

# Knowledge Learning

plateforme e-learning

**Presente par:** Laura González PINILLA

**Formation:** Titre Professionnel Développeur Web et Web Mobile (DWWM)

Centre Européen de Formation

**Date de présentation:** 01/12/25

## Table des matières

1. Introduction.....	2
2. Presentation du Projet .....	2
3. Maquettage .....	3
3.1. capture d'ecran.....	3
4. Résumé du projet .....	4
5. Répertoire Github.....	5
6. Captures d'écran page web .....	7
7. Conception de la Base de Données .....	9
5.1. Modèle Physique des Données .....	10
5.2. Diagramme de la Base de Données .....	11
5.3. Justification de la Conception . .....	12
8. Structure du Code et architecture.....	12
9. Documentation du Code .....	13
10. Tests et validation.....	14
11. Considérations de Sécurité .....	15
12. Conclusion .....	16

# Introduction

## 1. Contexte du projet

**Projet réalisé dans le cadre du Titre Professionnel DWWM Knowledge Learning** est une plateforme d'e-learning développée pour la société fictive *Knowledge*, spécialisée dans l'édition de livres de formation. Cette plateforme permet aux utilisateurs d'accéder à des formations en ligne dans divers domaines tels que la musique, l'informatique, le jardinage ou encore la cuisine. Le site propose un système complet d'achat et de réservation en ligne, permettant aux utilisateurs d'acquérir des cursus complets ou des leçons spécifiques, avec un accès sécurisé aux contenus achetés.

### 1.1. Objectifs

- Proposer un système de gestion de cours en ligne.
- Offrir une expérience utilisateur sécurisée et simple.
- Mettre en place un e-commerce intégré avec paiement Stripe.
- Permettre aux administrateurs de gérer les utilisateurs, contenus et commandes.

### 1.2. Public cible

- Clients : utilisateurs souhaitant suivre des cours en ligne.
- Administrateurs : responsables de la gestion du catalogue, des comptes et des commandes.

La sécurité est assurée par l'utilisation de **JWT** pour l'authentification, la protection **CSRF**, et le **chiffrement des mots de passe**.

## 2. Présentation du projet

### Fonctionnalités principales

- Création et activation de comptes utilisateurs (token email).
- Authentification sécurisée avec JWT + sessions PHP.
- Gestion des rôles (admin, client).
- Panier d'achat dynamique.
- Paiement sécurisé via Stripe (mode test).
- Accès aux cours/ leçons achetés.
- Validation des leçons et délivrance automatique de certifications.

- Interface d'administration : gestion des utilisateurs, contenus et commandes.

## Défis rencontrés

- Intégration de Stripe pour les paiements en mode test.
- Mise en place de la logique de certification.

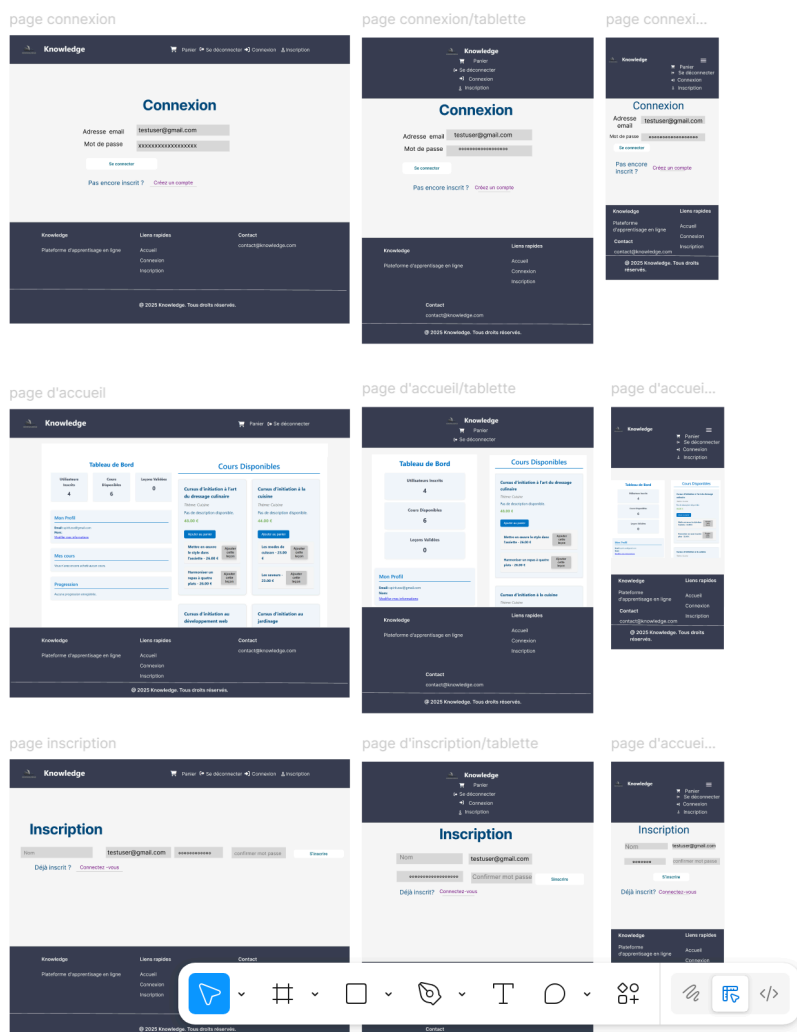
## Solutions apportées

- Recherche documentaire et utilisation de l'API officielle Stripe.
- Tests unitaires pour sécuriser la logique métier.

