DOSSIER DE PROJET POUR LE TITRE PROFESSIONNEL DE DÉVELOPPEUR WEB & WEB MOBILE

Développement d'une boutique en ligne from scratch

SOMMAIRE

Compétences du référentiel couvertes par le projet

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

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

Pour l'activité type 2, "**Développer la partie back-end d'une application web ou 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 œuvre des composants dans une application de gestion de contenu ou e-commerce

Résumé

Ce dossier présente le travail effectué en tant qu'étudiante à La Plateforme, à savoir le développement d'une boutique de jeux vidéo en ligne nommée Game Vault.

GameVault est une boutique en ligne spécialisée dans la vente de jeux vidéo au format physique. Notre objectif est de fournir aux joueurs une solution pratique pour obtenir des copies physiques de leurs jeux préférés, tout en offrant une expérience conviviale et sécurisée.

Spécifications fonctionnelles

Arborescence du site :

- **Page d'accueil** : la première page que voit l'utilisateur quand il ouvre notre site. Elle est composée d'un call-to-action (carrousel des jeux en précommande) et de plusieurs parties présentant des jeux par groupes : les derniers jeux sortis, quelques jeux aléatoires et les best sellers. Cette page sert à donner des idées d'achats à l'utilisateur.
- **Page admin** : accessible seulement aux utilisateurs ayant le rôle d'administrateur. Permet d'ajouter ou de supprimer une plateforme, une catégorie ou une sous-catégorie, de changer de rôle, de voir les infos ou de supprimer un utilisateur et de voir, d'ajouter ou de supprimer un jeu (et de modifier?).
- **Page all products** : une page présentant tous les produits, avec une possibilité de les filtrer par plateforme, catégorie et sous-catégorie et incluant une pagination.
- **Page d'authentification** : page d'authentification simple par login et mot de passe avec la possibilité de s'inscrire.
- **Page panier** : page présentant les articles présents dans le panier de l'utilisateur, avec la possibilité de changer la quantité et de supprimer un article du panier.
    - **Page de formulaire de commande** : page permettant de choisir l'adresse à laquelle l'utilisateur veut se faire livrer, de renseigner les informations de paiement et de voir les articles achetés.

    - **Page de d'ajout d'adresse** : formulaire renseignant un prénom, un nom, un numéro de téléphone, un pays, une adresse, une ville et un code postal à laquelle livrer les produits commandés.
    - **Page de résumé d'achat** : page récapitulant la commande qui vient d'être passée. Plusieurs informations y figurent : le numéro de la commande, les jeux achetés, la quantité de chaque jeu et le prix payé pour chaque jeu. Possibilité de cliquer sur un lien si on est pas redirigé automatiquement.
- **Page forbidden** : page informant l'utilisateur qu'il essaye d'accéder à une page qui lui est interdite.
- **Page de changement de profil** : Page permettant à l'utilisateur de changer son login, son mot de passe, son adresse mail, son prénom, son nom, sa date de naissance et son numéro de téléphone.

Fonctionnalités du site

Inscription et gestion des comptes utilisateurs :
- Création de compte pour les utilisateurs.
- Connexion sécurisée.
- Gestion des informations personnelles et des adresses de livraison.

Catalogue de jeux :
- Possibilité de filtrer les jeux par console, catégorie et sous-catégorie.
- Page de détail pour chaque jeu, comprenant des descriptions, des images et des caractéristiques techniques.

Panier et processus de commande :
- Ajout de jeux au panier d'achat.
- Gestion du panier avec la possibilité de modifier les quantités, de supprimer des articles, etc.
- Possibilité de choisir entre plusieurs adresses de livraison.

Gestion des commandes et suivi :
- Historique des commandes des utilisateurs (?)

Système de recherche avancée :
- Barre de recherche avec auto-complétion permettant aux utilisateurs de chercher des jeux par titre.

