

# DOSSIER DE PROJET

Graziella FOURNEL

Développeur Web et Web mobile



Application Eco Web



# Sommaire

<b>1.Compétences du référentiel couvertes par le projet .....</b>	<b>p.03</b>
a. Développer la partie frontend d'une application web ou web mobile en intégrant les recommandations de sécurité	
b. Développer la partie backend d'une application web ou web mobile en intégrant les recommandations de sécurité	
<b>2.Résumé du projet .....</b>	<b>p.04</b>
<b>3.Le cahier des charges.....</b>	<b>p.05</b>
a. Contexte	
b. Définition du besoin	
c. Objectif du projet	
d. Descriptif de toutes les fonctionnalités attendues	
e. délais	
<b>4.Spécification technique .....</b>	<b>p.06</b>
<b>5.Réalisation du projet.....</b>	<b>p.07</b>
a. Préparation du projet (Trello, GitHub) .....	p.07
b. Persona. ....	p.08
c. Récits utilisateurs.....	p.09
d. Diagrammes de cas d'utilisation .....	p.09
e. Diagrammes de séquences .....	p.10
f. Charte graphique.....	p.10
g. Maquettage.....	p.10
h. Création de la base de données (méthode merise) .....	p.11
i. initialisation du projet.....	p.14
j. Développement de la base de données.....	p.14
k. Création des Controller .....	p.15
l. Création des formulaires.....	p.16
m. La vue avec TWIG .....	p.17
m. Javascript.....	p.18
<b>6.Fonctionnalité la plus représentative : L'authentification .....</b>	<b>p.19</b>
<b>7.Veille sur les vulnérabilités de sécurité .....</b>	<b>p.26</b>
<b>8.Recherche Anglophone .....</b>	<b>p.26</b>
<b>9.Traduction d'un extrait anglophone .....</b>	<b>p.27</b>
<b>10.Conclusion.....</b>	<b>p.27</b>
<b>Annexe .....</b>	<b>p.28</b>

# 1-Compétences du référentiel couverte par le projet

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

- ➔ Maquetter une application.
- ➔ Réaliser une interface utilisateur web statique et adaptable.
- ➔ Développer une interface utilisateur web dynamique.

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

- ➔ Développer les composants d'accès aux données.
- ➔ Développer la partie back-end d'une application web ou web mobile.
- ➔ Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce.

## 2-Résumer du projet

Dans le cadre de ma formation développeur Web-Web mobile, et de mettre en pratique mes connaissances, je dois créer une plateforme de formation sous la forme d'une application web du 14 mars 2022 au 21 avril 2022. Ce projet a pour but de concevoir toutes les étapes de conception d'une application web, l'analyse des besoins, maquettage, intégration, développement et déploiement.

ECOIT est un centre de formation qui a pour objectifs d'être une plateforme d'éducation permettant à tout instructeur expert en accessibilité est en éco-conception web de présenter des modules de cours.

Les formateurs créer leur formation sous forme de plusieurs module composé de leçon, de vidéo et de ressources externe. Chaque apprenant inscrit a la possibilité de suivre une formation.

Afin de mener à bien ce projet, j'ai généré le maquettage sur le logiciel Adobe XD (Charte graphique, Wireframe et Mockup). J'ai développé la partie front avec HTML5 et le Framework Bootstrap. La partie back et la mise en place de la base de données relationnelle est conçue avec le Framework Symfony 6 qui a contribué de réaliser une interface utilisateurs MCV (modèle, vue, contrôleur). La plateforme d'hébergement Heroku a permis le déploiement de l'application.

J'ai développé ce projet en autonomie et avec l'aide de documentation, de forum et de vidéos dédiée, cela m'a donné l'occasion d'appliquer et d'approfondir mes connaissances sur le Framework Symfony.

## 3-Cahier des charges

### a. Contexte

Ce projet est effectué en autonomie dans le cadre d'une évaluation en cours de formation. Pour concevoir cette application web je m'aide des différentes documentations officielles ainsi que de l'aide d'autres apprenants par le biais de réseaux sociaux et de plateforme de discussion. Je m'inspire également de vidéos explicatives.

J'ai principalement travaillé de mon domicile depuis un ordinateur portable qui me permet d'être mobile et de pouvoir concevoir cette application de partout. Je possède également un écran supplémentaire qui me permet de voir en temps réel l'évolution de l'application sans devoir passer d'une fenêtre à l'autre.

Mon rôle dans ce projet est de concevoir l'application web en passant par toutes les étapes de conception, analyse des besoins, maquettage, intégration, développement.

### b. Définition du besoin

La crise écologique est devenue importante, le changement climatique se fait ressentir. Aujourd'hui, internet en matière d'émissions de CO<sup>2</sup> pollue 1.5 fois plus que le transport aérien, selon Greenit en 20 ans le poids d'une page web a été multiplié par 115.

En 2017 un centre de formation ECOIT a été fondée, aujourd'hui cet organisme souhaite développer une plateforme de formation permettant aux instructeurs experts en accessibilité et en éco-conception de présenter des modules de cours.

### c. Objectif du projet

L'application permet à toute personne souhaitant réduire leur impact digital sur l'environnement de partager leurs connaissances en proposant des cours accessibles à tout apprenant inscrit.

ECOIT souhaite devenir la référence française pour les développeurs soucieux de leur impact digital.

### d. Descriptif de toutes les fonctionnalités attendues

De plus un catalogue devra être créé afin de répartir les formations par grille, un filtre permettra aux apprenants connectés de classer les formations suivies ou non. Une barre de recherche sera accessible à tous.

Sur la page d'accueil sera présentée les 3 dernières formations créées.

Pour finir un quiz devra être proposé à la fin de chaque section d'une formation afin de vérifier les connaissances de l'apprenant.

Dans cette application, 3 rôles sont définis, (administrateur, instructeur, apprenant.)

Il ne peut y avoir qu'un unique administrateur, et seul ce dernier peut valider un instructeur qui fera la demande par le biais d'un formulaire présent sur une des pages web de l'application accessible à tout visiteur.

Tous les visiteurs pourront créer un compte apprenant qui leur permet de se connecter afin de suivre une formation qui aura été créée par un instructeur.

### e. Délais

## 4-Spécification technique

Voici les différentes technologies que j'ai utilisé pour ce projet :

### **BOUTSTRAP :**

Bootstrap est donc un framework proposant une bibliothèque d'outils HTML, CSS et JavaScript qui vont nous aider à concevoir nos applications web. Bootstrap est évidemment pensé pour le responsive web design (visualisation sur les différents supports : PC, tablette, smartphone). C'est l'un des frameworks HTML / CSS / JavaScript les plus utilisés au monde

### **GITHUB :**

Git est un système de gestion de version distribué, Sa base de données locale et son système de gestion des instantanés le rendent très performant.

Son système de branches et les trois états de fichiers lui permettent d'être robuste lorsqu'il s'agit de travailler sur un projet, que ce soit seul ou à plusieurs.

### **TRELLO :**

Trello est un outil de gestion de projet en ligne inspiré par la méthode Kanban de Toyota. Il repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement.

### **ADOBE XD :**

Adobe XD est un outil de conception vectorielle pour les applications Web et mobiles, développé et publié par Adobe Inc.. J'ai créé les différents documents de maquette avec ce logiciel.

### **PHPMYADMIN :**

Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL sur un serveur PHP. De nombreux hébergeurs, gratuits comme payants, le proposent ce qui évite à l'utilisateur d'avoir à l'installer, ce logiciel m'a permis de vérifier mes données tout au long de la création de cette application.

### **COMPOSER :**

Composer est un gestionnaire de dépendance pour PHP. Il permet d'ajouter et de gérer des dépendances. C'est-à-dire des librairies externes, déjà codées par d'autres développeurs et que tu peux intégrer à mon projet.

### **TWIG :**

Twig est un moteur de template, on le retrouve dans le Framework Symfony, mais il peut être utilisé sur la majorité des Framework ainsi que dans un environnement PHP . Il permet la séparation de la présentation des données, permet de rendre les pages plus lisibles et plus claires, il est rapide.

### **SYMFONY 6 :**

Framework PHP, qui permet un développement rapide qui se base sur le pattern MVC (Modèle Vue Contrôleur). Il permet de développer en ayant une structure de base et un cadre de travail qui définit les bonnes pratiques à mettre en place. Symfony met en place des mesures de sécurité par exemple pour les injections SQL.

# 5-Réalisation du projet

## a. Préparation du projet (Trello, GITHUB)

### TRELLO

Afin de gérer au mieux mon temps et les différentes étapes de conception, j'e créé un tableau sur un outil de gestion visuel Trello.

