

Application web de formation en ligne pour l'apprentissage autonome et accessible

Projet réalisé dans le cadre de la présentation au

Titre Professionnel Développeur Web et Web Mobile

Présenté par

David SAADOUN

Centre Européen de Formation

SOMMAIRE

SOMMAIRE	2
INTRODUCTION	4
LISTE DES COMPÉTENCES COUVERTES PAR LE PROJET	5
I. Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité	5
A. Maquetter une application	5
B. Réaliser une interface utilisateur web statique et adaptable	5
C. Développer une interface utilisateur web dynamique	6
II. Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité	6
A. Créer une base de données	6
B. Développer les composants d'accès aux données	6
C. Développer la partie back-end d'une application web ou web mobile	7
RÉSUMÉ DU PROJET	8
CAHIER DES CHARGES	9
I. Besoins et objectifs de l'application	9
A. Besoins	9
B. Objectifs	10
C. Cibles	11
II. Users Stories	12
III. Arborescence	12
IV. MVP	13
V. Fonctionnalités détaillées des pages	14
VI. Evolutions potentielles	15
A. Étudiant	16
B. Administrateur	16
VII. Wireframes	16
VIII. Charte graphique et logo	17-19
IX. Exemples de maquettes	20-21
SPÉCIFICATIONS TECHNIQUES	22
I. Technologies	22
II. Navigateurs compatibles	22
III. Possibilités de déploiement	23
IV. Création de la base de données	24
A. MCD	24
B. MLD	25
C. Dictionnaire des données	25-26
V. Routes front et back	26

A. Back-end.....	26-27
B. Front-end.....	27
VULNÉRABILITÉS DE SÉCURITÉ ET VEILLE.....	28
I. Veille technologique.....	28
A. <i>Symfony et Doctrine ORM</i>	28
B. <i>Symfony Mailer</i>	28
C. <i>Sécurisation des données utilisateurs</i>	29
II. Veille de sécurité.....	29
A. <i>Bonnes pratiques OWASP</i>	29
B. <i>Gestion des CORS et API sécurisée</i>	30
C. <i>Sécurisation des formulaires et protection XSS</i>	30-32
D. <i>PHPUnit</i>	33
III. Processus de recherche et traduction.....	34
A. <i>Processus</i>	34
B. <i>Ressource en anglais</i>	34-35
C. <i>Traduction effectuée</i>	36
CONCLUSION	37
ANNEXE.....	38

INTRODUCTION

Depuis mon plus jeune âge, je suis passionné par les technologies, et plus particulièrement par la programmation. Même si mon parcours scolaire et professionnel m'a éloigné de l'informatique, j'ai toujours continué à me former en autodidacte sur mon temps libre. Cette démarche m'a permis de développer mes compétences en développement et de rester à l'écoute des évolutions du secteur.

Souhaitant donner une nouvelle orientation à ma carrière, j'ai choisi d'intégrer la formation complète de développeur web et web mobile proposée par l'école Centre Européen de Formation. Ce parcours m'a permis d'acquérir des bases solides en HTML, CSS, JavaScript, avant d'approfondir des frameworks front-end comme Vue.js, React, Angular et WordPress. J'ai également renforcé mes connaissances en back-end avec PHP et Symfony, en travaillant notamment sur la conception et la gestion d'APIs, ainsi que sur des projets en Python.

En parallèle, j'ai appris à utiliser des outils essentiels comme Git et GitHub, afin de gérer le versionnement de mes projets et d'adopter les bonnes pratiques de travail collaboratif.

Pour valider cette formation, j'ai réalisé seul mon projet de fin d'études, intitulé Knowledge. Ce projet m'a permis de mobiliser l'ensemble des compétences acquises, de la conception à la réalisation technique, tout en gérant moi-même les aspects liés à l'organisation, à la résolution de problèmes, et au développement complet d'une application.

Cette expérience a été particulièrement enrichissante, car elle m'a permis de prouver ma capacité à mener un projet de A à Z de manière autonome, tout en respectant des contraintes de qualité et de délais. Cela a également renforcé ma rigueur, ma capacité d'adaptation et ma motivation à poursuivre dans le développement web et mobile.

Aujourd'hui, je souhaite mettre à profit ces compétences dans un cadre professionnel, continuer à progresser et m'impliquer dans des projets concrets.

LISTE DES COMPÉTENCES COUVERTES PAR LE PROJET

I. Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

A. Maquetter une application

Avant de commencer le développement technique de la plateforme e-learning "Knowledge Learning", j'ai d'abord consacré une phase essentielle à la conception. Cette étape préparatoire m'a permis de poser des bases solides pour la réalisation du projet, en cohérence avec les besoins exprimés par l'entreprise Knowledge.

Dans un premier temps, j'ai rédigé des user stories afin de définir clairement les actions possibles pour les différents types d'utilisateurs : création de compte, achat de leçons ou cursus, suivi des certifications, gestion des contenus pour les administrateurs, etc.

J'ai ensuite réalisé des wireframes (versions mobiles et desktop) pour visualiser la structure des pages de la plateforme : parcours utilisateur, navigation, et organisation des fonctionnalités clés.

Enfin, j'ai conçu une charte graphique ainsi que des maquettes visuelles pour donner une identité propre à l'application, en respectant la palette de couleurs et les éléments graphiques définis dans le cahier des charges (logo, favicon, etc.).

Cette démarche m'a permis d'assurer une vision claire du projet avant d'entamer le développement de la partie back-end, cœur de ma mission.

B. Réaliser une interface utilisateur web statique et adaptable

Ce projet a pour objectif de développer une plateforme e-learning, "Knowledge Learning", destinée à rendre accessibles en ligne les formations proposées par l'éditeur Knowledge, initialement diffusées sous forme de livres et de kits.

L'interface utilisateur de la plateforme a donc été pensée pour être simple, intuitive et accessible, afin de s'adresser à un public varié. Les utilisateurs peuvent en effet être des débutants souhaitant se former en autonomie, quel que soit leur niveau en informatique (particuliers, amateurs ou professionnels).

De plus, la plateforme doit permettre un accès aux cours depuis n'importe quel support, que ce soit sur ordinateur, tablette ou smartphone. Il était donc impératif de concevoir une interface responsive, s'adaptant automatiquement aux différents formats d'écran, pour garantir une expérience utilisateur optimale en toutes circonstances.

Cette attention portée à l'ergonomie et à l'adaptabilité de l'interface vise à faciliter la navigation, la lecture des cours, ainsi que la gestion des achats et certifications, tout en assurant un confort d'utilisation maximal.

C. Développer une interface utilisateur web dynamique

Dans le cadre du projet "Knowledge Learning", il était nécessaire de proposer une interface utilisateur dynamique et interactive pour répondre aux fonctionnalités prévues (gestion des comptes, achats, suivi des certifications).

J'ai choisi d'utiliser Symfony pour développer cette interface, en m'appuyant sur son moteur de templates Twig et les mécanismes de routing et de formulaires pour créer des pages interactives.

Ainsi, l'utilisateur peut naviguer de façon fluide entre les différentes sections du site, réaliser des actions (achat, validation de leçons) et interagir avec les données sans complexité technique, tout en assurant la robustesse du back-end.

II. Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

A. Créer une base de données

Dès le début du projet "Knowledge Learning", j'ai travaillé à structurer les données nécessaires au bon fonctionnement de la plateforme (utilisateurs, cursus, leçons, achats, certifications).

J'ai commencé par définir les entités principales et leurs relations, ce qui m'a permis de réaliser un MCD (Modèle Conceptuel de Données), suivi d'un MLD (Modèle Logique de Données), et d'un dictionnaire des données.

Pour gérer la base de données, j'ai choisi PostgreSQL, en raison de sa robustesse et de sa compatibilité avec Doctrine ORM, utilisé dans Symfony. Ce choix facilite la gestion des entités et des relations en respectant les bonnes pratiques de développement.

B. Développer les composants d'accès aux données

Pour la gestion de l'accès aux données dans le projet "Knowledge Learning", j'ai utilisé Doctrine ORM, intégré à Symfony. Cet outil permet de simplifier les interactions entre les entités de l'application et la base de données PostgreSQL.

Les entités (comme Utilisateur, Cursus, Leçon, Achat, Certification) ont été modélisées en PHP via Doctrine, ce qui facilite les opérations CRUD tout en respectant l'architecture MVC de Symfony.

Doctrine gère également la migration des schémas de base de données, permettant ainsi de suivre l'évolution de la structure des données tout au long du projet.

