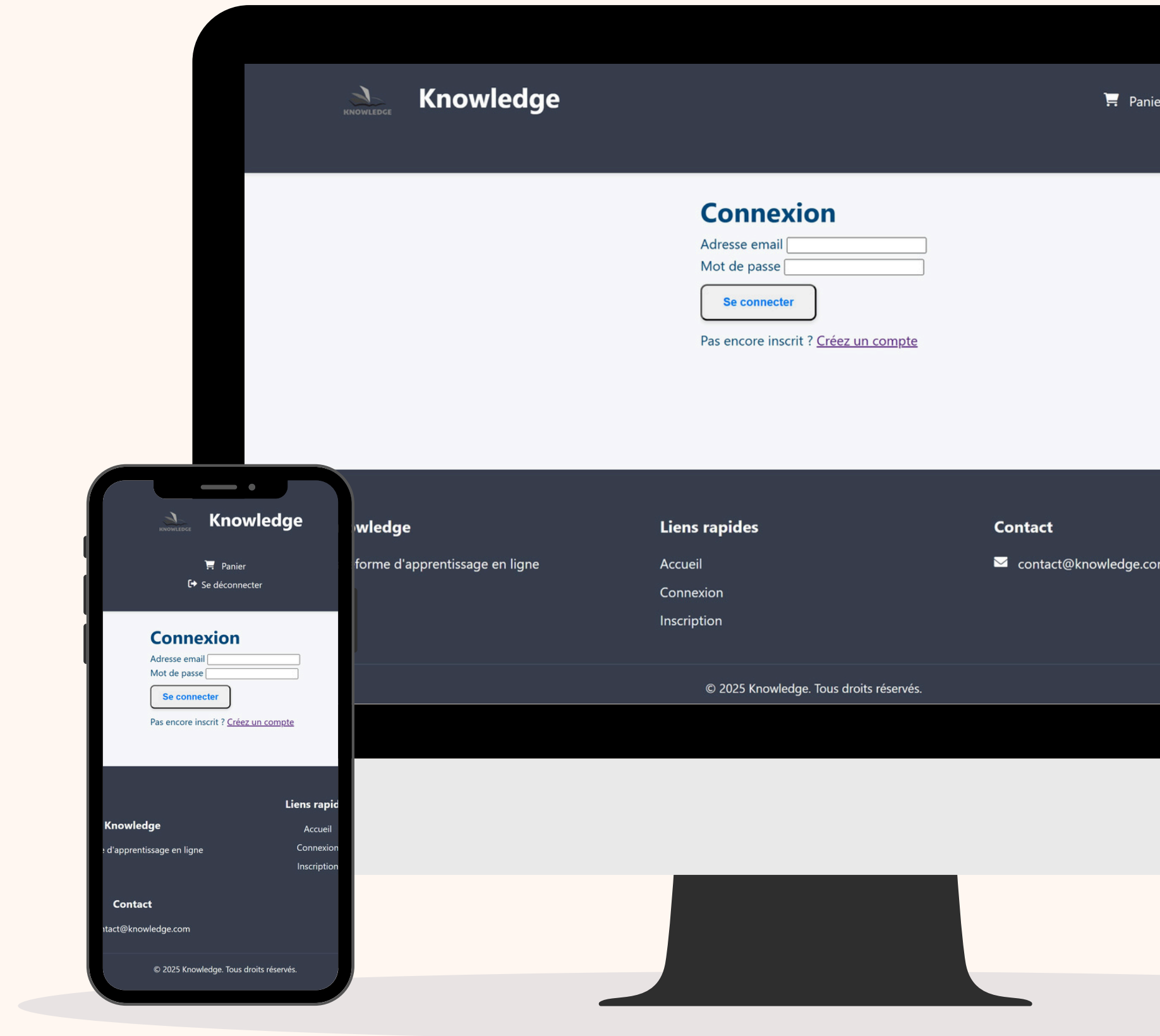
