

PROJET IBOOK

SESSION 2020/2021

GITHUB:HTTPS://GITHUB.COM/RAKTHAK/IBook

KIM Rakthak
8, Allée des hêtres pourpres
77400
Rakthak.kim@gmail.com

KAMRUZZAMAN
Habib Mohammed
65 bd Paul vaillant couturier
94200
Ivry sur seine
K.habib02@gmail.com

SOMMAIRE

REMERCIEMENTS	2
PRÉSENTATION DE L'ÉQUIPE	3
KIM Rakthak:	3
Kamruzzaman Habib:	3
PRÉSENTATION DU PROJET	4
OUTILS DE DÉVELOPPEMENT	4
PHP version 8.0.3	4
SYMFONY version 5.2.9	4
JAVASCRIPT	4
CSS	4
HTML	4
AJAX	4
COMPOSER	4
VS CODE	4
LOOPING	5
XAMPP	5
ANALYSE	5
Diagramme de Cas	5
Diagramme de Classe	6
MAQUETTAGE:	7
Zoning Desktop et Mobile	7
Wireframe Desktop et Mobile	9
BASE DE DONNÉES:	11
MCD (Modèle Conceptuel de Données)	11
MLD (Modèle Logique de Données)	12
SQL	13
Structure de la Base de Données	14
DÉVELOPPEMENT DU SITE WEB:	14
MVC (Model, Vue et Controller)	14
Architectures des dossiers	16
FRONT-END:	17
BACKEND:	18
Connection à la Base de Donnée	18
Réalisations comportant les extraits de code les plus significatifs	20
Construction de la vue pour les like	23

PRÉSENTATION DU JEU D'ESSAI	25
LA SÉCURITÉ DU SITE:	25
LES PAGES DU SITE:	27
CONCLUSION:	30

REMERCIEMENTS

Je tiens à remercier toutes les personnes qui ont contribué au bon déroulement de ma formation.

Tout d'abord un grand merci à toute l'équipe de T2 Technologie, de m'avoir donné la chance et la patience de travailler avec eux .

J'adresse mes remerciements à mes formateurs,

- **Mr Esat BYLYBASHI**
- **Mr Marc CHENEBAUX**
- **Mme Marie BEDO**
- **Mme Dany TAN**

qui m'ont beaucoup aidé tout le long de ce cursus.Et je remercie également toute l'équipe du GRETA de Noisy pour leur accueil et pour mon intégration dans le centre.

KIM RAKTHAK

PRÉSENTATION DE L'ÉQUIPE

KIM Rakthak:

J'ai 35 ans, marié et père de trois enfants. Ancien technicien frigoriste, je suis actuellement en reconversion professionnelle dans le domaine de l'informatique, spécialité WEB. Pourquoi développeur web? car ce métier m'offrira une certaine liberté dans ma vie et que j'adore avoir la satisfaction de résoudre des problèmes.

Kamruzzaman Habib:

Auparavant, j'ai travaillé comme réceptionniste d'hôtel. Actuellement, je suis auto-entrepreneur où je vends des produits personnalisés en ligne. Le secteur du développement Web est un secteur à croissance très rapide et il est prévu qu'il sera très demandé dans les années à venir. J'avais des connaissances préalables à ce sujet et j'ai trouvé que ce serait utile si je pouvais acquérir ces connaissances de manière professionnelle. Et cela donnera également l'opportunité de travailler de manière indépendante.

PRÉSENTATION DU PROJET

L' application IBOOK est un site de bibliothèque, où les visiteurs peuvent lire leurs livres préférés. Pour lire des livres, tous les visiteurs doivent créer un compte et se connecter à leur compte. Une fois connectés, ils peuvent lire et aimer tous les livres qu'ils veulent. Et dans l'application il y a des pages dédiées pour l'administrateur, où il/elle peut ajouter, éditer et supprimer les livres et autres informations.

et bien entendu seul l'administrateur valide a le droit d'apporter les modifications après l'authentification du compte.

OUTILS DE DÉVELOPPEMENT

PHP version 8.0.3: PHP est un langage de script de serveur et un outil puissant pour créer des pages Web dynamiques et interactives.

SYMFONY version 5.2.9: Symfony est un ensemble de composants PHP ainsi qu'un framework MVC libre écrit en PHP. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web.

JAVASCRIPT: JavaScript est un langage de programmation qui permet d'implémenter des mécanismes complexes sur une page web et de la rendre dynamique.

CSS: CSS est l'acronyme de « Cascading Style Sheets » ce qui signifie « feuille de style en cascade ». Le CSS correspond à un langage informatique permettant de mettre en forme des pages web HTML ou XML et contient des éléments de codage.

HTML: L'HyperText Markup Language, HTML, désigne un type de langage informatique descriptif. Il s'agit plus précisément d'un format de données utilisé dans l'univers d'Internet pour la mise en forme des pages Web.

AJAX: AJAX (Asynchronous JavaScript + XML) n'est pas une technologie en soi, mais un terme désignant une « nouvelle » approche utilisant un ensemble de technologies existantes, dont : HTML ou XHTML, les feuilles de styles CSS, JavaScript, le modèle objet de document (DOM), XML, XSLT, et l'objet XMLHttpRequest

COMPOSER: Composer est un logiciel gestionnaire de dépendances libre écrit en PHP. Il permet à ses utilisateurs de déclarer et d'installer les bibliothèques dont le projet principal a besoin. Le développement a débuté en avril 2011.

VSCODE: Visual Studio Code est un éditeur de code source gratuit créé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage,

de la mise en évidence de la syntaxe, de la complétion intelligente du code, des extraits de code, de la refactorisation du code et de Git intégré.

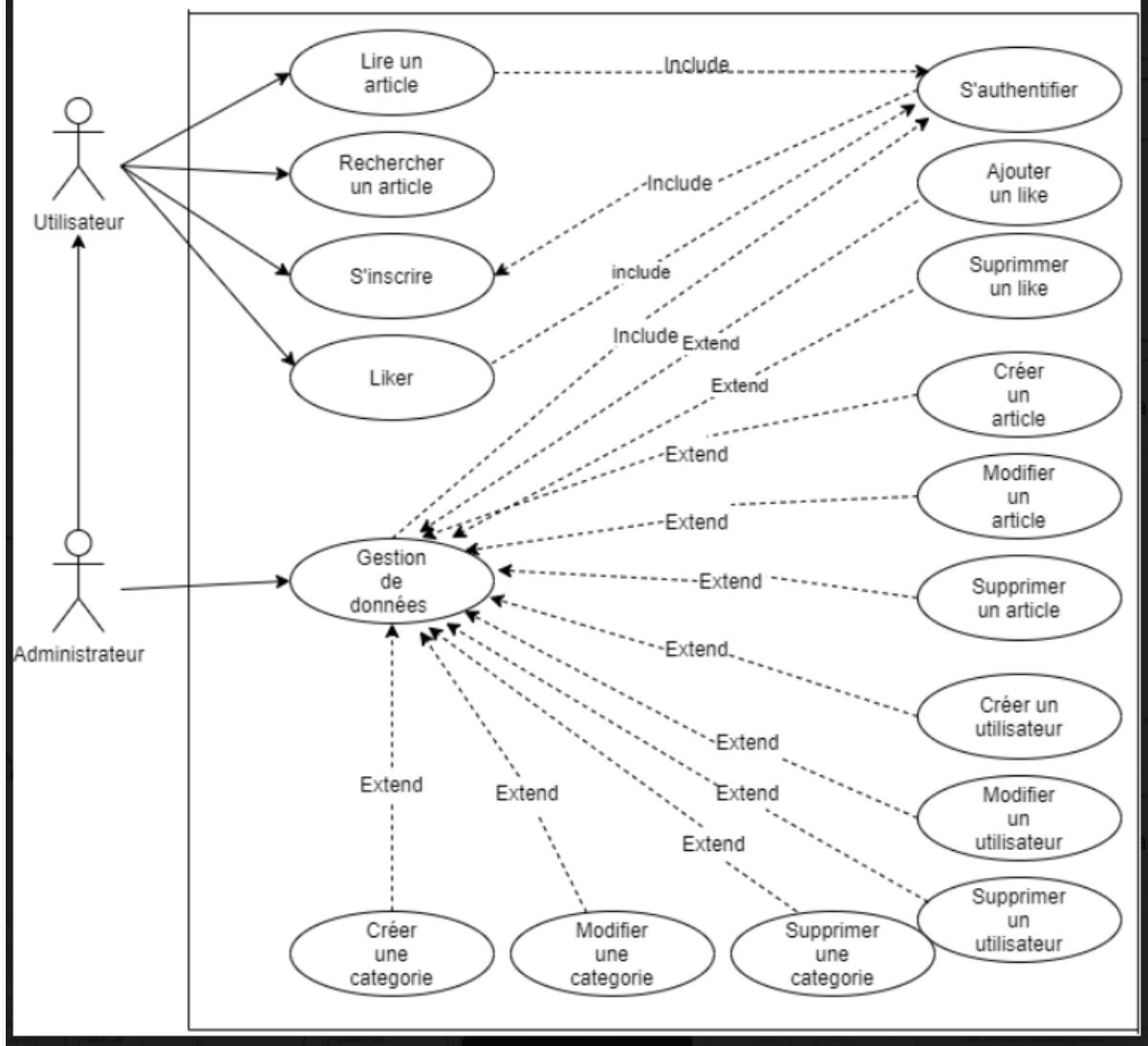
LOOPING: Looping est un logiciel permettant la mise en oeuvre intuitive et très rapide de modèles conceptuels de données (MCD) et la génération automatique temps-réel des modèles logiques (MLD) et des requêtes SQL de création des tables BD correspondantes.

