

# **TITRE PROFESSIONNEL DÉVELOPPEUR WEB ET WEB MOBILE**

**Soutenance**  
**David SAADOUN – Knowledge**

# Qui suis-je?

---

## Parcours :

### O'clock

➡ No-Code – Automatisation (Airtable, Make)

➡ 2024

### Centre Européen de Formation

➡ Spécialisation Front-end / Back-end

➡ 2024-2025

## Pourquoi ?

- Intérêt depuis tous jeune pour les nouvelles technologies
- J'aime le fait qu'avec le code, on peut créer des outils concrets qui servent les gens comme des plateformes d'apprentissage, des sites pour gagner du temps, c'est motivant de développer quelque chose de réellement utilisé.

# Sommaires

---



I. PRÉSENTATION  
DU PROJET



II. CONCEPTION



III. DÉVELOPPEMENT



IV. RÉALISATIONS  
PERSONNELLES



V. DÉMONSTRATION  
DU PROJET



VI. EXEMPLE DE  
RECHERCHE



VII. CONCLUSION

# Organisation du travail

---



David

***Product owner***



David

***Scrum master***



David

***Dev front***



David

***Dev back***



David

***Git master***





## **Missions réalisées**

- Conception de la base de données (MCD / MLD)
- Création des interfaces
- Développement Symfony (back-end & front-end)
- Sécurité, validation, gestion des rôles
- Envoi de mails, gestion des erreurs
- Tests, débogage et déploiement local

## CONSTAT

De nombreux élèves en formation ont du mal à suivre leurs parcours, accéder aux contenus ou interagir avec leurs formateurs via des outils simples et accessibles.

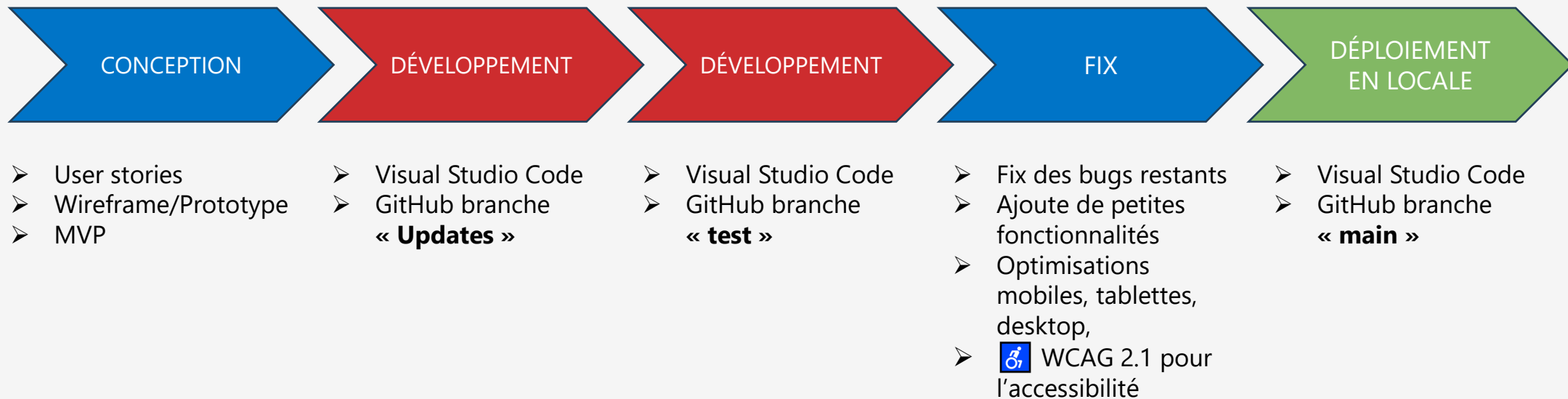
## AUDIENCE

-  Étudiants en formation
-  Formateurs
-  Centres de formation
-  Administrateurs pédagogiques

# Organisation du travail

---

- **5 sprints**, du vendredi au vendredi
- Utilisation de Teams, communication via Discord
- Daily meeting à 9 heure.
- Journée type de **9h à 19h**



