KNOWLEDGE

Application d'e-commerce et e-learning.

DATE : 19/06/2025

Projet réalisé dans le cadre de la présentation au

TITRE PROFESIONNEL Développeur Web et Web Mobile

présenté par

LECAVELIER DESETANGS YOANN

Centre Européen de Formation

KNOWLEDGE

Sommaire

LISTE DES COMPETENCES COUVERTES PAR LE PROJET 5

Développer la partie front-end d'une application web reponsive 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 web dynamique 5

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 6

CAHIER DES CHARGES 7

Besoins et objectifs de l'application 7

A. Besoin 7

B. Objectif 8

C. Cibles 9

II. Users Stories 10

III. Arborescence 10

IV. MVP 12

A . Utilisateur 12

B. Étudiant 12

C. Administrateur 13

V. Fonctionnalités détaillées des pages 13

VI. Evolutions potentielles 14

VII. Wireframes 14

VIII. Charte graphique et logo 17

IX. Exemples de maquettes. 18

X. Explication du côté responsive. 19

XI. Accessibilité. 19

XII. Optimisation SEO. 20

Spécifications techniques 20

I. Technologies 20
II. Navigateurs compatibles 20
III. Possibilités de déploiement.....21
1. Le déploiement local 21
2. Le déploiement par un hébergeur 21
IV. Création de la base de données 22
V. Routes Front et Back 23
A . Front-end.....23
B. Back-end.....24
VI. Fonctionnement global 25
A. Lien entre le back-end et le front-end. 25
B. Prise en compte des RGPD 25
C. Gestionnaire de version 26
D. Processus de déploiement 26
E. Failles de sécurité majeures.....27
Conclusion.....28
Annexe.....29

LISTE DES COMPETENCES COUVERTES PAR LE PROJET

Développer la partie front-end d'une application web reponsive en intégrant les recommandations de sécurité

A. Maquetter une application

1. Définition du concept d'application
2. Identification des fonctionnalités clés
3. Création de wireframes
4. Élaboration d'une charte graphique
5. Conception de maquettes visuelles

B. Réaliser une interface utilisateur web statique et adaptable

1. Analyse des cibles utilisateurs
2. Prise en compte de la diversité des profils utilisateurs
3. Choix d'une interface accessible et intuitive
4. Conception centrée sur l'expérience utilisateur

C. Développer une interface web dynamique

1. Analyse des besoins fonctionnels d'un projet hybride (E-Commerce & E-Learning)
2. Choix d'une solution adaptée aux objectifs du projet
3. Structuration d'une architecture adaptée à la double fonctionnalité
4. Choix du framework : Symfony.

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

A. Créer une base de données

Afin de réaliser la base de données du projet, j'ai utilisé MySQL, avec l'application Doctrine de Symfony et PhpMyAdmin, puis j'ai structuré les données grâce aux Entity de Symfony.

B. Développer les composants d'accès aux données.

Pour accéder aux données de l'application, j'ai utilisé Doctrine ORM, et afin d'assurer le versionning de l'application, j'ai utilisé l'invité de commande avec Doctrine Migration.

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

Pour la partie back-end de mon application, j'ai utilisé les extensions de symfony, tel que « form » pour réaliser les formulaires d'inscription.

CAHIER DES CHARGES

Besoins et objectifs de l'application

A. Besoin

Afin de connaître exactement les besoins d'un étudiant utilisant un site E-Learning, je me suis inspiré du site sur lequel je suis inscrit. (Centre Européen de Formation)

Ce dont nécessite un utilisateur classique :

- Se créer un compte.
- Se connecter à son compte
- Voir les formations disponibles.
- Voir les leçons dans les formations.
- Voir le prix de ses formations.
- Acheter une formation

Ce dont nécessite un étudiant :

- Se connecter à son compte.
- Récupérer ses leçons par rapport à sa formation.
- Étudier.

(L'étudiant à également le droit de voir les autres formations disponibles.)

Ce dont nécessite un administrateur :

- Voir les utilisateurs.
- Voir les formations.
- Modifier les formations.
- Créer une formation.
- Supprimer une formation.
- Créer une leçon.
- Modifier une leçon.
- Supprimer une leçon.
- Modifier les informations d'un utilisateur.
- Supprimer un utilisateur.

B. Objectif

Pour réaliser une bonne plateforme d'E-Learning il faut qu'elle s'adresse à tous, mais de manière simple.

Les informations à prendre en compte sont :

- L'utilisateur est peut-être un futur étudiant, c'est pour cela qu'il faut que son expérience soit toutes aussi agréable qu'un étudiant.
- L'étudiant à besoin d'accéder à ses leçons sans problème et sans incompréhension du logiciel.
- L'administrateur doit gérer le logiciel afin de le garder à jour et d'intervenir en cas de problème.

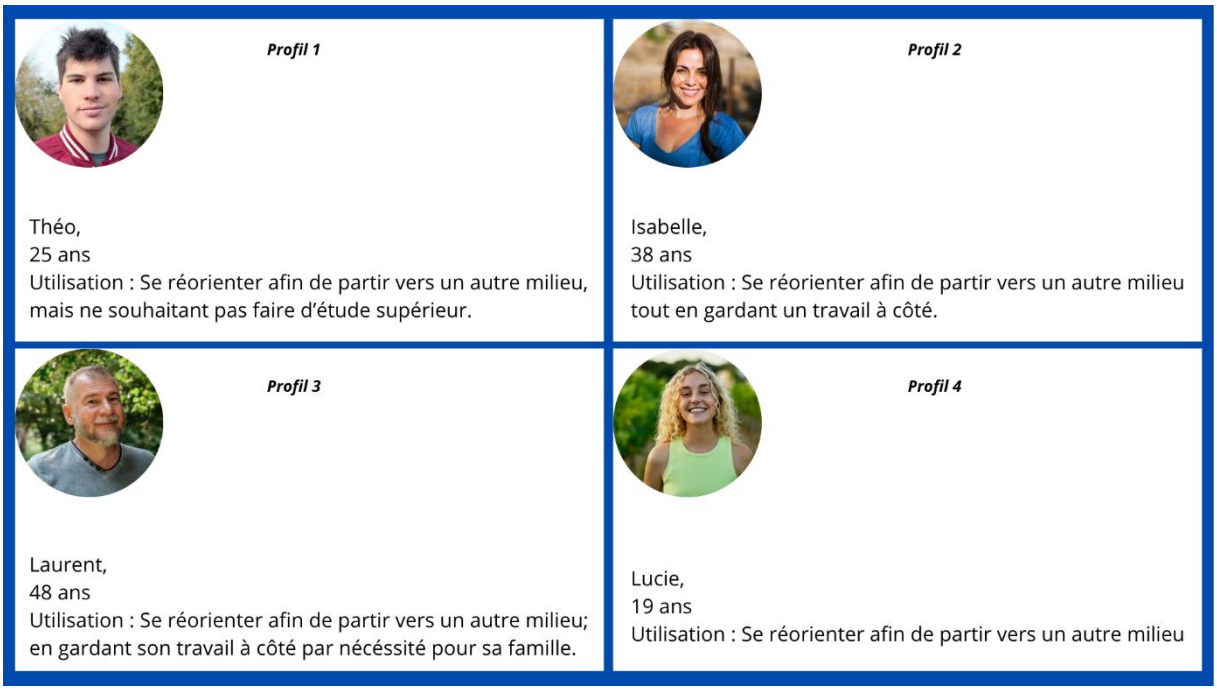
En analysant le site du CEF (Centre Européen de Formation), je me suis rendu compte que notre expérience était agréable, car l'apparence est jolie, cependant je me suis rendu compte pendant ma formation qu'il y avait énormément de soucis, car les pages sont parfois « chargés » ce qui créait des latences, avec un bon ordinateur et une bonne connexion, il y a parfois aussi des soucis en récupérant les ressources (tel que la page reste blanche pendant longtemps le temps de charger).

