Knowledge Learning » constitue une solution e-learning moderne et évolutive, offrant à la fois une expérience utilisateur complète et un socle technique solide, capable d'accompagner la société Knowledge dans le développement de son offre de formation en ligne.