Je le crée suivant le modèle KABAN qui permet une visualisation des tâches et de leurs avancements. Cette méthode se présente souvent en 3 colonnes « à faire », « en cours » et « fait ». Il suffit de placer l'action à effectuer sous forme d'étiquette et ensuite de la déplacer dans la colonne correspond à l'état de la tâche.

J'y rajoute plusieurs colonnes notamment la colonne « Product backlog » et j'y insère au fur et a mesure des tâches à accomplir. Un extrait du tableau est en annexe 1.

### GITHUB

Git est un DVCS (Distriburted Version Control Sytème), Il peut être utilisé aussi bien en local qu'avec un serveur distant, sa base de données est décentralisé. J'ai débuté par générer un répertoire sur GIT HUB puis j'ai créé un dossier sur mon ordinateur du même nom. Pour permettre le bon fonctionnement du dépôts, je l'ai initialisé avec la commande git init . Pour rétablir la connexion avec le répertoire distant j'ai exécuté la commande git add ce qui a permis d'ajouter les fichiers dans la base j'ai utilisé la commande git commit plus un message qui permet de faire comprendre rapidement les modifications au différent collaborateur qui pourra les valider en utilisant la commande git push.

```
fourn@LAPTOP-T5G9R9S0 MINGW64 /c/xampp/apps/ecfecoit (main)
$ git init
Reinitialized existing Git repository in C:/xampp/apps/ecfecoit/.git/

fourn@LAPTOP-T5G9R9S0 MINGW64 /c/xampp/apps/ecfecoit (main)
$ git remote add origin https://github.com/Graziella38690/ecfecoit.git

fourn@LAPTOP-T5G9R9S0 MINGW64 /c/xampp/apps/ecfecoit (main)
$ git add .

fourn@LAPTOP-T5G9R9S0 MINGW64 /c/xampp/apps/ecfecoit (main)
$ git comit -m "initial comit"
git: 'comit' is not a git command. See 'git --help'.

The most similar command is
    commit

fourn@LAPTOP-T5G9R9S0 MINGW64 /c/xampp/apps/ecfecoit (main)
$ git commit -m "initial commit"
[main 1b3f319] initial commit
1 file changed, 2 insertions(+)
create mode 100644 readme.md

fourn@LAPTOP-T5G9R9S0 MINGW64 /c/xampp/apps/ecfecoit (main)
$ git push -u origin main
Enumerating objects: 51, done.
Counting objects: 100% (51/51), done.
Delta compression using up to 8 threads
Compressing objects: 100% (46/46), done.
Writing objects: 100% (51/51), 46.35 KiB | 2.90 MiB/s, done.
Total 51 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Graziella38690/ecfecoit.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

*Capture console*

## b. Persona

La première étape de la réalisation du projet consiste à réaliser des persona qui sont des portrait-robot d'un être humains représentatif d'une catégorie ou d'un groupe. Ces persona permettent d'évaluer les attentes, besoins et ressentis des futurs utilisateurs, c'est un outil nécessaire à la réalisation des récits utilisateurs. (D'autre fiche persona sont disponible en Annexe 1 p.28)

Fiche persona n°1	
Administrateur	
	Nom: Arnaud Age: 41 ans Profession: Développer full stack
	<u>Maturité numérique</u> Arnaud est très à l'aise avec l'outil informatique, il collabore avec beaucoup d'entreprise, il est à l'origine de nombreuses applications sur le web.
<u>Caractère</u> Responsable Organisé Rigoureux Colérique	<u>Langage et logiciel informatique:</u> HTML 5, CSS 3, Javascript, PHP, MySQL : Niveau Expert Symfony & Wordpress: Niveau Avancée Suite Adobe : Niveau Intermédiaire