En ayant compris cela, je me suis rendu compte qu'on pouvait faire quelque chose d'esthétique, mais pas chargé en design, ce qui pourra permettre d'avoir une expérience agréable et fluide.

C. Cibles

Pour avoir une représentation correcte des utilisateurs de site E-Learning, j'ai regardé les statistiques sur Internet.

La moyenne d'âge se situe entre 25 à 45 ans.



Profil 1

Théo,
25 ans
Utilisation : Se réorienter afin de partir vers un autre milieu, mais ne souhaitant pas faire d'étude supérieur.

Profil 2

Isabelle,
38 ans
Utilisation : Se réorienter afin de partir vers un autre milieu tout en gardant un travail à côté.

Profil 3

Laurent,
48 ans
Utilisation : Se réorienter afin de partir vers un autre milieu; en gardant son travail à côté par nécéssité pour sa famille.

Profil 4

Lucie,
19 ans
Utilisation : Se réorienter afin de partir vers un autre milieu

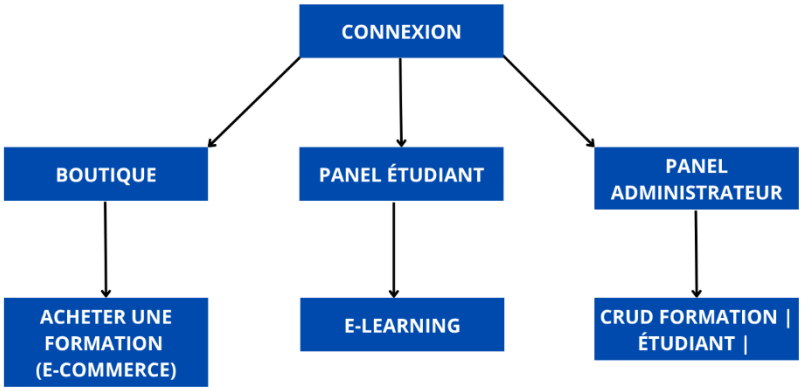
En conclusion :

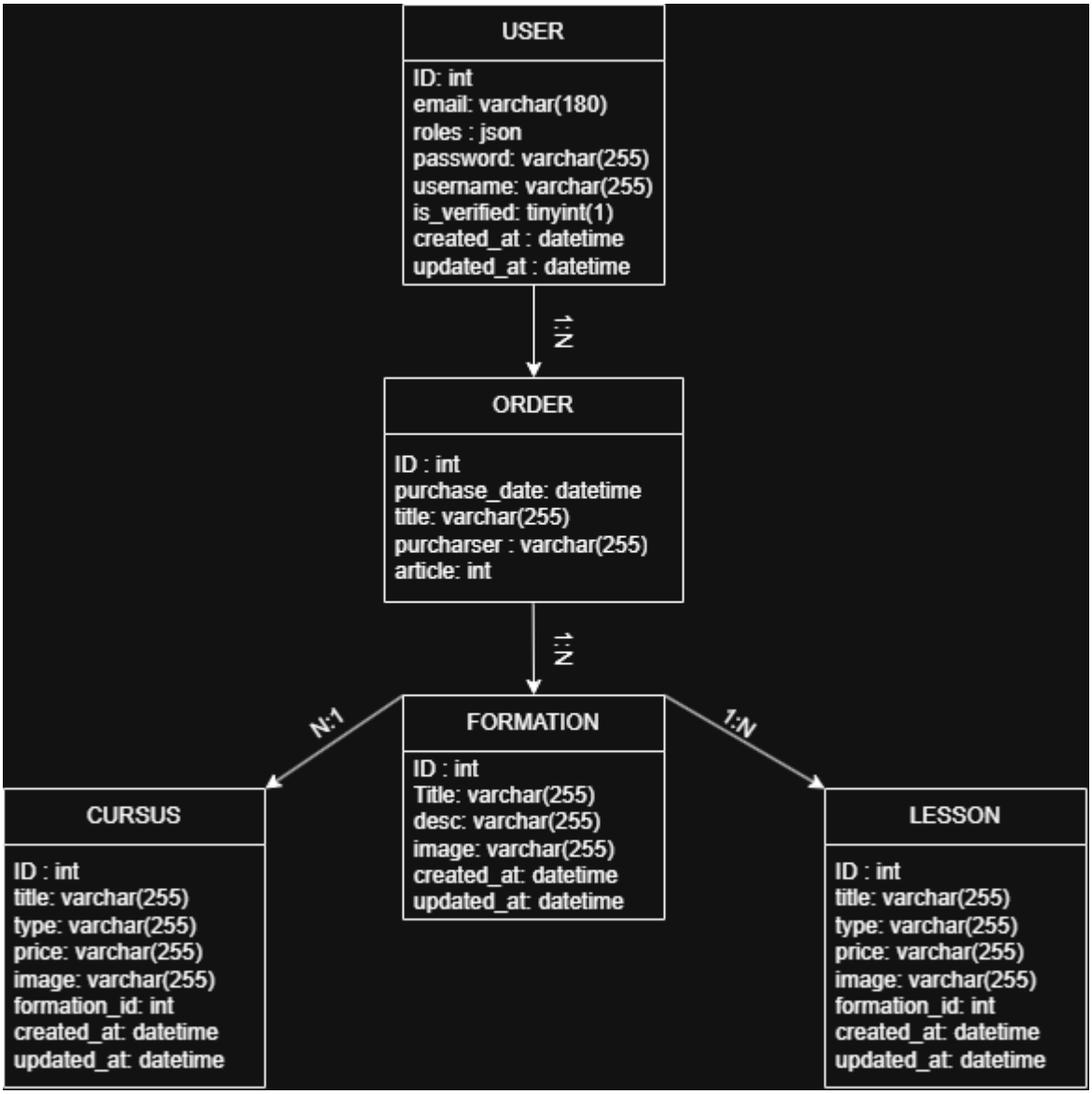
Les utilisateurs attendent à peu près la même chose du site internet, avec des situations différentes.