Merge sur la branche **main** et déploiement le **vendredi après-midi**

# Conception

---



# User stories

➤ 3 types d'utilisateurs

- Visiteur
- Étudiant
- Administrateur

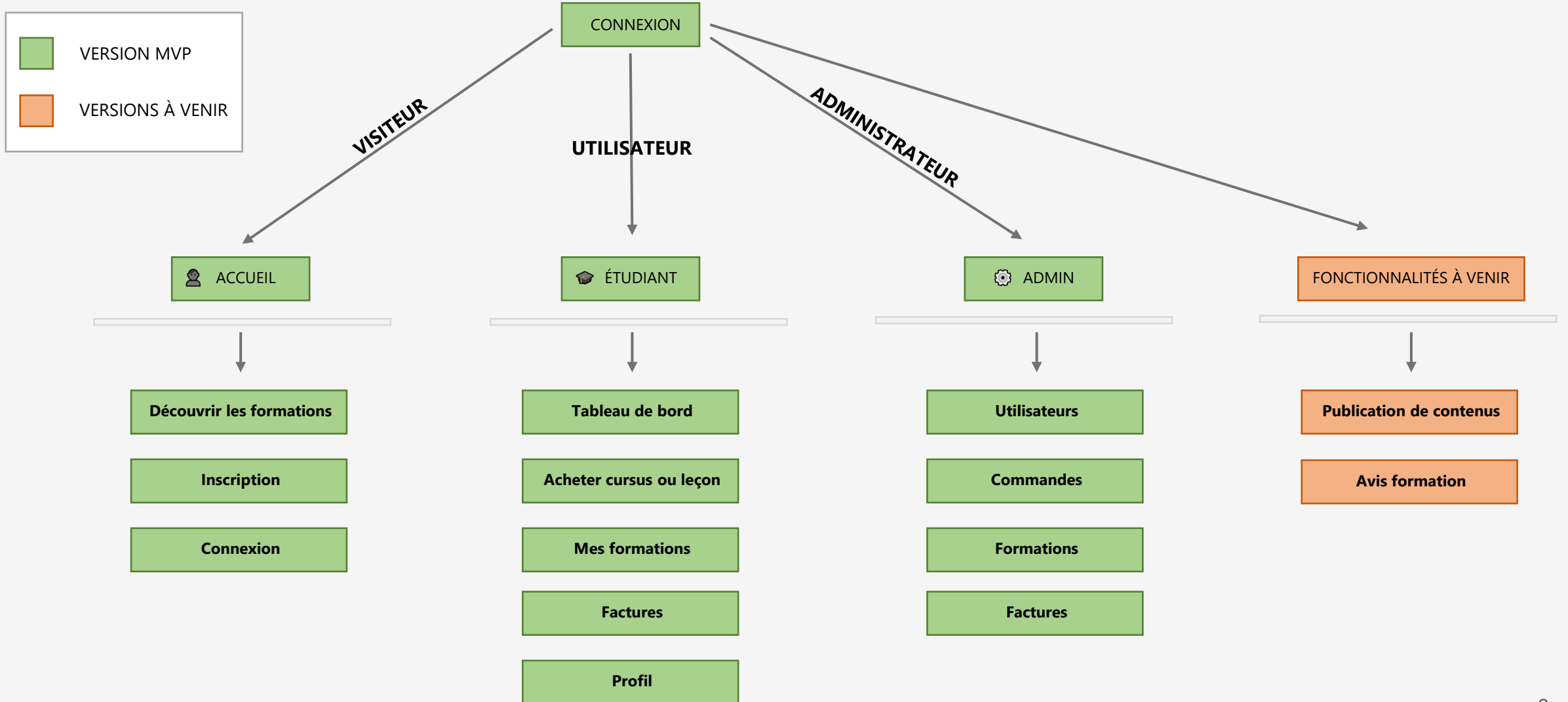
• En vert: **MVP**

• En orange: versions futures

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