XAMPP: C'est un package de solution de serveur Web multiplateforme gratuit et open source développé par Apache Friends, composé principalement du serveur HTTP Apache, de la base de données MariaDB et d'interpréteurs pour les scripts écrits dans les langages de programmation PHPet Perl

ANALYSE:

[Diagramme de Cas](#)

Diagramme de cas



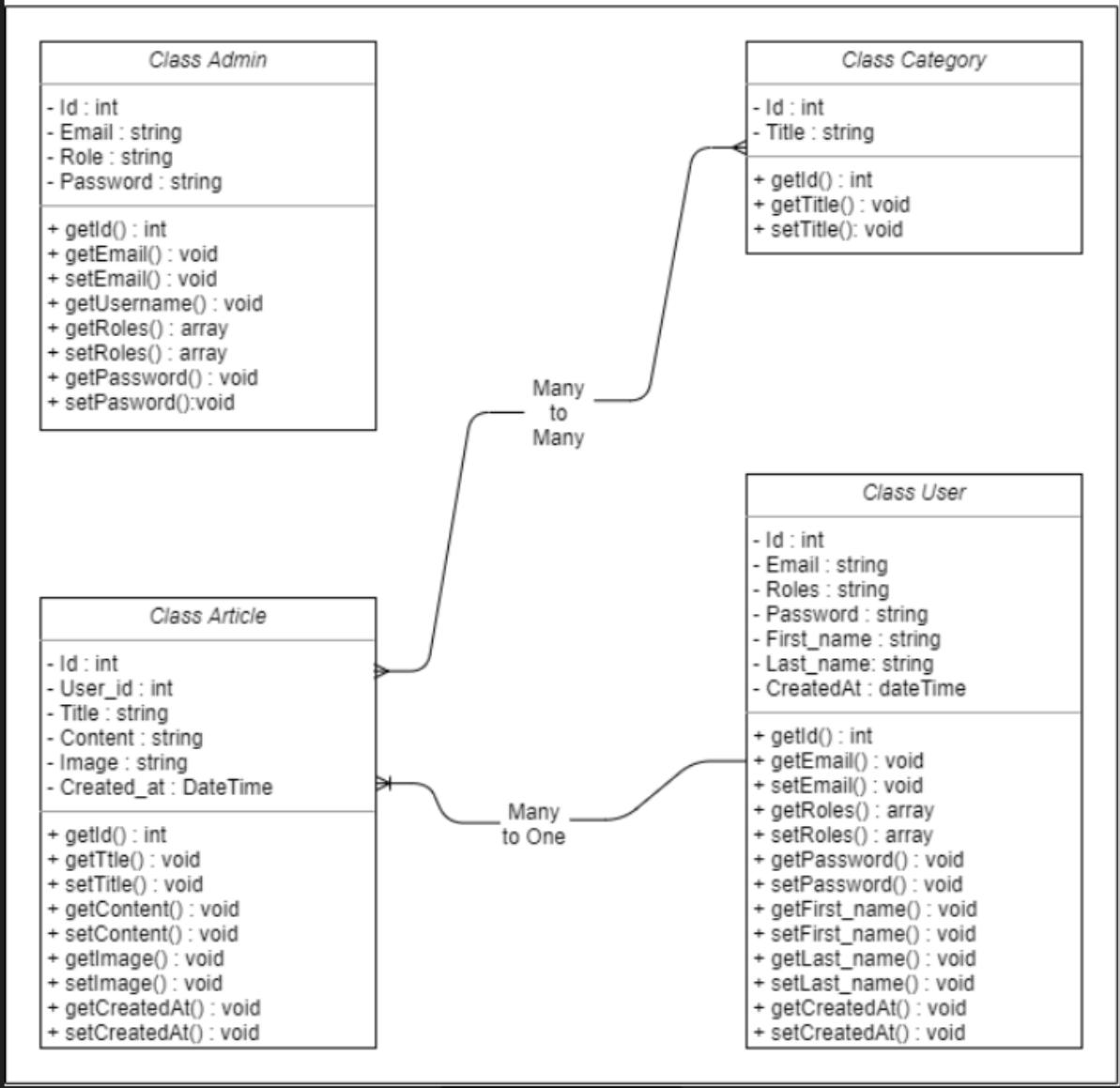
Ici ce diagramme explique la fonctionnalité des exigences d'un utilisateur.

Comme indiqué sur le schéma, il existe 2 types de connexion.

Les flèches pleines représentent les cas d'utilisation d'un utilisateur, sachant qu'un administrateur enregistré peut hériter de tous les droits d'un visiteur (utilisateur). Où les flèches en pointillés représentent les dépendances entre les cas d'utilisation.

Diagramme de Classe

DIAGRAMME DE CLASSE



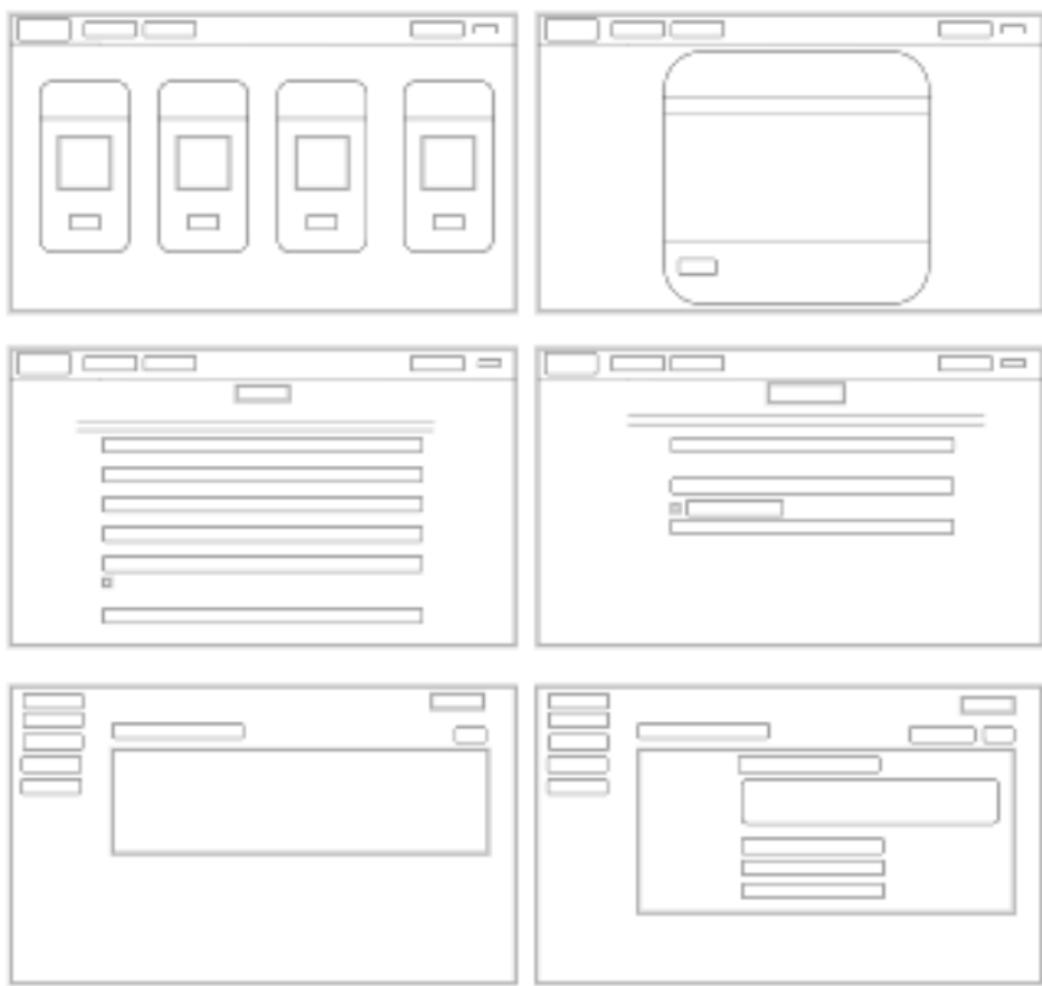
Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que leurs relations.

MAQUETTAGE:

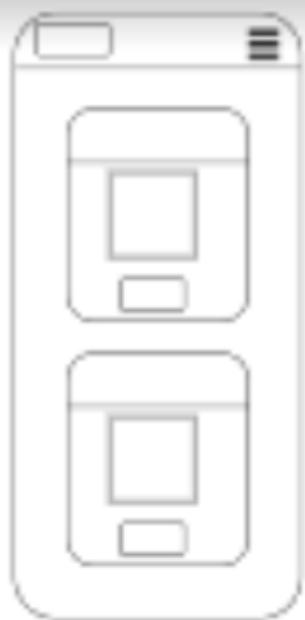
Zoning Desktop et Mobile

Zonning => dans le contexte de la création et du développement d'un site web, le zoning est la pratique qui consiste à "découper" et représenter les différentes zones et les types de contenus qui leur sont affectés pour un type de page du site.