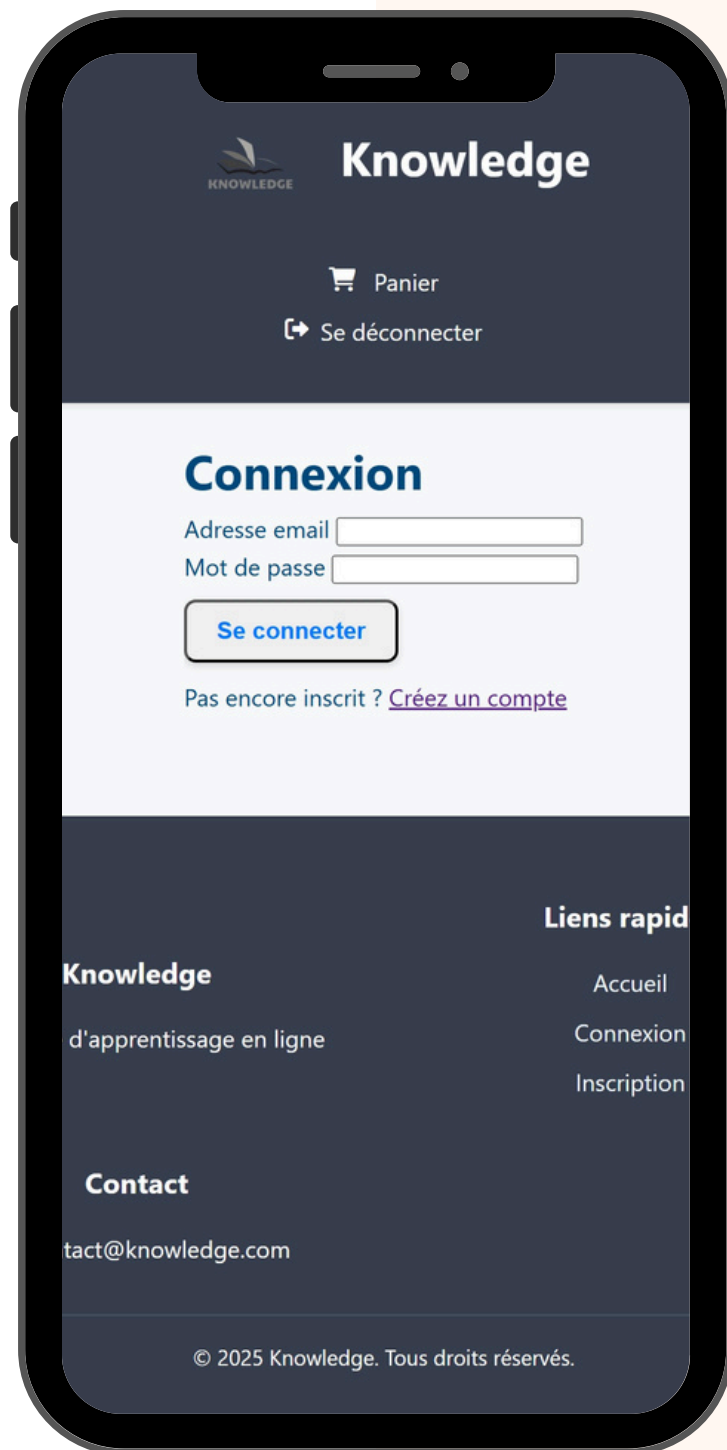