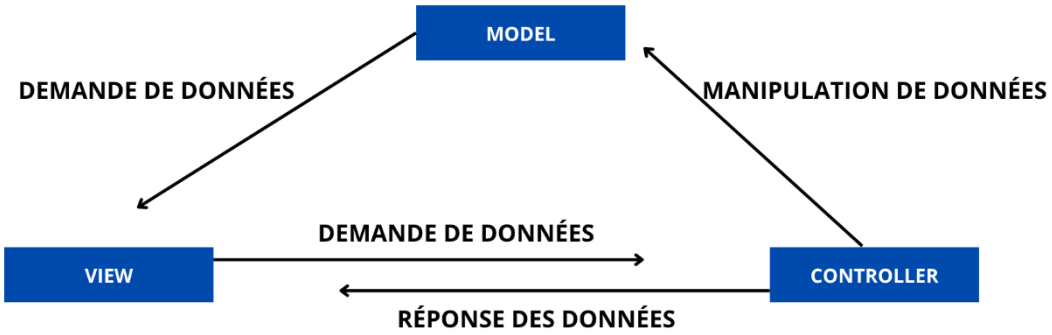
II. Users Stories

RÔLE	JE SOUHAITE	AFIN DE
Visiteur	M'inscrire	Visualiser les formations
Visiteur	Me connecter	Acheter une formation
Etudiant	Me connecter	Accéder à ma formation
Etudiant	Accéder à ma formation	Etudier
Administrateur	Me connecter	D'accéder à la plateforme
Administrateur	Accéder à la plateforme	Accéder au panel administrateur
Administrateur	Accéder au panel administrateur	Maintenir à jour le site, et les formations.

III. Arborescence




USER
ID: int
email: varchar(180)
roles : json
password: varchar(255)
username: varchar(255)
is_verified: tinyint(1)
created_at : datetime
updated_at : datetime
1:N
ORDER
ID : int
purchase_date: datetime
title: varchar(255)
purcharser : varchar(255)
article: int
1:N
N:1
1:N
FORMATION
ID : int
Title: varchar(255)
desc: varchar(255)
image: varchar(255)
created_at: datetime
updated_at: datetime
CURSUS
ID : int
title: varchar(255)
type: varchar(255)
price: varchar(255)
image: varchar(255)
formation_id: int
created_at: datetime
updated_at: datetime
LESSON
ID : int
title: varchar(255)
type: varchar(255)
price: varchar(255)
image: varchar(255)
formation_id: int
created_at: datetime
updated_at: datetime



IV. MVP

Mon Produit Minimum Viable s'est construit autour de l'achat de formation et le E-learning qui sont les buts principaux de cette application.

Pour cela, une connexion sécurisée est obligatoire afin de protéger les données de nos utilisateurs.

Fonctionnalités présentes dans l'application :

- L'utilisateur doit pouvoir s'inscrire et valider son inscription par mail.

A . Utilisateur

- Arrive sur la page d'accueil : accéder à la boutique.
- Voir les formations disponibles.
- Voir les détails des formations.
- Voir les leçons présentes dans les formations.
- Voir les prix des leçons.
- Acheter une leçon.

B. Étudiant

- Arrive sur le panel étudiant, il doit pouvoir :
- Consulter les formations disponibles.
- Voir ses leçons.
- Voir sa formation.

C. Administrateur

- Arriver sur le panel administrateur, il doit pouvoir :
- Consulter les formations disponibles.
- Mettre à jour les formations disponibles.
- Supprimer les formations disponibles.
- Ajouter des formations.
- Consulter les élèves.
- Vérifier les informations des élèves.
- Modifier les informations des élèves.
- Supprimer des élèves.
- Consulter les achats.

V. Fonctionnalités détaillées des pages

Les pages/routes autres que l'accueil sont sécurisées de manière à ce qu'un **utilisateur non vérifié** (*par mail*) **ne puisse pas s'y rendre**. Seules les pages « Accueil, Inscription, Connexion » sont accessibles à tous.

➢ **Page de connexion :**

Une fois l'utilisateur authentifié avec succès, il sera automatiquement redirigé par mail.
Cependant, s'il n'a pas vérifié son compte par mail, il n'aura toujours pas accès à la page de boutique.

➢**Page accueil :**
Cette page sert à préciser à l'utilisateur qu'il doit s'inscrire avant d'accéder aux services disponibles sur l'application, une fois fait, il a plusieurs choix :

- Vérifier son panier.
- Voir les produits dans la boutique.
- Se déconnecter.

➢**Page boutique :**

Sur cette page, nous retrouverons toutes les formations disponibles sur le site.

Avec un bouton de redirections afin de savoir les détails des formations

➢**Page Détails de formations :**

Sur cette page, nous retrouverons tous les détails de la formation sélectionnée afin d'en savoir davantage sur la formation.

➢**Page panier :**

Sur cette page, nous retrouverons les produits ajoutés dans notre panier afin de vérifier le prix et le choix du produits.

Et un bouton de redirection afin de payer le ou les produits sélectionnés.

➢**Page paiement :**

Sur cette page, nous retrouverons un script réalisé avec Stripe afin d'avoir un sécurisé et facile d'accès, qui évite également les fraudes.

➢**Page Panneau d'administration :**

Sur cette page, nous retrouverons plusieurs fonctionnalités telles que :

- CRUD Formations.
- CRUD Etudiant.
- Suivis des achats.