ZONING
DESKTOP

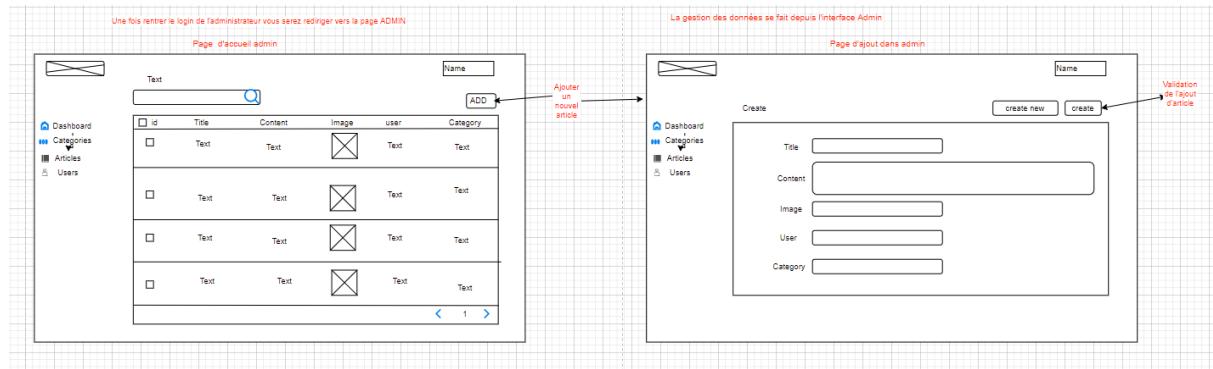
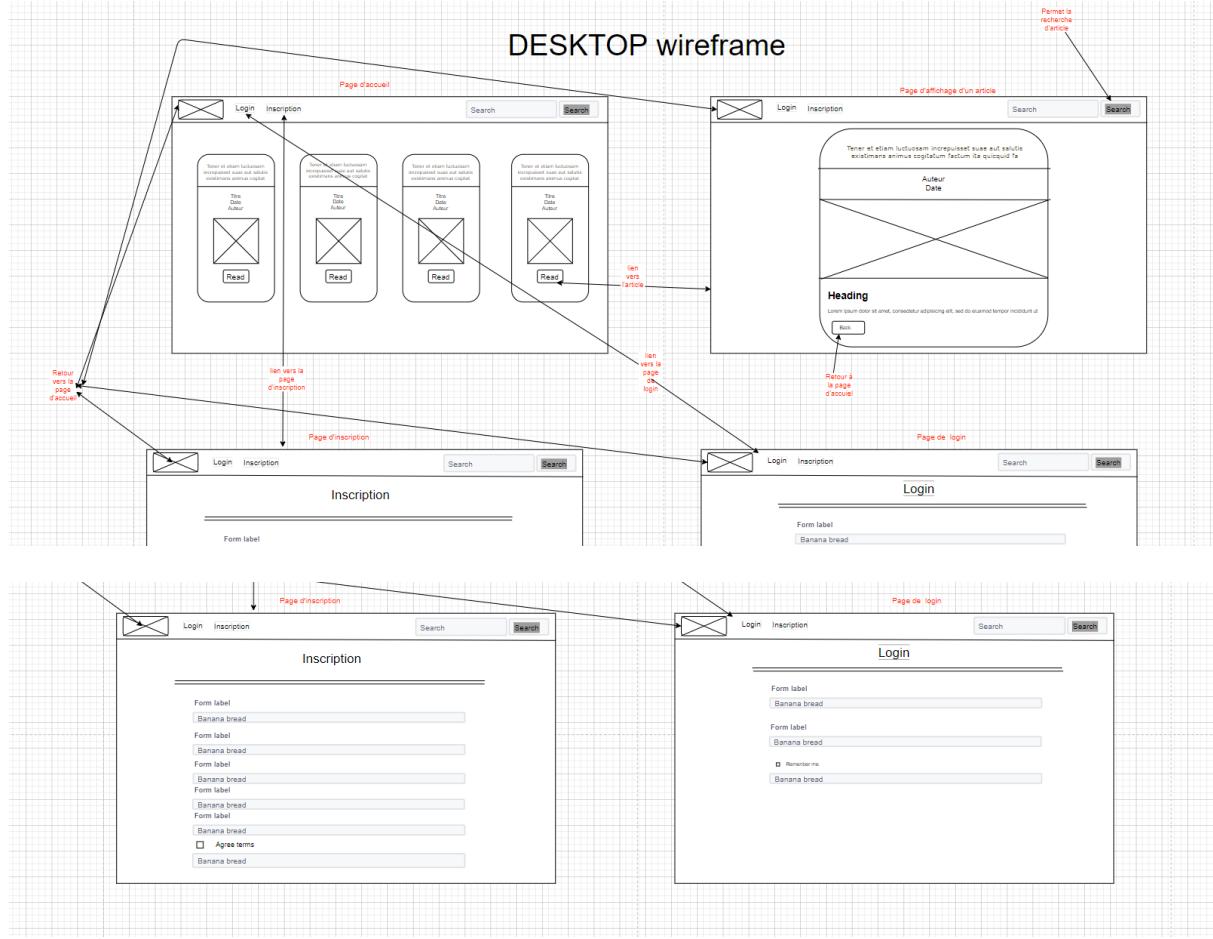


ZONING MOBILE

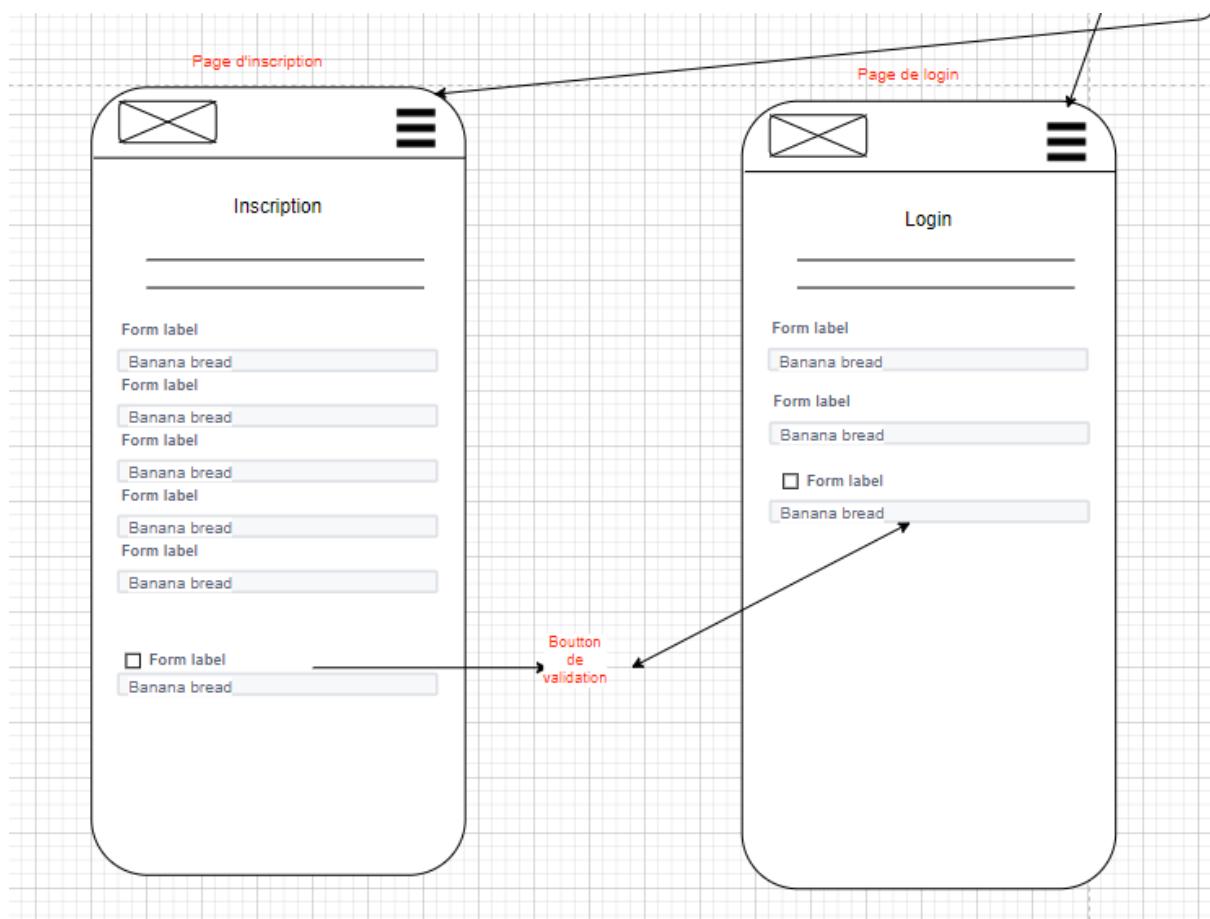
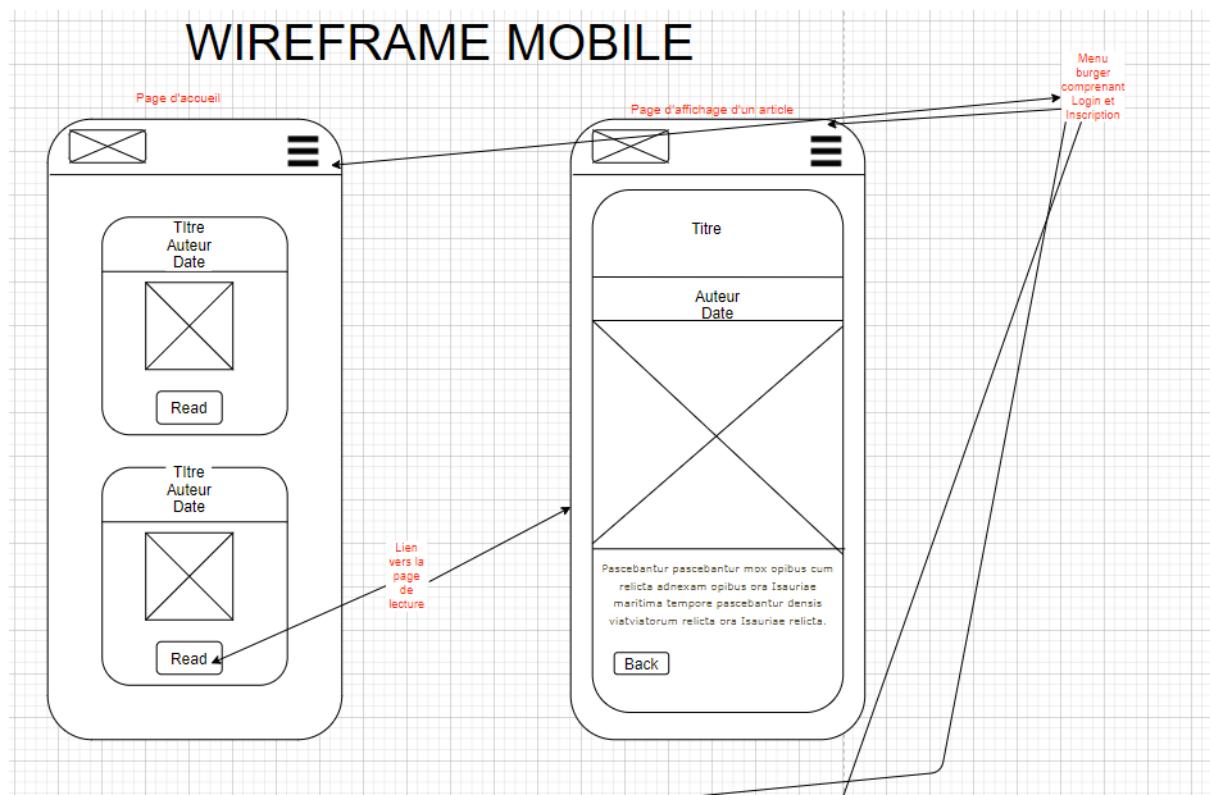