CRÉER LA  
PLATEFORME

# KNOWLEDGE LEARNING - PLATEFORME E-LEARNING

LAURA GONZALEZ





- 01 **Contexte de l'entreprise et du projet.**
- 02 **Justification de la conception de la base de données et Mise en place de la base de données.**
- 03 **Composants d'accès aux données et Composants e-commerce.**
- 04 **Sécurité et Tests.**
- 05 **Recherches et développement.**
- 06 **Conclusion et questions.**

# Contexte de l'entreprise et du projet

## La société fictive : Knowledge

- Mission : démocratiser l'apprentissage grâce à des supports accessibles
- Vision : passer du papier au numérique pour toucher un public plus large
- Positionnement : formations pratiques (musique, informatique, jardinage...)

## Pourquoi une plateforme e-learning ?

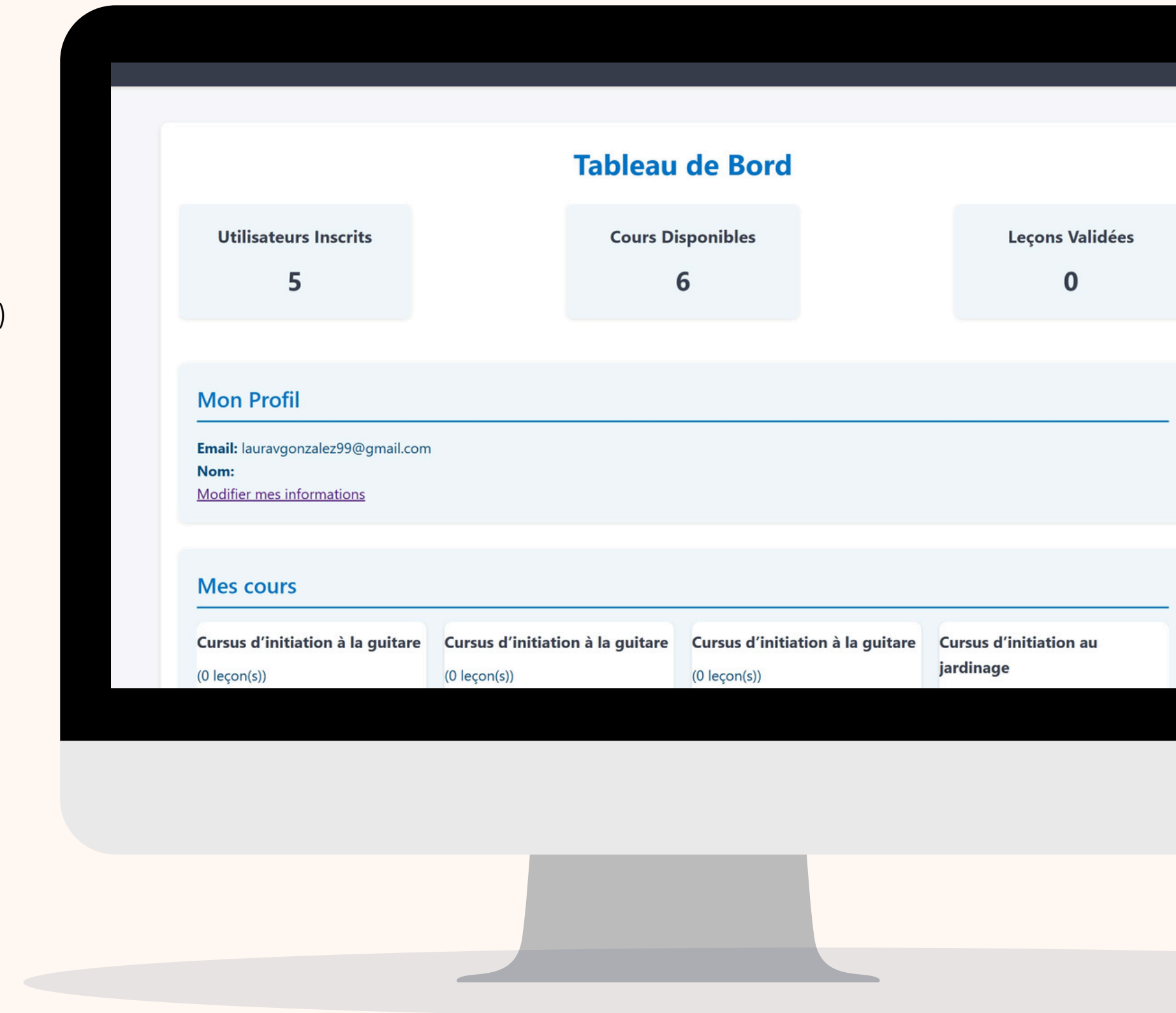
- Répondre à la demande croissante de formation à distance
- Compléter l'offre papier par des contenus vidéos et interactifs
- Suivre la progression des utilisateurs

## Contexte et objectifs du projet

- Besoins identifiés :
  - Vente de formations en ligne
  - Suivi de l'apprentissage
  - Automatisation des certifications

## Périmètre fonctionnel

- Utilisateurs :
  - Créer un compte, activer par mail
  - Acheter un cursus ou une leçon
  - Valider les leçons, obtenir une certification
- Administrateurs :
  - Gérer les comptes, contenus, commandes
  - Accès au backoffice sécurisé



# Justification de la conception de la base de données

## 1 Choix de la méthode de conception

- approche UML2.
- Explication succincte des avantages de la méthode choisie pour ce projet.