## 3. Maquettage

Réalisation de maquettes desktop, tablette et mobile avec Figma.

Interface simple et intuitive, pensée pour la lisibilité et l'ergonomie.



## 4. Résumé du Projet

La plateforme **Knowledge Learning** est un site web d'e-learning conçu pour proposer des cours et des leçons en ligne aux utilisateurs. Elle permet la création de comptes, l'achat de formations, et la délivrance de **certifications** après la complétion des cours.

### Développement Front-End

- Technologies utilisées : HTML, CSS, JavaScript.
- Architecture MVC : séparation entre logique, données et présentation.
- Partie statique : pages d'accueil, formulaires d'inscription/connexion.
- Partie dynamique : gestion du panier, formulaires interactifs, validation des cours.

### Développement Back-End

- Langage : PHP orienté objet.
- Architecture : MVC maison (contrôleurs, modèles, vues).
- Base de données : avec **phpMyAdmin** + MySQL (tables users, cursus, leçons, orders, validations, certifications).
- Sécurité :
  - Authentification avec JWT.
  - Hachage des mots de passe avec bcrypt, Une validation de mot de passe renforcée a été prévue (longueur, caractères spéciaux).
  - Protection CSRF.
  - Requêtes préparées (PDO) pour éviter les injections SQL.

Le projet se concentre principalement sur le développement **back-end**, incluant :

- La conception de la **base de données**,
- La gestion des **composants d'accès aux données**,
- L'intégration d'un système **e-commerce**,
- la **gestion de la sécurité** (CSRF, JWT, injection SQL),
- la **mise en place de rôles utilisateurs** (administrateur/client) avec contrôle d'accès,
- le traitement des commandes et la gestion des certifications.

Le tout a été développé en **PHP/avec phpMyAdmin + MySQL**, en suivant une approche **orientée objet** (modèle **MVC**), avec des **tests unitaires** pour assurer la robustesse du code.

### Défis rencontrés

Deux défis majeurs ont été relevés :

- L'intégration de la **passerelle de paiement** (mode test),
- La mise en œuvre de la logique de **certification**.

Ces difficultés ont été surmontées grâce à des phases de **recherche approfondie et de tests**.

### Enseignements tirés

Ce projet m'a permis de consolider mes connaissances sur :

- La **conception de bases de données** relationnelles,
- Les **principes de la programmation orientée objet**,
- Les **bonnes pratiques de sécurité web**,
- L'importance des **tests automatisés** dans un projet professionnel.

## 5. Répertoire GitHub

Le code source de la plateforme Knowledge Learning est disponible sur GitHub à l'URL suivante :

<https://github.com/gopileau/knowledge.git>

### README.md

#### Knowledge Learning Platform

##### Prérequis

- PHP 7.4 ou supérieur
- Composer
- Base de données MySQL ou compatible
- Serveur web Apache ou Nginx
- Clé secrète JWT (à définir dans le fichier .env)
- Clés API Stripe pour le paiement (mode sandbox recommandé)

##### Installation

1. Cloner le dépôt GitHub :

git clone <https://github.com/gopileau/knowledge.git>

markdown Copier le code 2. Installer les dépendances PHP avec Composer :  
composer install

markdown Copier le code 3. Configurer la base de données dans le fichier config/database.php ou .env

4. Lancer les migrations pour créer les tables :

php run\_migration.php

markdown Copier le code 5. Configurer la clé secrète JWT dans .env :

JWT\_SECRET=VotreCleSecrete

markdown Copier le code 6. Configurer les clés API Stripe dans config/config.php ou .env

## Authentification

Le système d'authentification utilise :

- **JWT (JSON Web Tokens)** pour une authentification stateless
- **Sessions PHP** pour rétrocompatibilité
- **Tokens d'activation par email**
- **Protection CSRF**

## Fonctionnalités JWT

- Tokens valables 1 heure
- Stockage sécurisé en cookies HttpOnly
- Compatible avec les applications SPA (Single Page Application)

## Configuration

- Ajouter JWT\_SECRET dans le fichier .env
- Installer la dépendance JWT avec Composer :  
composer require firebase/php-jwt

markdown Copier le code

- Les sessions existantes continuent de fonctionner
- Les nouveaux logins génèrent à la fois session et JWT

## Lancement

Pour lancer le serveur PHP intégré en développement :

php -S localhost:8000 -t public

markdown Copier le code Puis accéder à l'application via <http://localhost:8000>

## Tests

Exécuter les tests unitaires avec PHPUnit :

vendor/bin/phpunit tests/

markdown Copier le code Les tests couvrent :