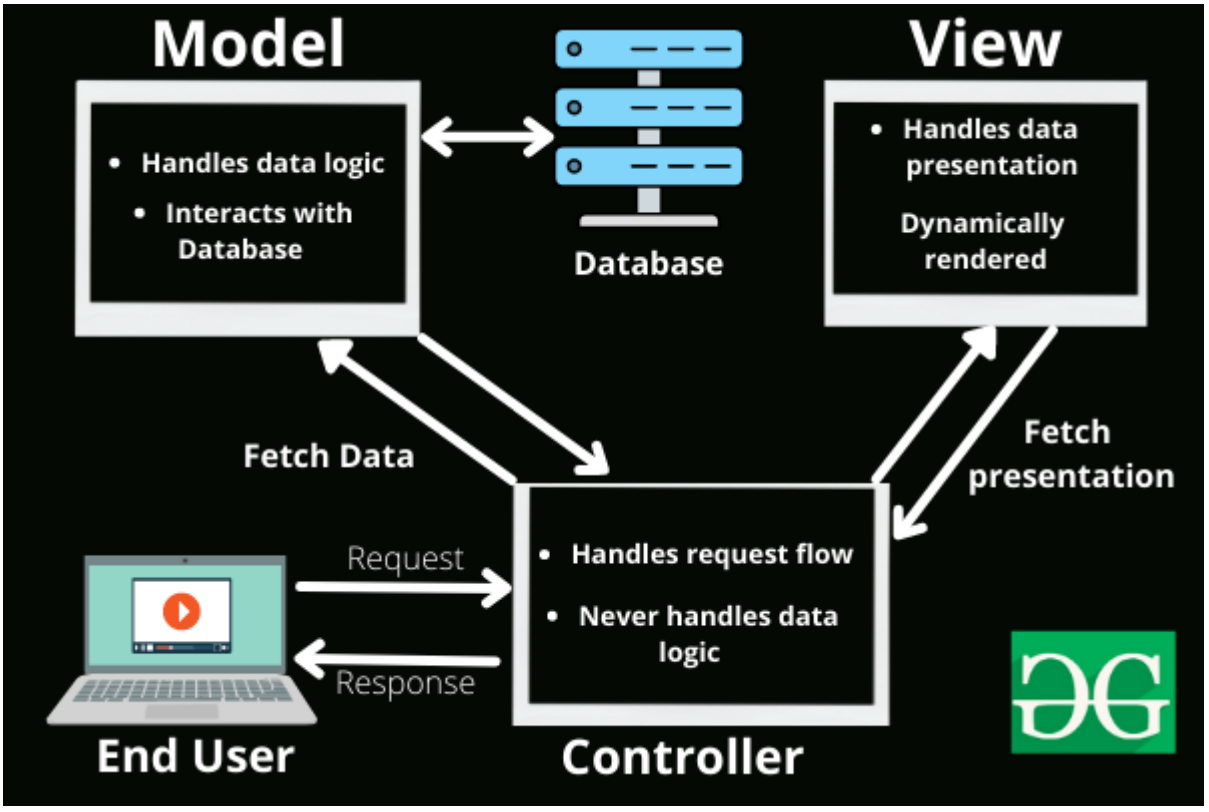
Langue du site :
- Anglais

Public cible :
- toutes les personnes qui aiment jouer
-

Spécification techniques

Le design pattern MVC (Modèle-Vue-Contrôleur) est un modèle architectural couramment utilisé dans le développement de logiciels et d'applications web. Il divise une application en trois composants principaux :

- Le Modèle (Model) : contient les données ainsi que de la logique en rapport avec les données : validation, lecture et enregistrement
- La Vue (View) : c'est la partie visible de l'application qui représente l'interface utilisateur. La vue affiche les données du modèle et interagit avec l'utilisateur. Elle peut être une page web, une fenêtre graphique ou tout autre moyen d'affichage des informations.
- Le Contrôleur (Controller) : C'est l'intermédiaire entre le modèle et la vue. Il reçoit les actions de l'utilisateur à travers la vue, traite les requêtes et met à jour le modèle en conséquence. Le contrôleur peut également mettre à jour la vue en fonction des changements dans le modèle.



