

Accrochet-vous

Un site de modèles de crochet



Tatiana Allery pour l'examen :
Développeur web et web mobile - Titre RNCP niveau V
Session : 11 septembre - 21 décembre 2024

Présentation : Tatiana ALLERY

Master Biologie-Agrosciences avec spécialisation mycologie & phytopathologie puis 1 an comme ingénieure d'études à l'INRAe (en foresterie à Cestas)



J'ai fais 6 mois de service civique à la Maison RoseUp de Bordeaux puis mi-temps pendant 1 an en tant que chargée d'accueil et de missions polyvalentes

En reconversion pour devenir développeuse web

Je fais du crochet depuis 5 ans

Compétences mises en oeuvre

- Développer la partie front-end d'une application web ou web mobile sécurisée
 - Installer et configurer son environnement de travail en fonction du projet web ou web mobile
 - Maquetter des interfaces utilisateur web ou web mobile
 - Réaliser des interfaces utilisateur statiques web ou web mobile
 - Développer la partie dynamique des interfaces utilisateur web ou web mobile
- Développer la partie back-end d'une application web ou web mobile sécurisée
 - Mettre en place une base de données relationnelle
 - Développer des composants d'accès aux données SQL et NoSQL
 - Développer des composants métier côté serveur
 - Documenter le déploiement d'une application dynamique web ou web mobile

I/ Présentation générale



1 - Présentation du projet

Constat :

Trouver des modèles (pattern) de crochet gratuits c'est compliqué et sur une multitude de sites.

Solution :

Un site qui regroupe uniquement des modèles gratuits en libre accès avec possibilité d'en soumettre de nouveau pour partager sa passion et la rendre accessible.

2 - Expression fonctionnelle des besoins

Catégorie d'utilisateur	Besoin
Utilisateur	Créer et modifier son compte et se connecter
	Accéder aux différents modèles gratuits qui sont proposés sur le site
	Proposer des modèles qu'il a créé
Administrateur	Gestion des utilisateurs
	Gestion des articles et des catégories
	Modération des articles et des commentaires

Contraintes :

- Sécuriser le contenu du site
- Gérer les droits d'accès pour différents types d'utilisateurs
- Responsive : ordinateurs et smartphones

3.1 - Cadre de réponse - front

Catégorie	Fonctionnalité	Rédacteur	Utilisateur	Visiteur
Visibilité sur les articles	Accès à la recherche par mot-clés	✓	✓	✓
	Accès aux filtres de recherche	✓	✓	✓
	Accès à tous les articles de la recherche	✓	✓	✗ limité à 5
Commentaires	Mettre un commentaire	✓	✓	✗
	Voir les commentaires	✓	✓	✗
Gestion de création de modèle	Soumettre un article ou un tuto	✓	✗	✗
	Modifier un article dont on est l'auteur	✓	✗	✗

3.2 - Cadre de réponse - back

Catégorie	Fonctionnalité	Super admin	Admin
Gestion des articles et tutos	Ajout d'article ou tutos	✓	✓
	Modification d'articles ou tutos	✓	✓
	Suppression d'articles/tutos	✓	✗
Gestion des commentaires	Accepter un commentaire	✓	✓
	Supprimer un commentaire	✓	✓
Gestion des comptes	Gestion des rôle admins (donner/enlever le rôle)	✓	✗
	Bloquer un compte	✓	✗
	Donner le statut rédacteur	✓	✓

3.3 - Cadre de réponse - contraintes spécifiques

Contraintes	Fonctionnalité
Sécuriser le contenu du site	<p>Mise en place des rôles vu précédemment</p> <p>Un côté admin vraiment séparé du site “client”</p>
Gérer les droits	<p>Système d'accès selon les droit avec <code>IsGranted</code> en back et boutons disabled en front</p> <p>Demande de droit pour le front pour devenir “rédacteur”</p>
Responsive	<p>Maquettes desktop & smartphone</p> <p>Media queries en SCSS pour le responsive</p>

4 - MoodBoard & charte graphique



#A589C6

#5F4B64

#AFC294

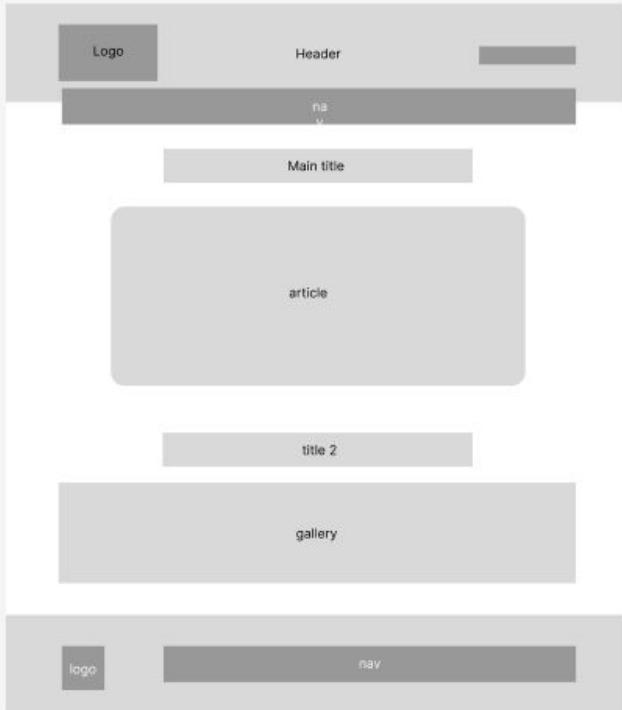
#6F7762

#EEEEEE

#181F1C

5 - Zoning et wireframe

welcome page - desktop

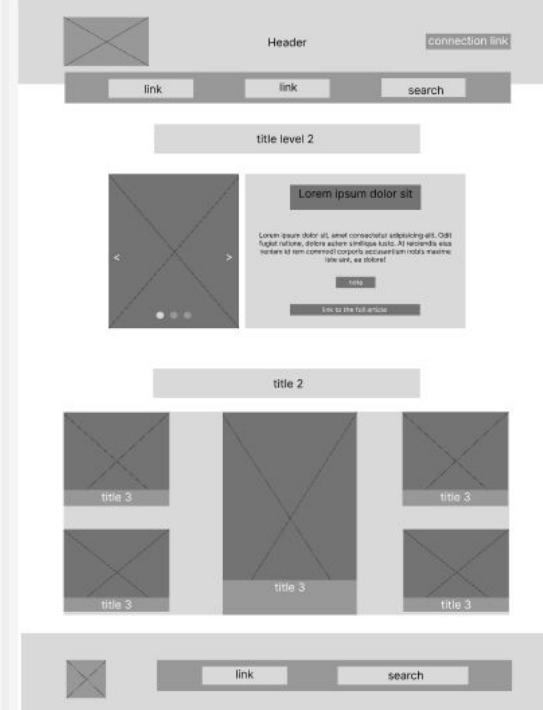


welcome pag...

