

Titre : Développeur web et web mobile

Dossier de projet



Bourega Fériale

Table des matières

Compétences du référentiel couvertes par le projet

Résumé

Spécifications fonctionnelles

Description de l'existant

Périmètre du projet

Cible adressée par le site internet

Arborescence du site

Description des fonctionnalités

1. Authentification
2. Catalogue produit et filtre
3. Fiche produit
4. Evaluation des produits et commentaires clients

Spécifications techniques

Choix techniques et environnement de travail

Architecture du projet

Réalisations

1. Conception de la base de données
2. Organisation du code
3. Extraits de code significatifs

Veille sur les vulnérabilités de sécurité

Recherches effectuées à partir d' un site anglophone

Annexes

Modèle Conceptuel de Données

Modèle Logique de Données

Modèle Physique de Données

Maquette

Compétences du référentiel couvertes par le projet

Le projet couvre les compétences énoncées ci-dessous.

Pour l'activité 1, **“Développer la partie front-end d'une application web et web mobile en intégrant les recommandations de sécurité”** :

- _ Maquetter une application
- _ Réaliser une interface utilisateur web ou mobile statique et adaptable
- _ Développer une interface utilisateur web dynamique
- _ Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce

Pour l'activité 2, **“Développer la partie back-end d'une application web et web mobile en intégrant les recommandations de sécurité”** :

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

Résumé

Wood & else est le nom du blog que j'ai créé avec ma camarade de classe Chadhilati Mansoibou dans le cadre d'un projet de classe à l'école de formation de la Plateforme à Marseille.

Ce site est une application de gestion de contenu, on peut donc considérer que c'est un projet professionnel.

L'objectif de ce travail était de créer un blog afin de publier des articles personnels et d'échanger avec nos visiteurs.

Il présente tout d'abord un catalogue de produits triés du plus récents au plus anciens. Pour pouvoir accéder au site, il faut tout d'abord s'authentifier à l'aide d'une page inscription et d'une page de connexion. Une fois, la connexion effectuée, on peut accéder à une page article pour chaque produit avec la possibilité d'y laisser des commentaires pour donner son avis sur l'article.

Les articles sont également filtrés par catégories à l'aide d'une liste déroulante, qui affiche une liste de produits pour chaque catégorie.

Ce site présente des articles d'ameublement.

Spécifications fonctionnelles

Description de l'existant

Créer un blog peut être utilisé dans différents contextes. Le choix de créer un blog qui présente des articles d'ameublement permet de cibler un large public et de réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce. Avec ma camarade de classe, nous avons convenu lors de nos différents entretiens quels étaient les besoins du projet, les fonctionnalités qui seraient développées mais également l'aspect graphique de la future application.

Périmètre du projet

Le site sera réalisé en français et ce dernier sera accessible sur différents supports, à savoir mobile, tablette et ordinateur.

Cible adressée par le site internet

Le site du blog Wood & else s'adresse à des particuliers ou des entreprises, et plus exactement des personnes qui voudraient meubler leur domicile, des chefs d'entreprise qui voudraient munir leurs locaux de meubles et de matériels, ou des personnes qui voudraient faire un cadeau à un proche.

Arborescence du site

L'arborescence du site se décline de la façon suivante :

- _ Une page d'accueil
- _ Une page contenant un formulaire d'inscription
- _ Une page contenant un formulaire de connexion

- _ Une page permettant de modifier son profil
- _ Une page contenant les articles
- _ Une page contenant les articles pour chaque catégorie
- _ Une page contenant un article et ses commentaires

Description des fonctionnalités

1. Authentification

L'authentification se fait grâce à la connexion à la base de données.

Le formulaire d'inscription doit contenir l'ensemble des champs présents dans la table "utilisateurs" ainsi qu'une confirmation de mot de passe. Dès qu'un utilisateur remplit ce formulaire, les données sont insérées dans la base de données et l'utilisateur est redirigé vers la page de connexion.