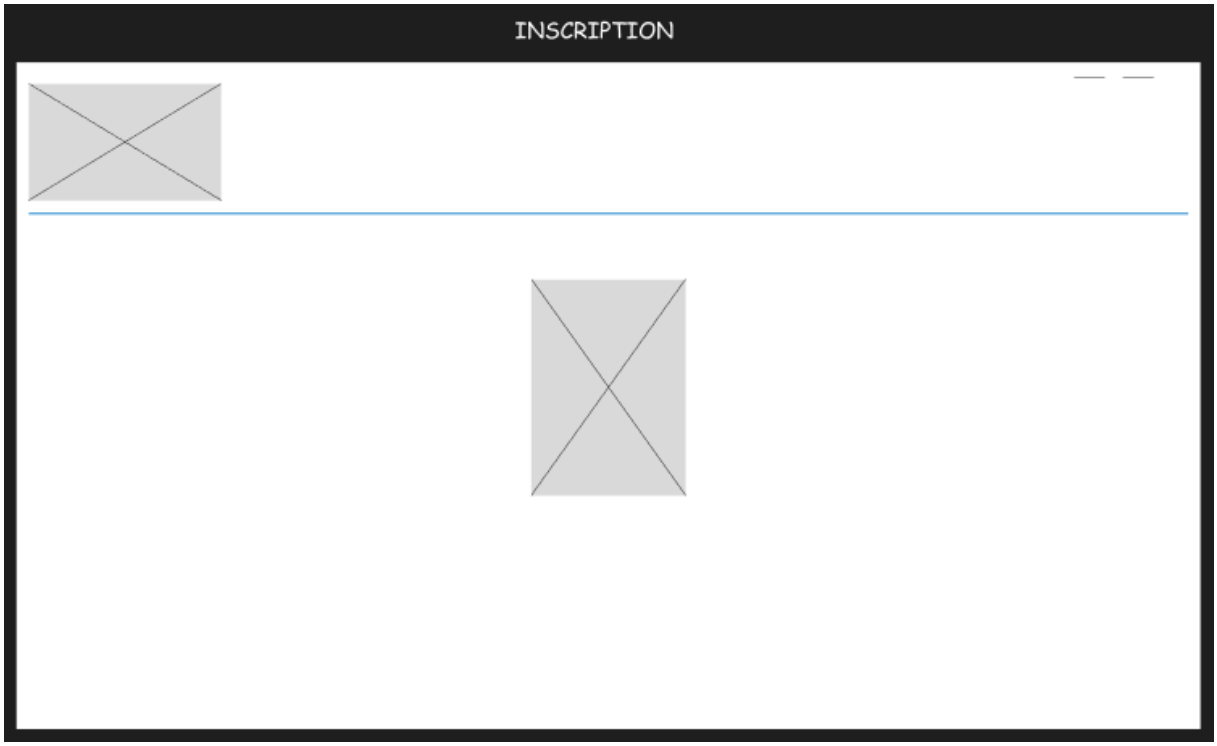
VI. Evolutions potentielles

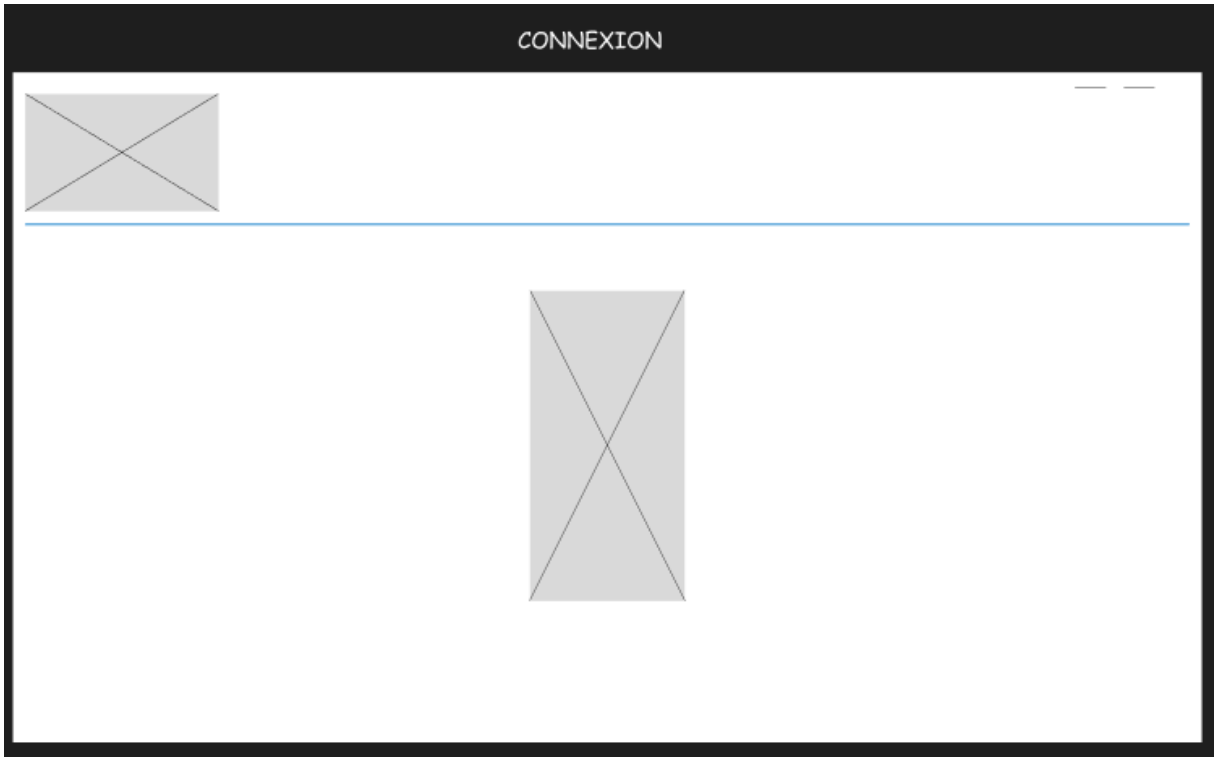
Il est sûr qu'un site comme celui-ci peut obtenir plusieurs évolutions telles que :