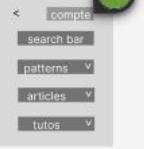
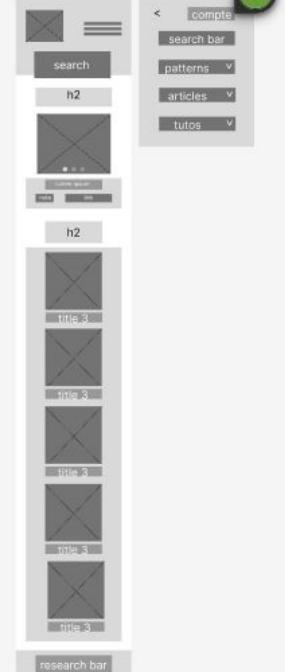


welcome pag...

wireframe - welcome page - desktop



wireframe - ...



6.1 - Maquette desktop

Home page - desktop



Accrochet-vous
Modèle de crochets gratuits

Se connecter

Modèles Tutos Recherche

Derniers ajouts



Arnold the acorn
Hooks & Heeler nous propose un modèle tout doux pour réaliser Arnold the Acorn. Un amigurumi en fil chenille, un patron qui se fait en une traite, ne nécessite pas de coudre les pièces entre elles.
amigurumi, débutant, anglais

4,8/5

Voir le modèle

Les modèles les mieux notés



T-shirt rayé riviera
vêtement, débutant, anglais



Sac de course sakura
maison, intermédiaire, anglais



Couverture pattoune
maison, débutant, français



Deadpool amigurumi
amigurumi, intermédiaire, anglais

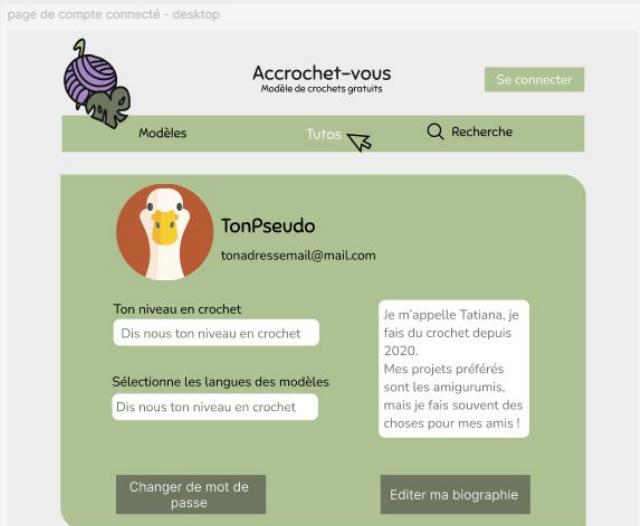


Sorcière Diablo IV
amigurumi, confirmé, anglais

Copyrights 2025 - Accrochet-vous Modèles Contact Demander le statut rédacteur

Tatiana ALLERY - Développeur web et web mobile

page de compte connecté - desktop



Accrochet-vous
Modèle de crochets gratuits

Se connecter

Modèles Tutos Recherche

TonPseudo
tonadressemall@mail.com

Ton niveau en crochet
Dis nous ton niveau en crochet

Sélectionne les langues des modèles
Dis nous ton niveau en crochet

Je m'appelle Tatiana, je fais du crochet depuis 2020.
Mes projets préférés sont les amigurumis, mais je fais souvent des choses pour mes amis !

Changer de mot de passe

Editer ma biographie

Search result - desktop

Accrochet-vous
Modèle de crochets gratuits

Se connecter

Modèles Tutos Recherche

Résultats pour la recherche "nom de la recherche"

Sorcière - Diablo IV



Un amigurumi inspiré du monde du jeu Diablo IV. Proposé par Charlotte de Rainbow crochet. amigurumi, confirmé, anglais

4,6 / 5

Voir le modèle

Arnold the acorn



Hooks & Heeler nous propose un modèle tout doux pour réaliser Arnold the Acorn. Un amigurumi en fil chenille. amigurumi, débutant, anglais

4,8 / 5

Voir le modèle

Deadpool en amigurumi



Un modèle proposé par le site theamigurumi.com qui représente un petit deadpool miniature. amigurumi, intermédiaire, anglais

4,9 / 5

Voir le modèle

Dino-triste



Nicole nous propose ce modèle d'amigurumi dinosaure tout mignon ! amigurumi, intermédiaire, anglais

4,6 / 5

Voir le modèle

Superman Amigurumi



Un amigurumi de superman, amigurumi, confirmé, anglais

4,2 / 5

Voir le modèle

1 sur 1

6.2 - Maquette mobile

Home page - phone - nav ...



Se connecter

Recherche

Les modèles

Tutos

Points de bases

Points avancés

Les modèles les mieux notés

T-shirt rayé riviera

Deadpool amigurumi

Couverture pattoune

espace compte - mobile



Search result - mobile

Recherche

filtres

Recherche pour "nom de la recherche"