Le formulaire de connexion doit avoir deux inputs "login" et "password" .

Lorsque le formulaire est validé, s'il existe un utilisateur en base de données correspondant à ces informations alors l'utilisateur devient connecté et une (ou plusieurs) variables de session sont créées.

La page profil possède un formulaire permettant à l'utilisateur de modifier l'ensemble de ces informations.

2. Catalogue produit et filtre

Une page doit permettre d'afficher l'ensemble des produits disponibles avec un système de pagination contenant 5 produits par pages et triés du plus récemment publiés au plus anciens.

Cet affichage devra comprendre le nom du produit, la photo du produit, la date et l'heure à laquelle l'article a été publié.

Un filtre doit être implémenté afin de trier les articles en fonction de leur catégorie.

3. Fiche produit

Une fois l'utilisateur connecté, l'utilisateur est en mesure d'accéder à une fiche produit. Cette dernière devra contenir :

Bourega Fériale

- _ Une photo du produit
- _ Le nom du produit
- _ La date et l'heure à laquelle l'article a été publié
- _ L'ensemble des commentaires liés à l'article avec le nom de l'utilisateur, la date et l'heure à laquelle le commentaire a été posté
- _ La possibilité d'écrire un nouveau commentaire

4. Evaluation des produits et commentaires clients

Cette page permet de voir un article, l'ensemble des commentaires associés et la possibilité d'en ajouter un nouveau.

L'utilisateur pourra laisser un commentaire sur le produit afin de faire part de ses appréciations sur ce dernier.

Spécifications techniques

Choix techniques et environnement de travail

Technologies utilisées pour la partie back-end :

- _ Le projet sera réalisé avec le langage PHP (Hypertext Preprocessor).
PHP est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web. Il peut être intégré facilement au HTML. PHP est un langage de script HTML, exécuté côté serveur. Sa syntaxe est emprunté aux langages C, Java et Perl, et est facile à apprendre. Le but de ce langage est de permettre aux développeurs web d'écrire des pages dynamiques rapidement, mais vous pouvez faire beaucoup plus avec PHP.
- _ PHPmyadmin et base de données SQL.
PHPmyadmin est un logiciel libre écrit en PHP qui a pour mission de s'occuper de l'administration d'un serveur de base de données MySQL ou MariaDB.

Technologies utilisées pour la partie front-end :

- _ Le projet sera réalisé avec du HTML et CSS, javascript.
Le HTML est un langage de balisage, c'est-à-dire un langage qui va nous permettre de définir les différents contenus d'une page.

Le CSS va servir à mettre en forme les différents contenus définis par le HTML en leur appliquant des styles.

L'environnement de développement est le suivant :

- _ Editeur de code : Visual studio code

- _ Outil de versionning : Git, Github

- _ Maquettage : Figma

Du point de vue de l'organisation, j'ai utilisé Trello afin de découper le projet en une multitude de tâches à réaliser et de définir leur ordre de priorité.

Architecture du projet

L'architecture du projet est organisée autour de classes, de pages PHP dans lesquelles est intégré du code HTML sous forme de formulaires et de la base de données SQL, ainsi que d'une page CSS pour le style.

Organisation du code

Les classes permettent de mieux organiser le code de façon plus simple.

On appelle classe la structure d'un objet, c'est à dire la déclaration de l'ensemble des entités qui composeront un objet.

Un objet est donc "issu" d'une classe, c'est le produit qui sort d'un moule.

En réalité, on dit qu'un objet est une instanciation d'une classe, c'est la raison pour laquelle on pourra parler indifféremment d'objet ou d'instance.

Une classe est composée de deux parties :

- _ Les attributs (parfois appelés données membres) : il s'agit des données représentant l'état de l'objet.

- _ Les méthodes (parfois appelées fonctions membres) : il s'agit des opérations applicables aux objets.