C. Développer la partie back-end d'une application web ou web mobile

Pour développer le back-end de la plateforme "Knowledge Learning", j'ai utilisé le framework Symfony, en suivant une architecture MVC.

J'ai structuré le projet avec des routes, associées à des contrôleurs, afin de gérer les différentes fonctionnalités : gestion des utilisateurs, achats de cursus ou de leçons, validation des certifications, etc.

La sécurité (authentification, gestion des rôles) a été assurée en utilisant les composants fournis par Symfony. J'ai également mis en place des services pour organiser le traitement des données et respecter le principe de séparation des responsabilités.

RÉSUMÉ DU PROJET

La demande en formations accessibles à distance ne cesse de croître, aussi bien pour les étudiants que pour les professionnels souhaitant acquérir de nouvelles compétences. Pour répondre à ces besoins, l'entreprise Knowledge, spécialisée dans l'édition de supports de formation, a décidé de créer une plateforme e-learning.

Cette plateforme, nommée "Knowledge Learning", a pour objectif de dématérialiser les formations initialement proposées sous forme de livres et de kits en librairie. Ces supports physiques sont désormais organisés sous forme de cursus, eux-mêmes composés de leçons interactives accessibles en ligne. Elle permet ainsi aux utilisateurs d'apprendre de manière autonome, à leur rythme, tout en garantissant un suivi personnalisé.

Le site propose différents thèmes de formation (musique, informatique, jardinage, cuisine...), chacun décliné en cursus et leçons. Les utilisateurs peuvent acheter ces formations et obtenir des certifications après validation des leçons.

Pour gérer efficacement les comptes utilisateurs, les achats et l'accès aux contenus, une interface d'administration est mise à disposition des administrateurs. La plateforme assure également la sécurisation des données et le suivi de la progression des utilisateurs.

En somme, "Knowledge Learning" vise à offrir un outil simple, intuitif et complet, accessible depuis tout support (ordinateur, mobile, tablette), favorisant l'apprentissage en ligne dans un environnement structuré et professionnel.

CAHIER DES CHARGES

I. Besoins et objectifs de l'application

A. Besoins

Pour bien comprendre les besoins liés au projet "Knowledge Learning", il a été nécessaire d'effectuer des recherches sur l'entreprise Knowledge et son évolution vers le e-learning.

L'entreprise est historiquement spécialisée dans l'édition de livres de formation et souhaite aujourd'hui proposer ces contenus sous forme numérique afin de répondre aux nouvelles attentes des utilisateurs : apprentissage à distance, accès facilité aux ressources, et suivi personnalisé.

Le projet vise donc à :

- Dématérialiser les cursus de formation existants.
- Permettre l'achat en ligne de cursus ou de leçons individuelles.
- Offrir un suivi de progression et des certifications.
- Mettre à disposition un espace d'administration pour gérer les utilisateurs et les contenus.

Les besoins techniques identifiés concernent principalement la gestion des comptes utilisateurs, la sécurisation des accès, la gestion des achats, ainsi que la structuration des contenus pédagogiques (thèmes, cursus, leçons).

Gestion des utilisateurs :

- Inscription d'un utilisateur
- Activation de compte par email
- Connexion et gestion des sessions
- Attribution de rôles (administrateur, client)

Gestion des formations :

- Création et gestion des thèmes de formation
- Création et gestion des cursus
- Création et gestion des leçons (contenus, vidéos)

Gestion des achats :

- Achat d'un cursus complet
- Achat d'une leçon individuelle
- Attribution des droits d'accès aux contenus achetés
- Simulation d'achat via un mode sandbox

Suivi des progressions et certifications :

- Validation des leçons par l'utilisateur
- Validation automatique d'un cursus complet
- Génération d'une certification
- Accès à un tableau de bord des certifications obtenues

Gestion back-office :

- Gestion des comptes utilisateurs
- Gestion des contenus (thèmes, cursus, leçons)
- Gestion des achats et des certifications
- Suivi des activités des utilisateurs

Sécurité et conformité :

- Sécurisation des données utilisateurs
- Cryptage des mots de passe
- Gestion des permissions selon les rôles
- Mise en place de bonnes pratiques de sécurité (CSRF, validations...)

B. Objectifs

Pour optimiser la diffusion des formations et proposer une solution accessible au plus grand nombre, les points essentiels à prendre en compte sont :

- L'utilisateur doit pouvoir consulter les formations disponibles proposées par Knowledge.
- Les cursus et leçons doivent être visibles via une plateforme en ligne intuitive.
- L'utilisateur doit pouvoir apprendre en autonomie, à son rythme, sans contrainte géographique.
- L'obtention des certifications doit être automatisée en fonction de la progression de l'utilisateur.
- Les administrateurs doivent pouvoir gérer facilement les comptes, les contenus et les ventes.
- La traçabilité des achats et des certifications doit être assurée pour chaque utilisateur.
- La plateforme doit garantir la sécurité des données personnelles des utilisateurs.

Après analyse du marché, il existe de nombreuses plateformes e-learning généralistes, mais peu adaptées aux éditeurs de contenus physiques souhaitant dématérialiser leur offre existante.

La solution "Knowledge Learning" permet donc de répondre à ce besoin spécifique : transformer les cursus et leçons de formation en parcours e-learning structurés. Contrairement aux plateformes généralistes, elle est pensée pour offrir une progression étape par étape, fidèle à la logique des supports physiques.

En outre, la gestion des achats individuels de leçons ou de cursus ainsi que la délivrance de certifications représentent des fonctionnalités clés, alignées avec les besoins de l'éditeur.

Ainsi, l'objectif principal du projet est de proposer un outil simple et efficace, qui permette à Knowledge d'élargir son offre de formation grâce au numérique, tout en facilitant l'accès et le suivi pour les utilisateurs.

C. Cibles

Afin de bien cerner les utilisateurs visés par la plateforme "Knowledge Learning", plusieurs profils types (personas) ont été définis. Ils représentent les différentes catégories d'utilisateurs susceptibles d'utiliser le service.

1. Étudiants en auto-formation

Ce profil regroupe des jeunes adultes, souvent en reconversion ou en complément de leurs études, cherchant à acquérir de nouvelles compétences (informatique, musique, cuisine...).

Équipement : Ordinateur portable, tablette, smartphone.

Attentes : Une interface simple, intuitive et rapide d'accès pour suivre des leçons de manière autonome, avec un suivi clair de la progression.

Contraintes : Faible temps disponible, besoin d'une navigation efficace et sans perte de temps.

2. Administrateurs de la plateforme

Ce sont les utilisateurs responsables de la gestion des contenus (cursus, leçons), des comptes utilisateurs et des achats.

Équipement : Ordinateur fixe ou portable.

Attentes : Une interface d'administration efficace, avec un accès rapide aux fonctionnalités de gestion, statistiques et suivi des utilisateurs.

Contraintes : Besoin d'ergonomie pour gérer de grandes quantités de données facilement.

II. Users Stories

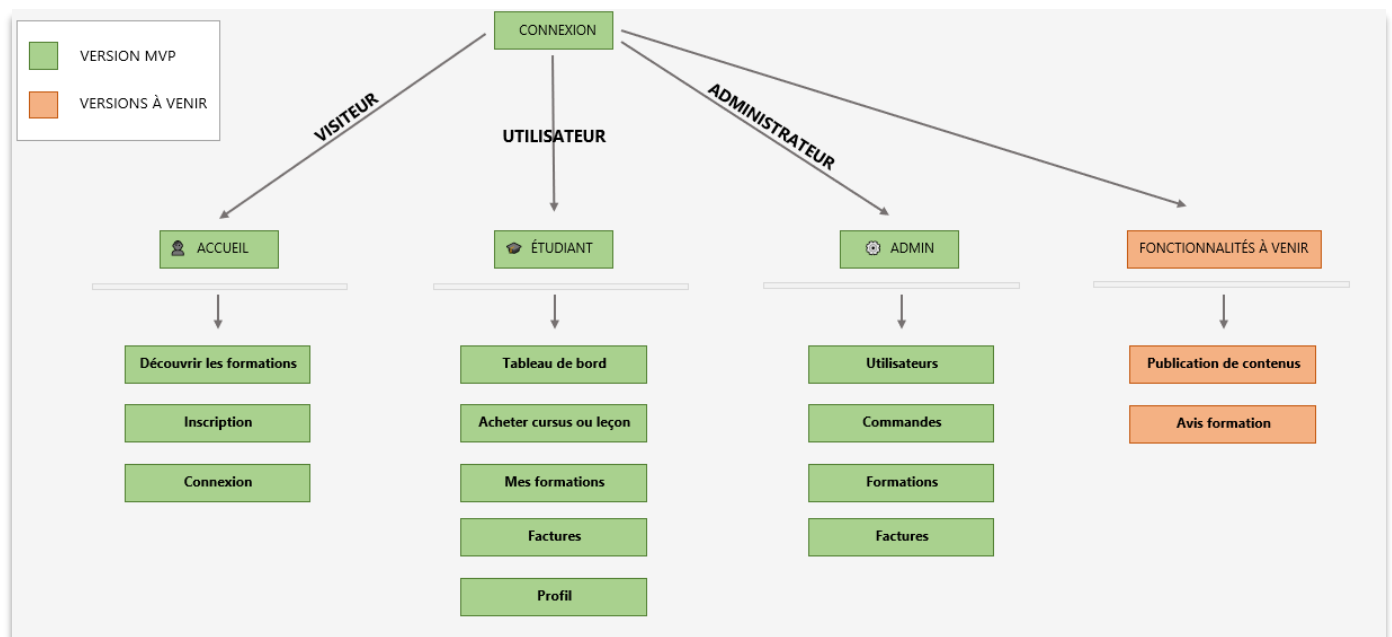
Légende :