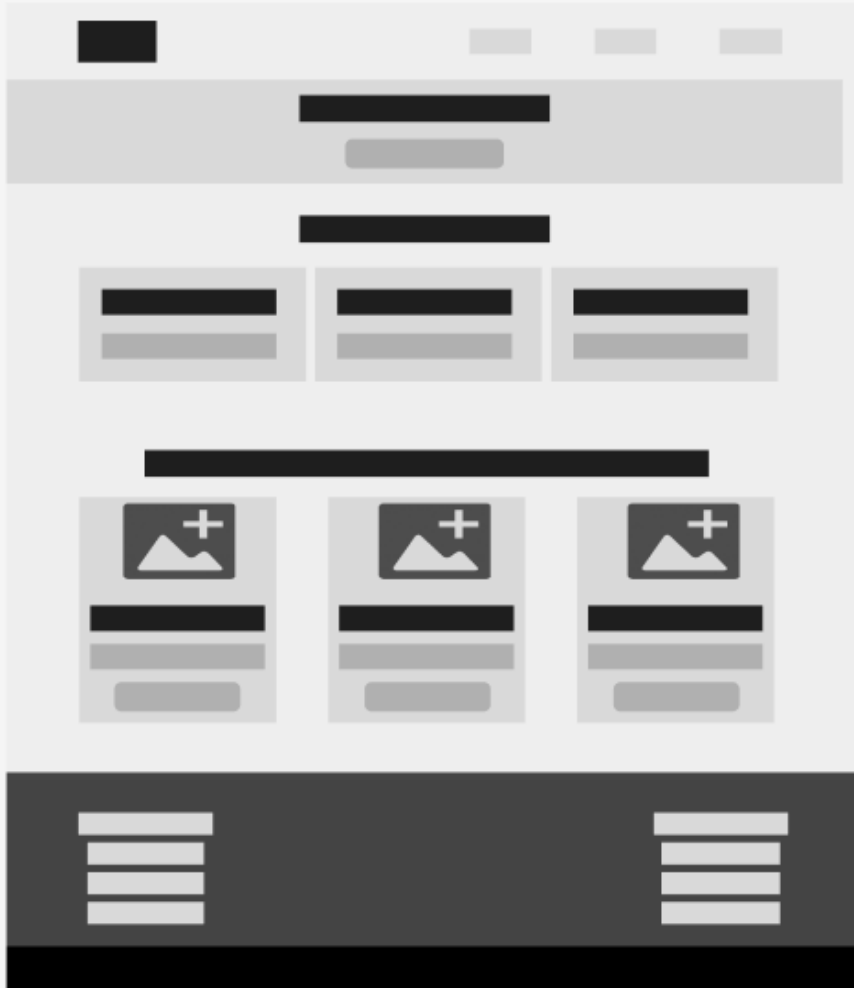


# Arborescence



# Wireframes

Desktop Wireframe

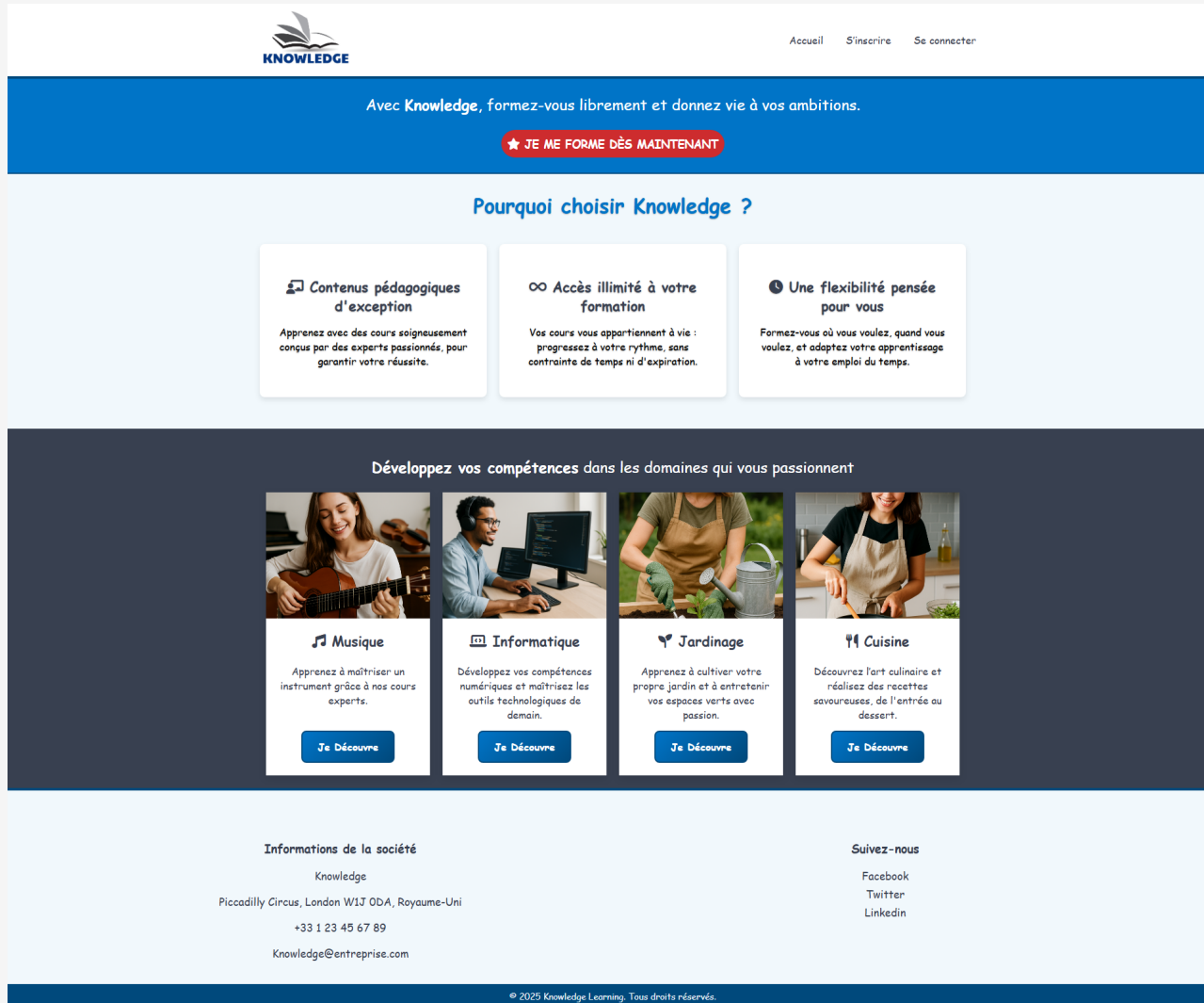


Mobile Wireframe

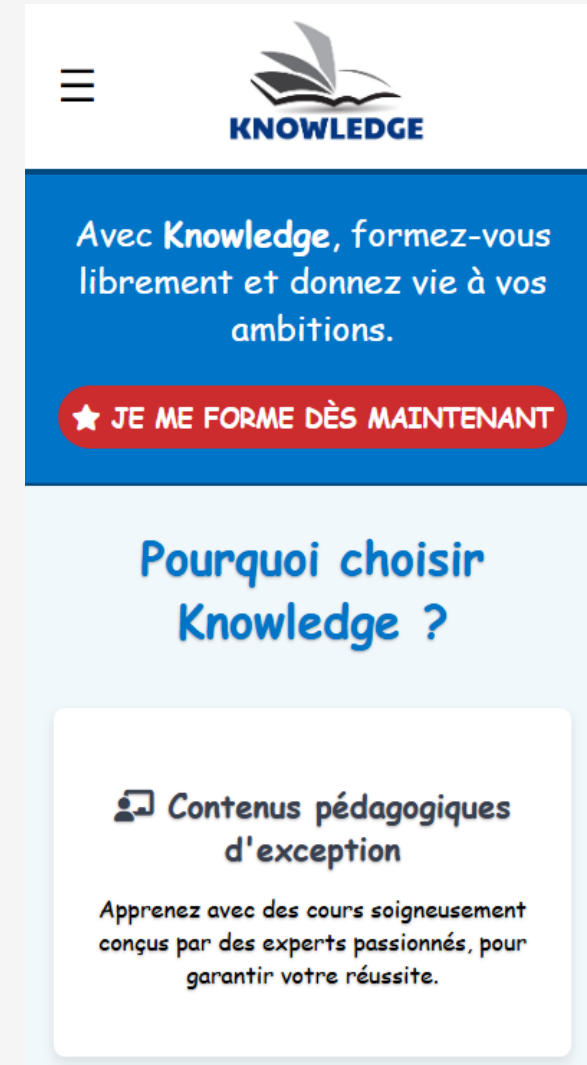


*Extrait des wireframes*

# Maquettes

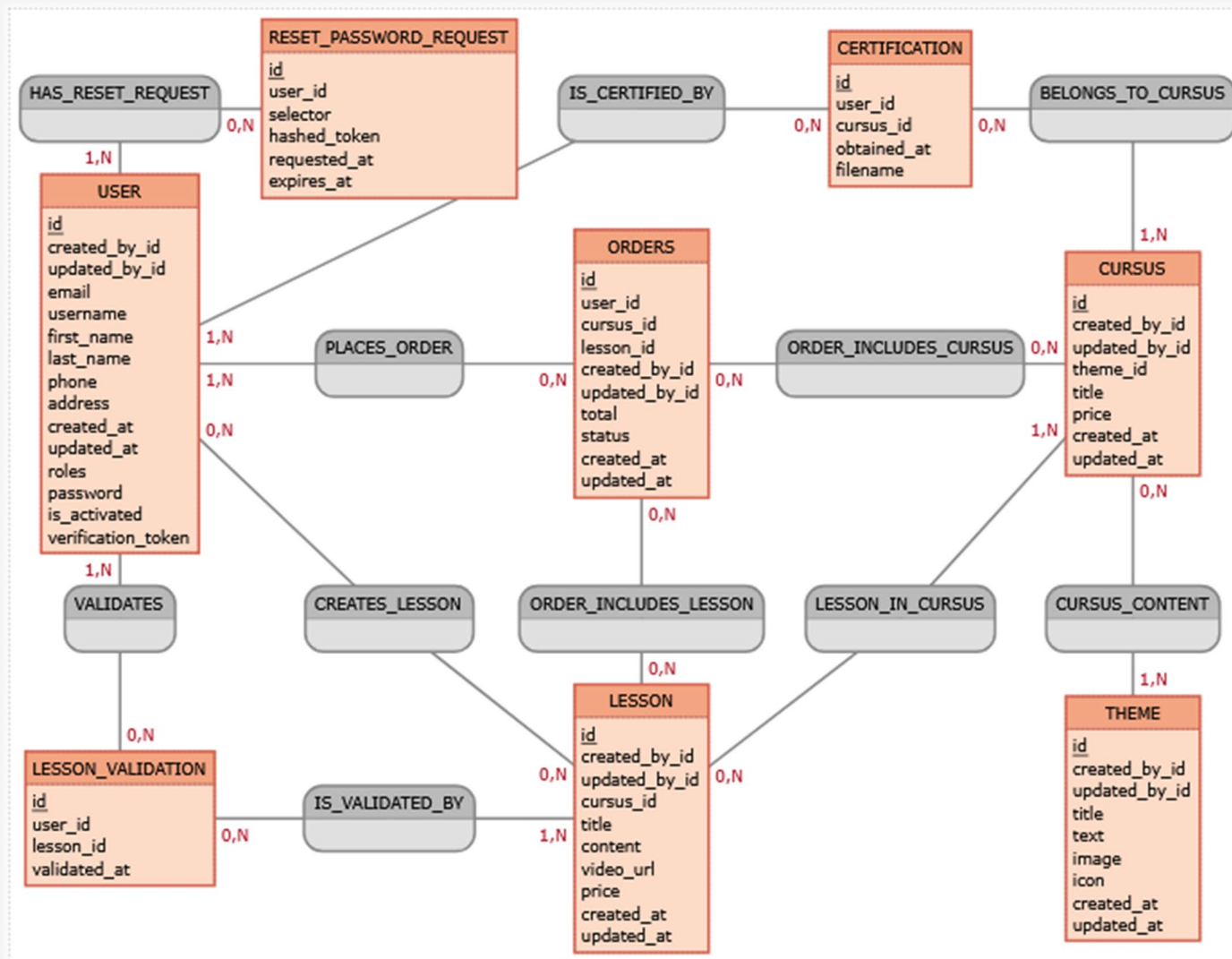


Desktop



Mobile (rogné)