Conception de la base de données

J'ai créé une base de données nommée "blog" à l'aide de phpmyadmin.

Ma base de données s'articule autour de 5 tables principales:

Bourega Fériale

- _ la table “utilisateurs” qui possèdent 5 champs (id , login, password, email, id_droits)
- _ la table “droits” qui possèdent deux champs (id, nom)
- _ la table “articles” qui possèdent 5 champs (id, article, id_utilisateur, id_categorie, date)
- _ la table “categories” qui possèdent deux champs (id, nom)
- _ la table “commentaires” qui possèdent 5 champs (id, commentaire, id_article, id_utilisateur, date)

Tout d’abord, la table “utilisateurs” va permettre d’identifier les clients.

Elle est liée à la table “droits” par l’association “appartient à”.

La table “categories” est liée à la table “articles” afin de déterminer à quelle catégorie appartient un article.

La table “articles”est liée à la table “utilisateurs”, afin de pouvoir associer un commentaire selon son id article.

La table “categories” est liée à la table “articles”, afin de pouvoir associer un commentaire selon son id article.

La table “commentaires” est liée à la table “utilisateurs” , afin de sélectionner des utilisateurs pour leurs propres commentaires.

Extraits de code significatifs

Eléments de la connexion à la base de données du db-config.php



```
$DB_SDN ='mysql:host=localhost;dbname=blog';  
$DB_USER = 'root';  
$DB_PASS = '';
```

Le db-config.php détermine les éléments qui permettent la connexion à la base de données:

- _ Le nom du serveur qui est dans ce cas le localhost
- _ Le nom de la base de données qui est dans ce cas “blog”
- _ Le nom de l'utilisateur qui est “root”
- _ le mot de passe qui est un espace vide

Le db-config.php est intégré dans la méthode construct avec require et PDO.

La méthode construct

```
public function __construct()
{
    $this->error = "";
    try {
        $options =
        [
            PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES utf8',
            PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
        ];
        $DB_SDN = 'mysql:host=localhost;dbname=blog';
        $DB_USER = 'root';
        $DB_PASS = '';

        //on va instancier donc créer un objet PDO
        $this->bdd = new PDO($DB_SDN, $DB_USER, $DB_PASS, $options);
    } catch (PDOException $exception) {
        echo 'ERREUR :' . $exception->getMessage();
    }
}
```

Le constructeur d'une classe est une méthode publique (dans la plupart des cas). Elle est appelée automatiquement au moment de l'instanciation. Elle sert généralement à initialiser les attributs et fournir à l'objet créé tout ce dont il a besoin pour fonctionner. Cette opération d'initialisation est connue sous le nom d'hydratation.

La méthode register

La méthode register permet d'enregistrer l'utilisateur en base de données.

Voici un extrait de code de cette méthode :

Bourega Fériale

```
if (!empty($login) && !empty($password) && !empty($passwordConfirm) && !empty($email))
{
    $infos = "SELECT * FROM utilisateurs WHERE login = :login AND email = :email";
    $result = $this->bdd->prepare($infos);
    $result->bindvalue(':login', $login);
    $result->bindvalue(':email', $email);
    $result->setFetchMode(PDO::FETCH_ASSOC); // j'utilise fetch_assoc pour récupérer les key
d'un tableau associatif
    $result->execute();
    $userData = $result->fetchAll();
    // var_dump($userData);
}
```

Cet extrait de code nous montre comment faire une requête et récupérer les données.

Tout d'abord, il faut vérifier que les champs ne soient pas vides.

Ensuite on écrit la requête, on la prépare, on lie les champs du formulaire aux champs de la base de données à l'aide de "bindvalue", on récupère les données dans un tableau associatif à l'aide de "fetch_assoc", on exécute la requête et on récupère le résultat à l'aide de "fetchAll".