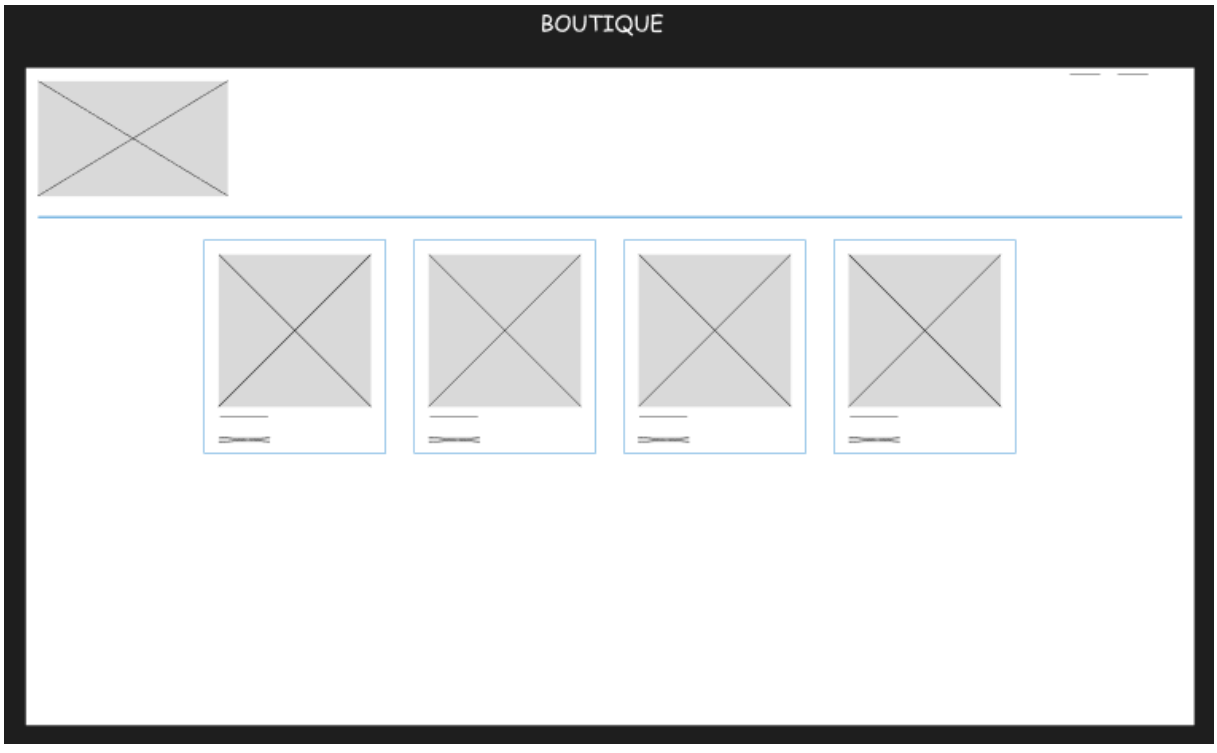
- Un espace support.
- Un chat en ligne afin de pouvoir communiquer avec d'autres élèves.
- Un espace mentorat afin de pouvoir y accueillir des mentors pouvant accompagner les élèves.
- Un bulletin en ligne.
- Paramètres de compte personnel (mettre à jour ses propres informations)

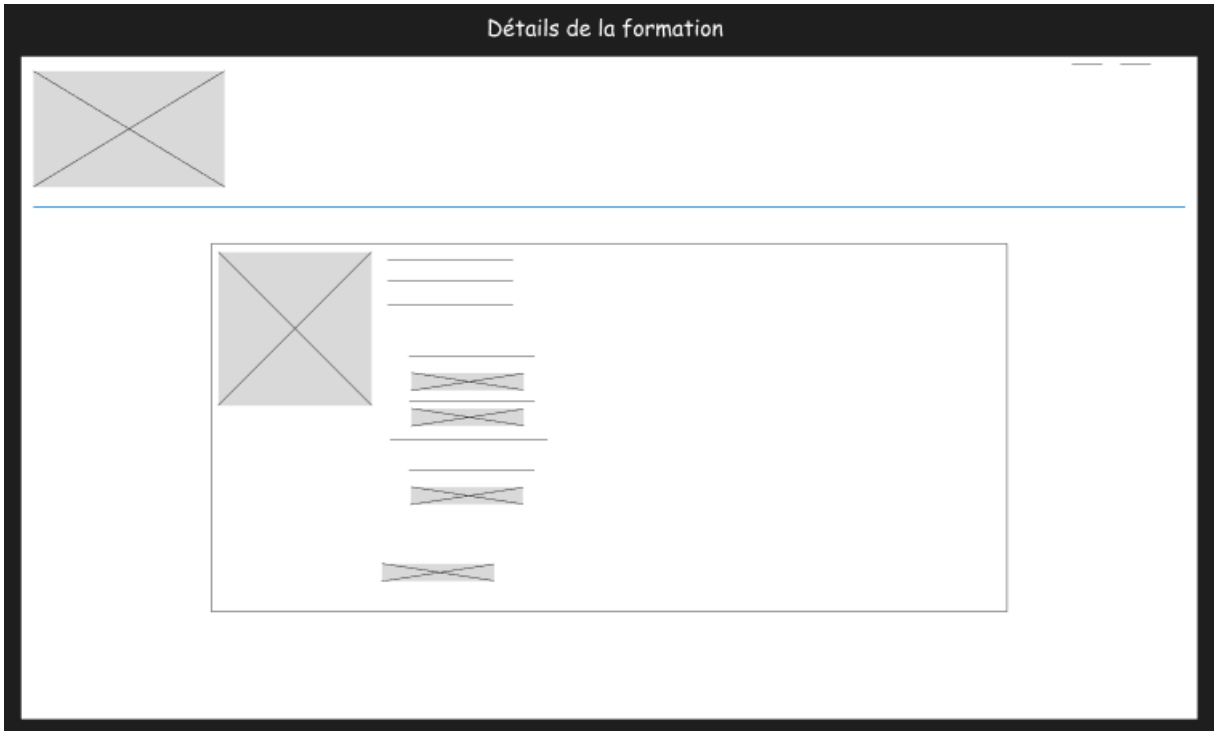
VII. Wireframes


ACCEUIL


INSCRIPTION


CONNEXION


BOUTIQUE



VIII. Charte graphique et logo

Pour une expérience agréable, j'ai décidé de réaliser une charte graphique « classique » afin de ne pas avoir différentes polices et couleurs pouvant déranger.