MVP

Version à venir

En Tant que	Je souhaite	Afin de
visiteur	créer un compte	accéder à l'espace élève
visiteur	me connecter	consulter mes formations
visiteur	découvrir le contenu de la plateforme	comprendre l'intérêt de m'inscrire
Étudiant	accéder à mes cours	suivre ma progression
Étudiant	acheter un cursus ou une leçon	enrichir mes connaissances
Étudiant	consulter mes factures	garder une trace de mes achats
Étudiant	modifier mes informations personnelles	mettre à jour mon profil
Étudiant	changer mon mot de passe	sécuriser mon compte
Étudiant	obtenir une certification	valider mes acquis
Administrateur	voir toutes les commandes et factures des utilisateurs	assurer le suivi administratif complet
Administrateur	gérer les comptes utilisateurs	activer, désactiver, ou corriger des accès
Formateur	publier des contenus	proposer des ressources pédagogiques
Formateur	suivre l'avancement des élèves	adapter son accompagnement pédagogique
Étudiant	laisser un avis sur une formation	aider les autres à faire leur choix

III. Arborescence



IV. MVP

Le Produit Minimum Viable (MVP) de la plateforme "Knowledge Learning" se concentre sur les fonctionnalités essentielles pour garantir une première version fonctionnelle, accessible aux utilisateurs et conforme aux besoins de l'entreprise Knowledge.

Objectifs du MVP :

- Dématérialiser les cursus et leçons sous forme de pages accessibles en ligne.
- Permettre l'achat de formations (cursus ou leçons individuelles) via un système simple.
- Gérer les comptes utilisateurs avec un système d'authentification sécurisé.
- Mettre en place un suivi de progression et de certifications.

Fonctionnalités incluses dans le MVP :

1. Authentification :

- Inscription d'un utilisateur avec validation par email.
- Connexion sécurisée via email et mot de passe.

2. Gestion des utilisateurs :

- Attribution d'un rôle (client ou administrateur).
- Accès à un profil utilisateur modifiable (informations personnelles, mot de passe).

3. Accès aux formations :

- Consultation des thèmes, cursus et leçons.
- Affichage d'une fiche détaillée pour chaque leçon.

4. Achats et accès :

- Achat d'un cursus complet ou d'une leçon à l'unité.
- Attribution des droits d'accès aux leçons achetées.
- Simulations d'achat en sandbox.

5. Suivi des certifications :

- Validation des leçons après consultation.
- Certification automatique à la fin d'un cursus validé.

6. Interface administrateur :

- Gestion des utilisateurs (création, modification).
- Gestion des contenus (création de thèmes, cursus, leçons).
- Suivi des achats et certifications.

Sécurité :

- Accès sécurisé aux pages (routes protégées).
- Cryptage des mots de passe.
- Respect des bonnes pratiques de sécurité.

V. Fonctionnalités détaillées des pages

La plateforme "Knowledge Learning" est structurée autour d'un parcours utilisateur simple et efficace, centré sur l'apprentissage en ligne et la gestion des formations.

1. Page d'accueil :

Présentation des thèmes de formation disponibles. L'utilisateur peut accéder aux cursus ou se connecter à son espace personnel.

2. Page de connexion :

Authentification des utilisateurs via email et mot de passe. Redirection vers l'espace client ou le tableau de bord administrateur selon le rôle.

- Nom
- Prénom
- Adresse email
- Mot de passe (crypté)
- Rôle (Client / Administrateur)
- Date d'inscription
- Dernière connexion

3. Page cursus :

Affichage des cursus disponibles. Chaque cursus contient plusieurs leçons accessibles selon les achats effectués. Les utilisateurs peuvent consulter les fiches leçon et valider leur progression.

- Titre
- Description
- Prix
- Liste des leçons associées
- Statut (disponible / indisponible)

4. Page leçons :

- Titre
- Description
- Vidéo (URL)
- Prix (si achat individuel)
- Statut de validation

5. Achat :

- Utilisateur concerné
- Type d'achat (Cursus / Leçon)
- Référence produit
- Date d'achat
- Statut de paiement (sandbox)

6. Page de certification :

Suivi des certifications obtenues par l'utilisateur en fonction des leçons et cursus validés.
Utilisateur concerné

- Utilisateur concerné
- Cursus associé
- Date d'obtention
- Statut

7. Espace profil utilisateur

Consultation et modification des informations personnelles (email, mot de passe, etc.).

8. Interface d'administration

Gestion des utilisateurs, des contenus (thèmes, cursus, leçons) et des achats. Accès réservé aux administrateurs.

Sécurité

Toutes les pages sont sécurisées et nécessitent une authentification pour accéder aux contenus achetés ou aux fonctions d'administration.

VI. Evolutions potentielles

Dans le cadre de l'amélioration continue de la plateforme "Knowledge Learning", plusieurs fonctionnalités complémentaires sont prévues pour enrichir l'expérience utilisateur et optimiser la gestion des contenus.

Ces évolutions concerneront à la fois la partie étudiant et la partie administrateur.

A. Étudiant

Publication d'avis : Les utilisateurs pourront laisser un avis sur les leçons ou cursus qu'ils ont suivis. Ces avis permettront de partager leur retour d'expérience et d'orienter les futurs apprenants.

Système de notation : Possibilité de noter les formations suivies sous forme d'étoiles ou de scores, visibles sur les fiches leçons.

Favoris : Les utilisateurs pourront ajouter des leçons ou cursus à une liste de favoris pour les retrouver facilement.

B. Administrateur

Publication de contenus : Ajout d'un module simplifié pour publier de nouveaux contenus (leçons, actualités, ressources) directement depuis l'interface d'administration.

Gestion des avis utilisateurs : L'administrateur pourra consulter, modérer ou supprimer les avis publiés par les utilisateurs sur les formations.