Cette extrait de code nous montre comment créer la page selon la catégorie :

```
public function articles_by_id_categ($id_categorie)
{
    $this->id_categorie = $id_categorie;

    $id_categorie = htmlspecialchars($id_categorie);
    $sql = "SELECT categories.nom, articles.article, articles.id_utilisateur, articles.images,
articles.id_categorie, articles.date
FROM `categories`
INNER JOIN articles ON articles.id_categorie = categories.id
WHERE categories.id = ? ";
    $request = $this->bdd->prepare($sql);
    $request->execute([$id_categorie]);
    // var_dump($request);

    $categ_id = $request->fetchAll(PDO::FETCH_ASSOC);
    // $categ_id = $request->fetchAll();
    // var_dump($categ_id);

    return $categ_id;
}
```

On commence par définir le paramètre “\$id_categorie” utilisé pour cette méthode et à y supprimer tous les caractères spéciaux à l’aide de “htmlspecialchars”.

On écrit ensuite la requête utilisée pour créer la page selon la catégorie.

Cette requête est un peu particulière car elle lie deux tables à l’aide de “inner join”.

Pour cette requête, on sélectionne d’abord les champs des tables utilisées en précisant bien à quelle table ils appartiennent, puis on termine en posant une égalité entre l’ “id_categorie” de la table “articles” et l’ “id” de la table “categories” où l’ “id” de la table “categories” est une valeur arbitraire.

Ensuite, on prépare la requête, on l’exécute à partir de la valeur entrée en paramètre, on récupère les données dans un tableau associatif, puis on récupère le résultat dans un “fetchAll”.

Pour que notre méthode fonctionne, il faut retourner le résultat de la requête.

L'instanciation de la classe Articles :



```
$articles = new Article();
```

L'instanciation est l'opération qui consiste à créer un objet.

Après avoir instancier une classe, on appelle les méthodes que contient la classe de la façon suivante :



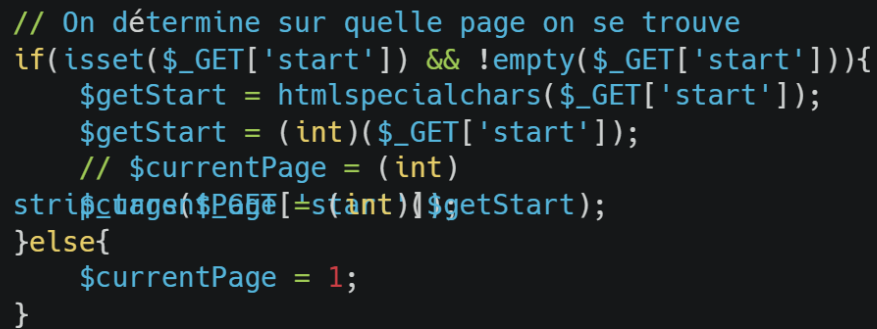
```
if (isset($_POST['submit'])) {  
    $datas = $user->register($_POST['login'], $_POST['password'],  
    $_POST['passwordConfirm'], $_POST['email']);  
}
```

Dans ce cas, après avoir validé le formulaire avec la fonction “isset”, on appelle la méthode register en entrant en paramètres les champs du formulaire.

La méthode register permet l'inscription, elle permet d'enregistrer l'utilisateur en base de données.

Bourega Fériale

Voici, deux extraits de code qui permettent la pagination :



```
// On détermine sur quelle page on se trouve
if(isset($_GET['start']) && !empty($_GET['start'])){
    $getStart = htmlspecialchars($_GET['start']);
    $getStart = (int)($_GET['start']);
    // $currentPage = (int)
    strip_tags($_GET['start']);
} else {
    $currentPage = 1;
}
```

Ici, on détermine sur quelle page on se trouve en déterminant un nombre entier pour cette page.



```
$nbArticles = $articles->total_number_articles();  
// var_dump($nbArticles); //OK  
  
// On détermine le nombre d'articles par page  
$parPage = 5;  
// On calcule le nombre de pages total  
$pages = ceil($nbArticles / $parPage);  
  
// Calcul du 1er article de la page  
$premier = ($currentPage * $parPage) - $parPage;  
  
$get_page = $articles->get_by_page($premier,$parPage);  
// var_dump($get_page);
```

Dans cette deuxième partie pour la pagination, on appelle la méthode qui permet de déterminer le nombre d'articles, on détermine le nombre d'articles par page en le posant égal à 5.