- Authentification
- Gestion des utilisateurs
- Fonctionnalités principales

## Déploiement

Le projet est déployé sur l'hébergement InfinityFree. Pour déployer :

- Transférez les fichiers du projet via FTP vers votre espace InfinityFree.
- Configurez la base de données MySQL sur InfinityFree et mettez à jour les paramètres dans config/database.php ou .env.
- Assurez-vous que les clés API Stripe et JWT sont configurées dans .env.
- Configurez les permissions nécessaires sur le serveur.
- Un fichier deploy.php est disponible pour un déploiement automatisé sur d'autres serveurs si besoin.

## Fonctionnalités principales

- Authentification sécurisée avec JWT et sessions PHP
- Gestion des rôles utilisateurs (admin, client)

- Panier d'achat et paiement via Stripe
- Accès aux cours et leçons achetés
- Validation des leçons et obtention de certifications
- Interface d'administration pour la gestion des utilisateurs, contenus et commandes

### Lien vers le dépôt GitHub

<https://github.com/gopileau/knowledge>

### Lien vers page web

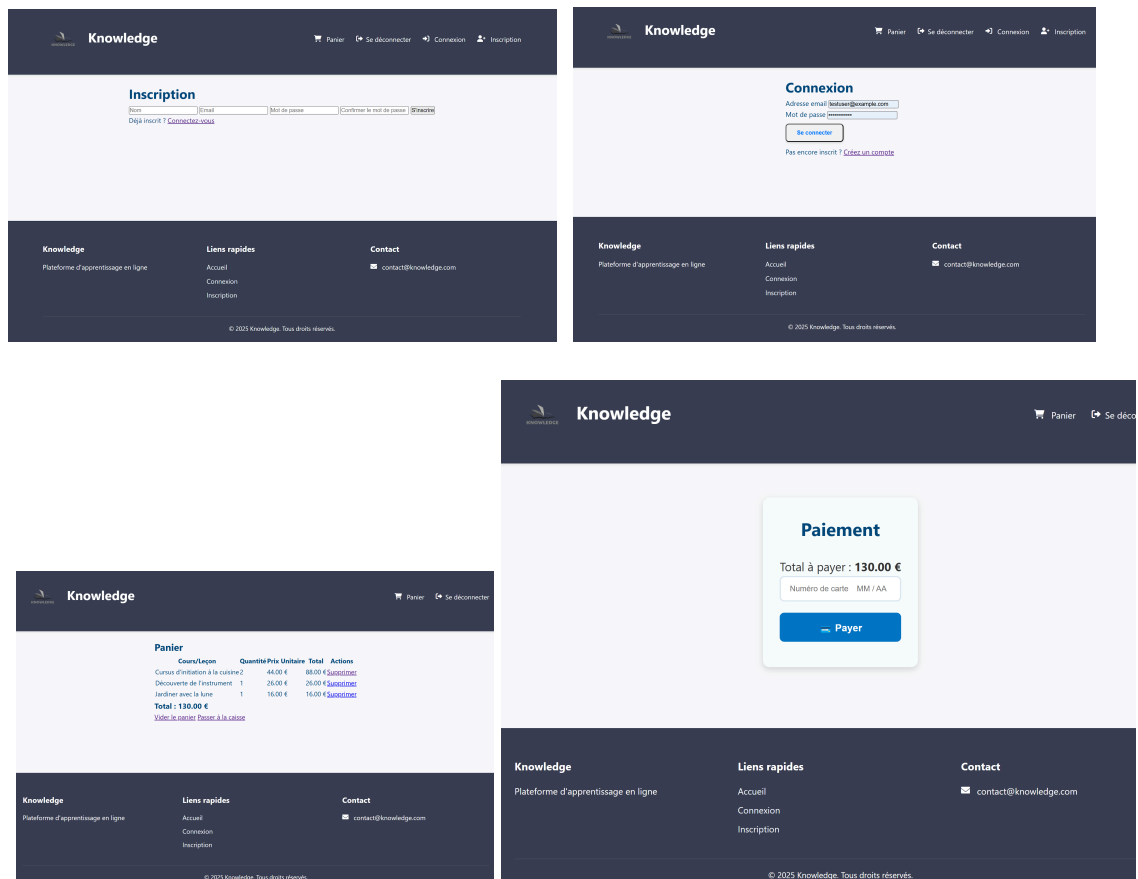
<https://knowledgelearn.rf.gd>

## Toute information complémentaire

### Difficultés rencontrées / solutions

- L'envoi d'e-mails d'activation n'est pas fonctionnel en production, car l'hébergeur gratuit InfinityFree bloque les connexions SMTP sortantes. Cependant, le système d'activation par token est bien implémenté côté serveur et peut être testé via un lien simulé généré après l'inscription.  
Ex : <https://knowledgelearn.rf.gd/activate?token=XYZ>