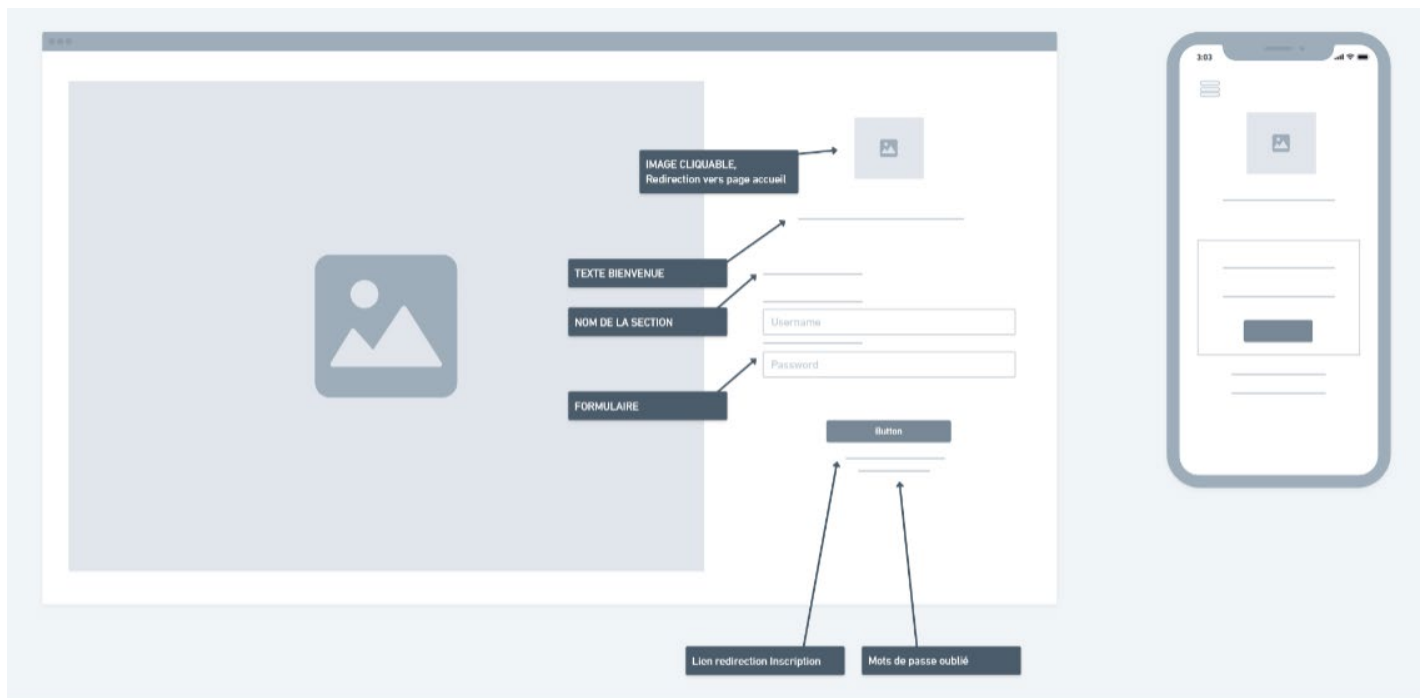
Statistiques avancées : Mise en place d'un tableau de bord affichant des statistiques détaillées sur les ventes, les certifications obtenues et la fréquentation des leçons.

VII. Wireframes

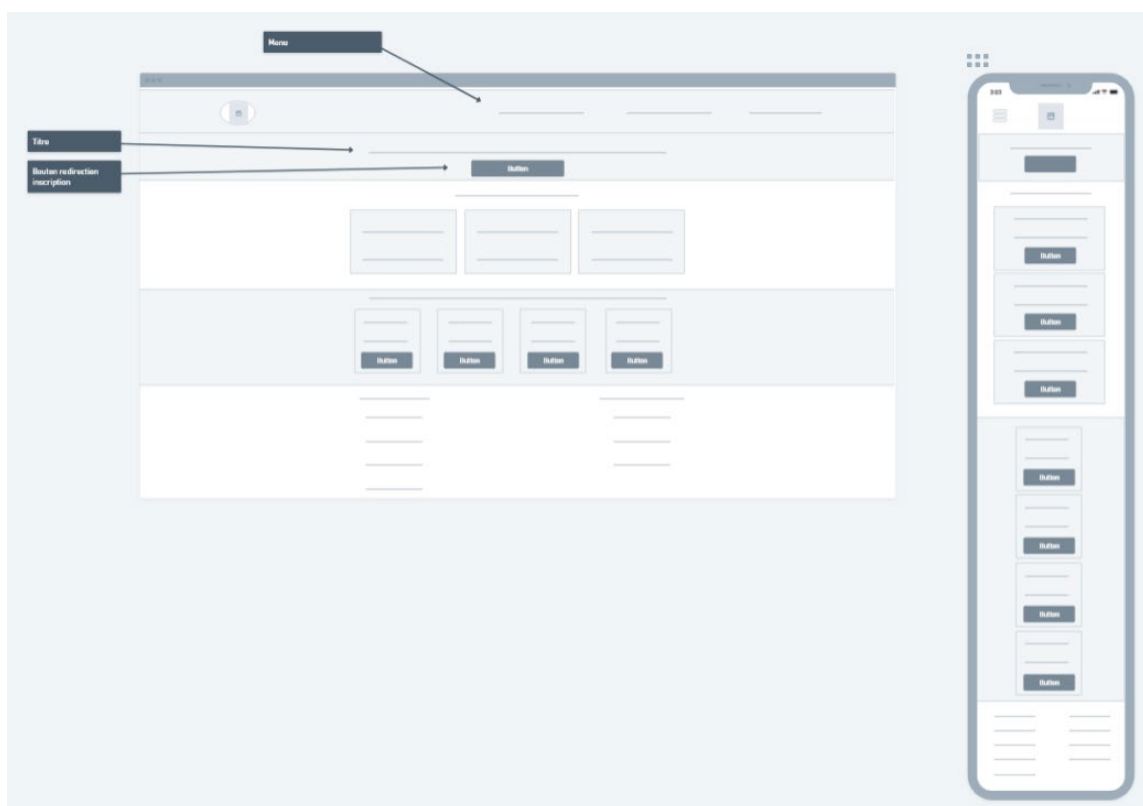
Pour plus de lisibilité, l'intégralité des wireframes est disponible à l'adresse suivante :

[Whimsical](#)

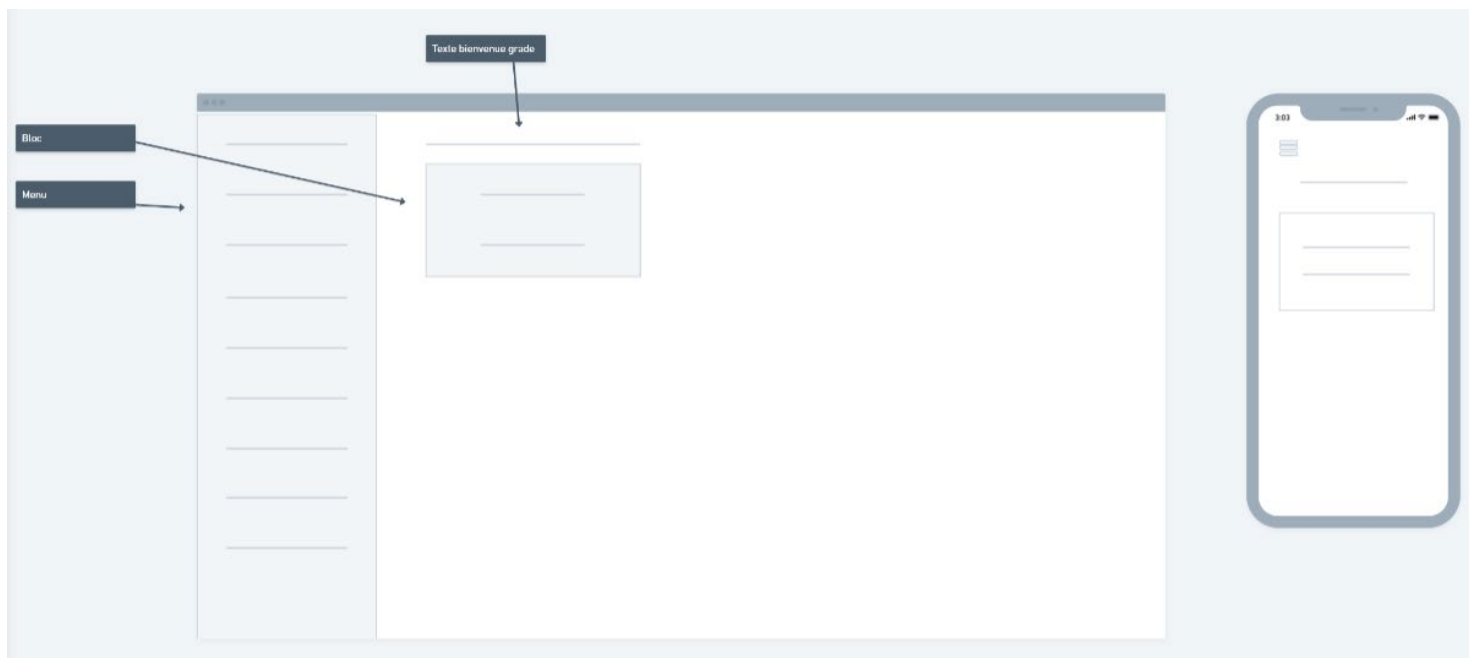
Écran de connexion



Accueil



Administration



VIII. Charte graphique et logo

L'application Knowledge Learning étant destinée aux utilisateurs de la plateforme de formation en ligne de l'éditeur Knowledge, l'objectif était de respecter les codes visuels de l'entreprise afin d'assurer une cohérence graphique et renforcer l'identité de marque.