# MCD



# Dictionnaire des données

Nom de la donnée	Désignation	Type	contrainte
id	Identifiant unique (clé primaire)	integer	auto_increment, primary key
username	Nom d'utilisateur	varchar(255)	not null, unique
email	Adresse email	varchar(255)	not null, unique
password	Mot de passe haché	varchar(255)	not null
roles	Liste des rôles (ex: ROLE_USER, ROLE_ADMIN)	json	not null
is_activated	Statut d'activation du compte	boolean	not null, default false
created_by_id	Créé par un autre utilisateur	int	foreign key to user.id, nullable
updated_by_id	Modifié par un autre utilisateur	int	foreign key to user.id, nullable
created_at	Date de création	datetime	not null, default current_timestamp
updated_at	Date de modification	datetime	nullable
verification_token	Jeton de vérification email	varchar(255)	nullable
first_name	Prénom de l'utilisateur	varchar(255)	nullable
last_name	Nom de famille de l'utilisateur	varchar(255)	nullable
phone	Numéro de téléphone	varchar(20)	nullable
address	Adresse postale	varchar(255)	nullable

*Extrait du dictionnaire des données*

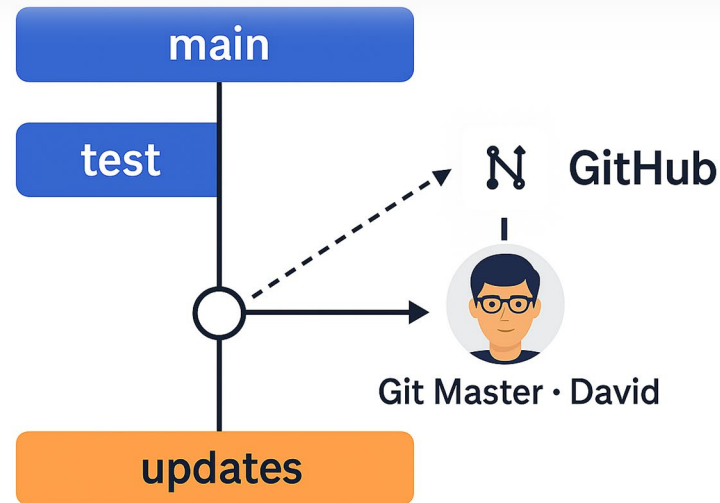
# Stack Technique

---

FRONT	BACK	DEVELOPER EXPERIENCE
<ul style="list-style-type: none"><li>➤ HTML5 / CSS3</li><li>➤ SCSS</li><li>➤ JavaScript</li><li>➤ Twig</li><li>➤ Webpack Encore</li><li>➤ Bootstrap 5</li></ul>	<ul style="list-style-type: none"><li>➤ Symfony 7(PHP 8.1)</li><li>➤ Doctrine ORM</li><li>➤ MySQL (via DATABASE_URL)</li><li>➤ JWT Auth (LexikJWTBundle)</li><li>➤ Stripe</li><li>➤ MailJet</li></ul>	<ul style="list-style-type: none"><li>➤ Docker (via compose.yaml)</li><li>➤ Git + GitHub</li><li>➤ Visual Studio Code</li><li>➤ Symfony Profiler</li><li>➤ PHPUnit</li><li>➤ .env et configuration Symfony</li><li>➤ Composer</li><li>➤ Symfony CLI</li></ul>

# Gitflow

Branche de **updates**  
*ex: add\_login\_user*



- 1 - Branche **updates**: contient les modifications sur lesquelles travaillent les membres de l'équipe.
- 2 - Branche **test**: fusionne les branches Updates pour effectuer des essais.
- 3 - Branche **main**: fusionne les branches validées, version destinée à la production

Le merge se fait uniquement via une **Pull Request**, suivie d'une revue de code par le Git Master « David »

# Développement

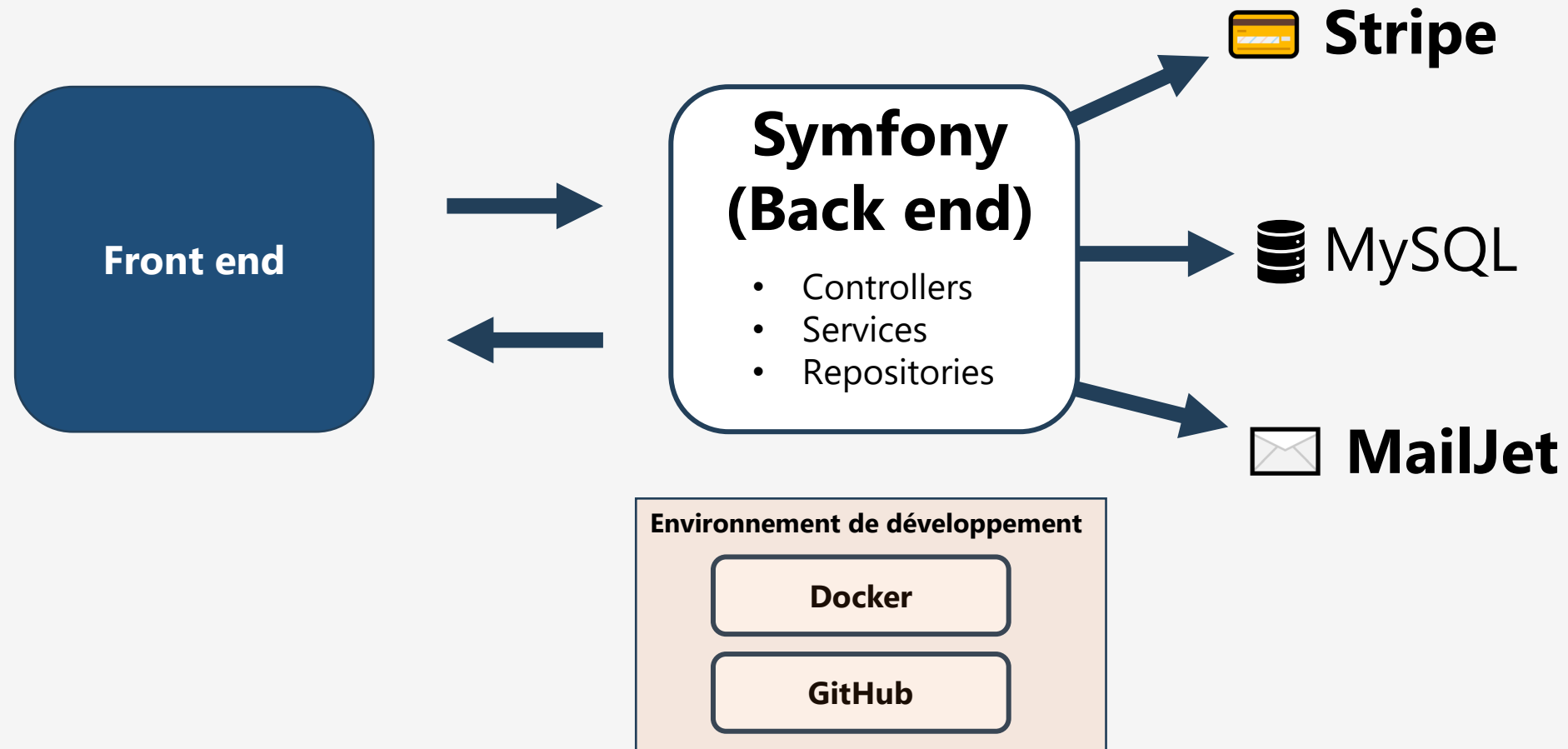
```

    check:function(view){
      if (this.$el.hasClass("close-full-overlay")){
        view.closeFullOverlay()
      }
      view.render()
      if (this.$el.hasClass("close-full-overlay")){
        view.navigate(c.router.baseUrl)
      }
      this.$el.addClass("iframe-ready")
      this.$el.removeClass("iframe-ready")
      this.$el.trigger("preview:close")
      this.$el.toggleClass("collapsed")
    }
  }
}