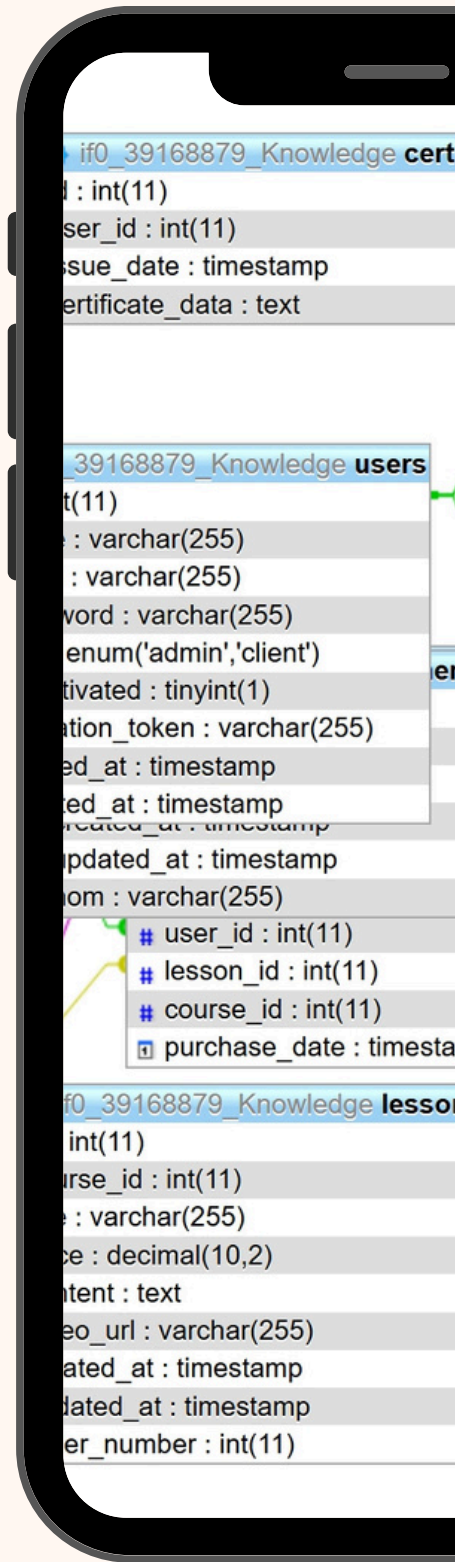
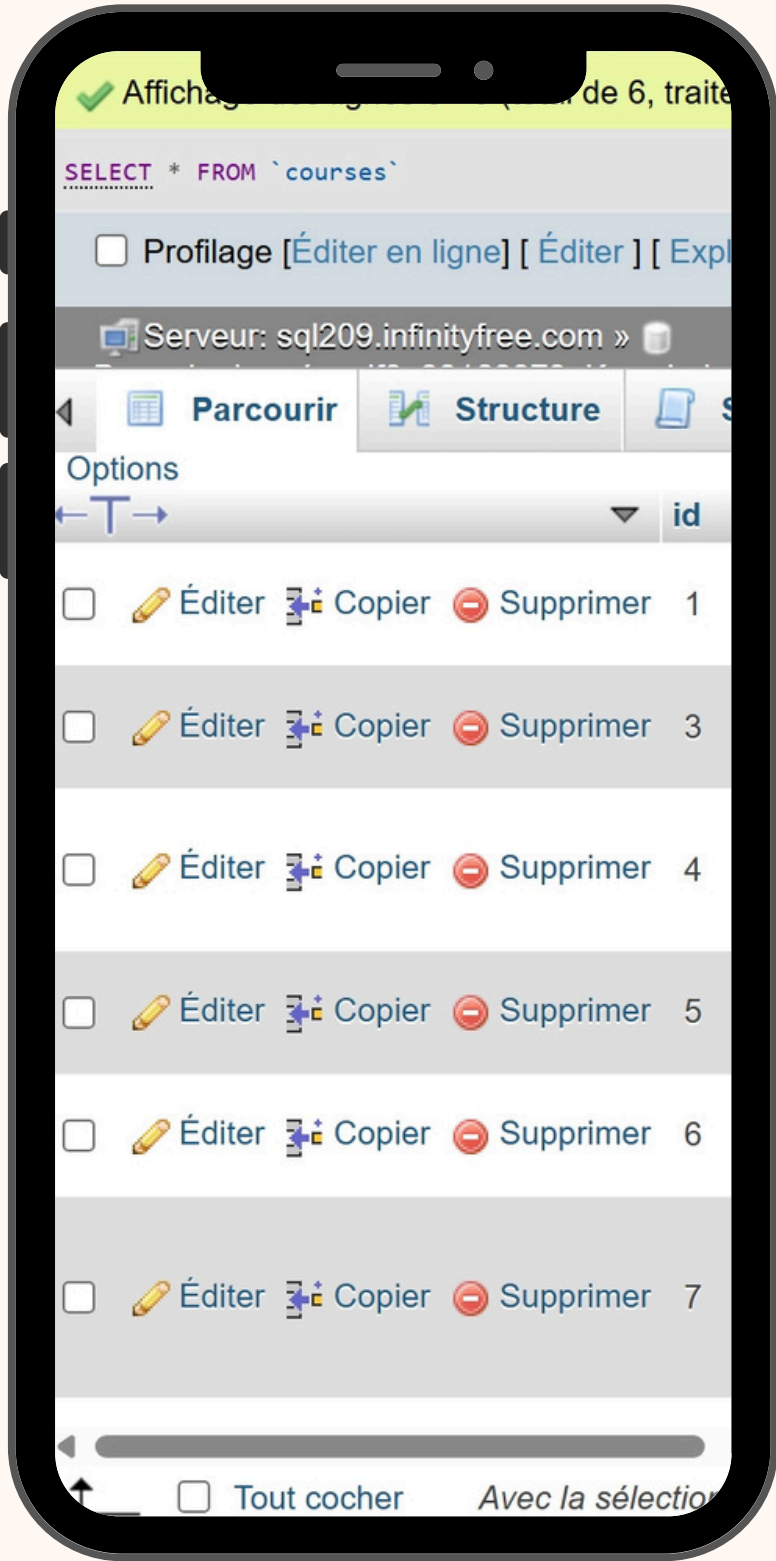