<u>Biographie:</u> Sorti du système scolaire à l'âge de 16 ans, il a enchaîné plusieurs jobs. À 21 ans une grande entreprise lui propose de le former au numérique, depuis il ne cesse d'apprendre et de réaliser des applications.	
<u>Intérêt pour l'écologie</u> Acteur majeur pour l'écologie, il évite de prendre sa voiture pour privilégier les transports en commun, utilise son vélo pour de courts trajets, et participe activement au tri sélectif de sa ville. Il assiste à de nombreuses conférences sur l'éco-conception.	
<u>Attente de l'application :</u> <ul style="list-style-type: none"><li>- Il souhaite que l'application lui permette d'être le seul à créer les comptes instructeurs.</li><li>- Il souhaite recevoir les demandes des visiteurs pour devenir instructeurs.</li><li>- Il souhaite pouvoir se connecter à l'application.</li></ul>	

Fiche persona



### c. Récits utilisateurs (User stories)

Le point de départ des tâches de développement sont les Users story, ce sont des explications générales, non formelle, d'une fonctionnalité logicielle. Elle est écrite du point de vue de l'utilisateur final. Elle contribue à fournir un système de référence pour le travail quotidien. Elle s'écrit suivant un format strict. Qui ? Quoi ? Pourquoi ?

Ex : Je suis Arthur étudiant, je souhaite répondre à un quizz pour évaluer mes connaissances.

↓  
Qui ?

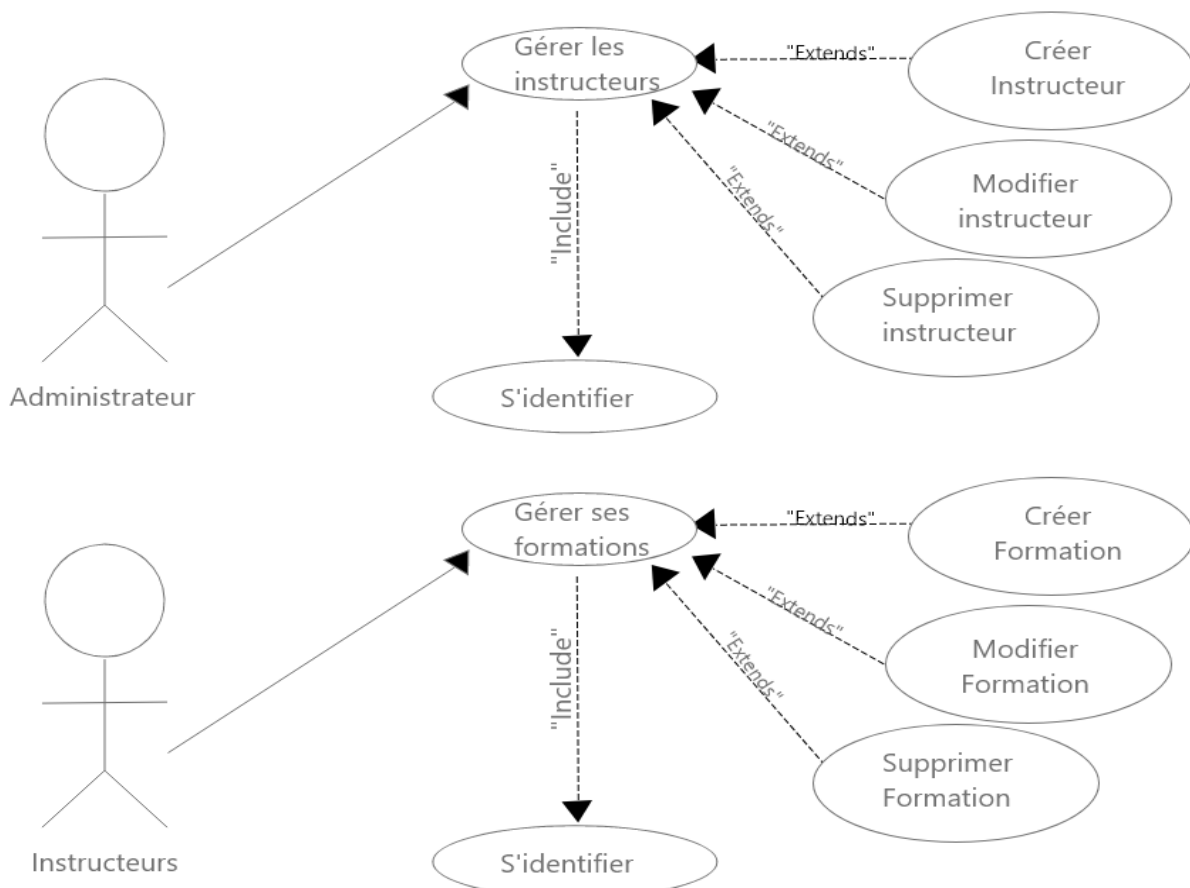
↓  
Quoi ?

↓  
Pourquoi ?

### d. Diagrammes de cas d'utilisation

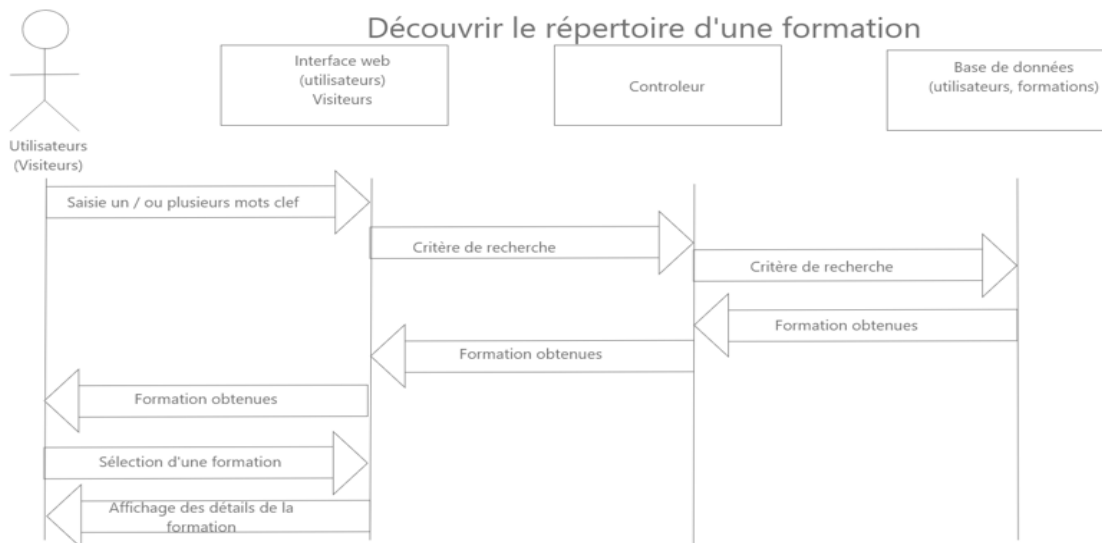
je réalise ensuite des diagrammes de cas d'utilisation, afin de représenter le comportement d'un système, cela permet de dresser une liste des fonctionnalités de l'application, de gérer les accès selon les différents utilisateurs. Il est composé de trois éléments principaux : Des acteurs, des cas d'utilisation et les relations entre eux.

Dans l'exemple ci-dessous l'acteur est l'administrateur, il peut gérer les instructeurs (le cas d'utilisation) mais il doit être connecté pour le faire, la fonctionnalité « s'identifier » est donc forte et doit être réalisé avant de pouvoir gérer les instructeurs.



### e. Diagrammes de séquences

Je conçois ensuite des diagrammes de séquence qui permettent d'illustrer toutes les interactions entre les différents acteurs ce qui permet de visualiser en détails toutes les communications effectuées pendant la fonctionnalité dans un ordre précis.



Dans cet exemple la fonctionnalité est « Découvrir le répertoire d'une formation », l'acteur est un utilisateur qui n'est pas inscrit, donc pas connecté. Ce visiteur a la possibilité de rechercher une formation par mots clefs dans une barre de recherche, il entre ces critères, mots clefs ect... Le Contrôleur envoie donc la requête à la base de données qui recherche et renvoie ensuite au contrôleur les résultats obtenus qui les renvoie à son tour à l'interface. A ce moment l'utilisateur pourra choisir plusieurs options. Dans ce diagramme cette fonction n'est pas représentée en détails.