Wireframe Desktop et Mobile

Wireframing => Le wireframe fonctionnel est un schéma utilisé lors de la conception d'une interface utilisateur pour définir les zones et composants qu'elle doit contenir. À partir d'un wireframe peut être réalisée l'interface proprement dite.

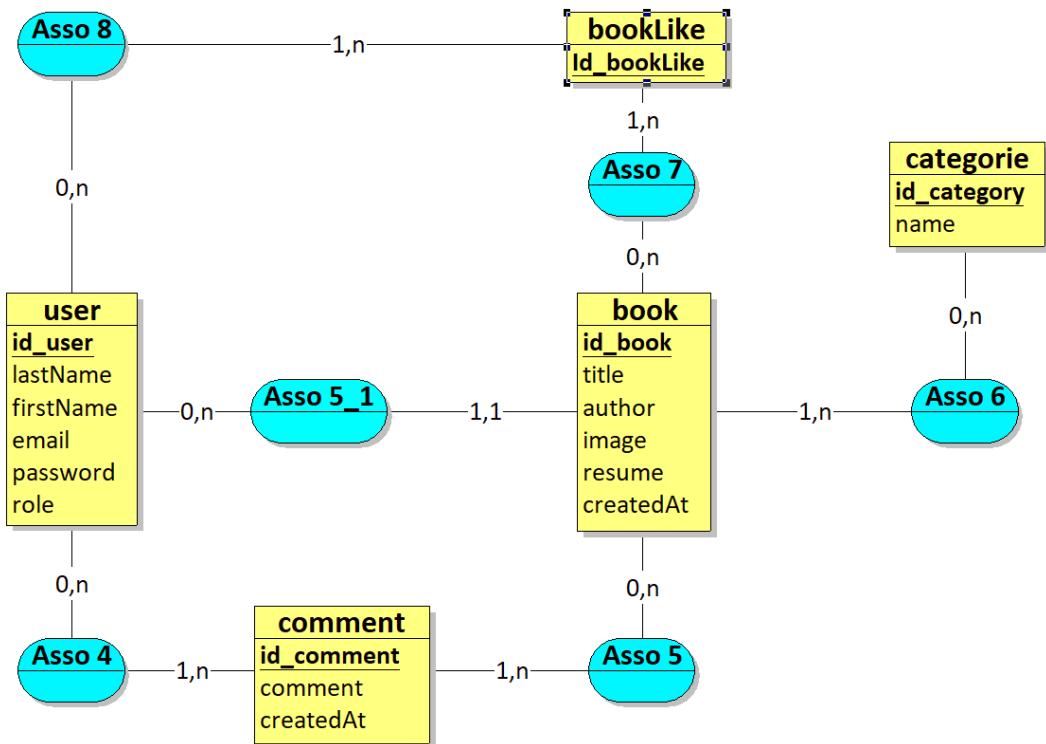


WIREFRAME MOBILE



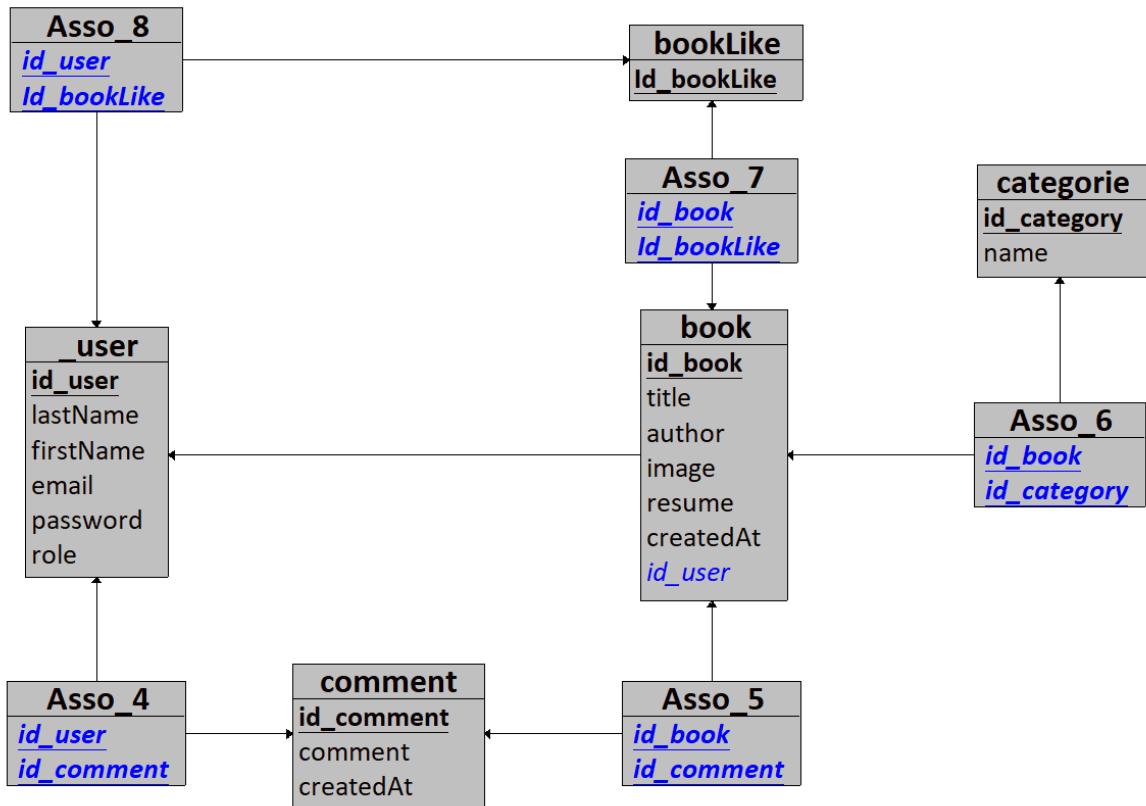
BASE DE DONNÉES:

MCD (Modèle Conceptuel de Données)



MCD => Le modèle conceptuel des données (MCD) a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités.

MLD (Modèle Logique de Données)



MLD => Modélisation logique des données (MLD) Le modèle logique de données est une étape de la conception qui consiste à décrire la structure des données utilisées sans faire référence à un langage de programmation. Il s'agit de préciser le type de données utilisées lors des traitements.