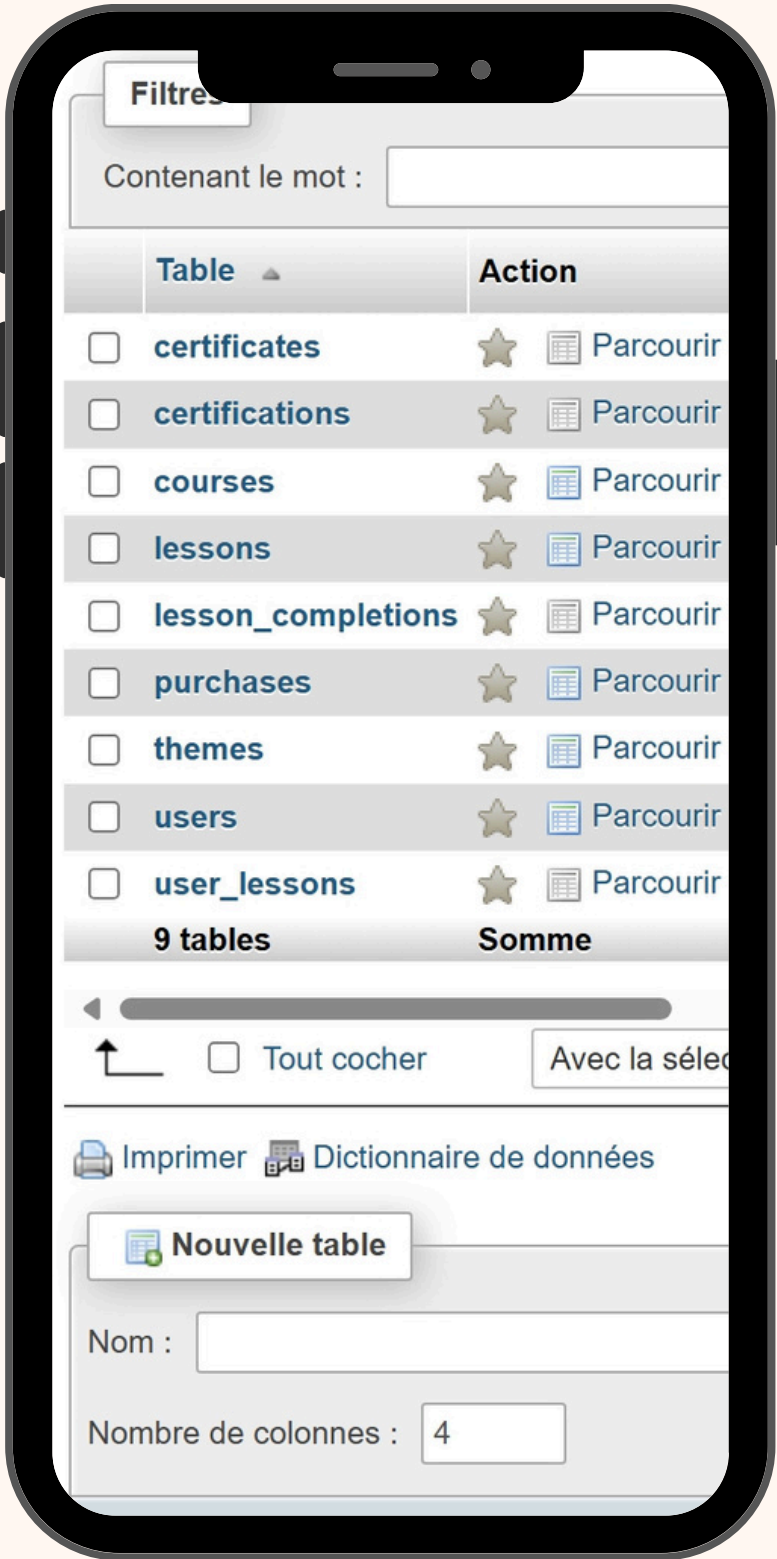
## 2 Structure et relations