En nous appuyant sur l'identité visuelle existante de Knowledge (présente sur leurs supports physiques et numériques), nous avons défini une charte graphique sobre et professionnelle, adaptée à l'univers de la formation.

L'objectif était de proposer une interface visuelle cohérente, moderne et fonctionnelle, fidèle à l'image de l'éditeur tout en offrant une expérience agréable aux utilisateurs.

La charte repose sur :

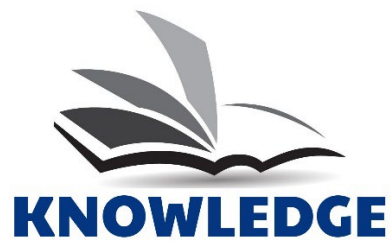
A. Typographie :

Une typographie claire et accessible, favorisant une lecture facile sur tous les supports (desktop et mobile).

- Comic sans MS

B. Logo

Un logo simple et épuré, représentant un livre ouvert, symbole de connaissance et d'apprentissage.



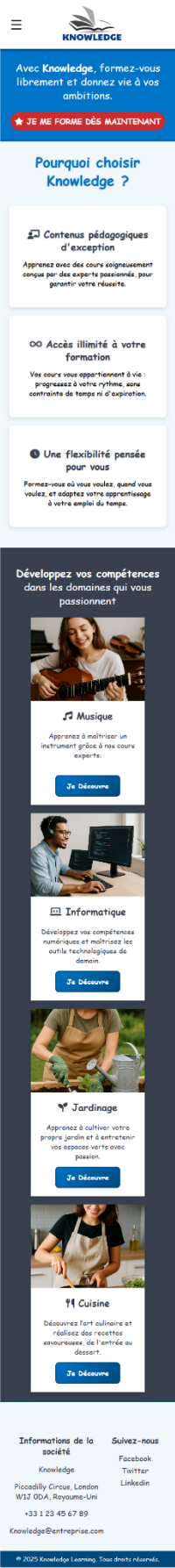
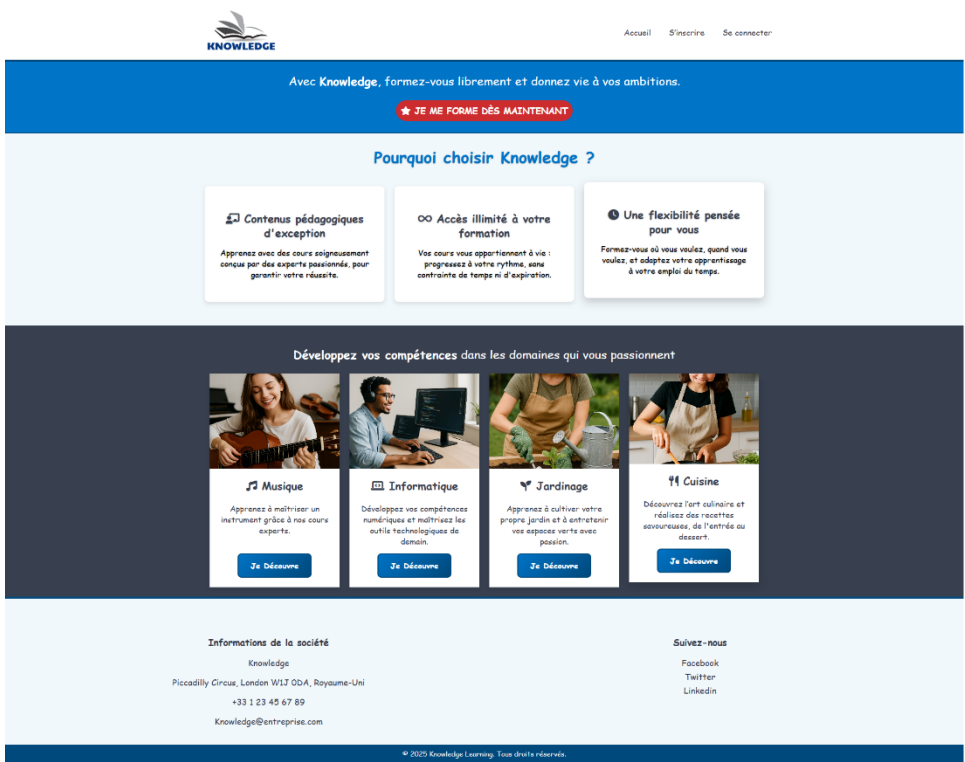
C. Palette de couleurs

Une palette de couleurs sobres, avec des teintes de bleu pour inspirer la confiance et la sérénité, accompagnée de gris pour la lisibilité.


#f1f8fc	#0074c7	#00497c	#384050	#cd2c2e	#82b864

Les deux dernières couleurs uniquement pour de l'accentuation avec parcimonie.

IX. Exemples de maquettes








Bienvenue sur Knowledge Learning, l'espace élève en ligne de formation.

Connexion

[Pas encore inscrit ? Créer un compte](#)

[Mot de passe oublié ?](#)



Bienvenue sur Knowledge Learning, l'espace élève en ligne de formation.

Connexion

[Pas encore inscrit ? Créer un compte](#)

[Mot de passe oublié ?](#)

Mon espace élève

Accueil

Boutique

Formations

Résultats

Certifications

Factures

Profil

Retour sur le site

Déconnexion

Bienvenue student !

Mes Formations

4

Mon espace élève

Accueil

Boutique

Formations

Résultats

Certifications

Factures

Profil

Retour sur le site

Déconnexion

nt !

SPÉCIFICATIONS TECHNIQUES

Pour réaliser les spécifications techniques de la plateforme Knowledge Learning, j'ai tout d'abord défini les technologies adaptées aux besoins du projet (e-learning, gestion des utilisateurs, achats en ligne, certifications).

Ensuite, j'ai identifié les navigateurs à privilégier pour assurer une compatibilité optimale. Enfin, j'ai établi la modélisation de la base de données, l'architecture back-end, ainsi que la gestion des routes front-end.

I. Technologies utilisées

Outils de développement :

- Composer : Gestionnaire de dépendances PHP.
- Symfony CLI : Outils en ligne de commande pour gérer le projet Symfony.

Back-end :

- Symfony (PHP 8.x) : Framework principal, architecture MVC.
- Doctrine ORM : Gestion des entités et accès aux données.
- Security Bundle : Authentification, gestion des rôles et sécurisation des routes.
- Symfony Mailer : Envoi d'emails (validation compte, notifications).

Front-end :

- Twig : Moteur de templates intégré à Symfony.
- Bootstrap : Framework CSS pour un design responsive.

Tests et qualité :

- PHPUnit : Réalisation de tests unitaires pour les fonctionnalités principales.
- Git / GitHub : Gestion de versions et hébergement du code source.

II. Navigateurs compatibles

Sur la base des parts de marché actuelles, l'application Knowledge Learning est optimisée pour fonctionner sur les navigateurs suivants :

Desktop :

- Google Chrome - 68,88 %
- Mozilla Firefox - 2,46 %
- Microsoft Edge - 4,92 %

Mobile :

- Google Chrome - 66,88 %
- Safari - 8,48 %

Source : <https://www.blogdumoderateur.com/navigateurs-web-plus-utilises-france-monde-2025/>

III. Possibilités de déploiement

Pour rendre la plateforme Knowledge Learning accessible aux utilisateurs, plusieurs solutions de déploiement sont envisageables.

1. Hébergement mutualisé (OVH, o2switch...)

- Solution simple et économique pour héberger le projet Symfony.
- Permet d'avoir un nom de domaine et une base MariaDB / MySQL (comme proposé chez o2switch). Convient à une application avec un trafic modéré.

2. Serveur VPS (Scaleway, OVH, Hetzner...)

- Plus flexible qu'un hébergement mutualisé.
- Possibilité d'installer un environnement optimisé pour Symfony (PHP, NGINX, MariaDB).
- Recommandé pour un site évolutif avec gestion fine des performances et de la sécurité.

3. Plateformes spécialisées (Heroku, Render, Railway...)

- Déploiement rapide sans gestion de l'infrastructure.
- Intégration continue facilitée avec GitHub.
- Idéal pour des prototypes ou projets pédagogiques.