```



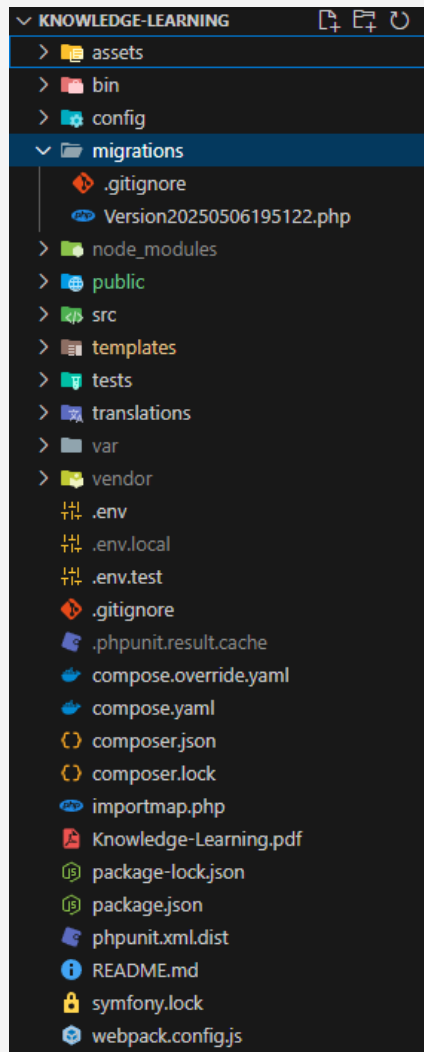
# Architecture



# Création de la base de données

```
1 <?php
2
3 namespace App\Entity;
4
5 use App\Entity\Order;
6 use App\Entity\Certification;
7 use App\Repository\UserRepository;
8 use Doctrine\Common\Collections\ArrayCollection;
9 use Doctrine\Common\Collections\Collection;
10 use Symfony\Bridge\Doctrine\Validator\Constraints\UniqueEntity;
11 use Doctrine\ORM\Mapping as ORM;
12 use Doctrine\ORM\Mapping\EntityListeners;
13 use Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface;
14 use Symfony\Component\Security\Core\User\UserInterface;
15
16 #[ORM\Entity(repositoryClass: UserRepository::class)]
17 #[ORM\EntityListeners(['App\EventListener\UserListener'])]
18 #[ORM\Table(name: 'user')]
19 #[UniqueEntity(fields: ['username'], message: 'Cet identifiant existe déjà.')]
20 #[UniqueEntity(fields: ['email'], message: 'Cette adresse email est déjà utilisée.')]
21 class User implements UserInterface, PasswordAuthenticatedUserInterface
22 {
23     #[ORM\Id]
24     #[ORM\GeneratedValue]
25     #[ORM\Column]
26     private ?int $id = null;
27
28     #[ORM\Column(length: 255, unique: true)]
29     private ?string $email = null;
30
```

*Extrait du fichier user.php*



```
1 <?php
2
3 namespace App\DataFixtures;
4
5 use App\Entity\Theme;
6 use App\Entity\Cursus;
7 use App\Entity\Lesson;
8 use App\Entity\User;
9 use Doctrine\Bundle\FixturesBundle\Fixture;
10 use Doctrine\Persistence\ObjectManager;
11 use Symfony\Component\PasswordHasher\Hasher\UserPasswordHasherInterface;
12
13 class AppFixtures extends Fixture
14 {
15     private UserPasswordHasherInterface $passwordHasher;
16
17     public function __construct(UserPasswordHasherInterface $passwordHasher)
18     {
19         $this->passwordHasher = $passwordHasher;
20     }
21
22     public function load(ObjectManager $manager): void
23     {
24         // 1. Themes with associated metadata and curricula
25         $themesMeta = [
26             'Musique' => [
27                 'text' => 'Apprenez à maîtriser un instrument grâce à nos cours experts.',
28                 'image' => 'music.webp',
29                 'icon' => 'fa-music',
30                 'cursus' => [
31                     ['Cursus d'initiation à la guitare', 50, [
32                         ['Découverte de l'instrument', 26],
33                         ['Les accords et les gammes', 26],
34                     ]],
35                     ['Cursus d'initiation au piano', 50, [
36                         ['Découverte de l'instrument', 26],
37                         ['Les accords et les gammes', 26],
38                     ]],
39                 ],
40             ],
41         ],
42     }
43 }
```

*Extrait du fichier AppFixtures.php*

18

\* À chaque modification de l'entité: **php bin/console doctrine:migrations:diff** et **php bin/console doctrine:migrations:migrate** \*

# Accès aux données

- Lecture (Read) via Repository
- Récupération des données filtrées
- Doctrine ORM

```
1 // 1) Retrieves all user commands
2 $orders = $orderRepository->findBy(['user' => $user]);
```