### f. Charte Graphiques

Avant de réaliser le maquetage du projet, j'ai conçu une charte graphique à partir d'éléments déjà en ma possession. Ce document permet de visualiser et de définir l'identité visuelle de l'entreprise, l'association concernée par le projet. Ce sont les règles à respecter pour la conception du produit. On n'y retrouve, les jeux de couleurs, le logo, les polices d'écriture à respecter. On n'y retrouve aussi des images, photos inspirantes pour la conception. La charte graphique est disponible en Annexe 2 p.29

### g. Maquettage

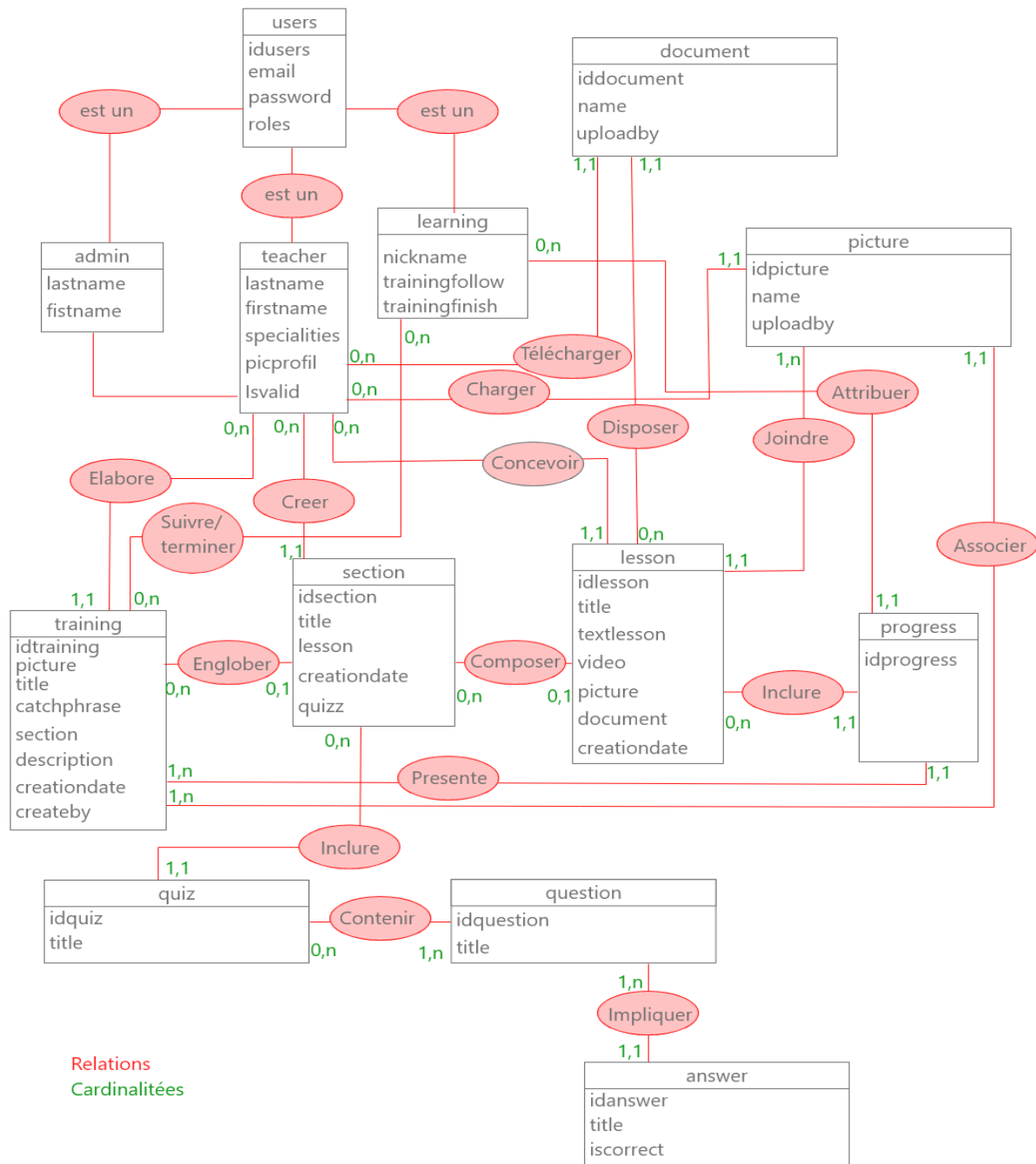
Une fois la charte graphique en main, je commence l'étape du maquetage, qui consiste à réaliser plusieurs documents graphiques à l'aide du logiciel Adobe XD. Le zoning, les wireframes puis les Mockup. Le zoning est un document peu détaillé qui représente les différentes zones d'une page, elle est composée de formes simples en noir, blanc et gris. Je ne réalise pas cette étape volontairement pour gagner du temps et je passe directement à la conception des wireframes qui comme le zoning est en noir et blanc mais représente plus en détails les différents éléments d'une page, comme les boutons de navigation, les proportions des différents blocs, des textes ect puis je termine avec la réalisation des Mockup qui représente le plus fidèlement possible ce que sera le produit final on n'y ajoute la couleur, la police d'écriture en respectant la charte graphique. Différents exemples sont présents en annexe 3 p30 et les différents rendus sont disponibles en annexe 4 p 31.

## h. Conception de la base de données (méthode merise)

Pour concevoir la base de données, je rassemble toutes les données que j'ai besoin de stocker, je me pose la question de quelles entités j'ai besoin. Par exemple : Dans cette application il y a des formation (training) qui sont caractérisé par un id, une image, un titre ...

Je commence par établir le Modèle Conceptuel de Données (MCD). Cette étape consiste à établir des relations entre les différentes entités avec des cardinalités qui exprime le nombre minimum et maximum de fois où l'entité participe à la relation.

Une entité doit contenir une ou plusieurs propriétés et un identifiant.

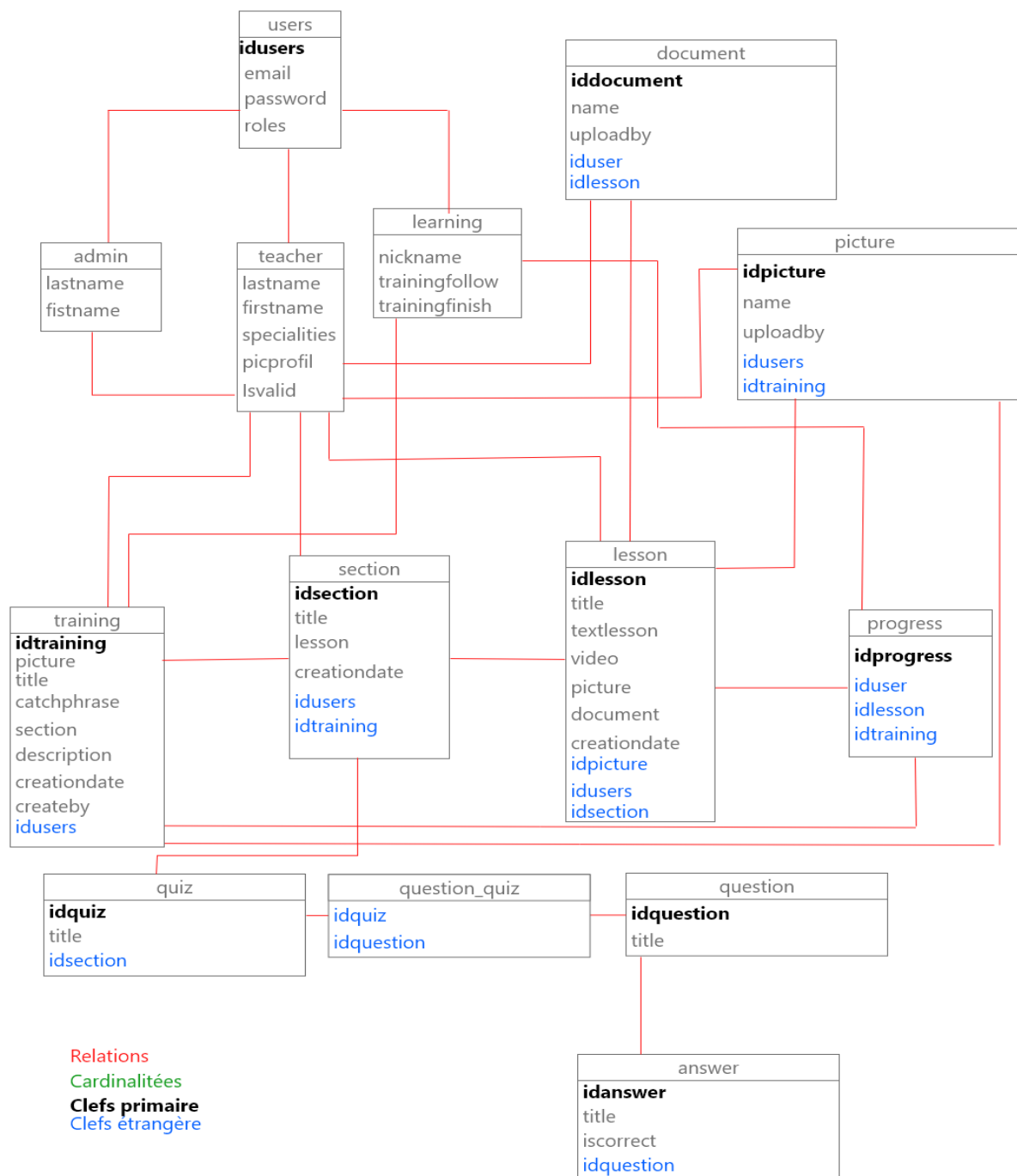


Modèle conceptuel de données

Le modèle conceptuel de donnée terminé, je passe à l'élaboration du Modèle Logique de Données (MLD). Chaque entité devient une table et leur identifiant devient la clé primaire de la table. Dans ce modèle les associations sont transformé selon le type des relations. Une entité est faible si sa cardinalité maximale dans une relation est de 0 ou 1 et une entité est forte si sa cardinalité maximale est de n.

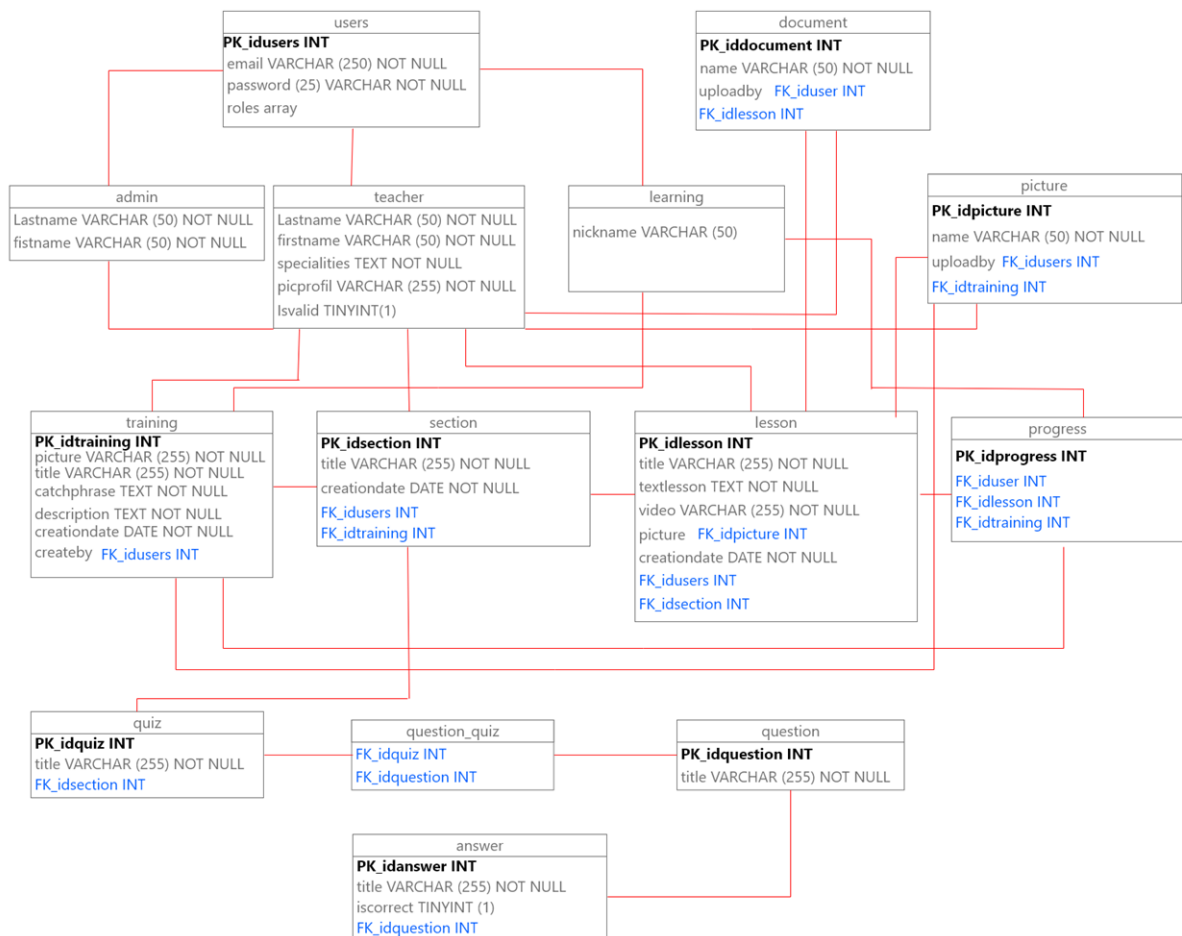
L'entité faible copie l'identifiant de l'entité forte est devient une clé étrangère qui est représentés par un #.

Si deux entités sont forte une nouvelle entité sera créer avec l'identifiant des deux entités fortes.



## Modèle Logique de données

Ensuite je réalise le Modèle Physique de Données (MPD), les entités vont devenir des tables, toutes les propriétés vont obtenir un type de données (varchar, int, date ect...). Les identifiants vont devenir des clé primaires (PK) ou des clés étrangères (FK).

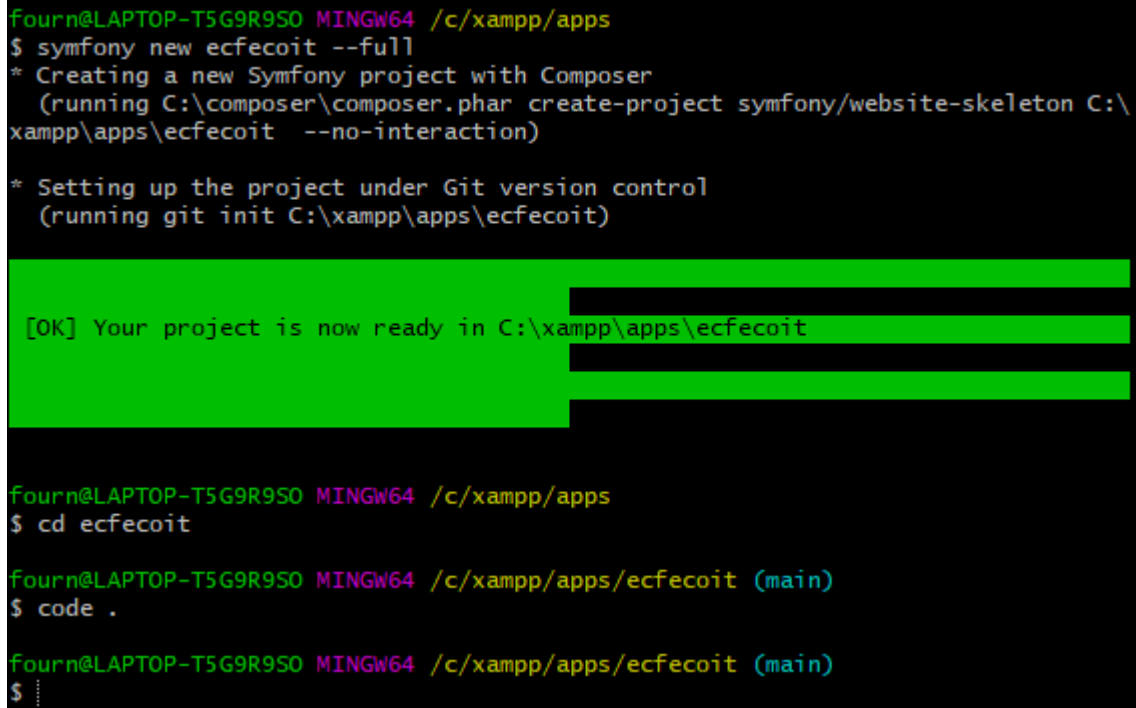


Modèle Physique de Données

### i. Initialisation du projet

Je choisis de développer cette application avec Symfony 6, en initialisant un nouveau projet Avec la commande :