4. Domaine et accès

- Le projet est déployé sous le nom de domaine : <https://knowledge-learning.fr/>
- Accès sécurisé via HTTPS.
- Utilisation d'un serveur MariaDB / MySQL distant ou managé.

Estimation des coûts

L'estimation précise des coûts pour la plateforme Knowledge Learning dépend de plusieurs facteurs, notamment le nombre d'utilisateurs actifs, le volume de contenus pédagogiques hébergés (vidéos, documents), et la fréquence d'utilisation.

Les principaux postes de dépenses incluent :

- **Hébergement web** (mutualisé ou VPS selon le trafic attendu).
- **Stockage des contenus multimédias** (vidéos, images) via des solutions comme Amazon S3 ou le stockage associé à l'hébergeur.
- **Base de données MariaDB / MySQL** pour la gestion des utilisateurs, achats et certifications.
- **Sécurité et certificats SSL** pour garantir la confidentialité des données utilisateurs

Pour un lancement initial, un hébergement mutualisé (comme o2switch) offre une solution économique, parfaitement adaptée à un site avec un trafic modéré. Cette solution permet de garantir l'accessibilité de la plateforme tout en maîtrisant les coûts.

Le déploiement progressif et la centralisation des ressources via une infrastructure cloud peuvent être envisagés dans un second temps, selon l'évolution des besoins de Knowledge et le succès de la plateforme.

A. MCD

B. MLD

user (id, created_by_id, updated_by_id, email, username, first_name, last_name, phone, address, roles, is_activated, verification_token, password, created_at, updated_at)

reset_password_request (id, user_id, selector, hashed_token, requested_at, expires_at)

certification (id, user_id, cursus_id, obtained_at, filename)

orders (id, user_id, cursus_id, lesson_id, total, status, created_at, updated_at)

cursus (id, created_by_id, updated_by_id, theme_id, title, price, created_at, updated_at)

lesson (id, created_by_id, updated_by_id, cursus_id, title, content, video_url, price, created_at, updated_at)

lesson_validation (id, user_id, lesson_id, validated_at)

theme (id, created_by_id, updated_by_id, title, text, image, icon, created_at, updated_at)

C. Dictionnaire des données

Nom de la donnée	Désignation	Type	Contrainte
id_user	Identifiant unique de l'utilisateur	integer	primary key, auto_increment
email	Adresse email de l'utilisateur	text	not null, unique
username	Nom d'utilisateur	text	not null, unique
first_name	Prénom de l'utilisateur	text	not null
last_name	Nom de famille de l'utilisateur	text	not null
phone	Numéro de téléphone	text	nullable
address	Adresse de l'utilisateur	text	nullable
roles	Liste des rôles (ROLE_USER, ROLE_ADMIN)	text	not null, default 'user'
is_activated	Statut d'activation	boolean	not null, default false
password	Mot de passe crypté	text	not null
verification_token	Jeton d'activation	text	nullable
created_at	Date de création	timestamp	default current_timestamp
updated_at	Date de mise à jour	timestamp	nullable
id_cursus	Identifiant unique du cursus	integer	primary key, auto_increment
theme_id	Identifiant du thème associé	integer	foreign key
title	Titre du cursus	text	not null
price	Prix du cursus	decimal	not null
created_at	Date de création	timestamp	default current_timestamp
updated_at	Date de mise à jour	timestamp	nullable
id_lesson	Identifiant unique de la leçon	integer	primary key, auto_increment
cursus_id	Identifiant du cursus	integer	foreign key
title	Titre de la leçon	text	not null
content	Contenu texte de la leçon	text	not null
video_url	Lien vers la vidéo	text	nullable

price	Prix de la leçon (si achat unitaire)	decimal	nullable
created_at	Date de création	timestamp	default current_timestamp
updated_at	Date de mise à jour	timestamp	nullable
id_theme	Identifiant unique du thème	integer	primary key, auto_increment
title	Titre du thème	text	not null
text	Description du thème	text	nullable
image	URL de l'image	text	nullable
icon	URL de l'icône	text	nullable
created_at	Date de création	timestamp	default current_timestamp
updated_at	Date de mise à jour	timestamp	nullable
id_order	Identifiant unique de la commande	integer	primary key, auto_increment
user_id	Utilisateur ayant passé la commande	integer	foreign key
cursus_id	Cursus commandé (si achat de cursus)	integer	foreign key, nullable
lesson_id	Leçon commandée (si achat unitaire)	integer	foreign key, nullable
total	Montant total de la commande	decimal	not null
status	Statut de la commande (paid)	text	not null
created_at	Date de la commande	timestamp	default current_timestamp
updated_at	Date de mise à jour	timestamp	nullable
id_certification	Identifiant unique de la certification	integer	primary key, auto_increment
user_id	Utilisateur certifié	integer	foreign key
cursus_id	Cursus certifié	integer	foreign key
obtained_at	Date d'obtention	timestamp	not null
filename	Nom du fichier de certification	text	nullable

V. Routes front et back

A. Back-end

ENDPOINT	OUTPUT	CONTROLLER	AUTH
Connexion			
POST /login	Authentifie l'utilisateur et génère un token de session	authController.login	Non-connecté
POST /register	Inscrit un nouvel utilisateur et envoie un mail de validation	authController.register	Non-connecté
GET /logout	Déconnecte l'utilisateur	authController.logout	Connecté
Users			
GET /users	Retourne la liste des utilisateurs	UserController.getAll	Administrateur
GET /users/:id	Retourne les informations d'un utilisateur	UserController.getOne	Connecté
PATCH /users/:id	Met à jour les infos d'un utilisateur	UserController.update	Connecté
DELETE /users/:id	Supprime un utilisateur	UserController.delete	Administrateur

Cursus			
GET /cursus	Retourne la liste des cursus disponibles	cursusController.getAll	Connecté
GET /cursus/:id	Retourne le détail d'un cursus	cursusController.getOne	Connecté
POST /cursus	Crée un nouveau cursus	cursusController.create	Administrateur
PATCH /cursus/:id	Met à jour un cursus existant	cursusController.update	Administrateur
DELETE /cursus/:id	Supprime un cursus	cursusController.delete	Administrateur
Leçons			
GET /lessons	Retourne la liste des leçons	lessonController.getAll	Connecté
GET /lessons/:id	Retourne le détail d'une leçon	lessonController.getOne	Connecté
POST /lessons	Crée une nouvelle leçon	lessonController.create	Administrateur
PATCH /lessons/:id	Met à jour une leçon	lessonController.update	Administrateur
DELETE /lessons/:id	Supprime une leçon	lessonController.delete	Administrateur
Commandes (Achat)			
POST /orders	Crée une nouvelle commande (cursus ou leçon)	orderController.create	Connecté
GET /orders	Liste des commandes de l'utilisateur connecté	orderController.getUserOrders	Connecté
GET /orders/:id	Détail d'une commande	orderController.getOne	Connecté

B. Front-end

URL	PAGE / VUE	DESCRIPTION	ACCÈS
/**	Accueil	Page d'accueil, présentation des formations, accès connexion / inscription	Public
/login	Connexion	Formulaire de connexion utilisateur	Public
/register	Inscription	Formulaire de création de compte	Public
/dashboard	Tableau de bord utilisateur	Vue principale après connexion (accès aux cursus, commandes, profil)	Connecté
/cursus	Liste des cursus	Page listant les cursus disponibles	Connecté
/cursus/:id	Détail du cursus	Affiche les leçons d'un cursus spécifique	Connecté
/lesson/:id	Détail d'une leçon	Consultation d'une leçon (contenu + vidéo)	Connecté
/certifications	Mes certifications	Vue récapitulative des cursus validés et certifications obtenues	Connecté
/profile	Profil utilisateur	Page pour modifier les informations personnelles	Connecté
/admin	Tableau de bord admin	Gestion des utilisateurs, cursus, leçons, commandes	Administrateur

I. Veille technologique

A. *Symfony et Doctrine ORM*

N'ayant jamais réalisé de projet professionnel avec le framework Symfony, j'ai dû me documenter sur son fonctionnement, notamment sur la gestion des entités et des bases de données avec Doctrine ORM.