## 6. Images de la page web





Knowledge

Menu

Se connecter

Utilisateurs inscrits3

Cours Disponibles6

Leçons Validées0

Mon Profil

Profil

lauravgonzalez@gmail.com

Modifier mes informations

Mon cours

Cours d'initiation à la guitare

(3 leçons)

Ajouter à mon panier

Cours d'initiation à la cuisine

(4 leçons)

Ajouter à mon panier

Cours d'initiation à la guitare

(3 leçons)

Ajouter à mon panier

Cours d'initiation au développement web

(5 leçons)

Ajouter à mon panier

Cours d'initiation au jardinage

(3 leçons)

Ajouter à mon panier

Cours d'initiation au piano

(3 leçons)

Ajouter à mon panier

Progression

Ajouter à mon panier

Cours Disponibles

Cours d'initiation à la cuisine

Thème: Cuisine

Par de développement web

45,00 €

Ajouter à mon panier

Cours d'initiation à la guitare

Thème: Guitare

Par de développement web

45,00 €

Ajouter à mon panier

Cours d'initiation au développement web

Thème: Développement web

Par de développement web

60,00 €

Ajouter à mon panier

Cours d'initiation au jardinage

Thème: Jardinage

Par de développement web

30,00 €

Ajouter à mon panier

Cours d'initiation au piano

Thème: Piano

Par de développement web

30,00 €

Ajouter à mon panier

Mon Achats

Achat du titre: Cuisine - 45,00 €

Achat du titre: Guitare - 45,00 €

Achat du titre: Développement web - 60,00 €

Achat du titre: Jardinage - 30,00 €

Achat du titre: Piano - 30,00 €

Activité Récente

Ajouter à mon panier

Notifications

Ajouter à mon panier

Se déconnecter

Knowledge

Plateforme d'apprentissage en ligne

Liens rapides

Accueil

Connexion

Inscription

Contact

contact@knowledge.com

© 2025 Knowledge. Tous droits réservés.

Knowledge

Menu

Se connecter

Tableau de bord Admin

Utilisateurs3

Cours6

Commandes7

Certifications0

Gestion utilisateurs

Gestion contenus

Historique achats

Certifications

Dernières commandes

ID	UTILISATEUR	DATE	TOTAL (€)
7	<a href="mailto:lauravgonzalez99@gmail.com">lauravgonzalez99@gmail.com</a>	16/05/2025 12:34	0.00
4	<a href="mailto:lauravgonzalez99@gmail.com">lauravgonzalez99@gmail.com</a>	16/05/2025 12:32	0.00
5	<a href="mailto:lauravgonzalez99@gmail.com">lauravgonzalez99@gmail.com</a>	16/05/2025 12:32	0.00
6	<a href="mailto:lauravgonzalez99@gmail.com">lauravgonzalez99@gmail.com</a>	16/05/2025 12:32	0.00
3	<a href="mailto:lauravgonzalez99@gmail.com">lauravgonzalez99@gmail.com</a>	16/05/2025 12:29	0.00

Se déconnecter

Knowledge

Plateforme d'apprentissage en ligne

Liens rapides

Accueil

Connexion

Inscription

Contact

contact@knowledge.com

© 2025 Knowledge. Tous droits réservés.

Knowledge

Menu

Se connecter

Gestion des Contenus

Nouveaux contenus

Coursus

ID	Titre	Type	Statut	Actions
7	Cours d'initiation à la cuisine	Course Active	Modifier   Supprimer	
6	Cours d'initiation à la cuisine	Course Active	Modifier   Supprimer	
4	Cours d'initiation au développement web	Course Active	Modifier   Supprimer	
5	Cours d'initiation au jardinage	Course Active	Modifier   Supprimer	
1	Cours d'initiation à la guitare	Course Active	Modifier   Supprimer	
3	Cours d'initiation au piano	Course Active	Modifier   Supprimer	

Leçons

ID	Titre	Type	Statut	Actions
1	Découverte de l'instrument	Lesson Active	Modifier   Supprimer	
2	Les accords et les gammes	Lesson Active	Modifier   Supprimer	
3	Découverte de l'instrument	Lesson Active	Modifier   Supprimer	
4	Les accords et les gammes	Lesson Active	Modifier   Supprimer	
7	Les langages HTML et CSS	Lesson Active	Modifier   Supprimer	
8	Dynamiser votre site avec JavaScript	Lesson Active	Modifier   Supprimer	
9	Les outils du jardinier	Lesson Active	Modifier   Supprimer	
10	Jardiner avec la lune	Lesson Active	Modifier   Supprimer	
11	Les modes de cuisson	Lesson Active	Modifier   Supprimer	
12	Les saveurs	Lesson Active	Modifier   Supprimer	
13	Mettre en œuvre le style dans l'assiette	Lesson Active	Modifier   Supprimer	
14	Hammoniser un repas à quatre plats	Lesson Active	Modifier   Supprimer	
15	Découverte de l'instrument	Lesson Active	Modifier   Supprimer	
16	Les accords et les gammes	Lesson Active	Modifier   Supprimer	
19	Les langages HTML et CSS	Lesson Active	Modifier   Supprimer	
20	Dynamiser votre site avec JavaScript	Lesson Active	Modifier   Supprimer	
21	Les outils du jardinier	Lesson Active	Modifier   Supprimer	
22	Jardiner avec la lune	Lesson Active	Modifier   Supprimer	
23	Les modes de cuisson	Lesson Active	Modifier   Supprimer	
24	Les saveurs	Lesson Active	Modifier   Supprimer	
25	Mettre en œuvre le style dans l'assiette	Lesson Active	Modifier   Supprimer	
26	Hammoniser un repas à quatre plats	Lesson Active	Modifier   Supprimer	