```
Symfony new ecfecoit --full
```



```
fourn@LAPTOP-T5G9R9SO MINGW64 /c/xampp/apps
$ symfony new ecfecoit --full
* Creating a new Symfony project with Composer
  (running C:\composer\composer.phar create-project symfony/website-skeleton C:\
xampp\apps\ecfecoit --no-interaction)

* Setting up the project under Git version control
  (running git init C:\xampp\apps\ecfecoit)

[OK] Your project is now ready in C:\xampp\apps\ecfecoit

fourn@LAPTOP-T5G9R9SO MINGW64 /c/xampp/apps
$ cd ecfecoit

fourn@LAPTOP-T5G9R9SO MINGW64 /c/xampp/apps/ecfecoit (main)
$ code .

fourn@LAPTOP-T5G9R9SO MINGW64 /c/xampp/apps/ecfecoit (main)
$
```

Cette commande permet de créer un dossier qui contiendra toutes l'arborescence et tous les fichiers de Symfony. Quand la commande d'exécute on peut voir l'indication :

```
(running C : / Composer/Composer.phar create-projet symfony/website-
skeleton C : /xampp/apps/ecfecoit -no-interaction)
```

Cela signifie que composer est déjà installé sur mon ordinateur et que c'est le paquet « symfony/website-skeleton » qui est utilisé.

### j. Développement de la base de données

Je commence par créer la base de données en effectuant la commande :

```
php bin/console make:entity
```

Symfony me demande ensuite de configurer les différents attributs qui corresponde aux champs, ainsi que les restrictions. Avec Symfony la création de tables est simplifiée car elle génère automatiquement un fichier. PHP ainsi qu'un repository. Pour terminer la base de données et la créer j'exécute dans l'ordre les commandes suivantes :

```
php bin/console doctrine:database:create
php bin/console make:migration
php bin/console doctrine:migration:migrate
```

### k. Création des Controller

Pour éviter de définir mes itinéraires dans le YAML, mais dans le Controller directement j'ai installé le package d'annotations avec la commande :

```
Composer require annotations
```

```
class AdminController extends AbstractController
{
    #[Route('/admin', name: 'app_admin')]
}
```

Pour mettre en place les différents contrôleurs j'exécute la commande :

```
Php bin/console make:Controller
```

Symfony me demande alors des informations complémentaires afin de créer la base du controller.

Cette qui permet de créer les fichier HomeController.php et index.html.twig.

Je modifie ensuite le fichier HomeController pour qu'il corresponde à mes besoins.

```
1  <?php
2
3  namespace App\Controller;
4  use App\Entity\User;
5  use App\Repository\TrainingRepository;
6  use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
7  use Symfony\Component\HttpFoundation\Response;
8  use Symfony\Component\Routing\Annotation\Route;
9
10 class HomeController extends AbstractController
11 {
12     #[Route('/index', name: 'app_home')]
13     public function index(TrainingRepository $TrainingRepository): Response
14     {
15         $this->getUser();
16         return $this->render('home/index.html.twig', [
17             'Trainings' => $TrainingRepository->findLastTraining(),
18         ]);
19     }
20 }
```

Dans ce fichier on peut voir que la classe HomeController herite de la classe Abstract Controller, ensuite nous avons une méthode index() qui retourne un object Response qui est défini par symfony. La fonction index () est décoré d'une annotation #[route('/index', name : 'app\_home')], cette annotation permet de définir une route /index qui est rattaché au controleur index.

J'ai donc procédé plusieurs cette commande afin de créer tous les Controller nécessaires.

## I. Création des formulaires

Pour gérer les données dans la base de données, créer ou supprimer des données j'ai créé des formulaires avec la commande :

```
php bin/console make:form
```

Symfony me demande alors quel nom de classes je souhaite pour ce formulaire, par convention il doit terminer par « Type » je le renomme donc « TrainingType ». Ensuite je dois préciser quelle entité on veut en référence pour le formulaire, je lui indique donc « Training », une classe « TrainingType » vient de se créer dans le dossier form.

```
1  <?php
2
3  namespace App\Form;
4
5  use App\Entity\section;
6  use App\Entity\Training;
7  use Symfony\Component\Form\AbstractType;
8  use Symfony\Component\Form\FormBuilderInterface;
9  use Symfony\Bridge\Doctrine\Form\Type\EntityType;
10 use Symfony\Component\OptionsResolver\OptionsResolver;
11 use Symfony\Component\Form\Extension\Core\Type\SubmitType;
12 use Symfony\Component\Form\Extension\Core\Type\CollectionType;
13
14
15
16 class TrainingType extends AbstractType
17 {
18     public function buildForm(FormBuilderInterface $builder, array $options): void
19     {
20         $builder
21             ->add('Title')
22             ->add('catchphrase')
23             ->add('Description')
24
25             ->add('Sections', CollectionType::class, array(
26                 'entry_type' => SectionType::class,
27                 'by_reference' => false,
28                 'allow_add' => true,
29                 'allow_delete' => true,
30                 'prototype' => true,
31             ))
32             ->add('valider', SubmitType::class);
33     };
34 }
35
36 public function configureOptions(OptionsResolver $resolver): void
37 {
38     $resolver->setDefaults([
39         'data_class' => Training::class,
40     ]);
41 }
```

Dans ce formulaire on peut voir la fonction configureOption , cette fonction permet de configurer les options du formulaire. C'est ici que l'on y indique à quelle entité est rattaché à ce formulaire. Il est bien rattaché à l'entité « Training »



### m. La vue avec TWIG

Ce moteur de Template me permet d'afficher les résultats d'une requête, j'ai intégré dans la base mes différents bloc que je voulais retrouver sur l'ensemble de page de l'applications.

INSI

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title>{% block title %}Welcome!{% endblock %}</title>
6
7          {# Run `composer require symfony/webpack-encore-bundle` to start using Symfony UX #}
8          {% block stylesheets %}
9
10
11
12          {{ encore_entry_link_tags('app') }}
13          {% endblock %}
14
15          {% block javascripts %}
16              {{ encore_entry_script_tags('app') }}
17          {% endblock %}
18
19      </head>
20      <body>
21
22
23      {{ include('navbar.html.twig') }}
24      <div class="container text-center">
25          </img>
26      </div>
27
28
29
30          {% block body %}
31          {% endblock %}
32      </body>
33  </html>
```

Si le fichier "base.html" contient l'en-tête et le pied de page HTML ainsi que la barre de navigation communes à toutes les pages, ces dernières peuvent en hériter avec le mot extends.