Sorcière - Diablo IV

Voir le modèle

Arnold the acorn

Voir le modèle

Deadpool - amigurumi

Voir le modèle

Dino-triste

Voir le modèle

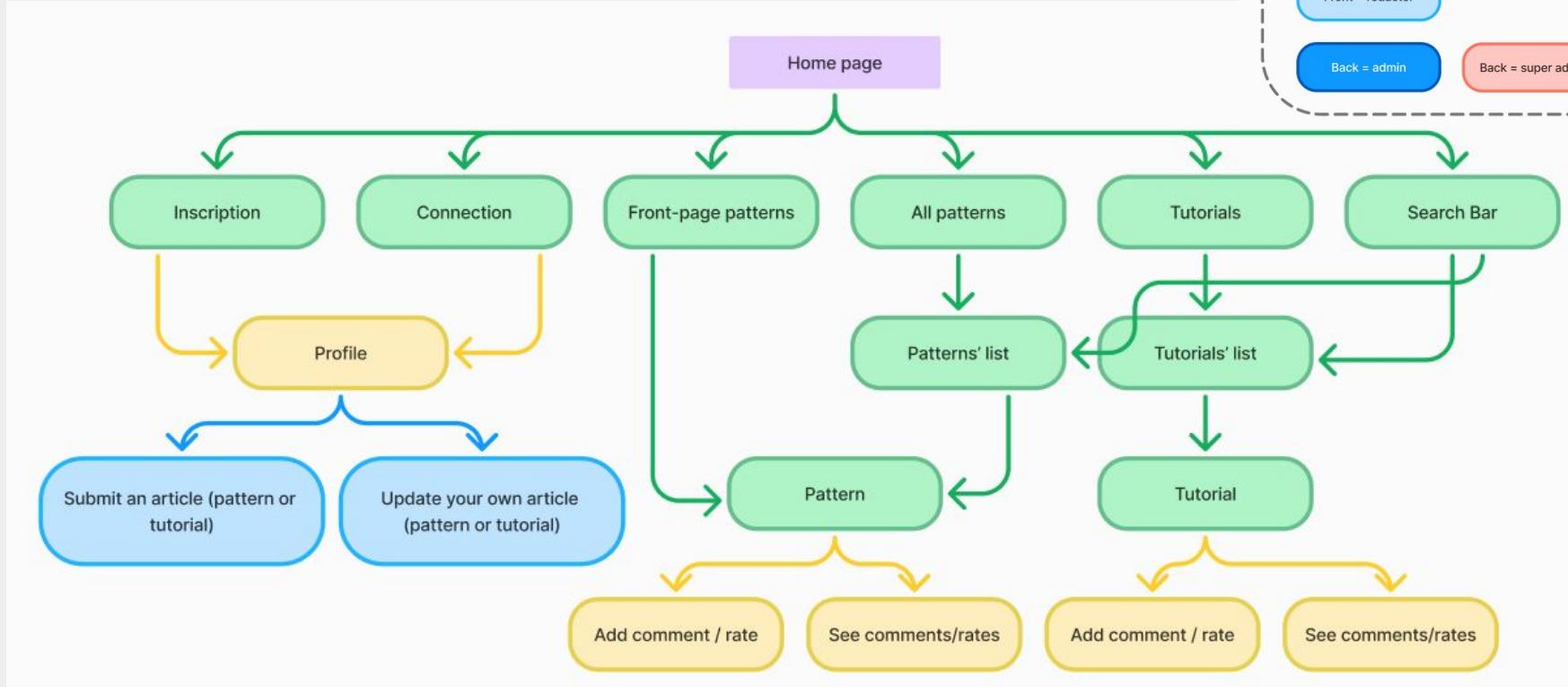
Superman

Voir le modèle

next page

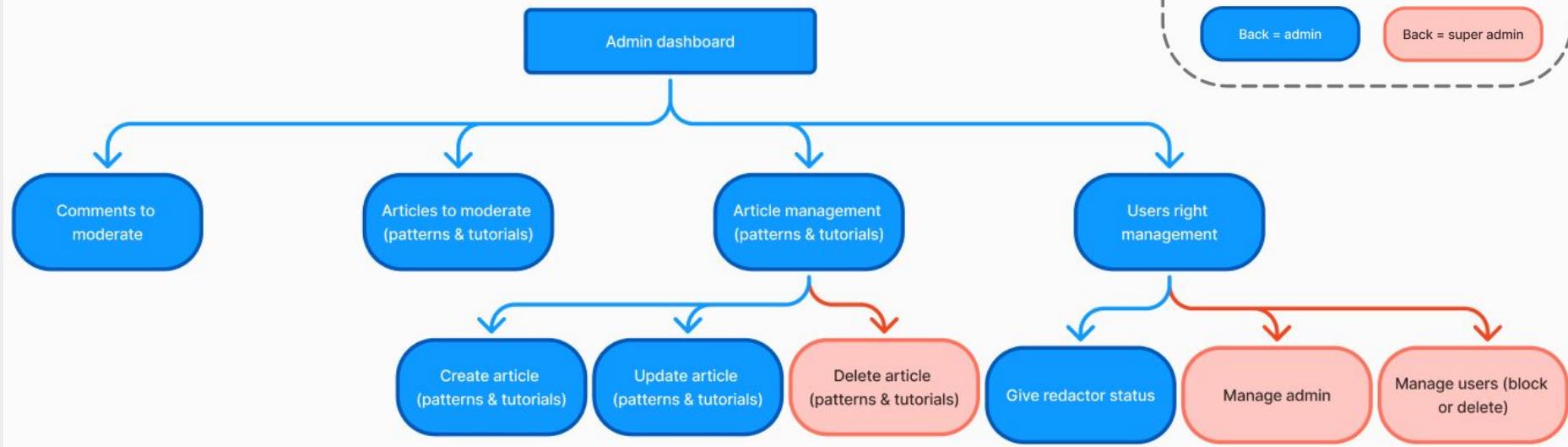
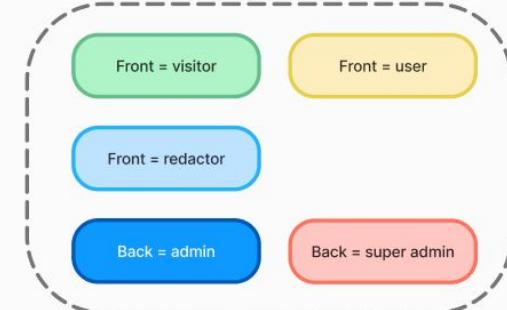