SQL

```


.sql
1 CREATE TABLE _user(
2     id_user INT,
3     lastName VARCHAR(50) NOT NULL,
4     firstName VARCHAR(50) NOT NULL,
5     email VARCHAR(50) NOT NULL,
6     password VARCHAR(50) NOT NULL,
7     role VARCHAR(50) NOT NULL,
8     PRIMARY KEY(id_user)
9 );
10
11 CREATE TABLE book(
12     id_book INT,
13     title VARCHAR(50) NOT NULL,
14     author VARCHAR(50) NOT NULL,
15     image VARCHAR(50) NOT NULL,
16     resume VARCHAR(250) NOT NULL,
17     createdat DATE NOT NULL,
18     id_user INT NOT NULL,
19     PRIMARY KEY(id_book),
20     FOREIGN KEY(id_user) REFERENCES _user(id_user)
21 );
22
23 CREATE TABLE comment(
24     id_comment INT,
25     comment TEXT,
26     createdat DATE NOT NULL,
27     PRIMARY KEY(id_comment)
28 );
29
30 CREATE TABLE categorie(
31     id_category INT,
32     name VARCHAR(50) NOT NULL,
33     PRIMARY KEY(id_category)
34 );
35
36 CREATE TABLE bookLike(
37     Id_bookLike INT,
38     PRIMARY KEY(Id_bookLike)


```



```


.sql
40
41 CREATE TABLE Asso_4(
42     id_user INT,
43     id_comment INT,
44     PRIMARY KEY(id_user, id_comment),
45     FOREIGN KEY(id_user) REFERENCES _user(id_user),
46     FOREIGN KEY(id_comment) REFERENCES comment(id_comment)
47 );
48
49 CREATE TABLE Asso_5(
50     id_book INT,
51     id_comment INT,
52     PRIMARY KEY(id_book, id_comment),
53     FOREIGN KEY(id_book) REFERENCES book(id_book),
54     FOREIGN KEY(id_comment) REFERENCES comment(id_comment)
55 );
56
57 CREATE TABLE Asso_6(
58     id_book INT,
59     id_category INT,
60     PRIMARY KEY(id_book, id_category),
61     FOREIGN KEY(id_book) REFERENCES book(id_book),
62     FOREIGN KEY(id_category) REFERENCES categorie(id_category)
63 );
64
65 CREATE TABLE Asso_7(
66     id_book INT,
67     Id_booklike INT,
68     PRIMARY KEY(id_book, Id_bookLike),
69     FOREIGN KEY(id_book) REFERENCES book(id_book),
70     FOREIGN KEY(Id_bookLike) REFERENCES bookLike(Id_bookLike)
71 );
72
73 CREATE TABLE Asso_8(
74     id_user INT,
75     Id_bookLike INT,
76     PRIMARY KEY(id_user, Id_bookLike),
77     FOREIGN KEY(id_user) REFERENCES user(id_user),
78     FOREIGN KEY(Id_bookLike) REFERENCES bookLike(Id_bookLike)


```

```


    return '';
}

public function up(Schema $schema): void
{
    // this up() migration is auto-generated, please modify it to your needs
    $this->addSql('CREATE TABLE book (id INT AUTO_INCREMENT NOT NULL, user_id INT DEFAULT NULL, title VARCHAR(255) NOT NULL, author VARCHAR(255) NOT NULL, image VARCHAR(255) NOT NULL, resume VARCHAR(250) NOT NULL, createdat DATE NOT NULL, id_user INT NOT NULL, PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci`');
    $this->addSql('CREATE TABLE book_category (book_id INT NOT NULL, category_id INT NOT NULL, INDEX IDX_1FB30F9816A2B381 (book_id), INDEX IDX_1FB30F9812469DE2 (category_id));
    $this->addSql('CREATE TABLE category (id INT AUTO_INCREMENT NOT NULL, name VARCHAR(255) NOT NULL, PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci`');
    $this->addSql('CREATE TABLE comment (id INT AUTO_INCREMENT NOT NULL, book_id INT DEFAULT NULL, user_id INT DEFAULT NULL, content VARCHAR(255) NOT NULL, content_longtext LONGTEXT NOT NULL COMMENT \'\\DC2type:json\', createdat DATE NOT NULL, id_user INT NOT NULL, email VARCHAR(180) NOT NULL, roles LONGTEXT NOT NULL COMMENT \'\\DC2type:json\' , password VARCHAR(255) NOT NULL, PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci`');
    $this->addSql('ALTER TABLE book ADD CONSTRAINT FK_CBE5A331A76ED395 FOREIGN KEY (user_id) REFERENCES user (id)');
    $this->addSql('ALTER TABLE book_category ADD CONSTRAINT FK_1FB30F9816A2B381 FOREIGN KEY (book_id) REFERENCES book (id) ON DELETE CASCADE');
    $this->addSql('ALTER TABLE book_category ADD CONSTRAINT FK_1FB30F9812469DE2 FOREIGN KEY (category_id) REFERENCES category (id) ON DELETE CASCADE');
    $this->addSql('ALTER TABLE comment ADD CONSTRAINT FK_9474526C16A2B381 FOREIGN KEY (book_id) REFERENCES book (id)');
    $this->addSql('ALTER TABLE comment ADD CONSTRAINT FK_9474526CA76ED395 FOREIGN KEY (user_id) REFERENCES user (id)');
}

public function down(Schema $schema): void
{
    // this down() migration is auto-generated, please modify it to your needs
    $this->addSql('ALTER TABLE book_category DROP FOREIGN KEY FK_1FB30F9816A2B381');
    $this->addSql('ALTER TABLE comment DROP FOREIGN KEY FK_9474526C16A2B381');
    $this->addSql('ALTER TABLE book_category DROP FOREIGN KEY FK_1FB30F9812469DE2');
    $this->addSql('ALTER TABLE book DROP FOREIGN KEY FK_CBE5A331A76ED395');
    $this->addSql('ALTER TABLE comment DROP FOREIGN KEY FK_9474526CA76ED395');
    $this->addSql('DROP TABLE book');
    $this->addSql('DROP TABLE book_category');
    $this->addSql('DROP TABLE category');
    $this->addSql('DROP TABLE comment');
    $this->addSql('DROP TABLE user');
}


```

SQL => C'est un langage spécifique à un domaine utilisé dans la programmation et conçu pour gérer les données détenues dans un système de gestion de base de données relationnelle, ou pour le traitement de flux dans un système de gestion de flux de données relationnelles.

Structure de la Base de Données

Table	Action	Lignes	Type	Interclassement	Taille	Perte
book	Parcourir Structure Rechercher Insérer Vider Supprimer	41	InnoDB	utf8mb4_unicode_ci	64,0 kio	-
book_category	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	48,0 kio	-
book_like	Parcourir Structure Rechercher Insérer Vider Supprimer	199	InnoDB	utf8mb4_unicode_ci	48,0 kio	-
category	Parcourir Structure Rechercher Insérer Vider Supprimer	4	InnoDB	utf8mb4_unicode_ci	16,0 kio	-
comment	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	48,0 kio	-
doctrine_migration_versions	Parcourir Structure Rechercher Insérer Vider Supprimer	2	InnoDB	utf8_unicode_ci	16,0 kio	-
user	Parcourir Structure Rechercher Insérer Vider Supprimer	12	InnoDB	utf8mb4_unicode_ci	32,0 kio	-
Somme						
258 InnoDB utf8mb4_general_ci 272,0 kio						

BDD => Une base de données est un ensemble structuré et organisé permettant le stockage de grandes quantités d'informations afin d'en faciliter l'exploitation (ajout, mise à jour, recherche de données). Nous utilisons ici MySQL, sur l'interface de phpMyAdmin.

DÉVELOPPEMENT DU SITE WEB:

MVC (Model, Vue et Controller)

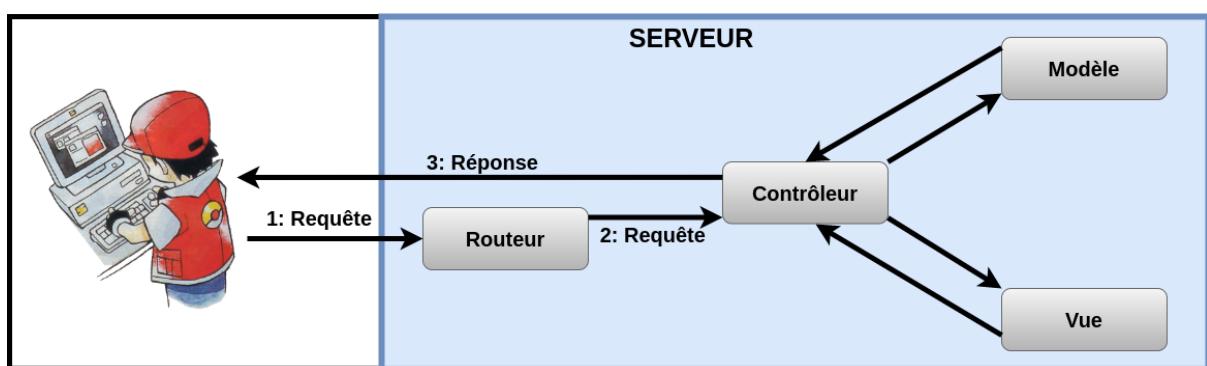
Le pattern MVC permet de bien organiser son code source. Il nous aide à savoir quels fichiers créer, mais surtout à définir leur rôle. Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts, comme l'explique la description qui suit.

Modèle: cette partie gère les données de notre site. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. On y trouve donc les requêtes SQL.