- Création (Create) via setters
- Hydratation via setters (ex. : setEmail)
- Insertion avec persist() + flush()

```
1 // Create users
2 $student = new User();
3 $student->setEmail('student@example.com')
4         ->setUsername('student')
5         ->setRoles(['ROLE_USER'])
6         ->setPassword($this->passwordHasher->hashPassword($student, 'password'))
7         ->setActivated(true);
8 $manager->persist($student);
9
10 $manager->flush();
```

- Suppression (Delete) d'un utilisateur
- Doctrine ORM
- remove() + flush()

```
1 $em->remove($user);
2 $em->flush();
```

\* Les entités Doctrine font office de modèles. Elles sont automatiquement liées à la base via Doctrine ORM, ce qui permet les opérations CRUD sans SQL manuel \*

# Réalisation du back-end



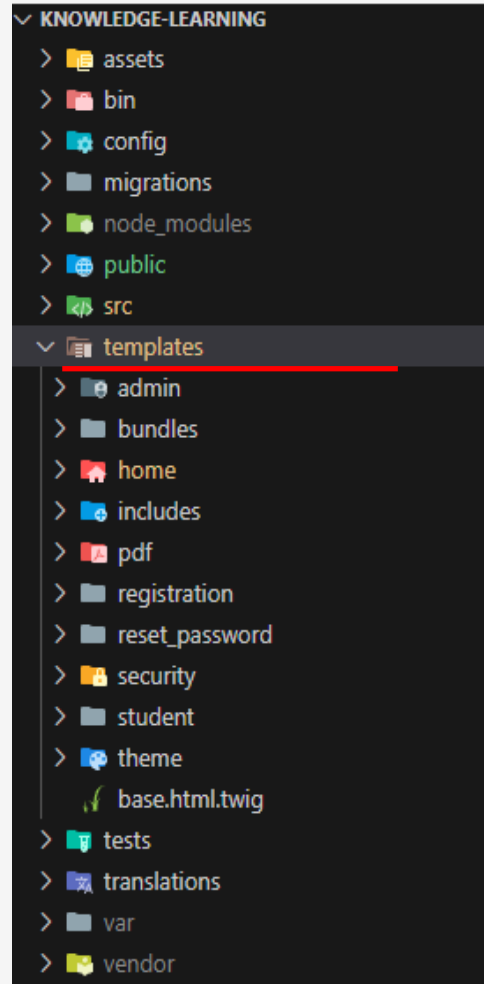
**Exemple du fichier:** src/Controller/StudentCoursesController.php

```
1  foreach ($orders as $order) {
2      // only processes pack orders
3      if (!$c = $order->getCursus()) {
4          continue;
5      }
6
7      $totalLessons = count($c->getLessons());
8
9      // count lessons validated for this course
10     $validatedCount = 0;
11     foreach ($allValidations as $val) {
12         if ($val->getLesson()->getCursus()->getId() === $c->getId()) {
13             $validatedCount++;
14         }
15     }
16
17     $pct = $totalLessons > 0
18         ? (int) round($validatedCount / $totalLessons * 100)
19         : 0;
20
21     $progress[$c->getId()] = $pct;
22 }
```

# Réalisation du front-end

Partie dynamique généré via Twig

- Templates organisés par sections (admin, student, etc.) dans templates/
- Application structurée avec des templates Twig
- Routage géré par les annotations dans les contrôleurs Symfony



Structure du projet

```
1
2 <title>
3     {% block title %}Knowledge-Learning
4     {% endblock %}
5 </title>
6
7
8 {% block stylesheets %}
9     {{ encore_entry_link_tags('style') }}
10 {% endblock %}
11
12 <main>
13     {% block body %}{% endblock %}
14 </main>
15
16 {% block javascripts %}
17     {{ encore_entry_script_tags('app') }}
18 {% endblock %}
```

Extrait de composant

# Réalisation du front-end

Le front-end repose sur Twig pour générer des pages dynamiques, stylées avec SCSS modulaire.

```
1 {% for theme in themes %}
2     <div class="card2">
3         
4         <h3 class="{{ theme.title|lower|replace({' ': '-'}) }}-title">
5             <i class="fas {{ theme.icon }}"></i>
6             {{ theme.title }}
7         </h3>
8         <p>{{ theme.text }}</p>
9         <a href="{{ path('visitor_themes_filter', { 'filter': theme.id }) }}" class="btn-custom">Je Découvre</a>
10    </div>
11 {% endfor %}
```

Extrait de la page **Home.html.twig**

```
1 @import "../admin.scss"; /* Import scss admin */
2 @import "../mobile.scss"; /* Import scss mobile */
3 @import "../tablet.scss"; /* Import scss tablet */
4
5 /* Colors Template*/
6 $color-1: #f1f8fc;
7 $color-2: #0074c7;
8 $color-3: #00497c;
9 $color-4: #384050;
10 $color-5: #cd2c2e;
11 $color-6: #82b864;
12 $color-7: #fff;
13
14 html,
15 body {
16     margin: 0;
17     padding: 0;
18     font-family: "Comic Sans MS", cursive, sans-serif;
19     height: 100%;
20     background-color: $color-1;
21 }
22
23 main {
24     padding: 0;
25     width: 100%;
26 }
27
28 header {
29     background-color: $color-7;
30     padding: 10px 0;
31     border-bottom: 4px solid $color-3;
32 }
```