7.1 - Arborescence du site

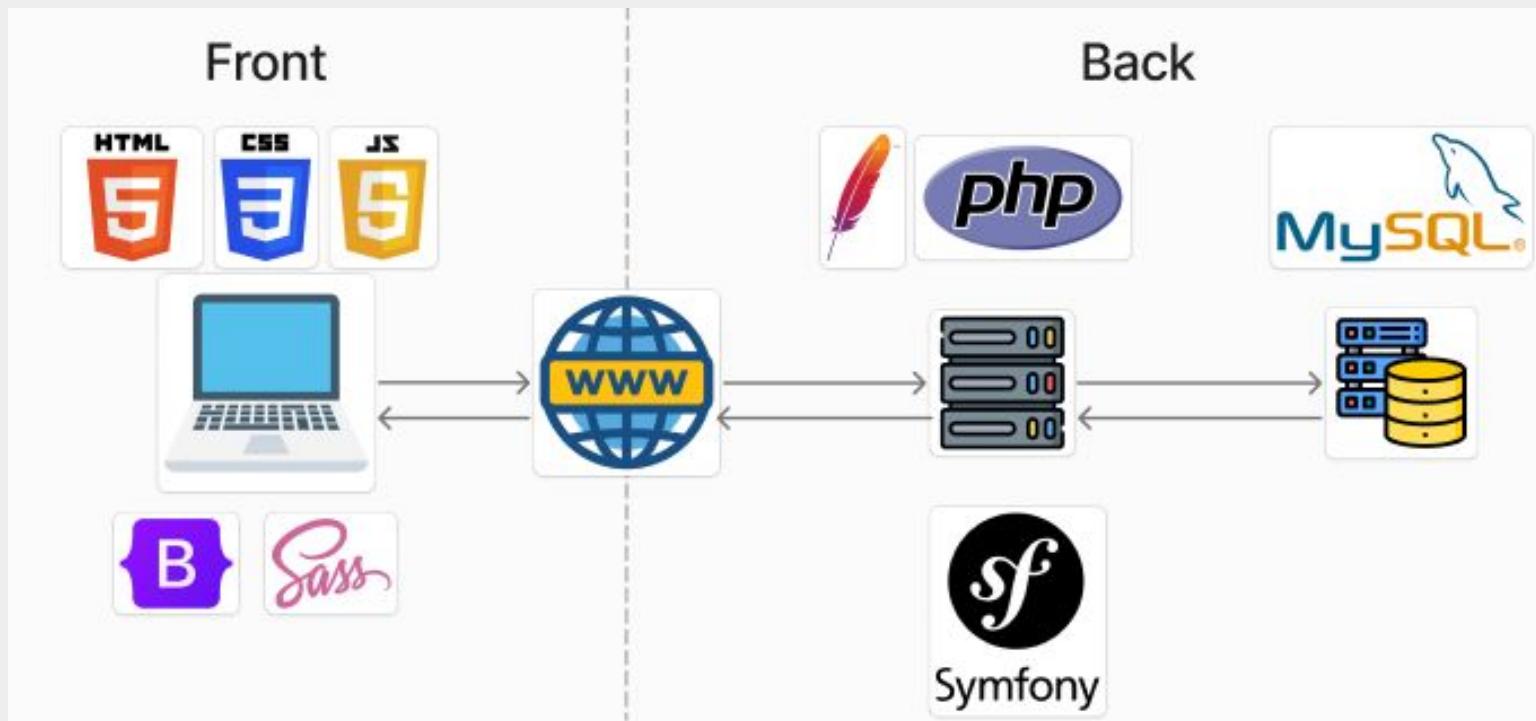


7.2 - Arborescence du site

Légende



8.1 - Architecture technique

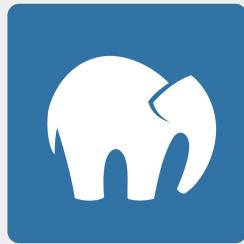


8.2 - Environnement de développement

Versionnement :



Serveur local :



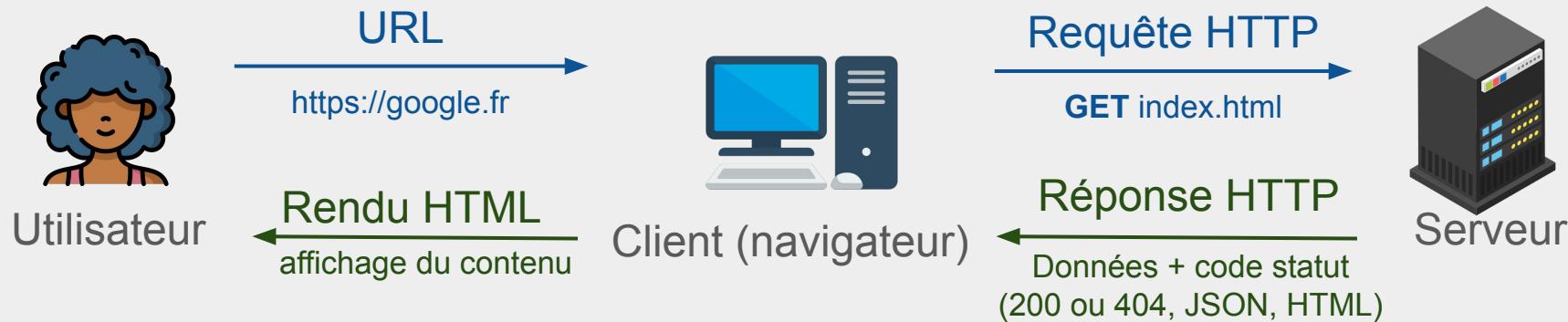
IDE :



9 - Schémas de fonctionnements basiques

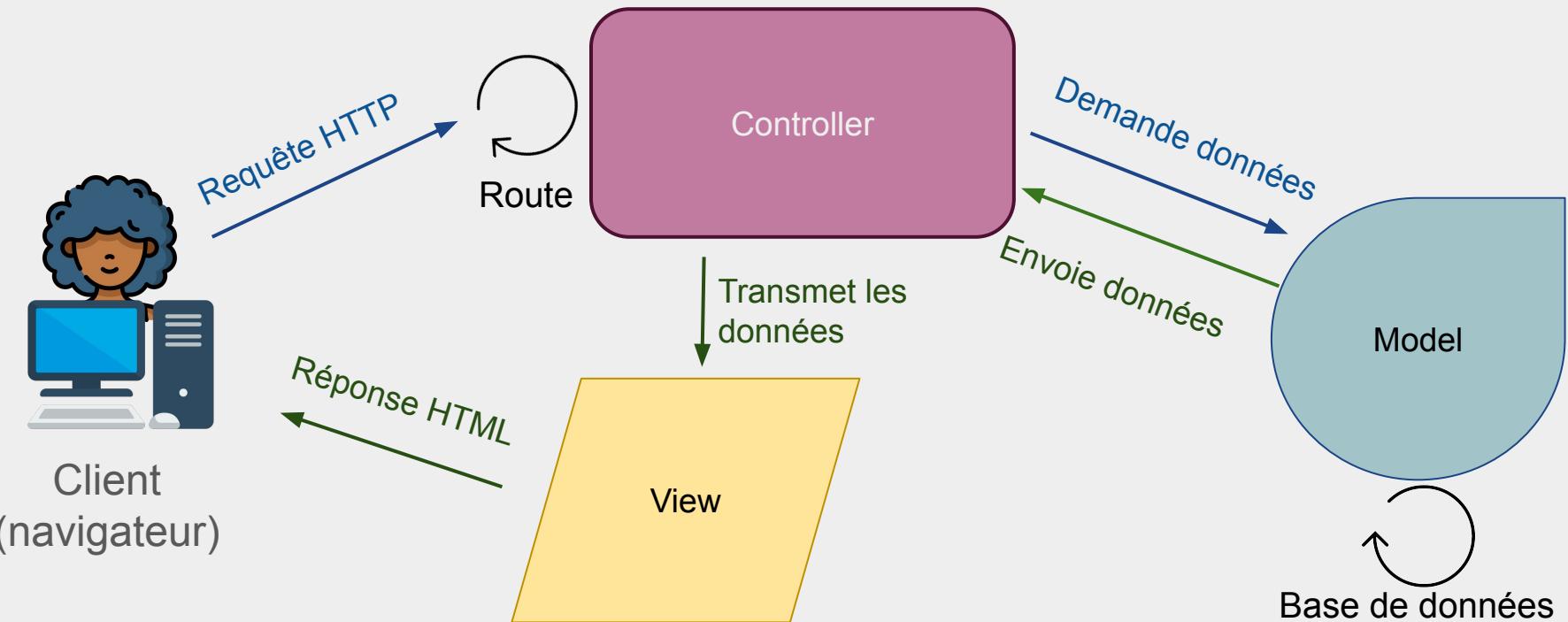
1 ~ Les requêtes HTTP(S)