Puis, on calcule le nombre de pages total, en divisant ces deux valeurs déterminées précédemment. Enfin, on calcule le premier article de la page et on appelle la méthode qui permet d'obtenir 5 articles par page.

Voici, l'extrait de code qui permet d'afficher le nom du produit, la photo du produit et la date et l'heure à laquelle il a été publié :

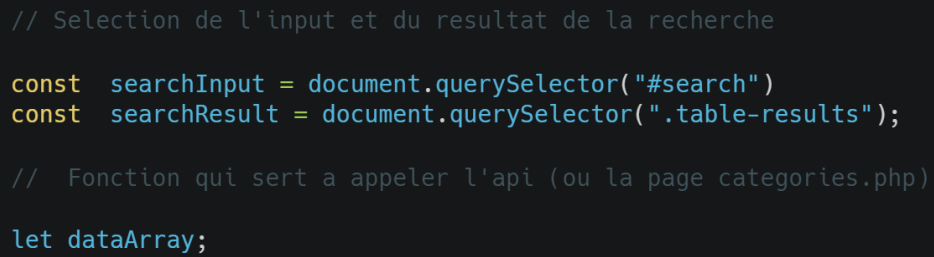
```
<?php
    foreach($get_page as $article){
        // var_dump($article);
    ?>
    <tr>
        <td><a href="article.php?id=<?= $article['id'] ?>"><?= $article['article']
?></a></td>
        <td><a href="article.php?id=<?= $article['id'] ?>"><?=
date_format(date_create($article['date']), 'd/m/Y H:i:s') ?></a></td>
        <td><a href="article.php?id=<?= $article['id'] ?>"></a></td>
    </tr>
<?php
```

Pour la barre de recherche, j'ai utilisé le javascript.

Il y a 5 étapes pour créer une barre de recherche, on doit :

- _ Appeler les données
- _ Formater les données
- _ Les trier par ordre alphabétique
- _ Les faire apparaître
- _ Les filtrer quand on écrit dans l'input

Sélection de l'input et résultat de la recherche :



```
// Selection de l'input et du resultat de la recherche

const searchInput = document.querySelector("#search")
const searchResult = document.querySelector(".table-results");

// Fonction qui sert a appeler l'api (ou la page categories.php)
let dataArray;
```

Ici, on définit deux constantes qui sélectionnent les balises de la page HTML qui contiennent la recherche et le résultat de la recherche, avec la méthode “querySelector” qui est un sélecteur.



```
async function getArt(){
  const res = await fetch("")

  // Analyse le corps de la requete et prend ce qu'on veut
  const { results } = await res.json()

  // Tableau qui trie les donnees par ordre alphabetique
  dataArray = orderList(results)
  // Cree une liste
  createArtList(dataArray)

  // Execute la fonction
  getArt()
}
```

Cette fonction est une fonction asynchrone. Le mot “async” permet de rendre l’exécution d’une fonction asynchrone. Il retourne toujours une promesse contenue dans le “fetch”. Il permet de grandement simplifier l’écriture des promesses. L’opérateur “await” permet d’attendre la résolution d’une promesse avant le reste de l’exécution du code. Il s’utilise uniquement dans les fonctions asynchrones.

```

// Fonction qui fait apparaître la liste
function createArtList(ArtList){
  ArtList.forEach(Articles=>{
    const listItem = document.createElement("div");
    listItem.setAttribute("class","table-item");
    listItem.innerHTML = '<div class="container-img"/>
                        <img src = "${Article.picture.medium}">
                        <p class="name">${Article.name}</p>
                        </div>
                        <p class="description">${Article.description}</p>
                        <p class="prix">${Article.prix}</p>'

    searchResult.appendChild(listItem);
  })
}

```

Cette fonction fait apparaître la liste de la recherche.