J'ai comparé les avantages d'utiliser Doctrine face à des requêtes SQL manuelles :

- **Abstraction des requêtes** et gestion facilitée des relations entre tables.
- **Migration et versionnage du schéma** de base de données.
- **Productivité accrue** grâce à l'intégration avec Symfony

Sources utilisées :

- <https://symfony.com/doc/current/doctrine.html>
- <https://www.doctrine-project.org/projects/doctrine-orm/en/current/index.html>
- <https://symfonycasts.com/screencast/doctrine>

B. *Envoi d'emails avec Symfony Mailer*

Pour la validation de compte et les notifications, j'ai choisi Symfony Mailer. La documentation officielle m'a permis d'implémenter l'envoi d'emails de manière sécurisée :

- Gestion des transports SMTP (en local puis en production).
- Séparation du template d'email et de la logique d'envoi.

Sources utilisées :

- <https://symfony.com/doc/current/mailer.html>

C. Sécurisation des données utilisateurs

Pour sécuriser les mots de passe des utilisateurs, j'ai utilisé le PasswordHasher de Symfony, qui repose sur des algorithmes de hashage robustes (bcrypt par défaut).

Concernant la gestion des sessions et de l'authentification, j'ai mis en place :

- **Firewall et Access Control** via le Security Bundle de Symfony.
- **Token de validation** pour l'activation des comptes.

Sources utilisées :

- <https://symfony.com/doc/current/security.html>
- <https://symfony.com/doc/current/security/passwords.html>

II. Veille de sécurité

A. Bonnes pratiques OWASP

Pour garantir la sécurité de l'application, je me suis appuyé sur les recommandations de l'OWASP :

- Sécurisation des formulaires contre les attaques **CSRF**.
- Validation et nettoyage des données utilisateur.
- Gestion des **rôles et permissions**.
- Mise en place de **rate limiting** sur les endpoints critiques.

Sources utilisées :

- <https://owasp.org/www-project-top-ten/>
- <https://symfony.com/doc/current/security.html>

B. Gestion des CORS et API sécurisée

Pour les interactions entre le front et le back, j'ai configuré les en-têtes CORS via un listener Symfony, afin de restreindre les origines autorisées et éviter les attaques XSS.

En complément, un rate limiting basique via NGINX (ou bundle Symfony) serait une évolution future pertinente.

Sources utilisées :

- https://cheatsheetseries.owasp.org/cheatsheets/Symfony_Cheat_Sheet.html
- <https://symfony.com/bundles/NelmioCorsBundle/current/index.html>

C. Sécurisation des formulaires & XSS

Pour sécuriser les formulaires, j'ai utilisé les constraints du Validator de Symfony afin de vérifier la validité des champs côté serveur (ex : Email, NotBlank, Length).

Côté front-end, les champs sont également protégés par les attributs HTML5 (required, type="email") et de simples scripts de validation JavaScript.

Les templates Twig sont sécurisés nativement via le système d'auto-escape, ce qui évite les failles XSS. J'ai également utilisé le filtre e() sur les données sensibles affichées.

Exemple de validation dans contrôleur Symfony :

src/Controller/CursusController.php

```
1  #[Route('/register', name: 'app_register')]
2  public function register(Request $request, EntityManagerInterface $entityManager,
3  UserPasswordHasherInterface $passwordHasher, UserAuthenticatorInterface $userAuthenticator,
4  LoginFormAuthenticator $authenticator): Response
5  {
6      // Block user when connected and redirect to student
7      if ($this->getUser()) {
8          return $this->redirectToRoute('student');
9      }
10
11      $user = new User();
12      $form = $this->createForm(RegistrationFormType::class, $user);
13      $form->handleRequest($request);
14
15      if ($form->isSubmitted() && $form->isValid()) {
16          // Générer un token de validation
17          $verificationToken = Uuid::v4()->toRfc4122();
18          $user->setActivated(false);
19          $user->setVerificationToken($verificationToken);
20
21          // Add the default "ROLE_USER" role
22          $user->setRoles(['ROLE_USER']);
23
24          // Encode password
25          $hashedPassword = $passwordHasher->hashPassword(
26              $user,
27              $form->get('plainPassword')->getData()
28          );
29          $user->setPassword($hashedPassword);
30
31          // Set creation date
32          $user->setCreatedAt(new \DateTimeImmutable());
33
34          if ($this->getUser()) {
35              $user->setCreatedBy($this->getUser());
36          }
37
38          // Save user
39          $entityManager->persist($user);
40          $entityManager->flush();
41
42          // Generate validation link
43          $verificationLink = $this->generateUrl('app_verify_email', [
44              'token' => $verificationToken
45          ], UrlGeneratorInterface::ABSOLUTE_URL);
46
47          // Send validation email
48          $this->mailerService->sendVerificationEmail($user->getEmail(), $verificationLink);
49
50          // Redirect user to login page
51          return $this->redirectToRoute('app_login');
52      }
53
54      return $this->render('registration/register.html.twig', [
55          'registrationForm' => $form->createView(),
56      ]);
57  }
```

Exemple Template Twig (XSS protection) :

```
1 {% extends 'base.html.twig' %}
2
3 {% block title %}Accueil - Formations
4 {% endblock %}
5
6 {% block body %}
7     <div class="container mt-5">
8         <h1 class="text-center mb-5">Bienvenue sur Knowledge Learning 🎓</h1>
9
10        <div class="row">
11            {% for theme in themes %}
12                <div class="col-md-6 mb-4">
13                    <div class="card shadow">
14                        <div
15                            class="card-body text-center">
16                            {# Display the title of the theme with XSS protection using auto-escaping #}
17                            <h2>{{ theme.title|e }}</h2>
18
19                            {# Link to the detailed page of the selected theme (cursus list) #}
20                            <a href="{{ path('theme_show', {'id': theme.id}) }}" class="btn btn-primary mt-3">
21                                Voir les cursus
22                            </a>
23                        </div>
24                    </div>
25                </div>
26            {% endfor %}
27        </div>
28    </div>
29 {% endblock %}
```

Grâce aux mécanismes de validation des formulaires de Symfony et à la fonction d'auto-échappement de Twig, la plateforme Knowledge Learning garantit à la fois l'intégrité des données saisies par les utilisateurs et la sécurité des contenus affichés.

En validant les données avant leur enregistrement et en assainissant les sorties côté interface, les risques d'attaques SQL Injection et XSS (Cross-Site Scripting) sont considérablement réduits.

Cela permet d'offrir une expérience utilisateur plus fiable et sécurisée.

D. PHPUnit

Pour garantir le bon fonctionnement des formulaires et des parcours utilisateurs, j'ai réalisé des tests fonctionnels avec PHPUnit sur des routes clés, comme l'inscription.

Ce test simule la soumission d'un formulaire d'inscription avec des données valides, et vérifie que l'utilisateur est bien redirigé vers la page de connexion après enregistrement.

Cela permet de s'assurer que le processus d'inscription est fonctionnel et sécurisé.

```
1  <?php
2
3  namespace App\Tests\Controller;
4
5  use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;
6
7  class RegistrationControllerTest extends WebTestCase
8  {
9      /**
10       * Test that the registration page loads,
11       * allows form submission with valid data,
12       * and redirects to the login page.
13       */
14     public function testRegisterPageLoadsCorrectly(): void
15     {
16         // 1) Create a new HTTP client for this test
17         $client = static::createClient();
18         // 2) Request the registration page
19         $crawler = $client->request('GET', '/register');
20
21         // 3) Generate unique credentials to avoid unique-key collisions
22         $username = 'user_' . uniqid();
23         $email = 'user_' . uniqid() . '@example.com';
24         // Ensure the password meets any minimum length requirements
25         $password = '7e]2Azvx7aCAG)';
26
27         // 4) Select the registration form by its submit button label
28         $form = $crawler->selectButton("S'inscrire")->form([
29             // Map your form field names to the test values
30             'registration_form[username]' => $username,
31             'registration_form[email]' => $email,
32             'registration_form[plainPassword]' => $password,
33         ]);
34
35         // 5) Submit the form
36         $client->submit($form);
37
38         // 6) Assert that after submission, we are redirected to the login page
39         $this->assertResponseRedirects('/login');
40     }
41 }
```

Simulation de test unitaire :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\tyson\OneDrive\Bureau\knowledge-learning> php bin/phpunit C:\Users\tyson\OneDrive\Bureau\knowledge-learning\tests\Controller\RegistrationControllerTest.php
PHPUnit 9.6.21 by Sebastian Bergmann and contributors.