HTTP = Hyper Text Transfer Protocol (Secure)



9 - Schémas de fonctionnements basiques

2 ~ Architecture Modèle - Vue - Contrôleur (MVC)



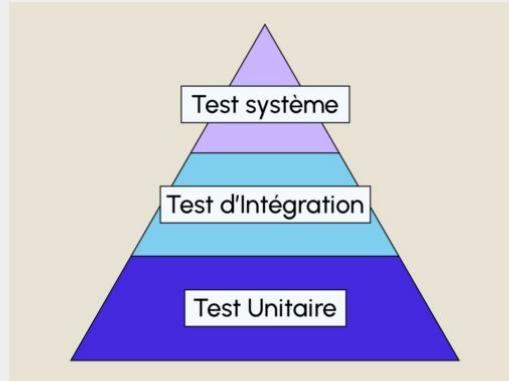
10 - Les Tests

Le **test unitaire** (ou « **T.U.** », ou « **U.T.** » en anglais) est une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme (appelée « **unité** » ou « **module** »).

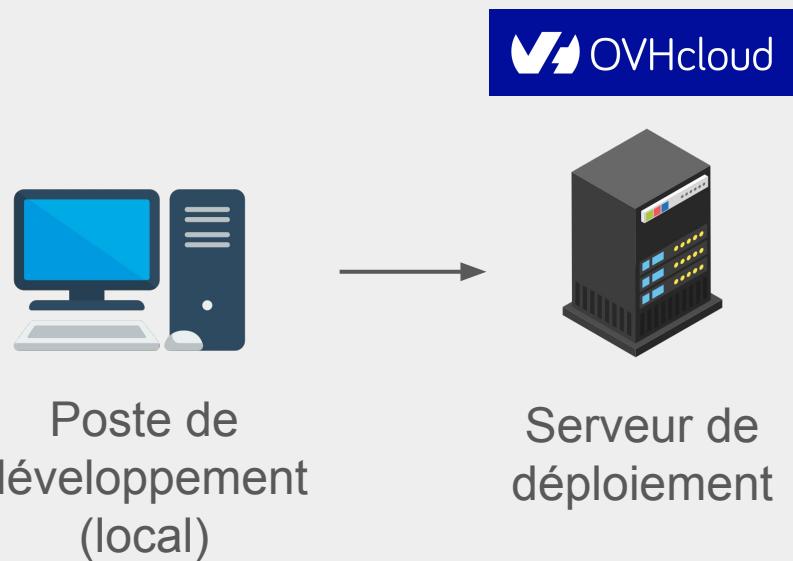
https://fr.wikipedia.org/wiki/Test_unitaire

Le test d'intégration est une phase de tests, précédée par les tests unitaires et généralement suivie par les tests de validation, vérifiant le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme (appelée « **unité** » ou « **module** ») ; dans le test d'intégration, chacun des modules indépendants du logiciel est assemblé et testé dans l'ensemble.

https://wikipedia.org/wiki/Test_d'intégration



11 - Le déploiement



En bare metal (ou serveur dédié)
⇒ serveur qu'il faut configurer

En hébergement mutualisé
⇒ un serveur préconfiguré, en partage avec d'autres

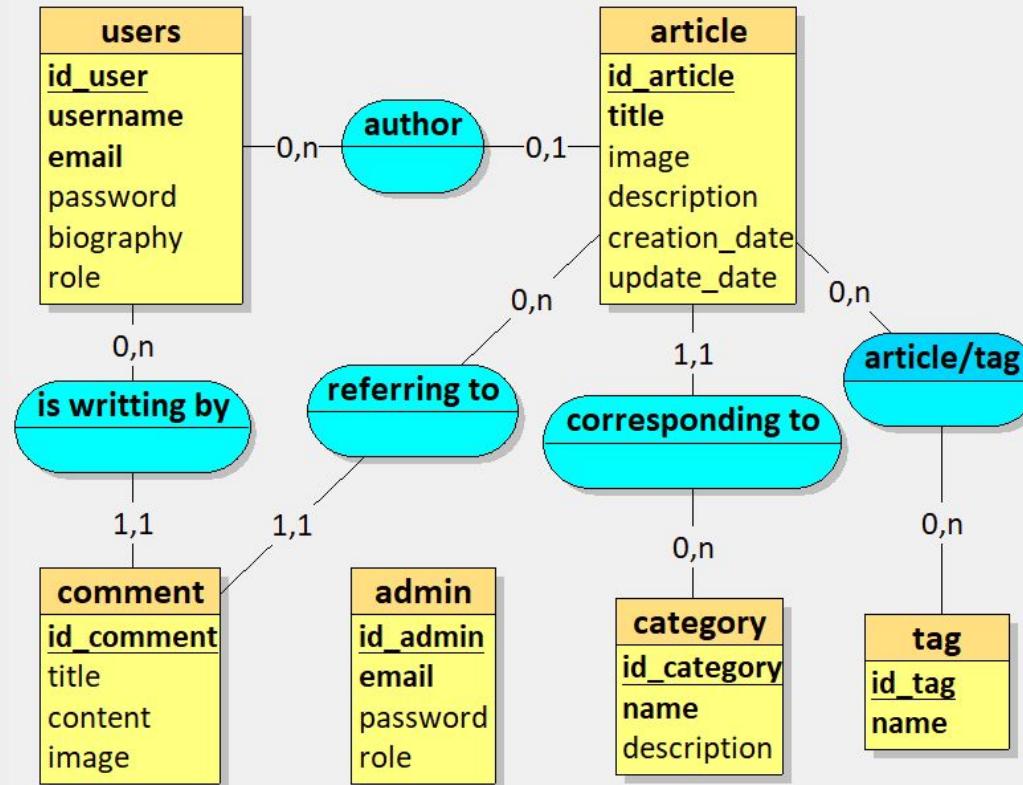
Les VPS
⇒ un fragment de serveur qui agit comme un bare metal