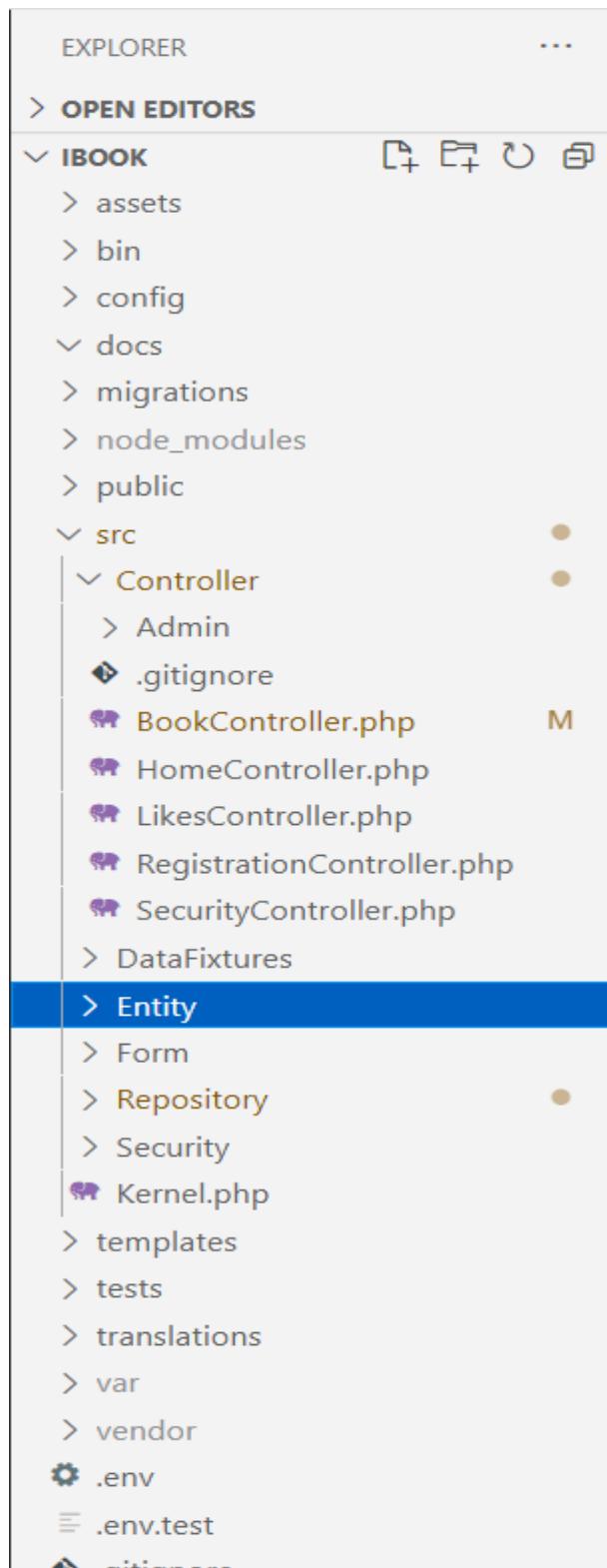
Vue: cette partie se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples, pour afficher par exemple la liste des messages des forums.

Contrôleur: cette partie gère la logique du code qui prend des décisions. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. Le

contrôleur contient exclusivement du PHP. C'est notamment lui qui détermine si le visiteur a le droit de voir la page ou non (gestion des droits d'accès).



Architectures des dossiers



Le MVC dans le framework symfony est constitué ainsi, les templates font partis de la vue, les models dans les entités et les controllers dans leurs propres dossiers controllers.

FRONT-END:

Dans le framework symfony il y a plusieurs bundles que j'ai du télécharger pour le front. Tout d'abord WEBPACK qui me permet d'organiser et de stocker mon CSS ainsi que mon JS, et Bootstrap 5 qui un framework de style.

Pour le front nous utilisons des fichiers au format twig qui est un moteur de template pour le langage PHP.

```
1  {% extends 'base.html.twig' %} 
2 
3  {% block title %}IBOOK
4  {% endblock %}
5 
6  {% block body %}
7 
8 
9  <nav class="navbar navbar-expand-lg navbar-light bg-secondary">
10     <div class="container-fluid">
11         <a class="navbar-brand" href="#">IBOOK</a>
12         <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls=""
13             <span class="navbar-toggler-icon"></span>
14         </button>
15         <div class="collapse navbar-collapse" id="navbarSupportedContent">
16             <ul class="navbar-nav me-auto mb-2 mb-lg-0">
17 
18                 <li class="nav-item">
19                     <a class="nav-link active" aria-current="page" href="{{ path ('allBooks') }}>LIVRES</a>
20                 </li>
21                 <li class="nav-item">
22                     <a class="nav-link active" aria-current="page" href="{{ path ('app_logout') }}>DECONNEXION</a>
23                 </li>
24                 <li class="nav-item">
25                     <a class="nav-link active" aria-current="page" href="{{ path ('admin') }}>ADMINISTRATION</a>
26                 </li>
27             </ul>
28 
29         </div>
30     </div>
31 </nav>
```

```

37 <div class="row justify-content-md-center mt-4 cardHeight ">
38   {% for book in lastBooks %}
39     <div class="col-3 mb-4">
40       <div class="card mb-4 cardHeight">
41         <h4 class="card-header text-center bg-primary">{{book.title}}</h4>
42         <div class="card-body">
43           <div class="card-body text-center">
44             <h6 class="text-center">Ecrit par
45             | {{book.author}}
46             </h6>
47             {% h6 class="card-subtitle text-muted ">from the</h6 %}
48             <p class="card-subtitle text-muted ">Created on
49             | {{book.createdAt | date('d/m/Y' ) }}</p>
50           </div>
51           
52           <div class="card-body text-center ">
53             <a href="{{path('read', {id: book.id })}}" class="card-link btn btn-primary">LIRE</a>
54           </div>
55           <a href="{{ path('liker', {'id':book.id})}}" class="btn btn-link js-like">
56             {% if app.user and book.likeByUser(app.user) %}
57               <i class="fas fa-thumbs-up"></i>
58             {% else %}
59               <i class="far fa-thumbs-up"></i>
60             {% endif %}
61             <span class="js-likes">{{ book.likes|length }}</span>
62             <span class="js-label">j'aime</span>
63           </a>
64         </div>
65       </div>
66     </div>
67   {% endfor %}
68
69

```

BACKEND:

Connection à la Base de Donnée

Création de la base de données. Pour créer la base de données, Symfony nous met à disposition une commande de terminal par l'intermédiaire de l'extension Doctrine . Les informations de connexion s'inscrivent dans le fichier .env situé à la racine du projet. Voici la ligne de ce fichier qui correspond à notre connexion.

DATABASE_URL="mysql://root:@127.0.0.1:3306/IBook?Version du serveur : 10.4.18-MariaDB"

```
⚙ .env
1 # In all environments, the following files are loaded if they exist,
2 # the latter taking precedence over the former:
3 #
4 # * .env           contains default values for the environment variables needed by the app
5 # * .env.local     uncommitted file with local overrides
6 # * .env.$APP_ENV  committed environment-specific defaults
7 # * .env.$APP_ENV.local uncommitted environment-specific overrides
8 #
9 # Real environment variables win over .env files.
10 #
11 # DO NOT DEFINE PRODUCTION SECRETS IN THIS FILE NOR IN ANY OTHER COMMITTED FILES.
12 #
13 # Run "composer dump-env prod" to compile .env files for production use (requires symfony/flex >=1.2).
14 # https://symfony.com/doc/current/best\_practices.html#use-environment-variables-for-infrastructure-configuration
15
16 ##### symfony/framework-bundle #####
17 APP_ENV=dev
18 APP_SECRET=c8fd890cba7d8da6a9ba1f52a066a80a
19 #####< symfony/framework-bundle #####
20
21 #####> symfony/mailer #####
22 # MAILER_DSN=smtp://localhost
23 #####< symfony/mailer #####
24
25 #####> doctrine/doctrine-bundle #####
26 # Format described at https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html#connecting-using-a-url
27 # IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml
28 #
29 # DATABASE_URL="sqlite:///kernel.project_dir%/var/data.db"
30 DATABASE_URL="mysql://root:@127.0.0.1:3306/IBook?Version du serveur : 10.4.18-MariaDB"
31 # DATABASE_URL="postgresql://db_user:db_password@127.0.0.1:5432/db_name?serverVersion=13&charset=utf8"
32 #####< doctrine/doctrine-bundle #####
--
```

Réalisations comportant les extraits de code les plus significatifs

```
1  <?php
2
3  namespace App\Repository;
4
5  use App\Entity\Book;
6  use Doctrine\Bundle\DoctrineBundle\Repository\ServiceEntityRepository;
7  use Doctrine\Persistence\ManagerRegistry;
8
9  /**
10  * @method Book|null find($id, $lockMode = null, $lockVersion = null)
11  * @method Book|null findOneBy(array $criteria, array $orderBy = null)
12  * @method Book[]    findAll()
13  * @method Book[]    findBy(array $criteria, array $orderBy = null, $limit = null, $offset = null)
14  */
15 class BookRepository extends ServiceEntityRepository
16 {
17     public function __construct(ManagerRegistry $registry)
18     {
19         parent::__construct($registry, Book::class);
20     }
21
22
23     /**
24      * @return Book[] Returns an array of Book objects
25      */
26     //retourne les dernieres publications
27     public function lastBook()
28     {
29         return $this->createQueryBuilder('k')
30             ->orderBy('k.id', 'DESC')
31             ->setMaxresults(12)
32             ->getQuery()
33             ->getResult();
34     }
35 }
```