8 tables principales : users, themes, cursus, lessons, orders, order\_items, validations, certifications

- 1→N : un thème contient plusieurs cursus ; un cursus contient plusieurs leçons
- N→N simulé : via table order\_items (un utilisateur peut acheter plusieurs éléments)

## 3 Qualité du modèle

- Données traçables : created\_at, updated\_at, created\_by
- Modèle normalisé : pas de redondance, clé primaire/étrangère bien définies
- Compatible avec les règles du e-learning (validation des leçons, certifications)



# Mise en place de la base de données

Implémentation de la base de données :

- Processus de création et de structuration: Doctrine Migrations pour PHP et manuellement.
- Utilisation des migrations pour la gestion du schéma de la base de données.

Avantages des migrations:

- Historique des modifications du schéma (versioning)
- Partage du schéma en équipe (collaboration)
- Réversibilité : rollback si besoin
- Automatisation de la structure en dev/test/prod

Technologies utilisées :

- SGBD : avec phpMyAdmin + MySQL.



# Couche d'accès aux données & séparation des responsabilités

## 1 Architecture de la couche DAO

- Utilisation du pattern Repository
- Accès à la base centralisé et structuré
- Code regroupé dans des classes comme UserRepository, CoursRepository...


## 2 Séparation des responsabilités



- Les contrôleurs appellent les services
- Les services utilisent les repositories
- Chaque couche a un rôle clair :
- Contrôleur : gère la requête
- Service : contient la logique métier
- Repository : communique avec la base.

## 3 Exemples de composants clés



```
public function findByEmail($email): User {  
    $stmt = $this->db->prepare("SELECT * FROM users WHERE email = :email");  
    $stmt->execute();  
    return $stmt->fetchObject(User::class);  
}
```

 Knowledge

 Panier  Se déconnecter

### Panier

Cours/Leçon	Quantité	Prix Unitaire	Total	Actions
Les modes de cuisson	1	23.00 €	23.00 €	<a href="#">Supprimer</a>
Découverte de l'instrument 1	1	26.00 €	26.00 €	<a href="#">Supprimer</a>
Cursus d'initiation au piano 1	1	50.00 €	50.00 €	<a href="#">Supprimer</a>

**Total : 99.00 €**

[Vider le panier](#) [Passer à la caisse](#)

**Knowledge**  
Plateforme d'apprentissage en ligne

**Liens rapides**  
[Accueil](#)  
[Connexion](#)  
[Inscription](#)

**Contact**  
 [contact@knowledge.com](mailto:contact@knowledge.com)

## Paie ment

Total à payer : **99.00 €**

Numéro de carte MM / AA

 Payer

## Paie ment

à payer : **99.00 €**

4000 0000 0000 0002 04 / 27

 Payer

**e carte a été refusée. Votre demande a été en attente de validation, mais a utilisé une carte de test**

# Composants e-commerce

Gestion du panier d'achat :

- Fonctionnalités : ajout/suppression d'articles, mise à jour des quantités, affichage du total.
- Persistance via sessions PHP (pas de base dédiée)

Intégration du paiement via Stripe :

- Paiement en mode sandbox (test)
- Création de session de paiement via API
- Traitement des réponses Stripe (webhook ou callback simplifié)
- Gestion des erreurs de paiement
- Sécurité des transactions.