III/ Présentation spécifique



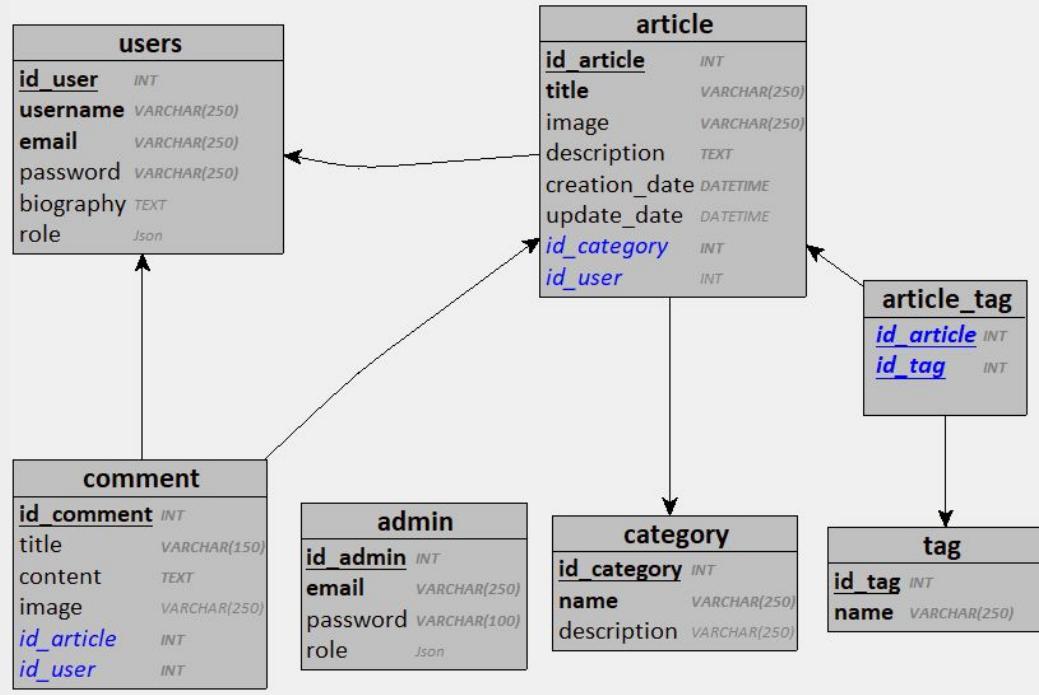
1. Conception de la base de données

1.1 - Modèle conceptuel des données (MCD)



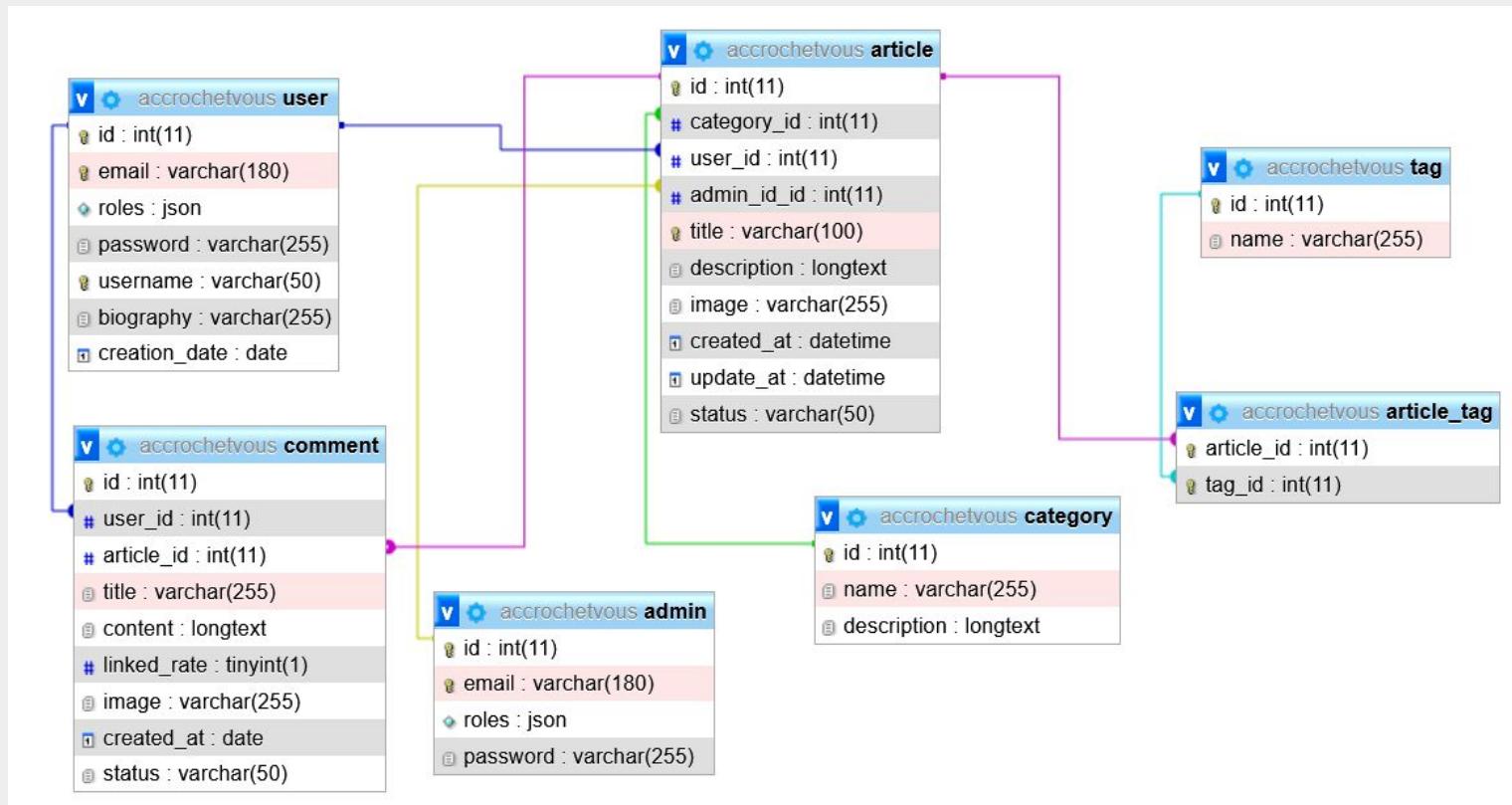
1. Conception de la base de données

1.2 - Modèle logique des données (MLD)



1. Conception de la base de données

1.3 - Base de donnée finale

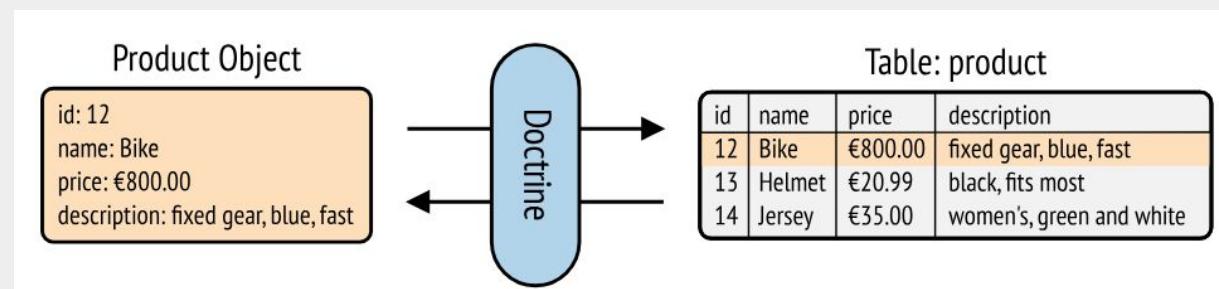


1. Conception de la base de données

1.4 - Utilisation d'un Object Relational Mapping (ORM)