Knowledge

Plateforme d'apprentissage en ligne

Liens rapides

Accueil

Connexion

Inscription

Contact

contact@knowledge.com

© 2025 Knowledge. Tous droits réservés.

Knowledge

Menu

Se connecter

Gestion des Utilisateurs

ID	Email	Rôle	Statut	Actions
1	<a href="mailto:lauravgonzalez99@gmail.com">lauravgonzalez99@gmail.com</a>	admin	client	Modifier   Supprimer
6	<a href="mailto:spitfirez@gmail.com">spitfirez@gmail.com</a>	client	client	Modifier   Supprimer
50	<a href="mailto:tesuser@example.com">tesuser@example.com</a>	client	client	Modifier   Supprimer

[Revenir à l'accueil admin](#)

Knowledge

Plateforme d'apprentissage en ligne

Liens rapides

Accueil

Connexion

Inscription

Contact

contact@knowledge.com

© 2025 Knowledge. Tous droits réservés.

## 7. Conception de la Base de Données

### Modèle Physique des Données (Schéma)

Le **Modèle Physique de Données (MPD)** représente **les tables** de ta base **telles qu'elles existent avec phpMyAdmin + MySQL**, avec leurs **champs, types, clés primaires, relations**.

Table	Action	Lignes	Type	Interclassement
<input type="checkbox"/> certificates	★	0	InnoDB	utf8mb4_unicode_ci
<input type="checkbox"/> certifications	★	0	InnoDB	utf8mb4_unicode_ci
<input type="checkbox"/> courses	★	6	InnoDB	utf8mb4_unicode_ci
<input type="checkbox"/> lessons	★	22	InnoDB	utf8mb4_unicode_ci
<input type="checkbox"/> lesson_completions	★	0	InnoDB	utf8mb4_unicode_ci
<input type="checkbox"/> purchases	★	7	InnoDB	utf8mb4_unicode_ci
<input type="checkbox"/> themes	★	4	InnoDB	utf8mb4_unicode_ci
<input type="checkbox"/> users	★	3	InnoDB	utf8mb4_unicode_ci
<input type="checkbox"/> user_lessons	★	0	InnoDB	utf8mb4_unicode_ci
9 tables	Somme	42	MyISAM	utf8mb4_unicode_ci

### Modèle Physique de Données - Plateforme Knowledge Learning

#### Tables et Champs

##### users

*Table : Stocke les informations de compte utilisateur.*

id : int (Clé primaire, auto-incrément)

name : varchar

email : varchar (Unique, Index)

password : varchar

role : varchar

is\_activated : boolean

activation\_token : varchar

created\_at : timestamp

updated\_at : timestamp

##### themes

*Table : Stocke les thèmes des cours.*

id : int (Clé primaire, auto-incrément)

title : varchar

##### cursus

*Table : Stocke les cours, chacun appartenant à un thème.*

id : int (Clé primaire, auto-incrément)

title : varchar

theme\_id : int (Clé étrangère vers themes.id)

price : decimal

##### lessons

*Table : Stocke les leçons, chacune appartenant à un cours.*

id : int (Clé primaire, auto-incrément)

title : varchar  
cursus\_id : int (Clé étrangère vers cursus.id)  
price : decimal  
video\_url : varchar  
content : text

#### **orders**

*Table : Stocke les informations de commande.*

id : int (Clé primaire, auto-incrément)  
user\_id : int (Clé étrangère vers users.id)  
total\_price : decimal  
created\_at : timestamp

#### **order\_items**

*Table : Stocke les éléments inclus dans une commande. Peut être des leçons ou des cours.*

id : int (Clé primaire, auto-incrément)  
order\_id : int (Clé étrangère vers orders.id)  
lesson\_id : int (Clé étrangère vers lessons.id, nullable)  
cursus\_id : int (Clé étrangère vers cursus.id, nullable)