Langages de programmation :

- backend : PHP, SQL
- frontend : HTML, CSS, JavaScript

Environnement de travail :

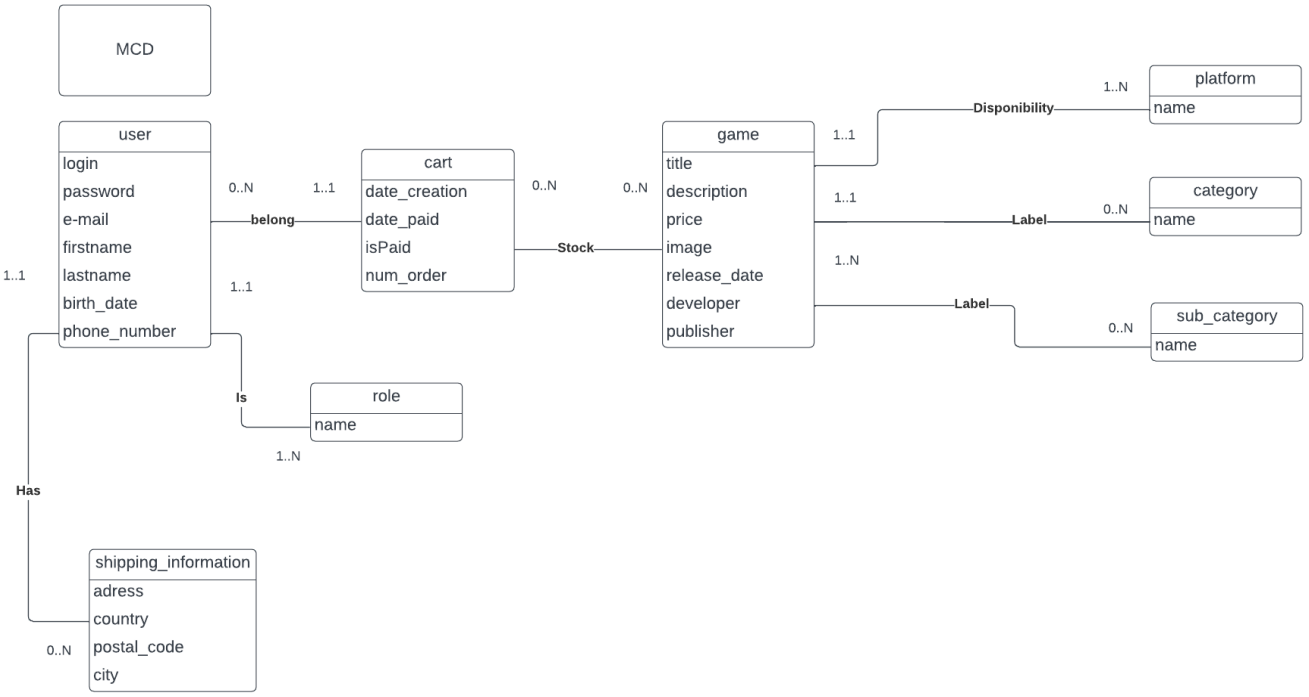
- éditeur de code : VSCode
- outil de versionning : GIT, GitHub
- outil de maquettage : figma

Base de données : Nous avons utilisé le système de gestion de base de données MySQL.