Base de donnée relationnelle ≠ PHP (Symfony) POO

Pour faire le lien = ORM



Avec Symfony l'ORM est Doctrine ⇒ permet de gérer la communication entre Symfony et notre base de données via `# [ORM\ ...]`

Va aussi gérer la sécurité (contrôles pour les injections SQL)

2.1 - Sécurité dans le projet - USERS

- Génération d'une entité User
- Role “User” quand même restrictif
- Mot de passe **hashé** dans la BD
- Fichier `security.yaml` avec le `firewall` renseigné

```
PS C:\MAMP\htdocs\accrochet-vous> php bin/console make:user
```

```
security:
    firewalls:
        user:
            lazy: true
            provider: app_user_provider
            form_login:
                login_path: login
                check_path: login
                default_target_path: home

            logout:
                path: user_logout
```

2.1 - Sécurité dans le projet - ADMINS

- Génération d'une entité Admin
- Role : Admin et Super_Admin
- pattern ^/admin ⇒ accessible seulement si connecté en tant que tel

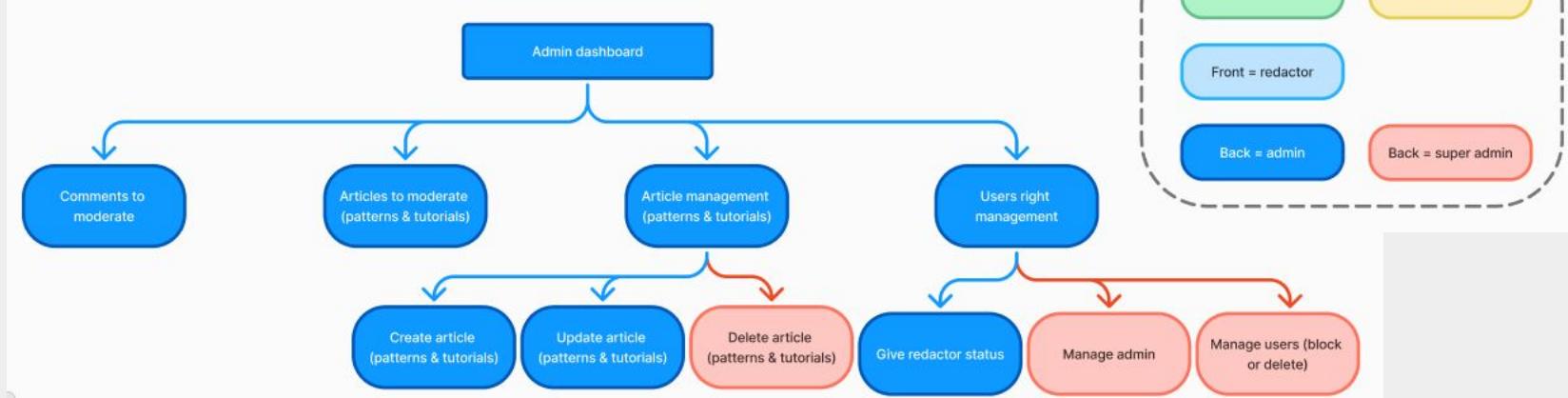
```
security:
    firewalls:
        admin:
            lazy: true
            provider: app_admin_provider
            pattern: ^/admin

            form_login:
                login_path: login_admin
                check_path: login_admin
                default_target_path: admin_dashboard

            logout:
                path: admin_logout
```

```
56     access_control:
57         - { path: ^/admin(?!/login), roles: [ROLE_ADMIN, ROLE_SUPER_ADMIN] }
```

2.3 - Sécurité dans le projet



```
##[IsGranted(new Expression{ expression: 'is_granted("ROLE_SUPER_ADMIN") or is_granted("ROLE_ADMIN")'})]
```

Restriction d'accès selon les rôles ⇒ renvoie à une page 403

3.1 - Exemples de fonctionnalités back

3.1.1 - La recherche : controller

```
12  class SearchResultController extends AbstractController
13  {
14      ▲ Tatiana +1*
15      #[Route(path: "/search", name: "user_search", methods: ["POST", "GET"])]
16      public function showResultSearch(Request $request, ArticleRepository $articleRepository,
17                                         CategoryRepository $categoryRepository): Response {
18          $search = $request->query->get('key', 'search');
19
20          //defines finding categories using the searchCategories method of the CategoryRepo
21          $categoriesFound = $categoryRepository->searchCategories($search);
22          $articlesFound = $articleRepository->searchArticles($search);
23
24          return $this->render('public/search-result.html.twig', [
25              "categoriesFound" => $categoriesFound,
26              "articlesFound" => $articlesFound, "search" => $search]);
27      }
28  }
```

3.1 - Exemples de fonctionnalités back

3.1.2 - La recherche : query-builder

```
12  class ArticleRepository extends ServiceEntityRepository
13
14  use Doctrine\ORM\QueryBuilder;
15
16  /**
17  * @param string $search
18  * @return \Doctrine\ORM\QueryBuilder
19  */
20  public function searchArticles(string $search): QueryBuilder
21  {
22      $qb = $this->createQueryBuilder('a')
23          //start the SQL request with the where
24          //a second condition for search in the description
25          //parameterising the variable 'search'
26          //construct the query
27          //get the result filtered by the DB
28          //SELECT *
29          //FROM article AS a
30          //WHERE a.title LIKE '%search%'
31          //OR a.description LIKE '%search%';
32
33      return $qb
34          //start the SQL request with the where
35          //a second condition for search in the description
36          //parameterising the variable 'search'
37          //construct the query
38          //get the result filtered by the DB
39          //SELECT *
40          //FROM article AS a
41          //WHERE a.title LIKE '%search%'
42          //OR a.description LIKE '%search%';
43
44      }
45  }
```

3.2 - Exemples de fonctionnalités back

3.2.1 - Inscription d'un nouvel user

```
19  class LoginController extends AbstractController
38  #[Route(path: "/inscription", name: "user_inscription")]
39  public function userInscription(UserRepository           $userRepository,
40                      Request                  $request,
41                      EntityManagerInterface    $entityManager,
42                      UserPasswordHasherInterface $passwordHasher,
43                      MailerInterface           $mailer): Response {
44
45      $newUser = new User();
46
47      $newUser->setCreationDate(new \DateTime());
48
49      $formCreateNewUser = $this->createForm(type: UserType::class, $newUser);
50
51
52      $formCreateNewUser->handleRequest($request);
53      $allUsers = $userRepository->findAll();
54
55      //check the other users for the username and email
56      foreach ($allUsers as $user) {
57
58          if ($formCreateNewUser->get('username')->getData() === $user->getUsername()) {
59              $formCreateNewUser->addError(new FormError(message: "Ce pseudo est déjà utilisé"));
56
57          if ($formCreateNewUser->get('email')->getData() === $user->getEmail()) {
58              $formCreateNewUser->addError(new FormError(message: "Cette adresse mail est déjà utilisée"));
59
59      }
```