Extrait du fichier **assets/css/style.scss**

# Sécurité

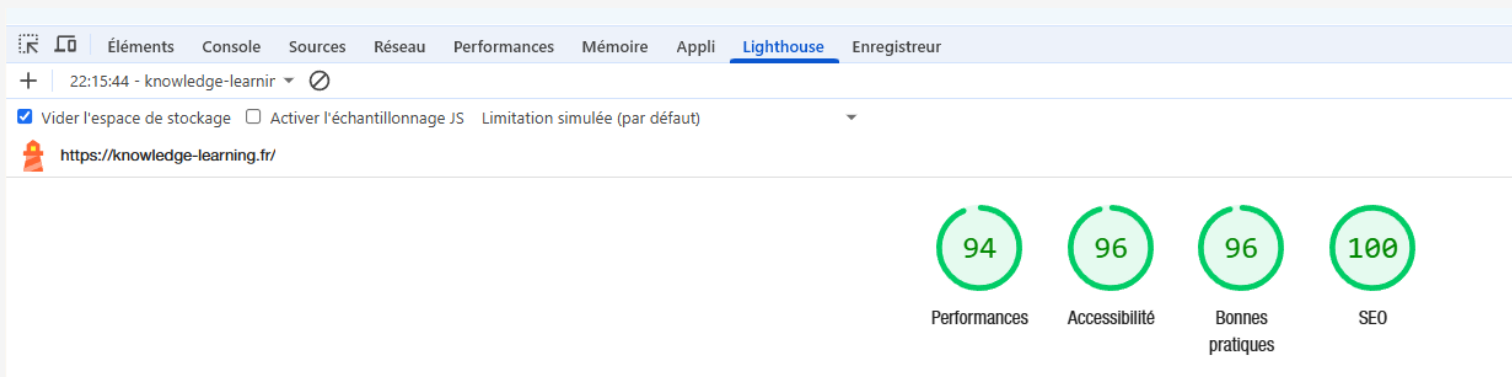
---

FAIT	À FAIRE
<ul style="list-style-type: none"><li>➤ Hachage des mots de passe</li><li>➤ Protection CSRF</li><li>➤ Validation des données côté serveur</li><li>➤ Filtrage des entrées utilisateurs (Validator + contraintes Symfony)</li><li>➤ Gestion des rôles et des accès</li><li>➤ Encodage automatique contre XSS → via Twig</li></ul>	<ul style="list-style-type: none"><li>➤ Limitation de requêtes (Rate limiting)</li><li>➤ Politique de sécurité HTTP stricte (CSP, HSTS)</li><li>➤ Vérification + nettoyage de fichiers uploadés</li><li>➤ Logs d'activités utilisateurs / tentatives de connexion</li></ul>

# Accessibilité & Performance

Les optimisations réalisées visent à améliorer l'accessibilité, les performances et la conformité aux standards web.

Ces résultats assurent une meilleure expérience utilisateur, une conformité aux standards d'accessibilité et un impact SEO renforcé.



*Résultats Lighthouse (Performances & Accessibilité)*



*Validation du code CSS selon la norme W3C (niveau 3 + SVG)*



# Démo du projet

---





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

### Connexion

Nom d'utilisateur

Nom d'utilisateur

Mot de passe

Mot de passe

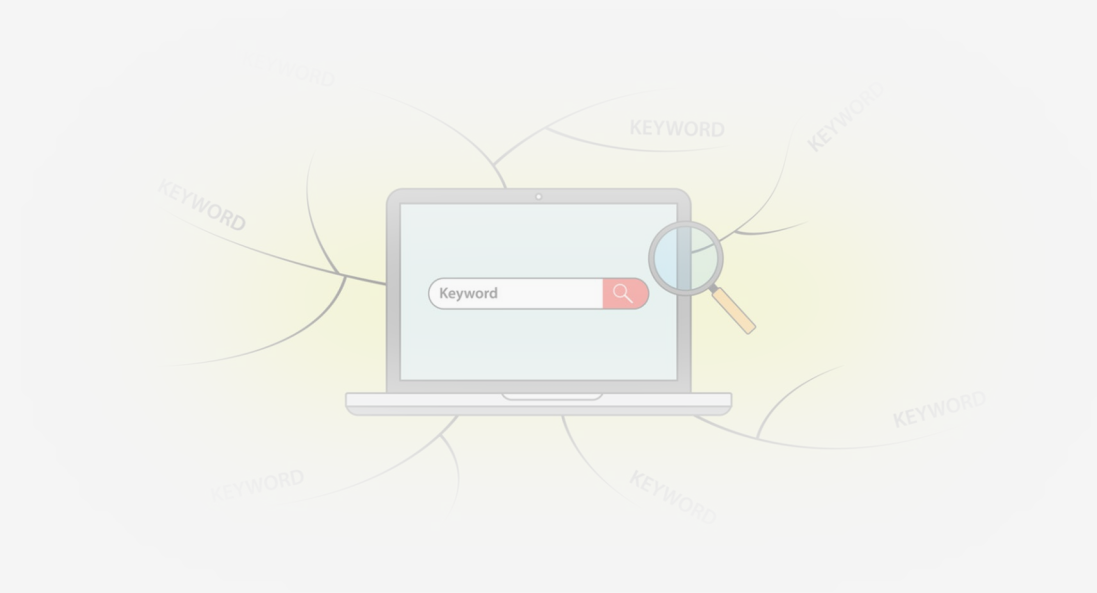
Valider

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

[Mot de passe oublié ?](#)

# Exemple de recherche

---



# Présentation d'une recherche

- Recherche en anglais, avec des mots simples
- Selection des sources:
  - Classement « personnel » en fonction de la clarté des comparaisons entre Symfony et Laravel.
- Vérification de la date de publication (résultats entre 2024 et 2025).
- Choix de sources fiables : blogs techniques, communautés (Reddit, DEV Community, Stackoverflow).
- Croisement des avis pour confirmer les points forts de Symfony (modularité, sécurité, scalabilité).

The screenshot shows a Google search for "symfony vs laravel stack". The search bar at the top contains the query. Below the search bar, there are tabs for "Tous", "Vidéos", "Images", "Vidéos courtes", "Web", "Actualités", "Livres", and "Plus". The "Tous" tab is selected. To the right of the tabs is a button labeled "Outils". Below the tabs, there are filters for "Toutes les langues", "Moins d'un an", "Tous les résultats", "Recherche avancée", and "Effacer". On the right side, it says "Environ 148 000 résultats (0,28 secondes)".

The first search result is from Glorywebs, titled "Laravel vs Symfony: Best PHP Framework for 2025 - Glorywebs". The snippet reads: "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." The date "27 févr. 2025" is shown. Below the snippet is the Glorywebs logo and the URL "https://www.glorywebs.com › blog › laravel-vs-symfony". There are also links for "À propos des extraits optimisés" and "Commentaires".

The second search result is from Reddit, titled "Symfony or Laravel? : r/PHP". The snippet reads: "Symfony has a lot of moving parts, but they're distinct parts. They are decoupled, can be tested in isolation, and are faster. Laravel is a big muddled pile of ...". It shows "90 réponses" and "Meilleure réponse: Reason? Advantages Forms: their docs is ...". Below the snippet are two more links: "Did I Make the Right Choice with PHP? What ..." with "104 réponses" and "6 févr. 2025", and "Someone still using Raw PHP over frameworks like ..." with "179 réponses" and "26 janv. 2025". At the bottom of the result is the link "Autres résultats sur www.reddit.com".

The third search result is from DEV Community, titled "Laravel VS Symfony: The PHP Framework Showdown". The snippet reads: "5 nov. 2024 — Both Laravel and Symfony are powerful frameworks, but the right choice depends on your project needs and team's expertise." Below the snippet is the DEV Community logo and the URL "https://dev.to › varzoeaa › laravel...". There is also a link for "Traduire cette page".

# Présentation d'une recherche sur la sécurité web

## Classement:

1. Toujours filtrer les entrées avec Validator
2. Protéger les formulaires avec des tokens CSRF
3. Restreindre les accès par rôles (IsGranted)
4. Éviter d'exposer les messages d'erreur sensibles

## Why Symfony?

On this page, you'll learn about the motivations for choosing Symfony and how it compares to other PHP frameworks like Laravel or Slim.