#### **validations**

*Table : Stocke les validations de leçons par les utilisateurs.*

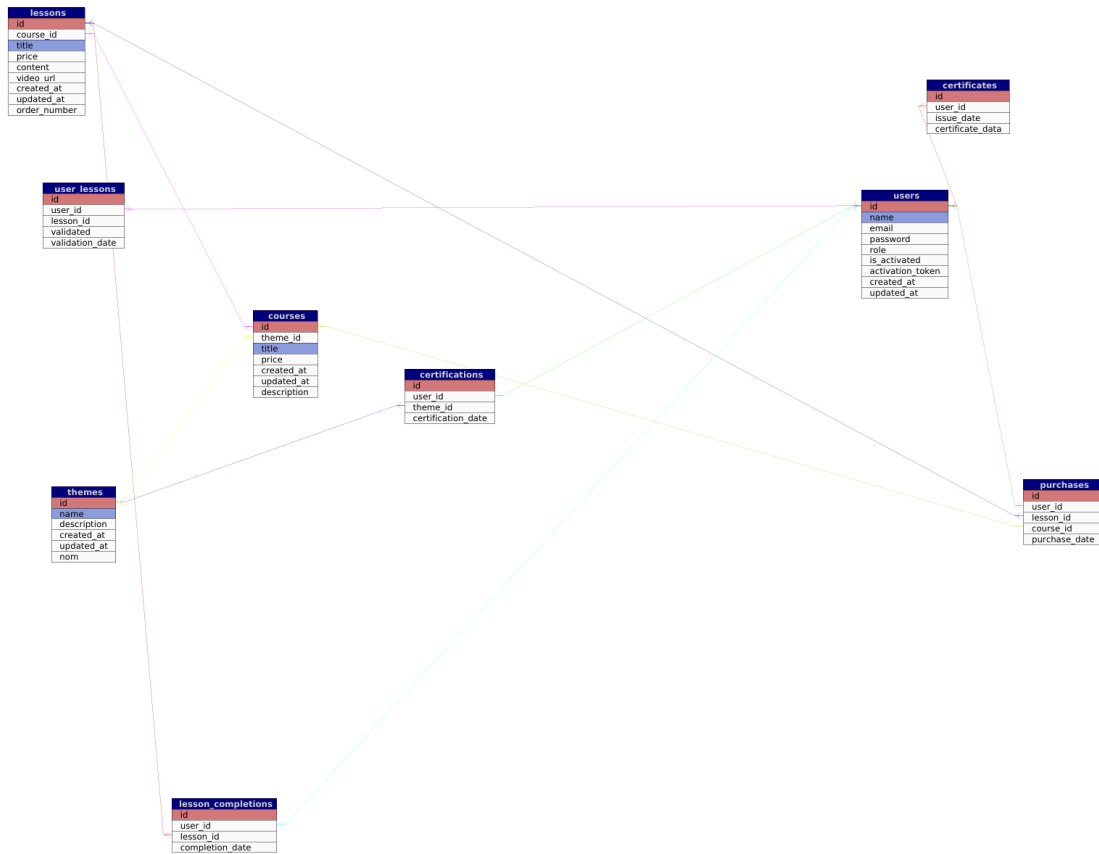
id : int (Clé primaire, auto-incrément)  
user\_id : int (Clé étrangère vers users.id)  
lesson\_id : int (Clé étrangère vers lessons.id)  
validated\_at : timestamp

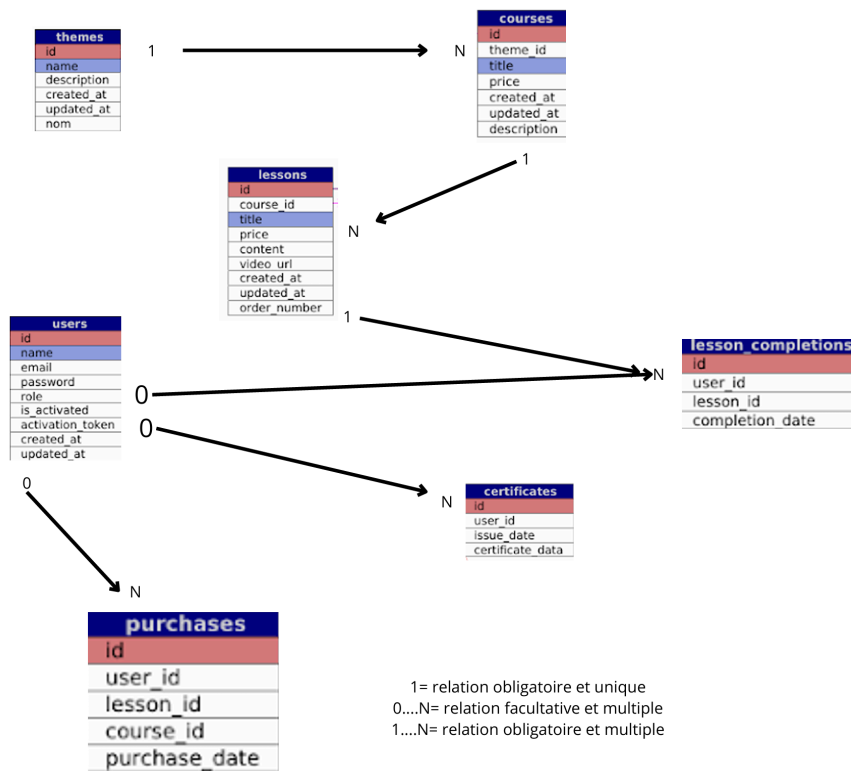
#### **certifications**

*Table : Stocke les certifications obtenues par les utilisateurs pour avoir terminé des cours.*

id : int (Clé primaire, auto-incrément)  
user\_id : int (Clé étrangère vers users.id)  
cursus\_id : int (Clé étrangère vers cursus.id)  
date\_obtained : timestamp