On crée ici une public function qu'on appelle lastBook dans le dossier bookRepository, on lui fait une commande en lui disant d'ordonnées les books par leurs id en ordre décroissant. Ensuite j'ajoute une méthode qui récupère seulement 12 livres "setMaxresults" enfin je récupère le résultat.

```

1 <?php
2
3 namespace App\Controller;
4
5 use App\Entity\Book;
6 use App\Repository\BookRepository;
7 use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
8 use Symfony\Component\HttpFoundation\Response;
9 use Symfony\Component\Routing\Annotation\Route;
10
11 class BookController extends AbstractController
12 {
13     //route vers les 12 derniers ajouts
14     #[Route('/book', name: 'book')]
15     public function index(BookRepository $bookRepository): Response
16     {
17         return $this->render('book/book.html.twig', [
18             'lastBooks' => $bookRepository->lastBook(),
19         ]);
20     }
21 }
```

je retourne ensuite dans mon controller pour créer ma fonction qui va me retourner les 12 derniers ajouts dans ma vue.

```

30
31 <div class="row justify-content-md-center mt-4 cardHeight " >
32     {% for book in lastBooks %}
33         <div class="col-3 mb-4" >
34             <div class="card mb-4 cardHeight" >
35                 <h4 class="card-header text-center bg-primary">{{book.title}}</h4>
36                 <div class="card-body text-center" >
37                     <div class="card-body text-center" >
38                         <h6 class="text-center">Ecrit par
39                         | {{book.author}}</h6>
40                         {% <h6 class="card-subtitle text-muted ">from the</h6> %}
41                         <p class="card-subtitle text-muted" >Created on
42                         | {{book.createdAt | date('d/m/Y' ) }}</p>
43                     </div>
44                     
45                     <div class="card-body text-center " >
46                         <a href="{{path('read', {'id': book.id })}}" class="card-link btn btn-primary" >LIRE</a>
47                     </div>
48                     <a href="{{ path('liker', {'id':book.id})}}" class="btn btn-link js-like" >
49                         {% if app.user and book.likeByUser(app.user) %}
50                             <i class="fas fa-thumbs-up"></i>
51                         {% else %}
52                             <i class="far fa-thumbs-up"></i>
53                         {% endif %}
54                         <span class="js-likes">{{ book.likes|length }}</span>
55                         <span class="js-label">j'aime</span>
56                     </a>
57                 </div>
58             </div>
59         <% endfor %>
60     </div>
61     {% endblock %}
```

2ème fonction:

```
1  <?php
2
3  namespace App\Controller;
4
5
6  use App\Entity\Book;
7  use App\Entity\BookLike;
8  use App\Repository\BookLikeRepository;
9  use Doctrine\Persistence\ObjectManager;
10 use Doctrine\ORM\EntityManagerInterface;
11 use Symfony\Component\HttpFoundation\Response;
12 use Symfony\Component\Routing\Annotation\Route;
13 use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
14
15
16
17 class LikesController extends AbstractController
18 {
19     /**
20      * cette fonction permet de liké ou disliké un book
21      *
22      */
23     #[Route('/book/{id}/like', name: 'liker')]
24     public function like(Book $book, EntityManagerInterface $manager, BookLikeRepository $bookLikeRepository): Response
25     {
26         $user = $this->getUser();
27         if (!$user) return $this->json([
28             'code' => 403,
29             'message' => "non autoriser"
30         ], 403);
31
32         if ($book->likeByUser($user)) {
33             $like = $bookLikeRepository->findOneBy([
34                 'book' => $book,
35                 'user' => $user
36             ]);
37             $manager->remove($like);
38
39             $manager->flush();
40
41             return $this->json([
42                 'code' => 200,
43                 'message' => 'le like est supprimé',
44                 'likes' => $bookLikeRepository->count(['book' => $book])
45             ], 200);
46
47             $like = new BookLike();
48             $like->setBook($book)
49                 ->setUser($user);
50             $manager->persist($like);
51             $manager->flush();
52             return $this->json([
53                 'code' => 200,
54                 'message' => 'le like est ajouté',
55                 'likes' => $bookLikeRepository->count(['book' => $book])
56             ], 200);
57         }
58     }
59 }
```

Cette méthode permet à un utilisateur de liké ou disliké un book, lorsqu'il est connecté .

Comme vous pouvez le voir ci dessus la méthode est directement gérée depuis le controller, elle a pour route “book/ {id}/like”.

Cette méthode comprend les classes book, entitymanager interface et booklikeRepository qui vont être stockés dans des variables.

elle va ensuite nous renvoyer une “response”.

Construction de la vue pour les like

```
35 </center>
36
37 <div class="row justify-content-md-center mt-4 cardHeight " >
38     {% for book in lastBooks %}
39         <div class="col-3 mb-4" >
40             <div class="card mb-4 cardHeight">
41                 <h4 class="card-header text-center bg-primary">{{book.title}}</h4>
42                 <div class="card-body">
43                     <div class="card-body text-center">
44                         <h6 class="text-center">Ecrit par
45                             {{book.author}}</h6>
46                         #<h6 class="card-subtitle text-muted ">from the</h6> #
47                         <p class="card-subtitle text-muted ">Created on
48                             {{book.createdAt | date('d/m/Y' ) }}</p>
49                     </div>
50                     
51                     <div class="card-body text-center " >
52                         <a href="{{path('read', {'id': book.id })}}" class="card-link btn btn-primary">LIRE</a>
53                         <a href="{{ path('liker', {'id':book.id})}}" class="btn btn-link js-like">
54                             {% if app.user and book.likeByUser(app.user) %}
55                                 <i class="fas fa-thumbs-up"></i>
56                             {% else %}
57                                 <i class="far fa-thumbs-up"></i>
58                             {% endif %}
59                             <span class="js-likes">{{ book.likes|length }}</span>
60                             <span class="js-label">j'aime</span>
61                         </a>
62                     </div>
63                 </div>
64             <% endfor %>
65         </div>
66     <% endfor %>
67 </div>
```

```
74
75  {% block javascripts %}
76      <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
77      <script>
78          |     function onClickBtnLike(event) {
79      event.preventDefault();
80
81      const url = this.href;
82      const spanCount = this.querySelector('span.js-likes');
83      const icone = this.querySelector('i');
84
85      axios.get(url).then(function (response) {
86
87          spanCount.textContent = response.data.likes;
88          if (icone.classList.contains('fas')) {
89              icone.classList.replace('fas', 'far');
90          } else {
91              icone.classList.replace('far', 'fas');
92          }
93
94      })
95  }
96
97
98  document.querySelectorAll('a.js-like').forEach(function (link) {
99      link.addEventListener('click', onClickBtnLike);
100  })
101  </script>
102  {% endblock %}
```

On utilise ici une librairie axios qui va nous servir à faire des requêtes ajax.

La partie javascript de la méthode likeBook va nous permettre que lorsqu'on appuie sur le pouce celle ci va être dynamisée cette à dire qu'elle va interagir, le pouce like grâce au fontawesome passe d'un pouce vide à un pouce plein. La variable spancount va permettre à la classe span.js-likes de rajouter ou enlever un like.

PRÉSENTATION DU JEU D'ESSAI

fonctionnalités	donnée en entré	donnée attendu	résultat
Inscription	nom,prénom, mail, password, repeat password <i>si chaque champs est respectés</i>	nom, prénom: alphabétique email: alphanumérique password: alphanumérique et caractères spéciaux	redirection vers la page d'accueil
	nom,prénom, mail, password, repeat password <i>si au moins un champs n'est pas respecté</i>		message d'erreur 'champs incorrect'

Login	email, password <i>si chaque champs est respectés</i>	email: alphanumérique password: alphanumérique et caractères spéciaux	redirection vers la page d'accueil
	email, password <i>si au moins un champs n'est pas respecté</i>		message: “veuillez renseigner ce champ”

LA SÉCURITÉ DU SITE:

```
# DATABASE_URL="sqlite:///kernel.project_dir/var/data.db"
DATABASE_URL="mysql://root:@127.0.0.1:3306/IBook?Version du serveur : 10.4.18-MariaDB"
# DATABASE_URL="mysql://root:password@127.0.0.1:3306/IBook?Version du serveur : 10.4.18-MariaDB"
```

Le fichier .env.local pour éviter de commiter nos données

```

public function getCredentials(Request $request)
{
    $credentials = [
        'email' => $request->request->get('email'),
        'password' => $request->request->get('password'),
        'csrf_token' => $request->request->get('_csrf_token'),
    ];
    $request->getSession()->set(
        Security::LAST_USERNAME,
        $credentials['email']
    );
}

return $credentials;
}

public function getUser($credentials, UserProviderInterface $userProvider)
{
    $token = new CsrfToken('authenticate', $credentials['csrf_token']);
    if (!$this->csrfTokenManager->isTokenValid($token)) {
        throw new InvalidCsrfTokenException();
    }

    $user = $this->entityManager->getRepository(User::class)->findOneBy(['email' => $credentials['email']]);
    if (!$user) {
        throw new UsernameNotFoundException('Email could not be found.');
    }

    return $user;
}