## n. Javascript

Afin de rendre plus dynamique et attractive la création d'une nouvelle formation, j'ai décidé de créer un sous formulaire dans le formulaire de création de formation. Ce script va permettre d'ajouter un nouveau bouton pour ajouter un sous formulaire.

Tout d'abord je vais modifier le fichier Template de création de formation.

Je crée d'abord une nouvelle ligne et j'y inclue modèle de formulaire pour l'ajout de section. Puis j'ajoute un exemplaire du formulaire.

```
<div id="Sections" data-prototype="{{form_row(form.Sections.vars.prototype)|e('html_attr')}}">

{{ form_row(form.Sections) }}
<span></span>
</div>
{{ form_end(form) }}
```

### Le script pour ajouter un bouton

```
let collection, boutonAjout, span;
window.onload = () => {
    collection = document.querySelector("#Sections");
    span = collection.querySelector("span");
    boutonAjout = document.createElement("button");
    boutonAjout.className = "btn-primary";
    boutonAjout.innerText = "Ajouter une section";
    let nouveauBouton = span.append(boutonAjout);
    collection.dataset.index = collection.querySelectorAll("input").length;
    boutonAjout.addEventListener("click", function(){
        addButton(collection, nouveauBouton);
    });
}
```

Je commence par déclarer une collection, boutonAjout puis span, pour attendre que le DOM soit chargé je fais window.onload... Dans collection je vais aller chercher la collection de formulaire avec querySelector c'est-à-dire la div complète, puis je vais aller la span, puis je crée le bouton ajout, je lui ajoute des paramètres, comme sa classe, son innerText, ceci crée le bouton. Il faut maintenant l'intégrer à la span, en créant un nouveau bouton. On va maintenant lui mettre un listener mais il faut savoir combien on n'a input à l'intérieur de la collection, pour cela on va lui ajouter un data set ce qui va mettre la longueur du formulaire à l'intérieur du span, on ajoute un event listener et on va attendre le clic qui va lancer la fonction, on lui passe la collection et les boutons.

# 6- Fonctionnalité la plus représentative

## L'authentification

### Développement du code

L'utilisateur a besoin d'être connecté pour avoir accès à toutes les fonctionnalités de l'application. Il faut donc d'abord exécuter la commande afin de créer le système d'authentification :

```
Symfony console make:auth
```

Cette commande va lancer un assistant qui va nous demander de renseigner les informations suivantes :

Le type d'authentification (avec ou sans formulaire de connexion), le nom de la classe contenant l'authentification (UserAuthenticator), le nom du contrôleur qui contiendra les routes de connexion et déconnexion (SecurityController), et la création ou non d'une route de déconnexion (/logout)

Après l'exécution de la commande, les fichiers suivants auront été créés ou modifiés.

### Le fichier security.yaml

Cette partie va gérer le nom de notre authenticator, puis le paramètre du logout qui contient le nom de la route qui contient le logout.

```
provider: app_user_provider
custom_authenticator: App\Security\UserAuthenticator
logout:
    path: app_logout
    # where to redirect after logout
    # target: app_any_route

# activate different ways to authenticate
# https://symfony.com/doc/current/security.html#the-firewall

# https://symfony.com/doc/current/security/impersonating_user.html
# switch_user: true
```

## Le fichier Sécurité Controller

```
1  <?php
2
3  namespace App\Controller;
4
5  use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
6  use Symfony\Component\HttpFoundation\Response;
7  use Symfony\Component\Routing\Annotation\Route;
8  use Symfony\Component\Security\Http\Authentication\AuthenticationUtils;
9
10 class SecurityController extends AbstractController
11 {
12     #[Route(path: '/login', name: 'app_login')]
13     public function login(AuthenticationUtils $authenticationUtils): Response
14     {
15         // if ($this->getUser()) {
16         //     return $this->redirectToRoute('target_path');
17         // }
18
19         // get the login error if there is one
20         $error = $authenticationUtils->getLastAuthenticationError();
21         // last username entered by the user
22         $lastUsername = $authenticationUtils->getLastUsername();
23
24         return $this->render('security/login.html.twig', ['last_username' => $lastUsername, 'error' => $error]);
25     }
26 }
```

Puis un Controller est créé le SecurityController qui contient une route app\_login et qui a le chemin /login. C'est une méthode qui s'appelle login, on n'a éventuellement une condition qui permet de rediriger l'utilisateur s'il est déjà connecté

On peut aussi avoir un message d'erreur, on peut le récupérer avec la méthode

« getLastAuthenticationError » puis on récupère le dernier nom utilisé par l'utilisateur, puis on charge le twig.

## Le fichier UserAuthenticator

```
class UserAuthenticator extends AbstractLoginFormAuthenticator
{
    use TargetPathTrait;

    public const LOGIN_ROUTE = 'app_login';

    private UrlGeneratorInterface $urlGenerator;

    public function __construct(UrlGeneratorInterface $urlGenerator)
    {
        $this->urlGenerator = $urlGenerator;
    }
}
```

Puis dans Security le UsersAuthenticator.php , ce fichier gère l'authentification, il récupère un trait le TargetPathTrait qui permet d'importer à l'intérieur de l'authentificateur des méthodes. On n'y retrouve aussi UrlGeneratorInterface qui permet de générer des URL.

```
public function authenticate(Request $request): Passport
{
    $email = $request->request->get('email', '');

    $request->getSession()->set(Security::LAST_USERNAME, $email);

    return new Passport(
        new UserBadge($email),
        new PasswordCredentials($request->request->get('password', '')),
        [
            new CsrfTokenBadge('authenticate', $request->request->get('_csrf_token')),
        ]
    );
}
```

La methode authenticate qui retourne un « Passport » qui permet de gérer l'identification des utilisateurs, il permet de récupérer l'utilisateur, le mot de passe et la valeur du token CSRF qui permet d'authentifier le formulaire.

Tout d'abord on récupère l'email de l'utilisateur, dans la session on insert le dernier utilisateur qui a était taper et on retourne un « passport » , dans le constructeur du « passport » on récupère l'utilisateur par son email avec la methode UserBadge, on récupère le mot de passe créer avec le PasswordCredentials puis on récupère le token Csrf qui permet de vérifier que le formulaire viens bien de l'application .

## Le fichier login.html.twig

La commande à générer un Template de formulaire, ce formulaire est déjà fait car le marker bundle utilise des classes affectées à Bootstrap. La vue de ce formulaire est disponible en annexe p

```
{% if error %}
    <div class="alert alert-danger">{{ error.messageKey|trans(error.messageData, 'security') }}</div>
{% endif %}
```

Cette partit du code concerne les erreurs s'il y a un problème avec l'identification.

```
{% if app.user %}
    <div class="mb-3">
        You are logged in as {{ app.user.username }}, <a href="{{ path('app_logout') }}">Logout</a>
    </div>
{% endif %}
```

Cette partit du code indique si un utilisateur est déjà connecté

```
<h1 class="h3 mb-3 font-weight-normal">Please sign in</h1>
<label for="inputEmail">Email</label>
<input type="email" value="{{ last_username }}" name="email" id="inputEmail" class="form-control" autocomplete="email" required autofocus>
<label for="inputPassword">Password</label>
<input type="password" name="password" id="inputPassword" class="form-control" autocomplete="current-password" required>

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

Cette partit de code concerne les informations de connexion à taper dans le formulaire, puis le token

```
    <div class="checkbox mb-3">
        <label>
            <input type="checkbox" name="_remember_me"> Remember me
        </label>
    </div>

    <button class="btn btn-lg btn-primary" type="submit">
        Sign in
    </button>
```

Pour terminer, ici l'utilisateur peut choisir de sauvegarder ces identifiants de connexion puis le bouton pour se connecter.

A ce stade nous avons une erreur de connexion, en effet aucun utilisateur est inscrit dans la base de données il faut donc créer le système d'inscription avec la commande :

```
Symfony console make:registrationform
```

Cette commande va lancer un assistant qui va nous demander de renseigner les informations. Si on souhaite ajouter une annotation « @UniqueEntity » dans notre classe Users pour les rendre uniques, si on souhaite envoyer un e-mail à l'utilisateur pour activer leur compte, si on souhaite connecter l'utilisateur après son inscription.

Après l'exécution de la commande les fichiers suivants vont être créés ou modifiés.

### **Le fichier RegistrationController.php**

```
class RegistrationController extends AbstractController
{
    #[Route('/register', name: 'app_register')]
    public function register(Request $request, UserPasswordHasherInterface $userPasswordHasher, EntityManagerInterface $entityManager): Response
    {
        $user = new User();
        $form = $this->createForm(RegistrationFormType::class, $user);
        $form->handleRequest($request);
        $user->setRoles(['ROLE_LAERNING']);
    }
}
```

On récupère la requête, on crée un nouvel utilisateur, on crée le formulaire, on gère le formulaire, on assigne le rôle Learning à l'utilisateur.