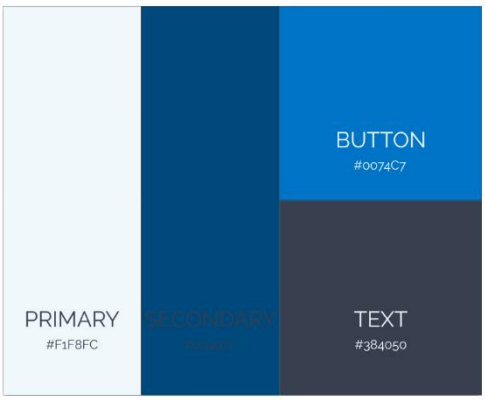
TYPOGRAPHIE

Comic Sans MS

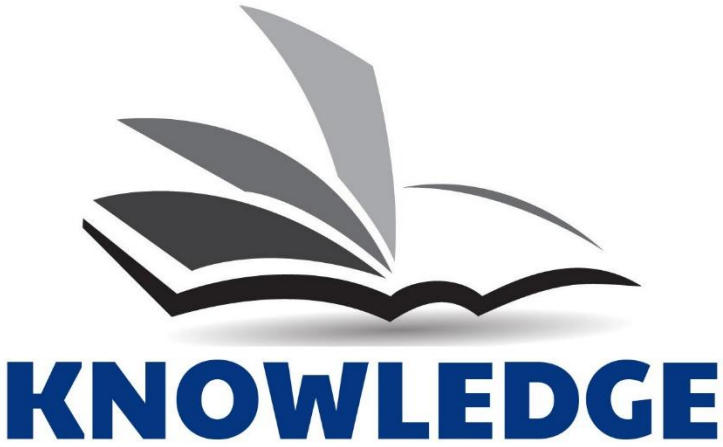
Comic Sans MS **Comic Sans MS**

Lorem ipsum dolor sit amet **Lorem ipsum dolor sit amet**

COLORS

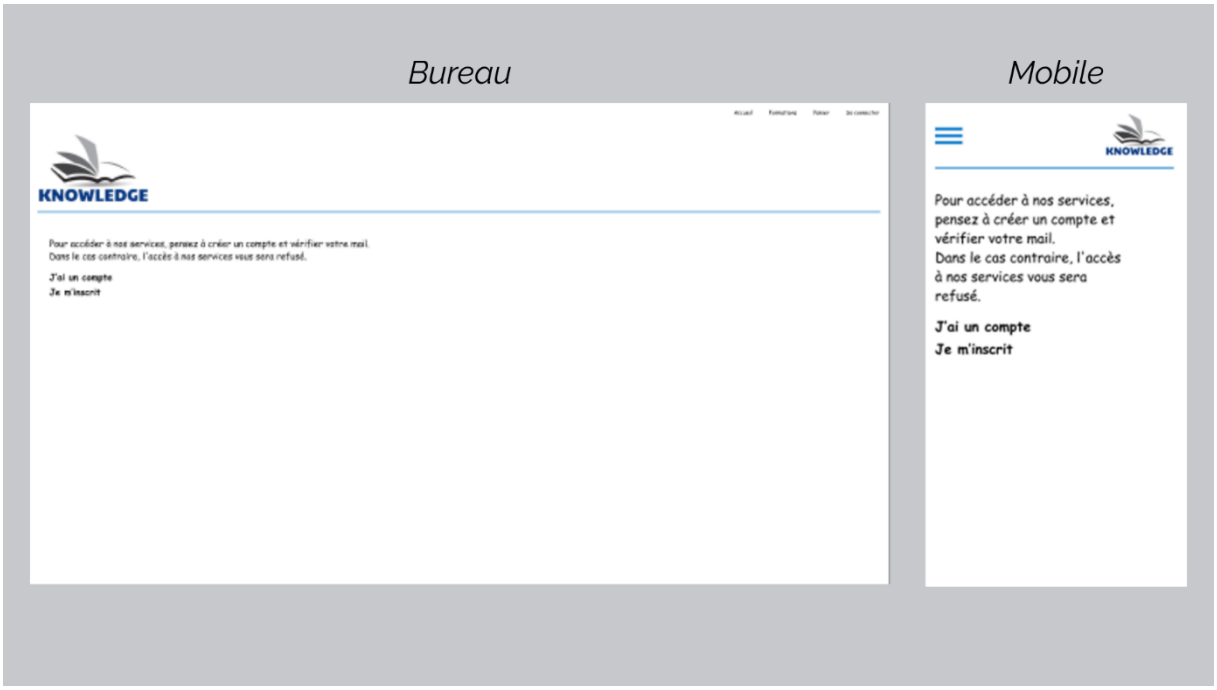


Pour le logo, j'ai fait le choix de gardé celui de mon précédent projet, étant donné qu'il était accordé à ses couleurs, et qu'il est simple et efficace.



IX. Exemples de maquettes.

Exemple de la maquette réalisé pour la page d'accueil.



Voici l'intégralité des maquettes via un lien cliquable :

https://www.figma.com/design/G9XqdzE4OoO5kTpW3C8rrh/Untitled?node-id=0-1&t=KLUASTyCgQ6Q3dcX-1

X. Explication du côté responsive.

Pour ce qui est du côté responsive, j'ai réalisé des breakpoint afin d'avoir un aperçu différent de la page entre un ordinateur, et un smartphone grâce aux media queries.

XI. Accessibilité.

Pour ce qui est de l'accessibilité, j'ai décidé de réaliser une navbar avec l'intégralité des liens (adapté à l'utilisateur), si l'utilisateur est un administrateur, il aura tous les liens avec les liens administrateur, s'il est étudiant, il aura les liens classiques, et un lien redirigeant sur le panel étudiant.

XII. Optimisation SEO.

Afin de réaliser un SEO efficace, réalisé le titre de chaque page sur les pages, avec une URL lisible.

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <link rel="stylesheet" href={{
      "styles/app.css"
    }}>