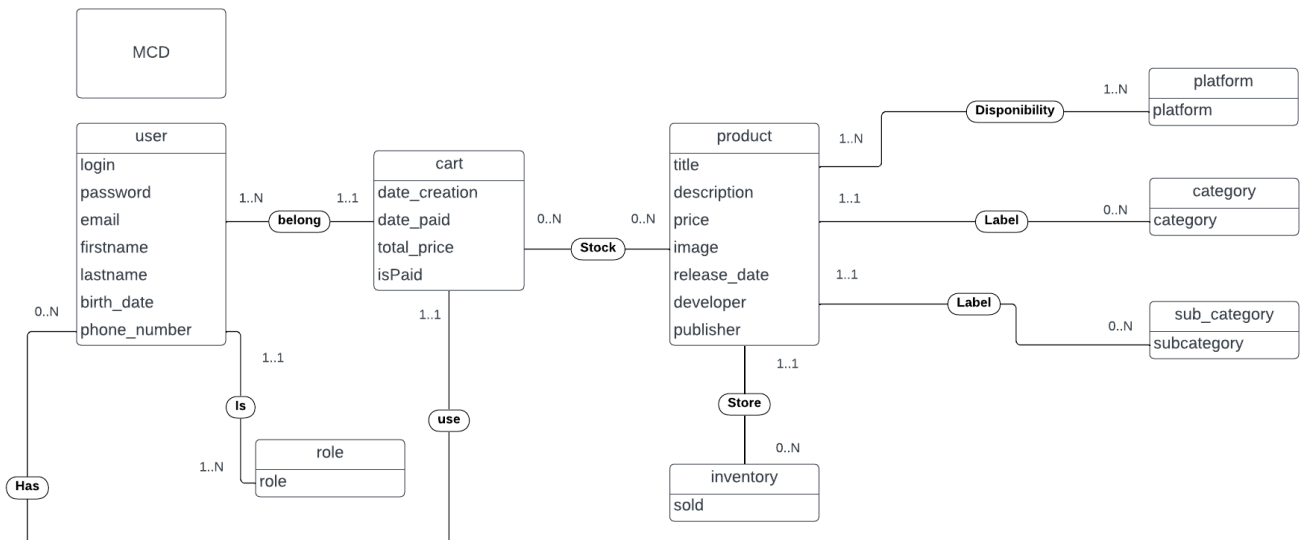
Schéma de la base de donnée (méthode merise) :