3.2 - Exemples de fonctionnalités back

3.2.2 - Inscription d'un nouvel user (suite)

```
61     if ($formCreateNewUser->isSubmitted() && $formCreateNewUser->isValid()) {
62         //check the password
63         $plainTextPassword = $formCreateNewUser->get('password')->getData();
64         $confirmPassword = $formCreateNewUser->get('passwordConfirmation')->getData();
65         if ($plainTextPassword === $confirmPassword) {
66             //hash the password with an implementation of UserPasswordHasherInterface
67             $hashedPassword = $passwordHasher->hashPassword(
68                 $newUser,
69                 $plainTextPassword
70             );
71             $newUser->setPassword($hashedPassword);
72             $newUser->setRoles(["ROLE_USER"]);
73
74             $entityManager->persist($newUser);
75             $entityManager->flush();
76
77         }
78     }
79
80     return $this->render('public/inscription.html.twig', ["formView" => $formCreateNewUser->createView()]);
81 }
```

4.1 - Exemples de fonctionnalités front

Le responsive

Media queries [@media](#)

→ pour gérer le responsive

Les breakpoints ou point de rupture

→ [max-width](#) ou [min-width](#)

Médias concernés

→ [only screen](#), [print](#), ou [all](#)

```
1  @import "_services.scss";
2
3  @media only screen and (max-width: $breakPoint) {
4    /* hamburger menu in mobile */
5    .menu.inMobile {
6      display: block;
7      list-style: none;
8      top: 7rem;
9      right: 2rem;
10     z-index: 5000;
11
12    li.hamburgerMenu {
13      █
14      background-color: $darkGreen;
15      height: 0.3rem;
16      width: 4.5rem;
17      margin: 0.8rem 0;
18      transform: rotate(0) translate(0, 0);
19      transition: all 1s;
20
21      &.liActive:first-of-type {
22        transform: rotate(-45deg);
23      }
24      &.liActive:last-of-type {
25        transform: rotate(45deg) translate(-1.5rem, -1.4rem);
26      }
27      &.liActive:nth-of-type(2) {
28        background-color: transparent;
29      }
30    }
31  }
```

4.2 - Exemples de fonctionnalités front

JavaScript

```

1  const hamb :Element = document.querySelector( selectors: '.menu.inMobile.isAbsolute');
2  const burgerMenu :NodeListOf<Element> = document.querySelectorAll( selectors: '.hamburgerMenu')
3  const nav :Element = document.querySelector( selectors: '.headerNavBar.dropdown-menu');
4
5  ▲ Tatiana
6  hamb.addEventListener( type: 'click', listener: () :void => {
7      nav.classList.toggle( token: 'navActive');
8      burgerMenu.forEach( callbackfn: (burger :Element) :void => {
9          burger.classList.toggle( token: 'liActive');
10     })
11 });

```

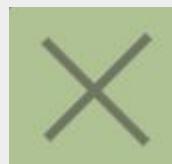
Sélectionner les éléments du DOM avec
querySelector

Ajouter un EventListener à l'évènement “click”

⇒ ajout d'une classe lors du clic et la retirer



Sans la classe “liActive”



Avec la classe “liActive”

5 - Tests unitaires

```
class UniqueFileNameGenerator
{
    3 usages ▲ TatianaAll +1
    public function generateUniqueFileName(string $imageName, string $imageExtension)
    {
        $currentTimestamp = time();
        $nameHashed = hash(algo: 'sha256', $imageName);

        return $imageNewName = $nameHashed . '-' . $currentTimestamp . '.' . $imageExtension;
    }
}
```

```
PS C:\MAMP\htdocs\accrochet-vous> php bin/phpunit src/Services/TestUniqueFilenameGenerator.php
PHPUnit 9.6.22 by Sebastian Bergmann and contributors.

Testing App\Services\TestUniqueFileNameGenerator
.
1 / 1 (100%)

Time: 00:00.535, Memory: 6.00 MB

OK (1 test, 2 assertions)
```

```
7 ▷ class TestUniqueFileNameGenerator extends TestCase {
    ▲ Tatiana *
8 ▷     public function testGenerateUniqueFilename() {
9         //new instance of the class to test
10        $uniqueFileNameGenerator = new UniqueFileNameGenerator();
11
12        //create a variable with the new instance to test and call it with the method
13        $uniqueFilename = $uniqueFileNameGenerator->generateUniqueFileName(imageName: 'hello', imageExtension: 'jpeg');
14
15        //test conditions given to the unique file name generate previously
16        $this->assertStringContainsString(needle: 'jpeg', $uniqueFilename);
17        $this->assertStringContainsString(time(), $uniqueFilename);
18        $this->assertStringNotContainsString(needle: 'hello', $uniqueFilename);
19
20        //php bin/phpunit src/Services/TestUniqueFilenameGenerator.php
21    }
22}
```

Démo du site !

Direction Accrochet-vous



Les modèles

Accrochet-vous

Modèles de crochet gratuit

Se connecter

Recherche



S'inscrire

Dernier ajout



Sac banane avec granny fleur

Pour réaliser ce sac banane il suffit de réaliser 4 granny squares comme ceux qui suivent :
Round 1: Ch 3 (counts as first Dc), Dc 1, ch 1, Repeat *2 dc, ch 1* 7 more times. Tie off and change to th ...

Catégorie : Accessoires

[Voir le modèle](#)



Accrochet-vous
Modèles de crochet gratuit



Dernier ajout



Sac banane avec granny fleur

Catégorie : Accessoires

[Voir le modèle](#)

16 - Axes d'améliorations et fonctionnalités manquantes

- Ajouter les tutoriels et les notations prévues initialement
- Modification d'un article par son auteur (avec validation mais l'article reste en ligne en v1 en attendant la validation)
- Ajout de la sécurité sur la longueur et complexité du mot de passe
- Pouvoir lier des pdf pour télécharger les patterns

Merci de votre attention !



Pagination

utilisation d'un bundle pour paginer ⇒ Paginator

composer require knplabs/knp-paginator-bundle

Rich text ou WYSIWYG avec Trix

What You See Is What You Get = rich text donc interface graphique des text area

Trix

8 - Architecture technique avec les versions