    <link rel="icon" type="image/png" href="{{ asset('favicon.png') }}" />
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Home</title>
  </head>
  <body>
```

Spécifications techniques

Pour réaliser les spécifications techniques, j'ai d'abord réfléchi aux besoins des utilisateurs en priorité afin d'adapter les technologies.

I. Technologies

- **Front-End**
  - **symfony/assets**
  - **symfony/form**
  - **symfony/twig-bundle**
- **Back-end**
  - **Symfony**
  - **Symfony/mailer**
  - **Doctrine**
  - **MySQL**

II. Navigateurs compatibles

Selon une étude effectuée sur Clubic :

En 2025, les trois navigateurs Web les plus utilisés dans le monde sont :

Google Chrome, **Safari** et **Microsoft Edge**.

Google Chrome domine largement la navigation Web mondiale avec une part de marché de 65,12%

Mon application devra donc être compatible avec ses navigateurs.

III. Possibilités de déploiement

Plusieurs possibilités sont exploitables pour le déploiement de l'application, de la plus à la moins coûteuse, et de la plus à la moins efficace.

1. Le déploiement local

Avantages :

- Faible coût.
- Possibilité de déployer l'ensemble de l'application assez rapidement.
- Peu de risque d'intrusion.
- Contrôle total des données.

Désavantages :

- Nécessite un ordinateur connecté au réseau local, allumé.
- Uniquement utilisable si connecté sur le même réseau local.
- Donne un URL illisible.
- Mise à jour manuel.

2. Le déploiement par un hébergeur

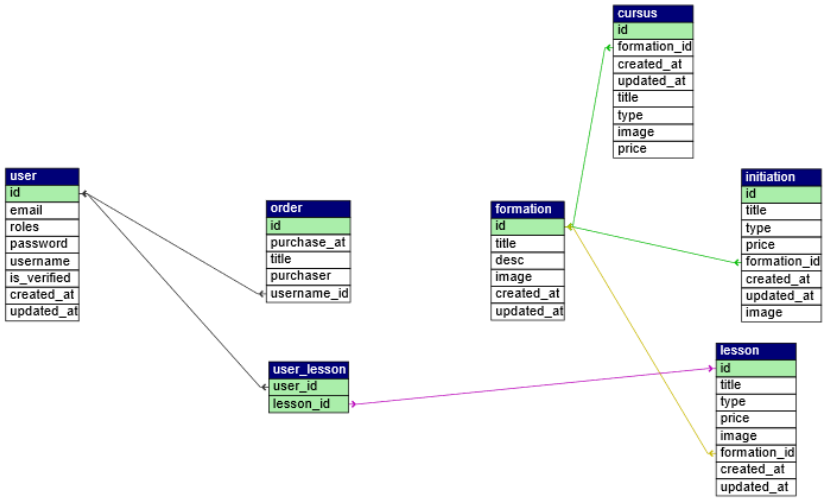
Avantages :

- Toujours disponible.
- Possibilité de déployer l'ensemble de l'application assez rapidement.
- Support pour le déploiement, et en cas de problème également.
- Mise à jour depuis un FTP.

Désavantages :

- Peu avoir un coût moyen ou élevé selon le choix de l'hébergeur.
- Nécessite un développeur ou une personne ayant les compétences de déployer / mettre à jour depuis un FTP.

IV. Création de la base de données



USER : (id,email,roles,password,username ,is_verified,created_at ,updated_at)
ORDER : (id,purchase_at,title,purchaser,username_id)
USER_LESSON : (user_id,lesson_id)
FORMATION : (id,title,desc,image,created_at,updated_at)

CURSUS : (id,formation_id,created_at,updated_at,title,type,image,price)
INITIATION : (id,title,type,price,formation_id,created_at,updated_at,image)
LESSON : (id,title,type,price,image,formation_id,created_at,updated_at)

Voici comme exemple, comment est réalisé une table de données.

User :

ID : ID de l'utilisateur (numéro propre à chaque utilisateur) (Integer)
Email : adresse e-mail. (varchar)
Roles : Rôle de l'utilisateur (json)
Password : Mot de passe (varchar)
Username : Nom d'utilisateur (varchar)
Is_verified : Vérifie si l'utilisateur est vérifié (boolean)
Create_at : Créer à (datetime)
Updated_at : Mise à jour à (datetime)

Mais, nous pouvons voir également que l'id de la table **User**, est relié à user_id de la table **User_Lesson** qui est également relié à l'id de la table **Lesson**.

Et que la table **Formation** a également un champ **formation_id** présent dans chaque composant d'une formation (**leçon**, **cursus**, **initiation**) .

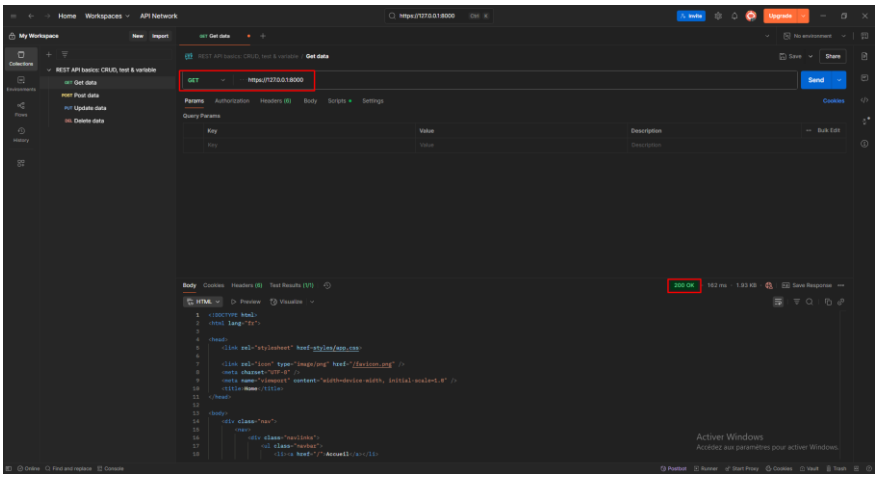
V. Routes Front et Back

A . Front-end

ROUTES FRONT :

URL	Title	Content
/	Acceuil	Acceuil du site web
/formations	Formations	Boutique de formation
/formations/{id}	Détails de la formation	Affiche les détails sur la formation
/cart	Panier	Affiche le contenu du panier
/login	Connexion	Affiche le formulaire de connexion
/register	Inscription	Affiche le formulaire d'inscription
/student	Élève	Affiche la page d'apprentissage
/logout	Déconnexion	Déconnecte l'utilisateur et retourne à l'acceuil

Test d'accès à la route via postman.



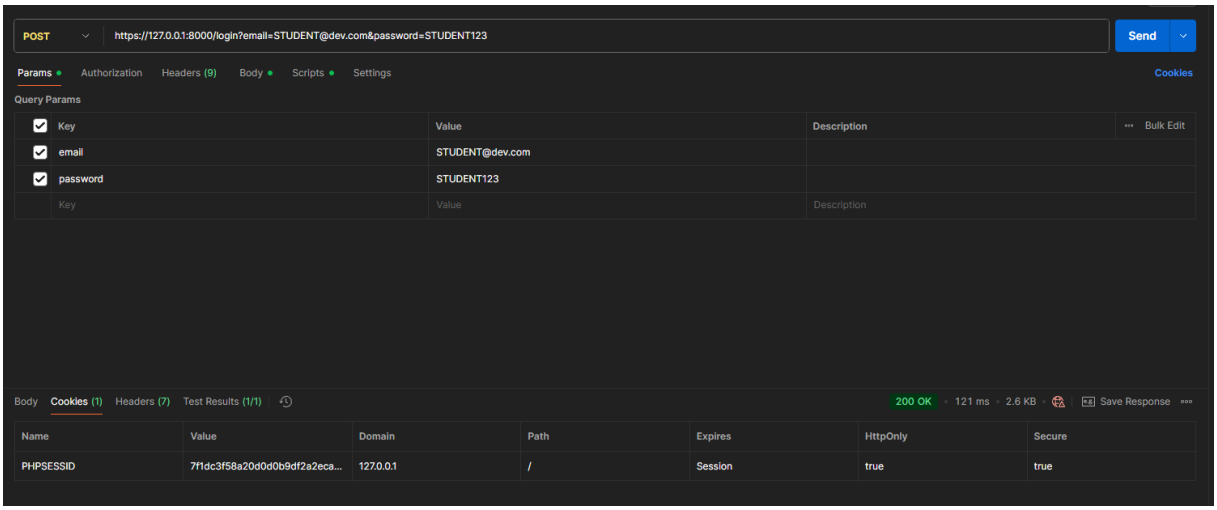
Test d'accès à la page d'accueil (Méthode GET) , validé avec un code « 200 OK ».

B. Back-end

ROUTES BACK :

URL	Title	Controlleur
/admin	Panel administrateur	AdminController
/formations	Formations	ProductsController
/formations/{id}	Détails de la formation	ProductsController
/cart	Panier	CartController
/login	Connexion	SecurityController
/register	Inscription	SecurityController
/student	Élève	StudentController
/logout	Déconnexion	SecurityController

Test de connexion à un compte depuis la page /login (Méthode POST), en utilisant un compté déjà créer, retour en 200 OK.



VI. Fonctionnement global

A. Lien entre le back-end et le front-end.

Le front-end est l'aspect visuel qu'un utilisateur va apercevoir sur son écran, par exemple un formulaire d'inscription, il va y entrer ses valeurs et appuyer sur un bouton « s'inscrire », ici interviens le back-end qui vas se charger d'enregistrer les informations dans la base de données si celles-ci ne sont pas déjà utilisées.

De retour sur le Front-end, le même utilisateur veux se connecter depuis le formulaire de connexion, il va y entrer son adresse e-mail et son mot de passe et appuyer sur « se connecter », une fois le bouton pressé, le back-end interviens afin de valider ou non la connexion selon la validité des données fournis.

Pour résumé : le front-end (côté client) envoie ou demande des données au back-end(côté serveur), et le back-end des données au front-end

B. Prise en compte des RGPD

Pour respecter la loi RGPD, un système de cookies à étais mis en place, permettant d'accepter ou de refuser la collecte des données essentiels au site, tel que : nom, prénom…

C. Gestionnaire de version

Afin d'avoir un gestionnaire de version pour le projet, j'ai créé un dépôt github, sur lequel j'ajoute grâce à Git, des mises à jour du projet avec des commentaires en anglais sur ce qu'il y a étais modifié.

J'écris en anglais, car c'est une langue internationale donc plus facile à comprendre pour les autres pays.

Exemple : [+] Adding new features for student (Settings)

Dépôt

D. Processus de déploiement

Pour déployer ce projet, il suffit d'installer :

- Visual Studio Code
- Composer
- Symfony
- Wampp/Xampp

Une fois installer, nous nous rendons dans le projet et ouvrons notre terminal afin d'exécuter la commande :

« Composer install », qui permet d'installer toutes les dépendances du projet.

Une fois réaliser, nous allons allumer Wampp ou Xampp, afin de nous rendre sur notre PhpMyAdmin, créer une nouvelle base de données au nom de « knowledge » et l'importer grâce à la fonction présente sur PhpMyAdmin.

Une fois importer, nous ouvrons un invité de commande dans la racine du projet Knowledge, puis nous exécutons : « symfony server :start ».

Et voilà, le projet est déployé en local !

E. Failles de sécurité majeures.

Les failles de sécurité majeures présente sur ce type de projet sont les injections SQL, le manque de token etc...

Pour remédier à ce type de problème j'ai mis en place plusieurs dispositif :

A. CRSF TOKEN.

Le fonctionnement d'un CRSF Token est de faire en sorte que l'action est bien d'un utilisateur autorisé à effectuer cette action, et non pas un site ou application externe.

Par exemple : les formulaires login et register sont tous les deux équipé d'un CRSF TOKEN afin d'être sûr que l'action ne viens pas d'autres choses que du formulaire.

