

Documentation du projet Knowledge

Projet : Knowledge - Plateforme de cours en ligne

Réalisé par : Yassine RAMDANE

Formation : Développeur Web et Web Mobile

École : Centre Européen de Formation

Date : Juin 2025

Documentation du projet Knowledge

Sommaire

1. Présentation du projet
2. Résumé du projet
3. Documentation backend :
4. server.js
5. coursesController.js
6. educatorController.js
7. userController.js
8. webhooks.js
9. authMiddleware.js
10. Course.js
11. CourseProgress.js
12. Purchase.js
13. User.js

Documentation du projet Knowledge

Résumé du projet

Knowledge est une application web complète dédiée à la gestion et à la diffusion de cours en ligne.

Conçue pour offrir une expérience fluide aux étudiants comme aux enseignants, elle permet d'accéder à des formations vidéo organisées par chapitres, d'en suivre la progression, et d'acheter des cours via un système de paiement sécurisé.

L'utilisateur peut visionner des leçons gratuites, suivre sa progression individuelle et noter les cours.

L'enseignant, quant à lui, peut créer et publier un cours complet, suivre les revenus générés, consulter les statistiques d'inscription et la progression de ses apprenants.

Le projet repose sur une architecture moderne : une interface frontend en React.js et Tailwind CSS, un backend en Node.js avec Express, une base MongoDB pour le stockage des données, Clerk pour l'authentification, et Stripe pour les paiements. L'hébergement est assuré par Vercel.

Ce projet, réalisé dans le cadre de la formation de Développeur Web et Web Mobile au Centre Européen de Formation, démontre une maîtrise des technologies web actuelles ainsi qu'une capacité à développer une plateforme pédagogique fonctionnelle et évolutive.

server.js

Initialise Express, configure les middlewares principaux (CORS, Clerk), et définit les routes principales :

- /api/course : accès aux cours
- /api/educator : accès enseignant (protégé)
- /api/user : accès utilisateur
- /clerk, /stripe : webhooks

Démarre le serveur sur le port spécifié.

coursesController.js

@getAllCourse()

- Récupère tous les cours publiés (isPublished: true)
- Retourne la liste sans contenu vidéo

@getCourseld(id)

- Récupère les détails complets d'un cours par son ID
- Supprime les URLs de vidéos non prévisualisables (preview)

educatorController.js

@updateRoleEducator()

- Définit le rôle 'educator' dans Clerk

@addCourse(courseData, image)

- Crée un nouveau cours avec image (upload Cloudinary)
- Enregistre contenu structuré (chapitres, vidéos)

@getEducatorCourses()

- Retourne les cours publiés par l'éducateur connecté

Documentation du projet Knowledge

@educatorDashboardData()

- Retourne : totalEarnings, totalCourses, enrolledStudentsData

@getEnrolledStudentsData()

- Liste détaillée des étudiants inscrits et date d'inscription

UserController.js

@getUserData()

- Retourne les données du compte actuel

@userEnrolledCourses()

- Retourne les cours où l'utilisateur est inscrit

@purchaseCourse(courseId)

- Crée une session Stripe Checkout
- Enregistre un achat en base avant le paiement

@updateUserCourseProgress(courseId, lectureId)

- Marque une leçon comme complétée

@getUserCourseProgress(courseId)

- Retourne la progression (lectures terminées)

@addUserRating(courseId, rating)

- Ajoute une note utilisateur à un cours

webhooks.js

@clerkWebhooks()

- Gère les webhooks Clerk : création, mise à jour, suppression utilisateur

@stripeWebhooks()

- Gère les paiements Stripe : success ou échec

authMiddleware.js

@protectEducatorClerk()

- Middleware pour restreindre l'accès aux routes éducateurs
- Vérifie que Clerk.publicMetadata.role === 'educator'

Course.js

Modèle Mongoose d'un cours :

- Titre, description, image, prix, contenu structuré (chapitres/leçons), avis
- Références à l'enseignant et aux étudiants inscrits

CourseProgress.js

Modèle de progression d'un utilisateur dans un cours :

- lectureCompleted[]

Documentation du projet Knowledge

- completed: true/false

Purchase.js

Modèle représentant un achat Stripe :

- courseId, userId, amount
- status : 'pending' | 'completed' | 'failed'

User.js

Utilisateur issu de Clerk, enrichi en base :

- _id (Clerk ID), email, name, imageUrl
- enrolledCourses[] : références MongoDB vers Course