## Diagramme de la Base de Données





## Justification de la Conception de la Base de Données

Le diagramme relationnel ci-dessus respecte les règles de modélisation relationnelle, avec des cardinalités explicites pour chaque lien. Les relations 1-N sont représentées par des flèches directionnelles accompagnées de leurs contraintes, permettant une lecture claire et conforme au Référentiel d'Évaluation.

Le schéma de la base de données est conçu pour prendre en charge les fonctionnalités de base de la plateforme Knowledge Learning, notamment la gestion des utilisateurs, le catalogue de cours, le e-commerce et la certification.

La table users stocke les informations de compte utilisateur, notamment le nom, l'adresse e-mail, le mot de passe, le rôle et l'état d'activation. Le champ email est unique pour garantir que chaque utilisateur possède un identifiant unique. Le champ role permet un contrôle d'accès basé sur les rôles, différents rôles ayant des autorisations différentes.

Les tables themes, cursus et lessons représentent le catalogue de cours. La table themes stocke les thèmes des cours, la table cursus stocke les cours et la table lessons stocke les leçons de chaque cours. Les relations entre ces tables sont de un-à-plusieurs.

Les tables `orders` et `order_items` prennent en charge la fonctionnalité de e-commerce. La table `orders` stocke les informations de commande, y compris l'utilisateur qui a passé la commande et le prix total. La table `order_items` stocke les éléments inclus dans chaque commande. La table `order_items` peut référencer soit une leçon, soit un cours, offrant une flexibilité dans ce que les utilisateurs peuvent acheter.

Les tables `validations` et `certifications` prennent en charge la fonctionnalité de certification. La table `validations` suit les leçons qu'un utilisateur

---

## Structure du Code et Patrons de Conception

Le projet **Knowledge Learning** est développé en **PHP orienté objet**, en suivant une architecture **MVC (Modèle-Vue-Contrôleur)**. Cette structure assure une séparation claire des responsabilités :

- **Modèle** : gestion des données et des entités,
- **Contrôleur** : logique métier et gestion des flux,
- **Vue** : présentation des données à l'utilisateur.

Plusieurs **patrons de conception** ont été mis en œuvre pour renforcer la maintenabilité et la modularité du code :

- **Patron MVC** : structure générale de l'application.
- **Singleton** : utilisé pour la classe `Database`, garantissant une unique instance de connexion à la base de données.
- **Repository** : pour centraliser et abstraire les accès aux données.

- **Service** : pour encapsuler la logique métier, comme dans CartService.
- Utilisation des **sessions PHP** : pour la gestion du panier sans base de données intermédiaire.

## 9.Documentation du code

Le code est entièrement documenté avec des **commentaires PHPDoc**, ce qui facilite sa lecture, sa compréhension et sa maintenance. Chaque classe et méthode comporte :

- une description claire de sa finalité,
- les paramètres attendus,
- les types de retour.

### Exemples de commentaires PHPDoc :

```
/**
 * Ajoute un élément au panier.
 *
 * @return void
 */
public function add() {
    // ...
}

/**
 * Supprime un élément du panier.
 *
 * @param int $id Identifiant de l'élément à supprimer.
 * @return void
 */
public function remove($id) {
    // ...
}
```

### Implémentation du E-Commerce

Le module e-commerce repose principalement sur le **CartController** et le **CartService**. Il propose les fonctionnalités suivantes :

- Ajout d'articles (cours ou leçons) au panier, avec vérification des doublons.
- Consultation, suppression et vidage du panier.
- Paiement sécurisé via l'API **Stripe** (mode test), avec gestion des PaymentIntents.
- Finalisation de la commande et réinitialisation du panier après paiement.

Le **panier est stocké en session**, ce qui permet de conserver les données de manière simple et efficace sans nécessiter de table dédiée.

Le paiement via **Stripe** est configuré avec une **clé API de test** définie dans le contrôleur. La gestion des erreurs et des statuts de paiement a été intégrée pour garantir une expérience utilisateur fluide et sécurisée.

Cette architecture modulaire facilite l'**évolutivité** et la **maintenance** du système e-commerce, tout en assurant une bonne séparation des responsabilités.

## 10. Tests

L'application **Knowledge Learning** intègre plusieurs tests unitaires afin d'assurer la stabilité et la fiabilité des fonctionnalités principales. Ces tests sont réalisés à l'aide de **PHPUnit**, et couvrent les composants critiques du système.