On commence par créer une “div” contenue dans une constante à l’aide de la méthode “document.createElement”. A cette constante ,on ajoute l’attribut “class” qu’on nomme “table-item”. Enfin, on insert dans cette constante les éléments de la recherche, puis on ajoute ces éléments à la constante du résultat de la recherche définit au départ (“searchResult”). Ceci fait apparaître la liste de la recherche.

En CSS, les **media queries** permettent de rendre le site responsive, et de répondre à la compétence du référentiel : “Réaliser une interface utilisateur web ou mobile statique et adaptable”.



```
.responsive {  
  /* width: 100%;  
    max-width: 200px;  
    height: auto; */  
  width: 71%;  
  max-width: 50%;  
  height: 50%;  
}
```

Veille sur les vulnérabilités de sécurité

Exposition des données sensibles

Lorsque vous surfez sur Internet, votre navigateur utilise le protocole HTTP (Hypertext Transfer Protocol) pour afficher les pages web, et le protocole Transmission Control Protocol/Internet Protocol (TCP/IP) pour les transmettre. Si le serveur web établit la connexion TCP avec le navigateur, une réponse avec le code status et le fichier demandé (généralement le fichier index.html pour la page web) sera transmise. Mais dans notre cas, les données transitent en HTTP et pas en HTTPS...

Les données transitant en HTTP peuvent être interceptées, car elles transitent en clair.

Comment éviter d'exposer les données sensibles en transit ?

- Utilisez le HTTPS pour l'ensemble de votre site, même s'il ne contient pas de données sensibles.
- Utilisez les requêtes GET pour récupérer les informations et POST pour modifier les informations.
- Sécurisez vos cookies pour qu'ils soient transmis par l'en-tête et via HTTPS.
- Sécurisez vos sessions en ajoutant une date d'expiration, en sécurisant l'ID et en ne mettant pas cet ID dans l'URL.

Les données sensibles ne sont pas seulement en transit, elles sont aussi stockées en base de données. Pour protéger certaines données stockées sur une application, il est possible d'utiliser des algorithmes de hachage.

L'intérêt des algorithmes de hachage est qu'ils permettent de calculer une empreinte (ou hash) d'une chaîne de caractères, par exemple. Cette empreinte est utile pour éviter de stocker en clair le mot de passe dans la base de données.

Comment éviter d'exposer les données stockées ?

- Sécurisez votre base de données avec le chiffrement.
- Utilisez des algorithmes de hachage sécurisés tels que Argon5, Scrypt, Bcrypt et PBKDF2.
- Le masquage des données peut être utilisé pour sécuriser les données sensibles d'une base de données.

Injection SQL

Cette vulnérabilité permet à un attaquant d'injecter des données non maîtrisées qui seront exécutées par l'application et qui permettent d'effectuer des actions qui ne sont normalement pas autorisées.

Ce type d'attaque s'effectue généralement grâce aux champs présents dans les formulaires.

Dans le cas d'une attaque par injection SQL, au lieu de mettre un nom d'utilisateur et un mot de passe sur une page de connexion, un utilisateur malveillant entrera des données directement interprétées par le moteur SQL, ce qui lui permettra de modifier le comportement de votre application.

Comment s'en prémunir ?

- Validez les entrées

Cela consiste à limiter ce que l'utilisateur peut mettre dans la zone de texte. Cela n'empêchera pas l'injection, mais c'est une mesure que vous pouvez mettre en place pour limiter des attaques de base. En effet, les caractères spéciaux spécifiques à certains langages ne pourront pas être utilisés.