```
<p class="title-form">Se connecter</p>
    <label for="inputEmail">Email</label> <br>
    <input
      type="email"
      value="{{ last_username }}"
      name="email"
      id="inputEmail"
      class="form-control"
      autocomplete="email"
      required
      autofocus
      placeholder="email"
    />
    <br />
    <label for="inputPassword">Mot de passe</label> <br>
    <input
    placeholder="mot de passe"
      type="password"
      name="password"
      id="inputPassword"
      class="form-control"
      autocomplete="current-password"
      required
    />

    <input
      type="hidden"
      name="_csrf_token"
      value="{{ csrf_token('authenticate') }}"
    />
```

À la fin de ce formulaire de connexion, il y a un input en hidden (donc caché) avec le crsf_token qui permet de valider que la connexion provient bien de ce formulaire et pas d'ailleurs.

B. Protections déjà mise en place par Symfony.

- Protection XSS (via Twig)
- Contrôle des accès et des rôles
- Hashage sécurisé des mots de passe
- Pare-feu configurable
- Protection contre les injections SQL (Doctrine)
- Validation stricte des données
- Limitation des tentatives (rate limiter)
- Sécurisation des sessions

Source : https://symfony.com/doc/current/security.html

Conclusion

En conclusion, ce projet m'a permis d'exploiter entièrement mes idées, et également de me donner et trouver les moyens de réussir les idées auxquelles j'avais pensé.

Il m'a aussi permis de réaliser que ce n'est pas parce que nous sommes seuls sur un projet qui nous parait gros que ce n'est pas possible.

Même si ce projet m'a donné beaucoup de difficulté certaines fois, il m'a permis d'évoluer sur mes compétences en termes de développement, mais également personnel.

J'ai également pris conscience du nombre de forums présents sur Internet pour remédier à un problème, et cela m'a permis également de comprendre qu'il fallait lire davantage certaines documentations.

Annexe

Le repository du projet est disponible à cette adresse :
https://github.com/Nylunn/KnowLedge

Voici des identifiants vous permettant de profiter pleinement de cette expérience.

Admin :

ADMIN@dev.com - ADMIN123

Etudiant :

Student@dev.com - STUDENT123