### Types de tests effectués

- **Tests unitaires**, couvrant :
  - L'authentification (création de compte, connexion, vérification des identifiants),
  - La gestion des utilisateurs (création, modification, suppression),
  - Le panier (ajout, suppression, vidage, calcul du total),
  - Les fonctionnalités administratives (gestion des cours, des leçons et des utilisateurs).
- **Tests de migration** : validation du bon déroulement des migrations de base de données et de l'intégrité des schémas.
- **Tests ciblés** :
  - tests/CartTest.php : tests spécifiques aux opérations du panier,
  - tests/AuthTest.php : tests liés à l'authentification (connexion, session, contrôle des rôles).

### Fonctionnalités principales couvertes

- Authentification et sécurité des accès,
- Gestion des comptes utilisateurs (création, rôles, modification),
- Fonctionnement du panier (ajouts, suppressions, vidage, cohérence),
- Fonctionnalités administratives (accès, droits, modifications).



## 11. Considérations de Sécurité

L'application **Knowledge Learning** intègre plusieurs mécanismes pour assurer la sécurité des données utilisateur et la protection contre les attaques courantes :

### Sécurité des comptes utilisateurs

- Utilisation de `password_hash()` avec l'algorithme **BCRYPT** pour un hachage sécurisé des mots de passe.
- Vérification du mot de passe à la connexion via `password_verify()`.
- Génération d'un **token d'activation unique** avec `bin2hex(random_bytes(32))` pour la validation des comptes.
- Activation de compte sécurisée via un lien contenant le token.

### Intégrité des données

- Vérification de l'existence d'un email avant l'enregistrement pour éviter les **comptes en doublon**.
- Utilisation de **requêtes préparées PDO** pour prévenir les **injections SQL**.

### Gestion des rôles

- Stockage des rôles (admin, client) et de l'état d'activation directement en base de données.
- Séparation claire des privilèges selon le rôle attribué à l'utilisateur (par défaut : client).

### Autres mesures

- **Jetons CSRF** intégrés dans les formulaires sensibles.
- Validation à améliorer : vérification de la **longueur minimale** et des **caractères spéciaux** dans les mots de passe (à faire si non encore implémenté).

### Critères RGPD et Sécurité Couverts

- **Hachage sécurisé** des mots de passe
- **Jetons CSRF** actifs dans les formulaires
- **Protection SQL** via requêtes préparées
- **Contrôle d'accès** selon les rôles
- **À faire** : validation renforcée des mots de passe côté serveur

+ Options											
				id	name	email	password		role	is_activated	activation_token
<input type="checkbox"/>				1		lauravgonzalez99@gmail.com	\$2y\$10\$Hw1bAeinJP9erX9lowHzOeZ.DthFrY29HW8lhbqKQ4F...		admin	1	576f0e98d64c34e5c4da96972f2f9ea71edd836294f390570...
<input type="checkbox"/>				6		spirit.zox@gmail.com	\$2y\$10\$Qd1tCA2qflobbDfHHRgPL3TBLMCIBYbS5iHWbJJxg...		client	1	NULL
<input type="checkbox"/>				50	user	testuser@example.com	\$2y\$10\$WNkwEt8oG8cySUHg12RT9OCHzoHArICsr6W7QnWiK6L...		client	0	56833df5309486dd51b21df668eb3f622346a95557469efa9b...

## 12. Conclusion

### Compétences acquises

- Conception d'une base de données relationnelle (Merise/UML).
- Développement front-end et back-end complet.
- Mise en place d'une architecture MVC avec PHP.
- Sécurisation d'une application web (JWT, CSRF, bcrypt).
- Déploiement sur un serveur distant (InfinityFree).

### Améliorations possibles

- Ajout d'un système d'e-mails d'activation pleinement fonctionnel.
- Optimisation de la validation des formulaires côté serveur.
- Modernisation de l'interface utilisateur (responsive design avancé).

Ce projet m'a permis de consolider mes compétences en développement web, en approfondissant notamment l'architecture MVC avec PHP. J'ai pu développer une application complète intégrant des fonctionnalités essentielles telles que l'inscription et la connexion sécurisées, la gestion des rôles utilisateurs, l'intégration d'un système de paiement (Stripe), ainsi que la documentation du code principalement réalisée avec PHPDoc.

Au-delà de l'aspect technique, ce projet m'a aussi permis de mieux appréhender l'importance de la sécurité, de la structuration du code, et de la documentation pour assurer la maintenabilité et la scalabilité d'une application.

Plusieurs pistes d'amélioration restent envisageables, notamment un renforcement de la validation des formulaires, une sécurisation accrue contre les attaques CSRF, ainsi que la création d'une interface utilisateur plus moderne et responsive.

En somme, ce projet a été une excellente opportunité de mettre en pratique mes connaissances dans un contexte proche de celui rencontré en entreprise. Il constitue une base solide sur laquelle je pourrai continuer à développer mes compétences, en intégrant par exemple des frameworks modernes ou en explorant d'autres architectures telles que REST ou API-first.

---