- Préparez les requêtes SQL

Ce sont des requêtes dans lesquelles les paramètres sont interprétés indépendamment de la requête elle-même. De cette manière, il est impossible d'effectuer des injections.

Recherches effectuées à partir d'un site anglophone

Pour déboguer mon site sur le localhost, j'ai par exemple utilisé un site anglophone qui s'appelle stackOverflow.



Stack Overflow est un site web de questions-réponses qui réunit des développeurs du monde entier.

Le site sert de ressources sur le développement informatique. Il doit son nom à l'erreur logicielle qui se produit lorsqu'un programme manque de mémoire.

Stack Overflow facilite la collaboration, la résolution de problèmes et le partage de connaissances entre individus, groupes ou entreprises.

Les avantages de Stack Overflow

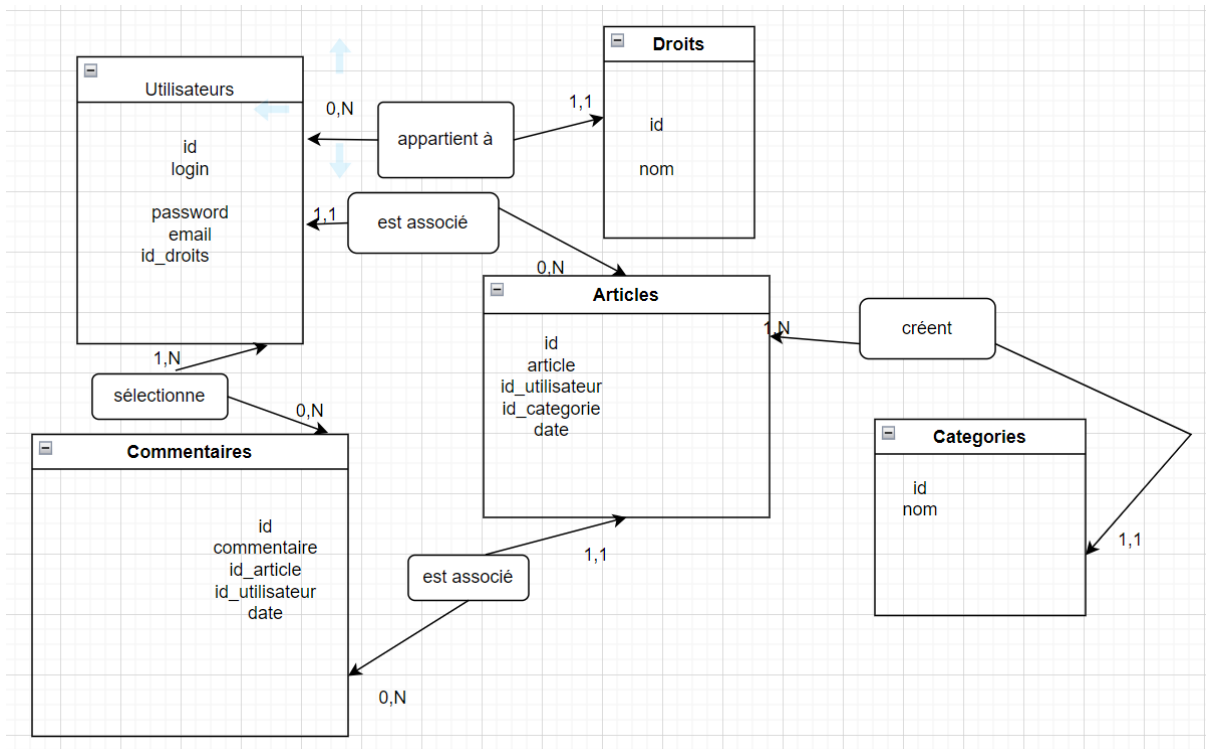
- Vous pouvez poser des questions et obtenir des réponses précises et détaillées.
- Les tags et les filtres permettent de trouver facilement les questions et les réponses appropriées.
- Il réunit des communautés de développeurs qui partagent les mêmes idées.
- Il n'y a pas de publicités ni de spams.
- Un système de vote permet de promouvoir les meilleures réponses.
- Des spécialistes vous donnent des réponses rapides.