```

```

CLASS AppAuthenticator EXTENDS AbstractFormLoginAuthenticator IMPLEMENTS Pass
{
    use TargetPathTrait;

    public const LOGIN_ROUTE = 'app_login';

    private $entityManager;
    private $urlGenerator;
    private $csrfTokenManager;
    private $passwordEncoder;

    public function __construct(EntityManagerInterface $entityManager, UrlGen
    {
        $this->entityManager = $entityManager;
        $this->urlGenerator = $urlGenerator;
        $this->csrfTokenManager = $csrfTokenManager;
        $this->passwordEncoder = $passwordEncoder;
    }

    public function supports(Request $request)
    {
        return self::LOGIN_ROUTE === $request->attributes->get('_route')
            && $request->isMethod('POST');
    }
}

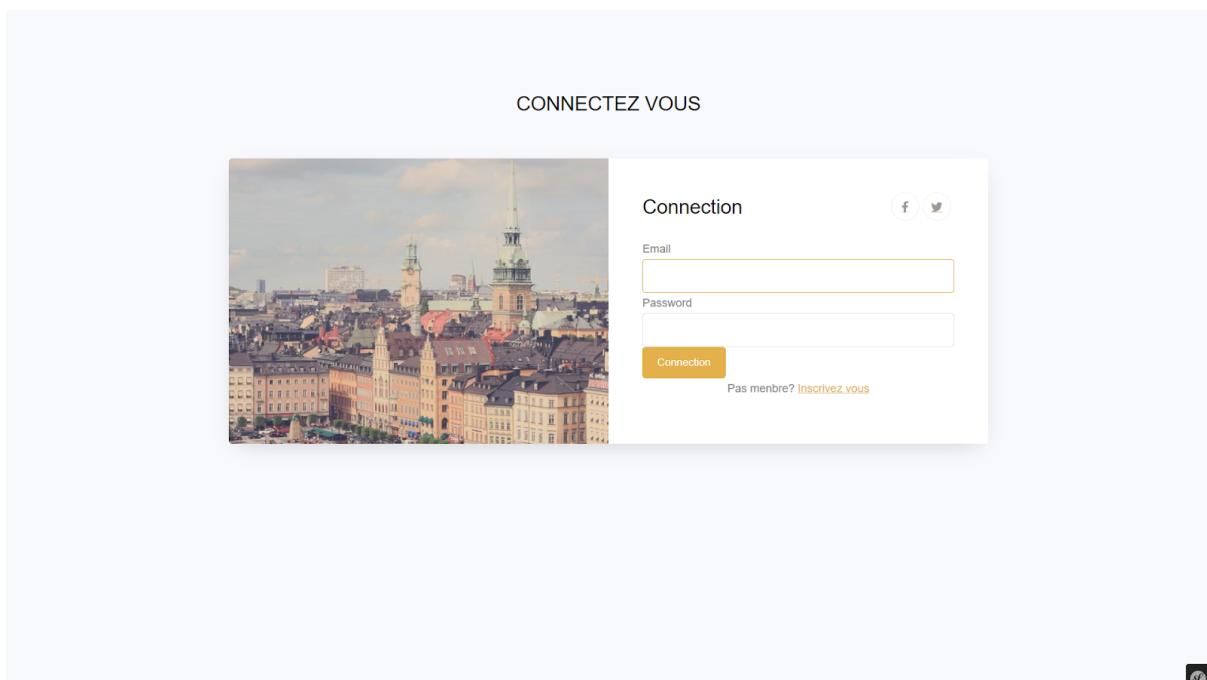
```

ENCODER pour le hachage des passwords

CSRFToken champ token caché qui va être stocké en session pour comparer le token reçu et le token généré

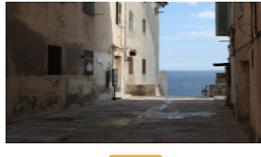
LES PAGES DU SITE:

The image shows the registration form page of the IBOOK website. The title "INSCRIPTION" is centered at the top. To the left of the form is a large, vertical photograph of a beach at sunset or sunrise. The sky is filled with warm, golden clouds, and the water reflects the light. In the foreground, there are wooden pilings and some birds flying over the water. The registration form itself is contained within a white box. It has a header "Inscrivez vous" with social media sharing icons for Facebook and Twitter. Below this are five input fields: "First name", "Last name", "Email", "Mot de passe", and "Repetez le mot de passe". There is also a checkbox labeled "Agree terms". At the bottom of the form is a large yellow "Envoyer" button. Below the button, a link says "déjà membre? Connectez vous". In the bottom right corner of the form area, there is a small circular logo with the letters "sf".



IBOOK LIVRES DECONNEXION ADMINISTRATION

Les derniers ajouts de livres

KIMSTREAM Ecrit par jolamy Created on 02/06/2021  LIRE 1 like	Voluptatibus ratione voluptatem ratione quaerat amet iste minima. Ecrit par William Gaillard Created on 24/05/2021  LIRE 6 likes	Unde et ea et aspernatur eaque voluptatibus. Ecrit par Laure Rousset Created on 20/05/2021  LIRE 3 likes	Amet tempora eligendi sit et voluptatem. Ecrit par Nath-Élisabeth Leveque Created on 06/04/2021  LIRE 5 likes
Rerum delectus est ducimus et voluptatum. Ecrit par Olivier Le Renaud Created on 25/03/2021  LIRE	Ea enim fugit illum molestias quas non. Ecrit par Margaret Levy-Teixeira Created on 04/03/2021  LIRE	Commodi ipsum ut ut autem occaecati laboriosam. Ecrit par Éléonore Auger-Bernard Created on 21/04/2021  LIRE	Blanditiis et harum ad officia. Ecrit par Andrée Brunet Created on 09/04/2021  LIRE

KIMSTREAM

Ecrit par jclamy
Crée le 02/06/2021

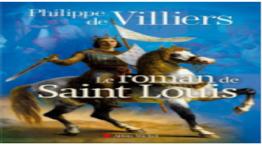


The text was full of spelling mistakes. Le texte était plein de fautes d'orthographe. On each page, the text is laid out in two columns. Sur chaque page, le texte est présenté en deux colonnes.

[Retour](#)

IBOOK Livres Nouveaux livres DECONNEXION

Tous les livres

Natus laudantium quo tenetur eveniet. Ecrit par Constance Regnier Created on 07/05/2021  LIRE 3 j'aime	Recusandae laborum et dolores ut. Ecrit par Gilbert du Bourdon Created on 22/03/2021  LIRE 4 j'aime	Doloribus et modi sequi voluptatem et. Ecrit par Eugène Leclerc Created on 03/04/2021  LIRE 7 j'aime	Sunt quis error illo dolores. Ecrit par Edith Cohen Created on 05/03/2021  LIRE 7 j'aime
Atque voluptatibus ad sit similique. Ecrit par Anne Gauthier Created on 30/05/2021  LIRE	Sapiente nam accusantium vel et quas qui beatiae. Ecrit par Adélaïde Costa Created on 28/05/2021  LIRE	Suscipit blanditiis sed explicabo qui quod. Ecrit par Valentine-Jeannine Pasquier Created on 29/05/2021  LIRE	Occaecati natus enim provident dolorem sed. Ecrit par Marie Michel-Gros Created on 09/03/2021  LIRE

Book						rakthak.kim@gmail.com
<input type="text"/> Search						Add Book
ID	Title	Author	Image	Created At		
251	Natus laudantium quo tenetur eveniet.	Constance Regnier	http://www.noblesseetroyautes.com/wp-content/uploads/2013/11/Gr...	May 7, 2021, 1:36:52 PM	Edit	Delete
252	Recusandae laborum et dolores ut.	Gilbert du Bourdon	https://picsum.photos/500	Mar 22, 2021, 12:35:42 PM	Edit	Delete
253	Doloribus et modi sequi voluptatem et.	Eugène Leclerc	https://picsum.photos/500	Apr 3, 2021, 8:44:23 PM	Edit	Delete
254	Sunt quis error illo dolores.	Édith Cohen	https://picsum.photos/500	Mar 5, 2021, 5:46:34 PM	Edit	Delete
255	Atque voluptatibus ad sit similique.	Anne Gauthier	https://picsum.photos/500	May 30, 2021, 10:00:41 AM	Edit	Delete
256	Sapiente nam accusantium vel et quas qui beatae.	Adélaïde Costa	https://picsum.photos/500	May 28, 2021, 12:41:49 AM	Edit	Delete
257	Suscipit blanditiis sed explicabo qui quod.	Valentine-Jeanne Pasquier	https://picsum.photos/500	May 29, 2021, 4:08:12 PM	Edit	Delete
258	Oiccaecati natus enim provident dolorem sed.	Marie Michel-Gros	https://picsum.photos/500	Mar 9, 2021, 2:07:04 AM	Edit	Delete
259	Blanditiis ut impedit ex rerum expedita corrupti.	Zoé Henry-Marechal	https://picsum.photos/500	May 30, 2021, 2:51:11 PM	Edit	Delete
260	Et vel et velit quo et.	Guy-Olivier Riou	https://picsum.photos/500	Apr 20, 2021, 10:25:19 AM	Edit	Delete
261	Aliquid omnis aut minus sint quisquam qui esse rerum.	Adrienne Loiseau	https://picsum.photos/500	Mar 26, 2021, 5:52:56 AM	Edit	Delete
262	Quia est et dolor corrupti quod iure quis.	Antoinette Legros-Blanchet	https://picsum.photos/500	Jun 1, 2021, 1:55:03 AM	Edit	Delete
263	Ut consequatur et dolorum laboriosam.	Jeanne Lefort	https://picsum.photos/500	Mar 13, 2021, 8:44:47 PM	Edit	Delete
				May 18, 2021, 1:41:11	Edit	

CONCLUSION:

Ce stage et ce projet ont été très enrichissant pour moi, car ils m'ont permis d'améliorer mes compétences dans le développement web. Fort de cette expérience et en réponse à ses enjeux, j'aimerai beaucoup par la suite essayer d'enrichir mes connaissances et de trouver au plus vite un emploi dans ce domaine.