Testing App\Tests\Controller\RegistrationControllerTest
.
1 / 1 (100%)

Time: 00:01.497, Memory: 42.00 MB

OK (1 test, 2 assertions)
PS C:\Users\tyson\OneDrive\Bureau\knowledge-learning>
```

III. Processus de recherche et traduction

A. *Processus*

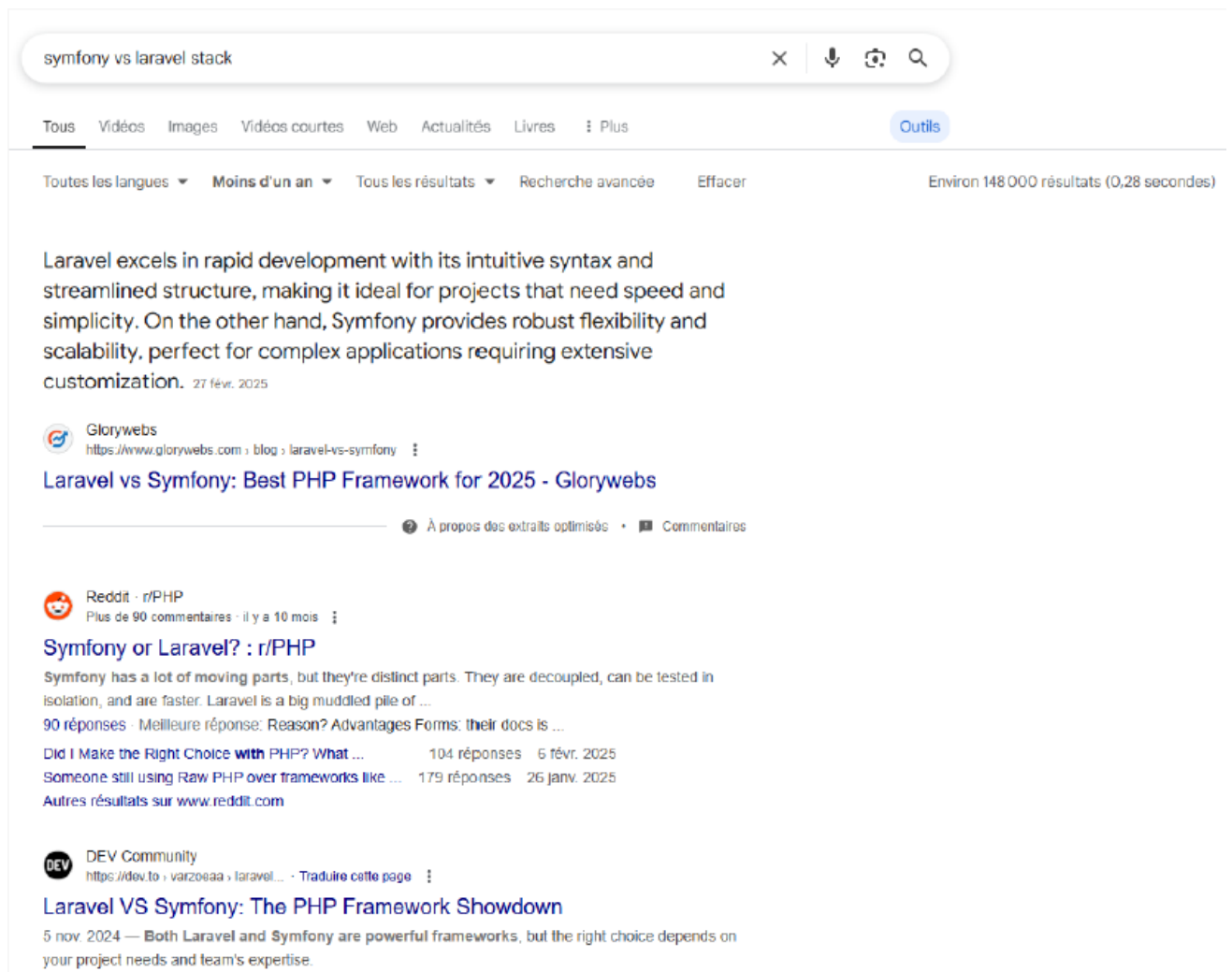
Pour effectuer mes recherches, j'adopte en grande partie le processus de recherche suivant:

1. Requête en anglais, plutôt qu'en français, sur google avec des termes simples
2. Sélection des sources par rapport au classement personnel, ainsi que la date de création / mise à jour de la source
3. Analyse critique des informations contenues dans les pages sélectionnées

Pour ma veille, j'ai principalement utilisé :

1. **Documentation officielle de Symfony et Doctrine.**
2. Les bonnes pratiques de **l'OWASP** pour la sécurité des applications web.
3. Des articles spécialisés (SymfonyCasts, Medium) pour des cas concrets.
4. Des échanges sur StackOverflow pour des retours d'expérience sur des problématiques spécifiques.

Dans un premier temps, j'effectue une requête simple pour aborder ce questionnement :



Dans le cadre de la réalisation de mon projet Knowledge Learning, j'ai souhaité approfondir mes connaissances sur le choix du framework Symfony face à d'autres alternatives, notamment Laravel.

J'ai donc effectué une recherche en anglais sur Google avec la requête : "symfony vs laravel stack".

Sur la page de résultats, le premier article proposé est celui de Glorywebs : "Laravel vs Symfony : Best PHP Framework for 2025". Cet article présente une comparaison des deux frameworks selon plusieurs critères : simplicité, rapidité de développement, flexibilité, évolutivité.

Je garde à l'esprit qu'il s'agit d'un article de blog et que ce sujet est souvent source de préférences personnelles et de débats dans la communauté PHP.

Pour croiser les points de vue, j'ai également consulté un fil de discussion sur Reddit (r/PHP). Ce type de forum permet de lire des retours d'expérience concrets d'utilisateurs ayant travaillé avec les deux frameworks. Ces échanges permettent d'obtenir des avis plus nuancés et moins "marketing".

Enfin, j'ai trouvé un article de la Dev Community intitulé : "Laravel vs Symfony : The PHP Framework Showdown". Cet article reprend les avantages et inconvénients des deux solutions selon le contexte d'utilisation (projet simple, projet complexe, rapidité vs robustesse).

Au vu de mon projet, nécessitant une bonne structure, une gestion avancée des utilisateurs et des rôles, et une évolutivité possible, j'ai confirmé que Symfony était le choix le plus adapté à mes besoins.

B. Ressource en anglais

Dans le cadre de cette recherche, voici l'extrait que j'ai choisi de traduire. Il est issu de l'article "Laravel vs Symfony: Best PHP Framework for 2025" sur Glorywebs :

- <https://www.glorywebs.com/blog/laravel-vs-symfony/>

Extrait original (en anglais) :

"Laravel excels in rapid development with its intuitive syntax and streamlined structure, making it ideal for projects that need speed and simplicity. On the other hand, Symfony provides robust flexibility and scalability, perfect for complex applications requiring extensive customization."

C. Traduction effectuée

Laravel excelle dans le développement rapide grâce à sa syntaxe intuitive et sa structure simplifiée, ce qui le rend idéal pour les projets nécessitant de la rapidité et de la simplicité.

En revanche, Symfony offre une flexibilité et une scalabilité robustes, le rendant parfaitement adapté aux applications complexes nécessitant une personnalisation avancée.

CONCLUSION

Ce projet Knowledge Learning m'a permis de consolider mes compétences en développement web, et plus particulièrement dans l'utilisation du framework Symfony et de Doctrine ORM pour la gestion des données.

Travailler seule sur ce projet a été un véritable défi, car j'ai dû gérer l'ensemble des aspects du développement : la conception, le back-end, le front-end, la sécurité, mais aussi le déploiement et la documentation. Cette autonomie m'a obligée à structurer rigoureusement ma méthode de travail et à approfondir mes recherches pour trouver les solutions adaptées.

L'objectif de transformer des supports de formation physiques en une plateforme e-learning fonctionnelle et évolutive a été atteint, tout en respectant les bonnes pratiques en termes de développement et de sécurité.

Ce projet m'a également permis d'améliorer ma capacité à faire de la veille technologique, à analyser des choix techniques (Symfony vs alternatives) et à appliquer des notions concrètes de gestion de projet web.

Enfin, cette expérience a renforcé ma polyvalence et m'a permis d'acquérir des compétences directement réutilisables dans un contexte professionnel.

ANNEXE

Les repos du projet sont accessibles à cette adresse : <https://github.com/devlivepro/Knowledge-Learning>

Le projet est accessible à l'adresse suivante : <https://knowledge-learning.fr/>

RÔLE	IDENTIFIANT	MOT DE PASSE
Administrateur	admin	password
Étudiant	student	password

Le front et le back sont déployés o2switch.