Les inconvénients de Stack Overflow

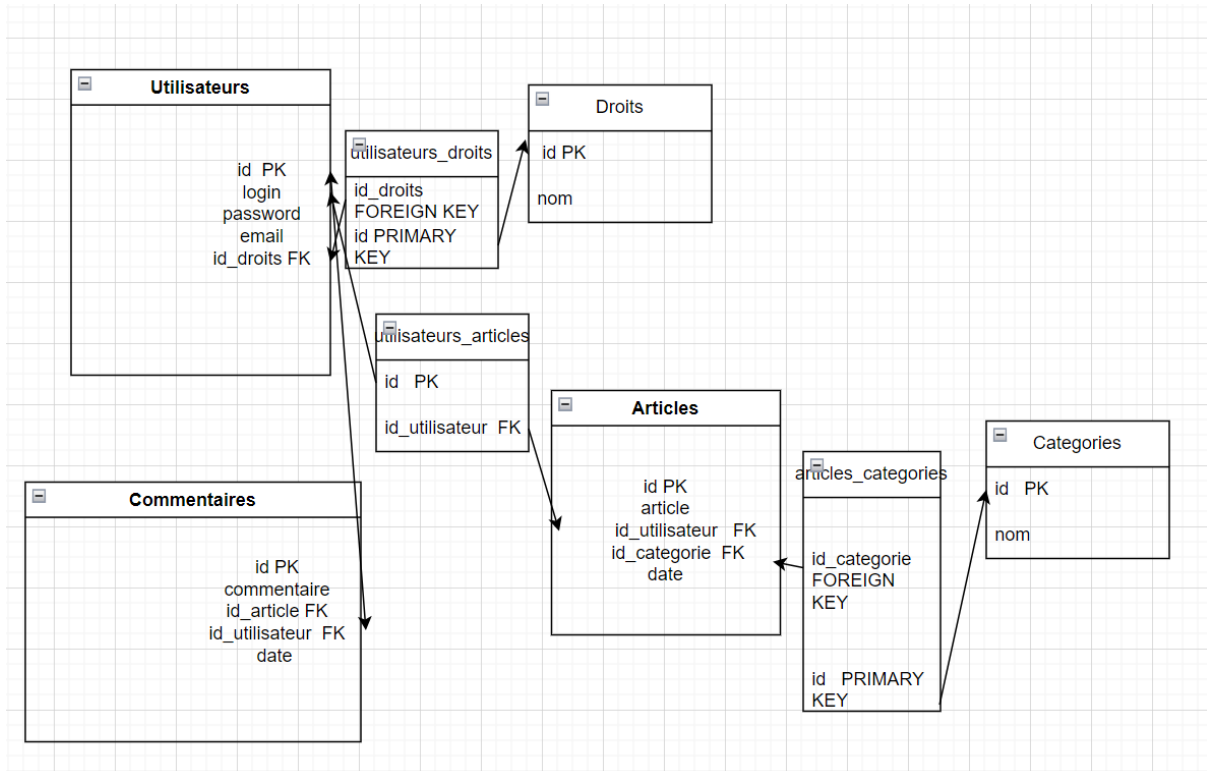
- Il est plutôt hostile aux questions des débutants, et les guides sur la bonne façon de poser une question sont difficiles d'accès pour les débutants.
- Vous pouvez obtenir un vote négatif par les autres utilisateurs sans explication ni justification.
- Il n'est pas possible d'affiner et de modifier les questions et les réponses.
- Les modérateurs ne facilitent pas toujours les discussions.

ANNEXES

Modèle Conceptuel de Données



Modèle Logique de Données



Modèle Physique de Données