```
if ($form->isSubmitted() && $form->isValid()) {
    // encode the plain password
    $user->setPassword(
        $userPasswordHasher->hashPassword(
            $user,
            $form->get('plainPassword')->getData()
        )
    );

    $entityManager->persist($user);
    $entityManager->flush();
    // do anything else you need here, like send an email
}
```

Si le formulaire est bon, on gère l'inscription, Pour des raisons de sécurité le mot de passe de l'utilisateur devra être hashé dans la base de données, on utilise donc le « userPasswordHasher »

### Le fichier Users.php

```
#[UniqueEntity(fields: ['email'], message: 'There is already an account with this email')]
```

Cette ligne est ajoutée à l'Entity Users, cela indique que si le champs Le mail a déjà été utilisé le message sera affiché et l'inscription ne pourra pas se faire.

### Le fichier RegistrationFormType.php

```
--
19 class RegistrationFormType extends AbstractType
20 {
21     public function buildForm(FormBuilderInterface $builder, array $options): void
22     {
23         $builder
24             ->add('Pseudo', TextType::class, [
25                 'attr' => [
26                     'class' => 'form-control'
27                 ],
28                 'label' => 'Pseudo'
29             ])
30
31             ->add('email', EmailType::class, [
32                 'attr' => [
33                     'class' => 'form-control'
34                 ],
35                 'label' => 'E-mail'
36             ])
```

### Le fichier navbar.html.twig

```
<a href="{{path('app_login')}}"><button type="button" class="btn btn-primary me-2">Connexion</button></a>
<a href="{{path('app_register')}}"><button type="button" class="btn btn-primary me-2">S'inscrire</button></a>
```

Pour finir j'ai inséré deux lignes dans la barre de navigations afin de pouvoir avoir accès à l'inscription et à la connexion.



### Essaie de l'authentification

Afin de tester si mon système d'authentification fonctionner j'ai essayé de me connecter avec un compte fictif. Afin de tester si on pouvait injecter du code dans mon formulaire je n'est pas mis de vrai adresse e-mail , l'interface m'envoie un message d'alerte. Il faut impérativement qu'il y est un @ dans ce champ. Je décide donc de mettre @ et là encore un message d'erreur s'affiche. Cette vérification est nécessaire afin d'éviter les failles XML.

The screenshot shows a login form titled "Me connecter". It has two input fields: "E-mail" containing "123456" and "Mot de passe" containing ".....". Below the password field is a checkbox labeled "Se souvenir de moi" which is unchecked. A green "Connection" button is at the bottom. A yellow warning message box is displayed, stating: "Insérez un symbole « @ » dans l'adresse e-mail. « 123456 » ne contient pas de « @ »."

The screenshot shows the same login form. The "E-mail" field now contains "123456@". A yellow warning message box is displayed, stating: "Entrer une partie après « @ ». « 123456@ » est incomplet."

The screenshot shows the login form with a red banner at the top that says "Identifiants invalides.". The "E-mail" field contains "123456@outlook.fr" and the "Mot de passe" field contains ".....". The "Se souvenir de moi" checkbox is now checked. The green "Connection" button is at the bottom.

Afin de tester la connexion avec de vrais informations je rentre les informations demandées correctement. Ayant aucun utilisateur d'inscrit l'interface me renvoie donc un autre message d'erreur.

Je me rends donc sur la page d'inscription, et j'entre les informations demandées. Une fois inscrit je me connecte de nouveau et je tombe sur la page d'accueil.

Afin de m'assurer qu'un utilisateur ne puis pas créer plusieurs comptes avec une même adresse E-mail je vérifie de nouveau l'inscription

The screenshot shows a registration form titled "S'inscrire". It has four input fields: "Pseudo" containing "123456", "E-mail" containing "123456@outlook.fr", and "Mot de passe" containing ".....". Below the password field is a checked checkbox labeled "En m'inscrivant à ce site j'accepte...". A green "M'inscrire" button is at the bottom.

The screenshot shows the registration form. The "Pseudo" field contains "123456" and the "E-mail" field contains "123456@outlook.fr". A red error message is displayed above the email field: "erreur: There is already an account with this email". The "Mot de passe" field is empty. The "En m'inscrivant à ce site j'accepte..." checkbox is checked. The green "M'inscrire" button is at the bottom.

## Vérification dans la base de données.

J'utilise php MyAdmin pour vérifier les informations envoyées on bien était inscrit dans la base de données.

	id	email	roles (DC2Type:json)	password	firstname	lastname	specialities	photo	pseudo	dateinscription (DC2Type:datetime_immutable)
r 41	123456	@outlook.fr	["ROLE_LAERNING"]	\$2y\$13\$eMUa7/Qn4hFlpEntFMR7heyoiErLY3yTyHVrde4nY1S...	NULL	NULL	NULL	NULL	123456	2022-11-13 17:46:55

Cette ligne correspond à l'utilisateur que je viens de créer, nous vérifions que l'adresse mail correspond, ainsi que le pseudo, le nom et prénom pourront être modifier par la suite si l'utilisateur souhaite le faire. Je vérifie aussi le rôle apprenant (ROLE\_LAERNING), et pour finir nous pouvons voir que le mots de passe a bien été hacher à l'aide de la fonction userPasswordHasher de Symfony. Cette fonction sera abordée dans ma recherche anglophone

## 7.Veille sur les vulnérabilités de sécurité

Afin de sécuriser au mieux mon application, j'effectue des recherches concernant les failles de sécurité possible, Symfony dispose de nombreux outils pour sécuriser une application ainsi que de nombreux paramètre de configuration afin de protéger l'application et les données saisie dans une base de données.

Je me rends donc dans la documentation de Symfony dans la partie sécurité, on peut se rendre compte que ce Framework est assez complet et propose beaucoup de solution concernant la sécurité au niveau des utilisateurs, de l'authentification, des pare-feux, des attributions des rôles, des limitations des tentatives de connexion ect ....

Site web consultées : <https://symfony.com/doc/current/security.html>

## 8.Recherche Anglophone

Toujours sur la documentation de Symfony, je me suis plus intéressée a la partie concernant le hachage des mots de passe.

### Registering the User: Hashing Passwords

Many applications require a user to log in with a password. For these applications, the SecurityBundle provides password hashing and verification functionality.

First, make sure your User class implements the PasswordAuthenticatedUserInterface:

Then, configure which password hasher should be used for this class. If your file wasn't already pre-configured, then should have done this for you:security.yamlmake:user

Now that Symfony knows how you want to hash the passwords, you can use the service to do this before saving your users to the database:UserPasswordHasherInterface

The maker command can help you set-up the registration controller and add features like email address verification using the SymfonyCastsVerifyEmailBundle.make:registration-form

```
$ composer require symfonycasts/verify-email-bundle
```

```
$ php bin/console make:registration-form
```

You can also manually hash a password by running:

```
$ php bin/console security:hash-password
```

Read more about all available hashers and password migration in Password Hashing and Verification.

## 9.Traduction d'un extrait anglophone

Enregistrement de l'utilisateur : hachage des mots de passe

Le SecurityBundle fournit la fonction du hachage et la vérification des mots de passe, pour qu'un utilisateur puisse se connecter à une application.

En premier il faut qu'on s'assure la classe User implémente l'interface Password AuthentucatedUserInterface.

Après nous devons configurer le hacheur de mot de passe à utiliser pour la class user, si le fichier n'est pas déjà préconfiguré on doit ajouter les lignes dans le security .yaml make :user.

Symfony sait comment on veut hacher les mots de passe, il nous reste plus qu'à utiliser le service avant d'enregistrer les utilisateurs dans la base de données : UserPasswordHasherInterface

On peut aussi utiliser la commande du maker pour configurer le contrôleur d'enregistrement et ajouter d'autre fonctionnalités tes que la vérification d'e-mail.


## 10.Conclusion

Concernant ce projet, toutes les fonctionnalités ne sont pas encore développées. Je travaille toujours sur ce projet afin de le rendre opérationnel le plus vite possible

Ce projet a pour moi était un défi, mes connaissances ne me permettait pas de réaliser ce projet dans un le temps imparti. Je me suis beaucoup aidé des forums, de la documentation officiels, des vidéos afin de rendre un projet de base fonctionnel.

# Annexe 1

Fiche persona n°2	
Instructeur	
	<b>Nom:</b> Céline <b>Age:</b> 28 ans <b>Profession:</b> Intégrateur web
	<p align="center"><u><b>Maturité numérique</b></u></p> <p>Céline est très à l'aise avec l'outil informatique, elle utilise énormément les outils de conception, graphisme, maquettage. Présente activement sur les réseaux sociaux.</p> <p align="center"><u><b>Langage et logiciel informatique:</b></u></p> <p>HTML 5, CSS 3, Javascript, Suite Adobe : Niveau Expert            PHP, My SQL : Niveau Intermédiaire            Wordpress: Niveau Avancée</p>
<p><u><b>Caractère</b></u></p> <p>Souriante            Organisée            Créative            Ambitieuse</p>	
<p><u><b>Biographie:</b></u></p> <p>Issue d'un parcours scolaire en numérique, ainsi que de plusieurs formations dans l'éco-conception. Elle travaille en free-lance et participe à de nombreux projets d'éco-conception. Elle souhaite devenir une actrice importante dans la diminution de l'impact digital.</p>	
<p><u><b>Intérêt pour l'écologie</b></u></p> <p>Elle pratique le zéro déchets, ainsi que le minimalisme. Auteur d'un blog reconnu ainsi que d'une chaîne YouTube traitant divers sujets de l'écologie, elle y présente aussi de nombreux tutoriels pour limiter la pollution.</p>	
<p><u><b>Attente de l'application :</b></u></p> <ul style="list-style-type: none"> <li>- Elle souhaite pouvoir se connecter à l'application.</li> <li>- Elle souhaite pouvoir créer ses formations sur l'application.</li> </ul>	

Fiche persona n°3	
Apprenant	
	<b>Nom:</b> Arthur <b>Age:</b> 21 ans <b>Profession:</b> Etudiant
	<p align="center"><u><b>Maturité numérique</b></u></p> <p>Arthur est un adepte des jeux vidéo, présent sur de nombreux réseaux sociaux. Il apprend le langage informatique depuis moins d'un an. Il est à l'aise avec l'outil informatique.</p> <p align="center"><u><b>Langage et logiciel informatique:</b></u></p> <p>HTML 5, CSS 3, Javascript, Suite Adobe : Niveau Intermédiaire            PHP, My SQL : Débutant            Wordpress: Débutant</p>
<p><u><b>Caractère</b></u></p> <p>Curieux            Dynamique            Inventif            Observateur</p>	
<p><u><b>Biographie:</b></u></p> <p>Passionné par les jeux vidéo, Arthur souhaite s'orienter à long terme vers la conception de jeux vidéo, pour l'instant, il entreprend des études dans le numérique.</p>	
<p><u><b>Intérêt pour l'écologie</b></u></p> <p>Il n'est pas concerné par l'écologie à 100%, mais il souhaite y participer dans ses prochaines créations numériques. Il fait le tri sélectif et fait attention à ses consommations d'électricité. Il prend son vélo pour de courts trajets.</p>	
<p><u><b>Attente de l'application :</b></u></p> <ul style="list-style-type: none"> <li>- Il souhaite pouvoir se créer un compte sur l'application.</li> <li>- Il souhaite suivre une formation, et connaître son avancement.</li> <li>- Il souhaite pouvoir connaître l'ensemble des formations proposées ainsi que faire une recherche.</li> <li>- Il souhaite pouvoir répondre à un petit quiz concernant le cours qu'il vient de suivre.</li> </ul>	

# Annexe 2

## Charte graphique



### Fonts:

Verdana Pro Light  
Verdana Pro Regular  
**Verdana Pro Bold**  
Sommet Rounded Light  
Sommet Rounded Regular  
**Sommet Rounded Bold**

### Palette de couleurs



### Titres formation

H1 font: Verdana pro Bold

### Titres des sections

H2 font: Sommet Rounded Regular

### Titres des leçons

H1 font: Sommet Rounded Regular

### Paragraphe

P font: Sommet Rounded Regular

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetet

[Lien](#)

[Lien survol](#)

font: Verdana pro Bold

font: Verdana pro Regular

Suivre

Bouton

Suivre

Bouton survol

### Titres formation

H1 font: Verdana pro Bold

### Titres des sections (H2)

H2 font: Sommet Rounded Regular

### Titres des leçons (H3)

H1 font: Sommet Rounded Regular

### Paragraphe (p)

P font: Sommet Rounded Regular

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetet

[Lien](#)

[Lien 2 survol](#)

font: Verdana pro Bold

font: Verdana pro Regular

Suivre

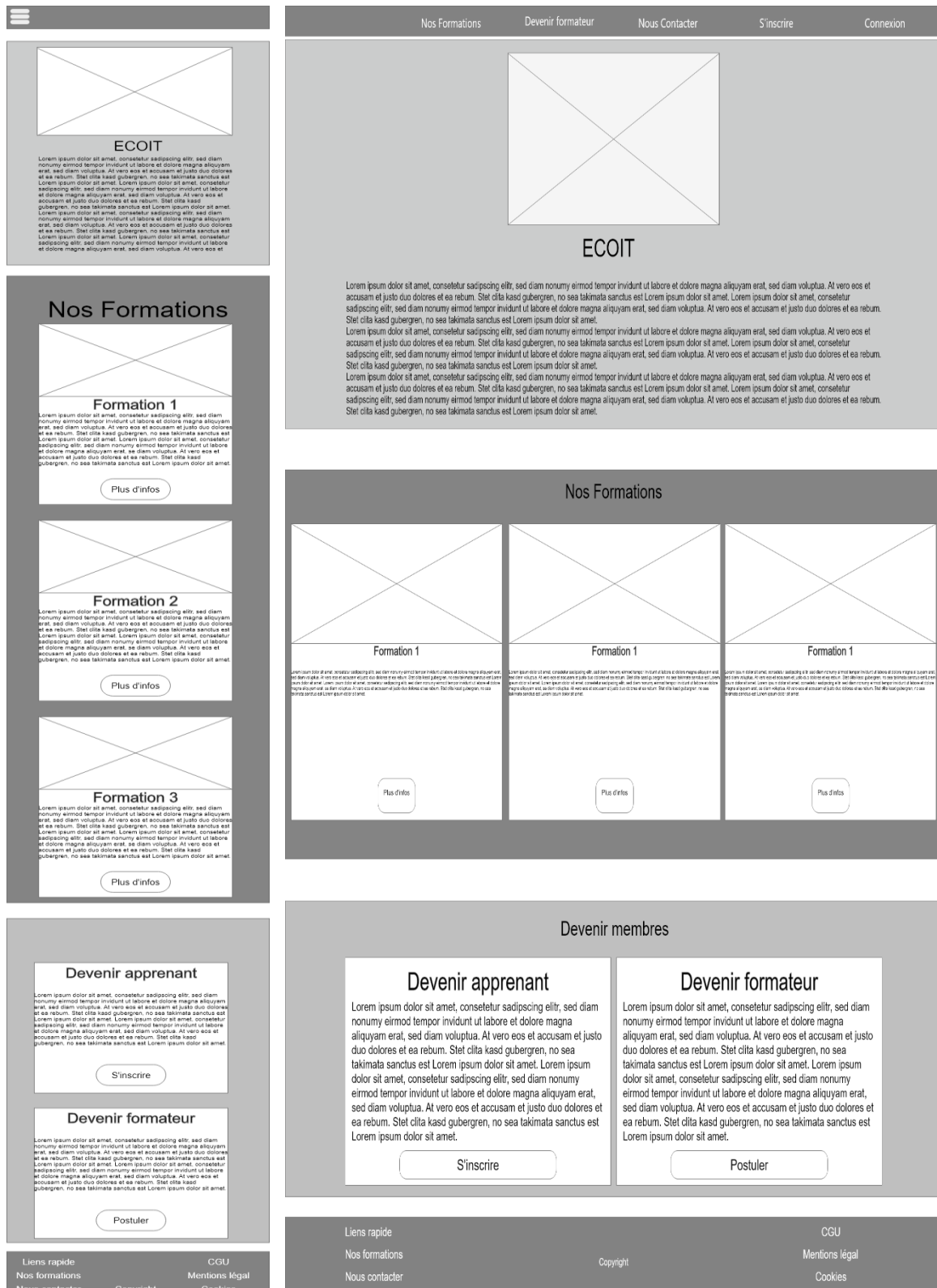
Bouton

Suivre

Bouton survol

# Annexe 3

## Différents wireframes de l'application réaliser avec Adobe XD



# Annexe 4

Différents wireframes de l'application réaliser avec Adobe XD