MCD

semaine 1

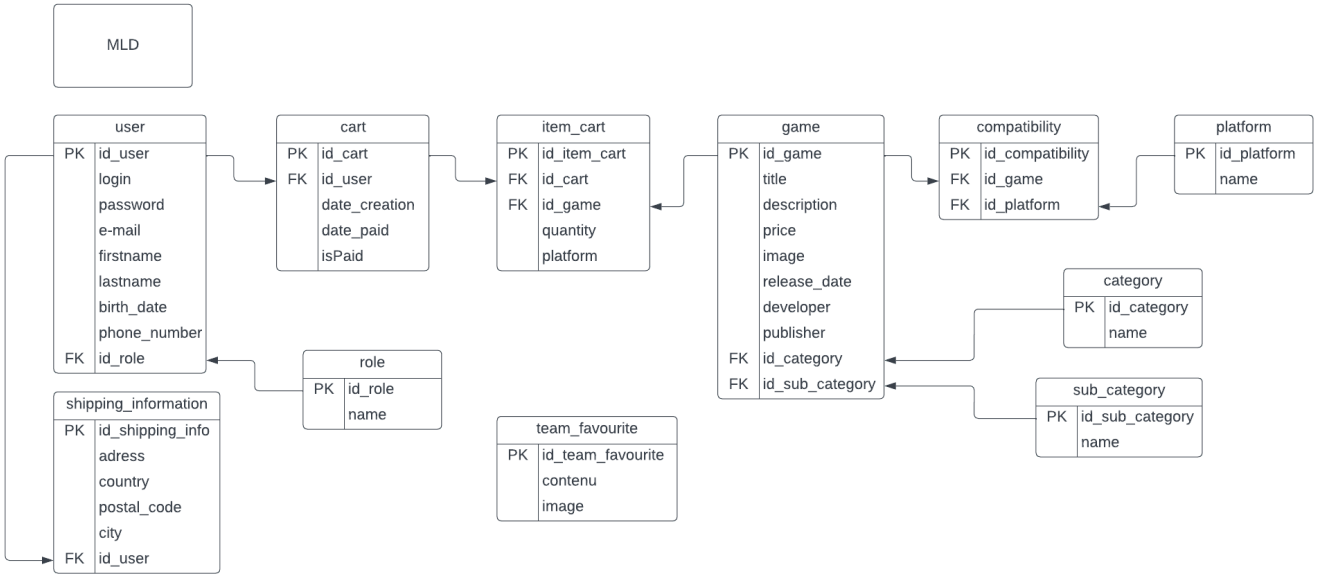


semaine 4

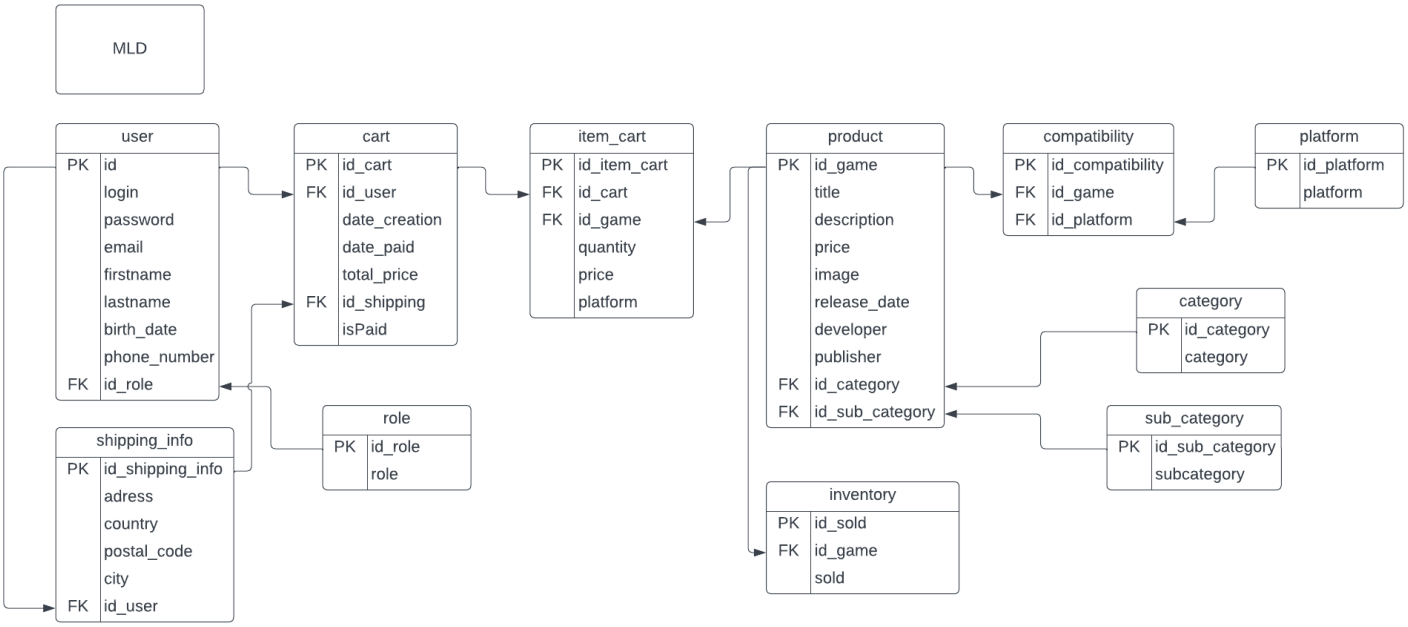


MLD

semaine 1

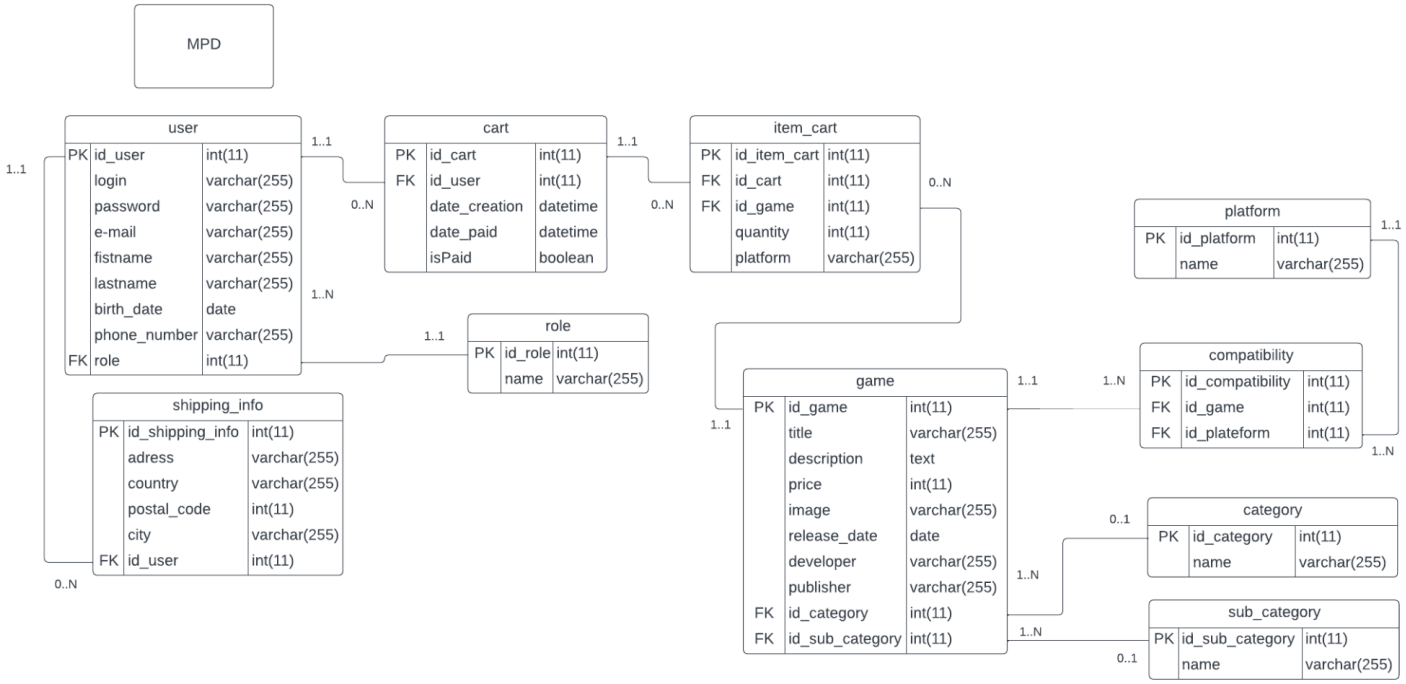


semaine 4

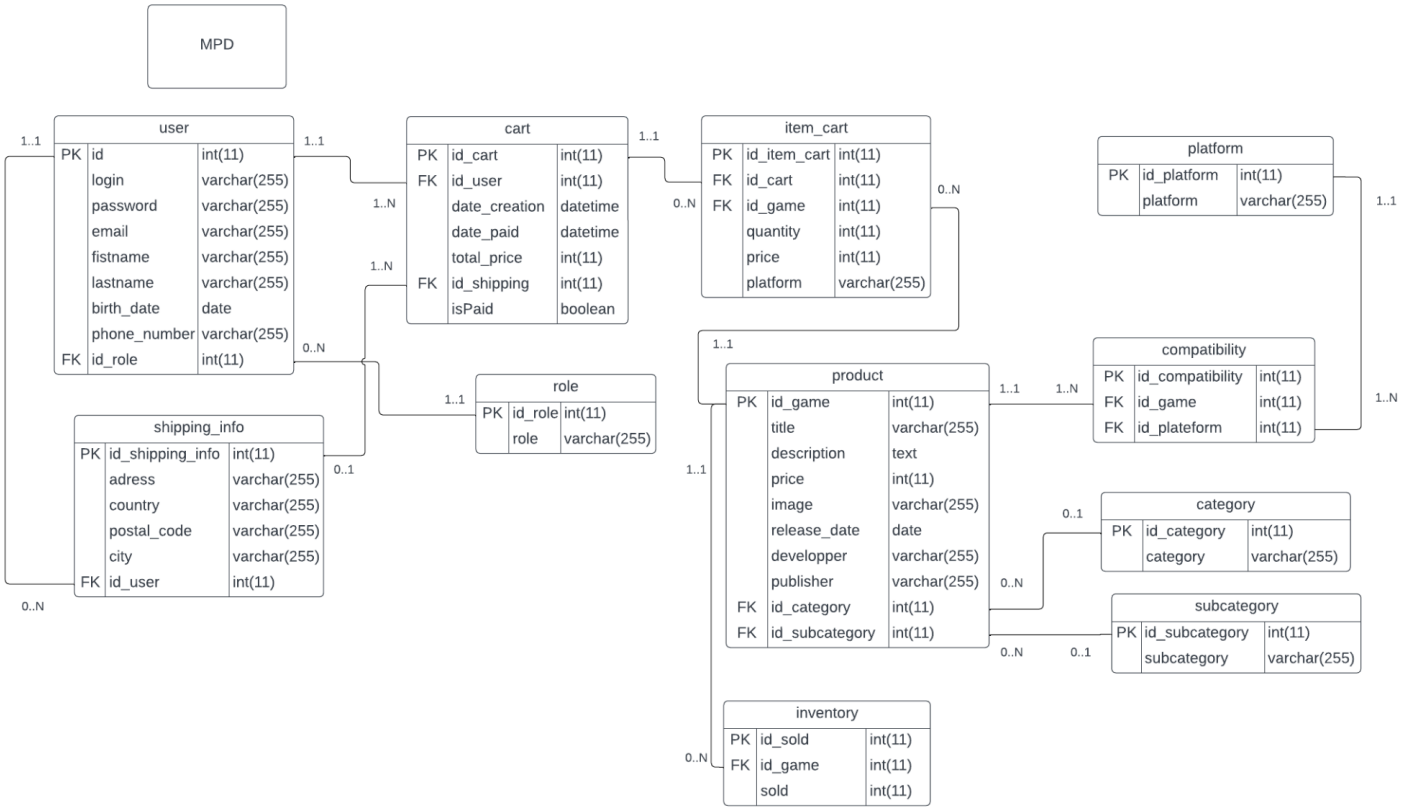


MPD

semaine 1