Processus d'achat et gestion des commandes :

- Validation du panier
- Paiement Stripe
- Enregistrement en base (commande + items)
- Suivi via la table orders + order\_items

# Sécurité

## Authentification :

- JWT : utilisé pour l'API, stocké en cookie HttpOnly
- Sessions PHP : pour les pages web classiques
- Activation de compte par email avec token.

## Protection CSRF (Cross-Site Request Forgery) :

- Mise en place de tokens CSRF.

## Gestion des mots de passe :

- Hash : password\_hash() (algorithme BCrypt)
- Vérification : password\_verify()
- Validation des mots de passe (longueur, caractères spéciaux)

## Autres mesures de sécurité :

- XSS : échappement des données dans les vues
- SQL injection : requêtes préparées PDO / DAO
- Contrôle d'accès :
- Vérification du rôle (admin ou client)
- Restriction des accès côté back & front

The image shows four smartphone screens displaying a user registration and login form. The first screen shows a registration form with fields for name, email, password, role, is\_activated, activation\_token, and created\_at. The second screen shows the 'Connexion' (Login) screen with fields for email and password. The third screen shows a user profile or settings page with fields for user, email, password, client, and a long alphanumeric string. The fourth screen shows a login screen with fields for email and password.

name varchar(255)

email varchar(255)

password varchar(255)

role enum

is\_activated tinyint(1)

activation\_token varchar(255)

created\_at timestamp

Connexion

email user@gmail.com

password .....

user

testuser@example.com

\$2y\$10\$WNkWEt8oG8cySUHg12RT90CHzoHAr1QnWiK6IBukVFHb.xK

client

0

56833df5309486dd51b21df668eb3f622346a69efa9be91bae0eeeee38

2025-06-10 03:44:41

Connexion

Adresse email

Mot de passe



# Tests unitaires & couverture fonctionnelle

## 1 Présentation des tests unitaires réalisés

- Valider automatiquement le bon fonctionnement du code
- Éviter les régressions lors des modifications
- Gagner du temps lors des vérifications

## 2 Outils utilisés

- PHPUnit : framework de tests unitaire en PHP
- Tests automatisés exécutables via terminal :
- `vendor/bin/phpunit tests/`

## 3 Couverture des fonctionnalités principales

- Authentification : inscription, activation par token, connexion
- Panier : ajout d'articles, suppression, total calculé, panier vidé
- Administration : ajout / modification / suppression d'un cours ou d'un utilisateur
- Migrations : test du bon déroulement des migrations de la base

## 4 Résultats des tests

- Nombre de tests passés : 100% des tests passent
- Métriques de couverture (si applicable) :
- Classes critiques testées (~60–70%)
- Tests exécutés automatiquement avant déploiement

# Recherches et développement

- Veille technologique : Pour choisir les bonnes technologies (PHP, MySQL, JWT, Stripe), j'ai consulté des documentations officielles, tutoriels, et forums spécialisés.
- Visionnage de tutoriels vidéo
- Apprentissage continu : UML pour représenter les entités sous forme de classes (User, Order, etc.) avec leurs attributs et relations.
- Merise pour formaliser la base de données relationnelle, avec le MPD (tables, clés, cardinalités). Cela permet d'avoir à la fois une vision objet et une vision relationnelle. »
- Résolution de problèmes : Face aux contraintes (ex: limitation SMTP sur l'hébergement), j'ai recherché des solutions alternatives (simulation du lien d'activation).
- Sources utilisées : Documentation officielle des APIs, tutoriels vidéo, articles de développeurs, échanges sur Stack Overflow.

L'envoi d'e-mails est implémenté avec PHPMailer et SMTP.

Cependant, en raison des restrictions de l'hébergement gratuit InfinityFree (blocage des connexions SMTP sortantes), les e-mails ne peuvent pas être envoyés en production.

✚ Un lien d'activation simulé est affiché après inscription, comme :

<https://knowledgelearn.rf.gd/activate?token=XYZ>

# **Conclusion et questions**

## **Résumé des réalisations**

Les objectifs atteints

MERCI POUR VOTRE ATTENTION