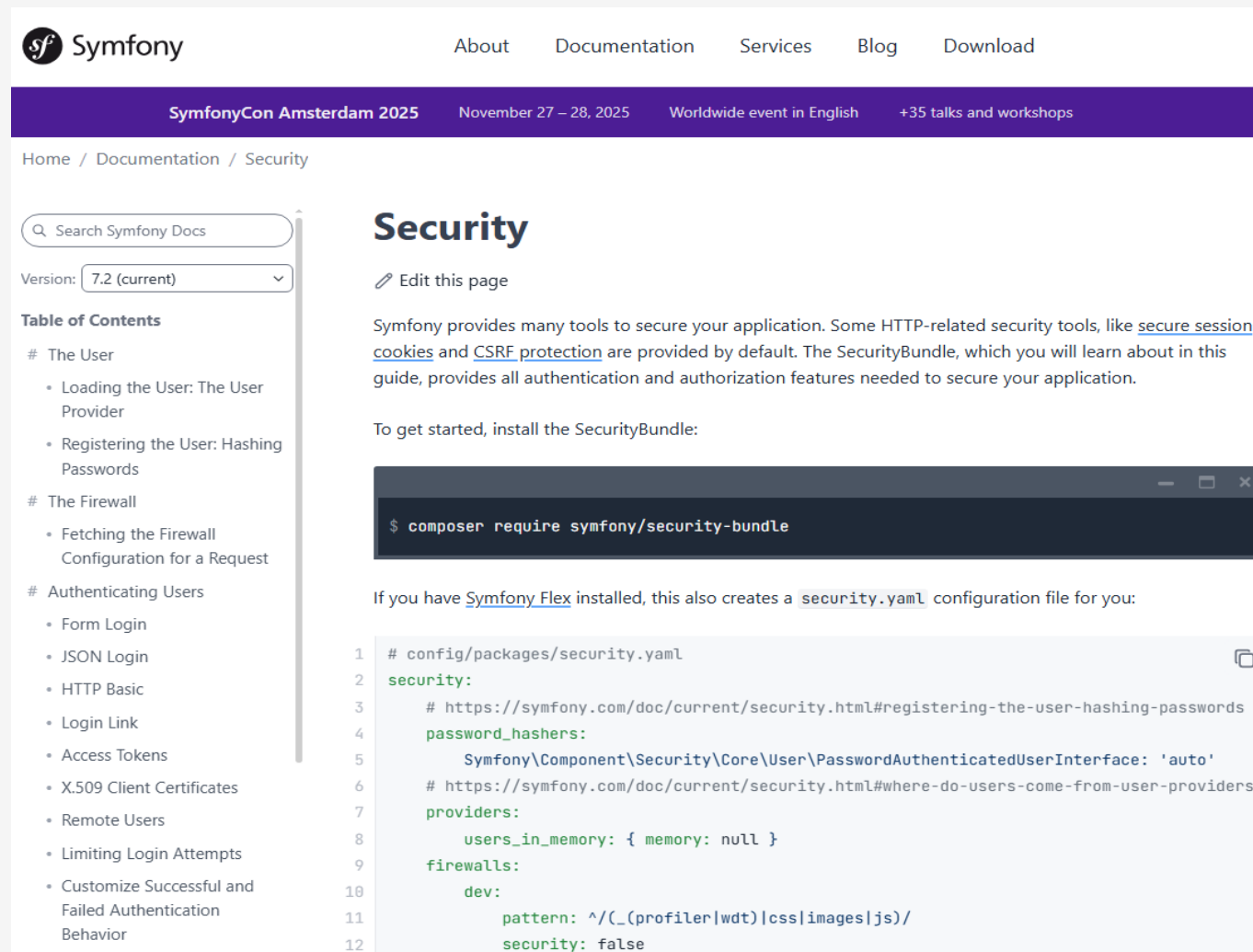
Symfony is designed for large-scale and complex web applications. Its modular component system, solid architecture, and extensive documentation make it a reliable choice for long-term projects.

Security, performance, and scalability are central to Symfony's ecosystem. With built-in tools such as routing, service containers, security firewalls, and Doctrine ORM, developers gain fine control over every aspect of the application.

Symfony is widely adopted in enterprise environments and follows best practices by default, making it ideal for professional-grade applications.

## TLDR

Symfony provides a robust, scalable, and secure framework with a strong community and long-term support. Its flexibility and adherence to standards make it a trusted choice for developing maintainable and enterprise-level web applications.



The screenshot shows the Symfony documentation page for Security. The page has a purple header with navigation links: About, Documentation, Services, Blog, and Download. Below the header, a purple banner promotes 'SymfonyCon Amsterdam 2025' from November 27-28, 2025, as a worldwide event in English with 35+ talks and workshops. The breadcrumb trail is 'Home / Documentation / Security'. A search bar and a version dropdown (set to 7.2 current) are present. A 'Table of Contents' sidebar lists topics like 'The User', 'The Firewall', and 'Authenticating Users'. The main content area is titled 'Security' and includes an 'Edit this page' link. It explains that Symfony provides tools for security, such as session cookies and CSRF protection, and mentions the SecurityBundle. A code block shows the command to install the bundle: `$ composer require symfony/security-bundle`. Below this, it states that if Symfony Flex is installed, a `security.yaml` file is created. A code snippet shows the configuration for the `security.yaml` file, including password hashers, providers, and firewalls. The `security` key is highlighted with a red underline.

**Security**

Edit this page

Symfony provides many tools to secure your application. Some HTTP-related security tools, like [secure session cookies](#) and [CSRF protection](#) are provided by default. The SecurityBundle, which you will learn about in this guide, provides all authentication and authorization features needed to secure your application.

To get started, install the SecurityBundle:

```
$ composer require symfony/security-bundle
```

If you have [Symfony Flex](#) installed, this also creates a `security.yaml` configuration file for you:

```
1 # config/packages/security.yaml
2 security:
3     # https://symfony.com/doc/current/security.html#registering-the-user-hashing-passwords
4     password_hashers:
5         Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface: 'auto'
6     # https://symfony.com/doc/current/security.html#where-do-users-come-from-user-providers
7     providers:
8         users_in_memory: { memory: null }
9     firewalls:
10         dev:
11             pattern: ^/(_(profiler|wdt)|css|images|js)/
12             security: false
```

Exemple d'une page officielle: <https://symfony.com/doc/current/security.html>

# Conclusion

---

- Fierté du travail accompli en autonomie
- Renforcement des compétences en développement web
- Apprentissage approfondi de Symfony et des bonnes pratiques
- Envie de continuer à faire évoluer le projet (ajout de fonctionnalités, tests, sécurité...)

**À vos questions !**