semaine 4



Hébergement : Le site est hébergé sur Plesk.

Responsive design : Le site a été conçu de manière à être adaptatif et à s'ajuster automatiquement à différents appareils et résolutions d'écran, offrant une expérience utilisateur cohérente sur ordinateurs de bureau, smartphones et tablettes.

Organisation au sein de l'équipe : chat gmail, Discord, communication orale

Réalisations

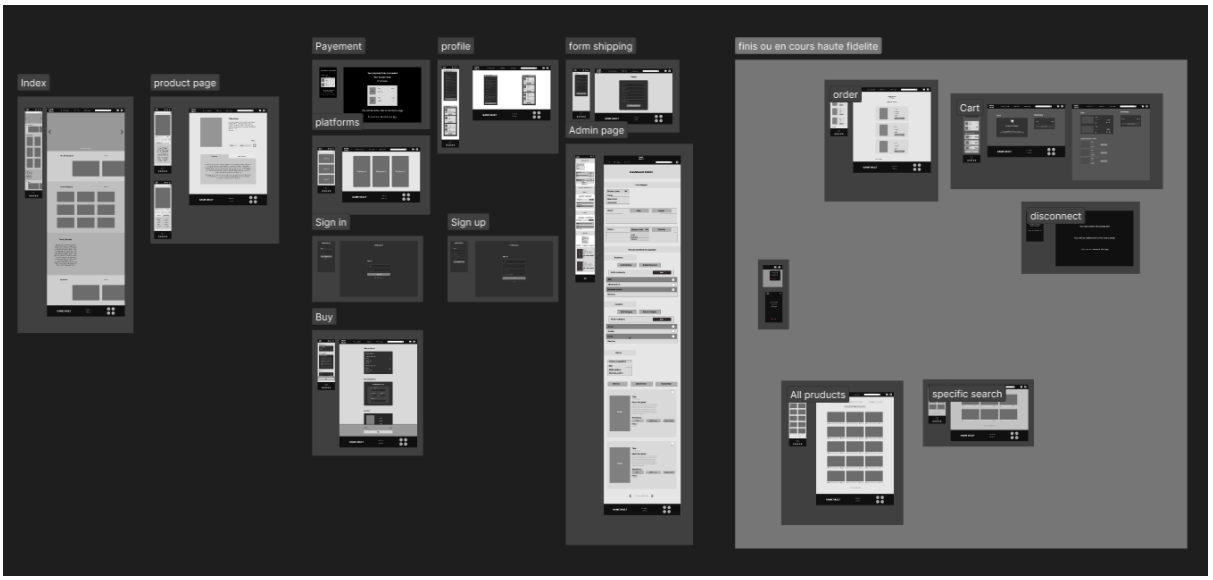
Charte graphique et maquettes

Charte graphique :

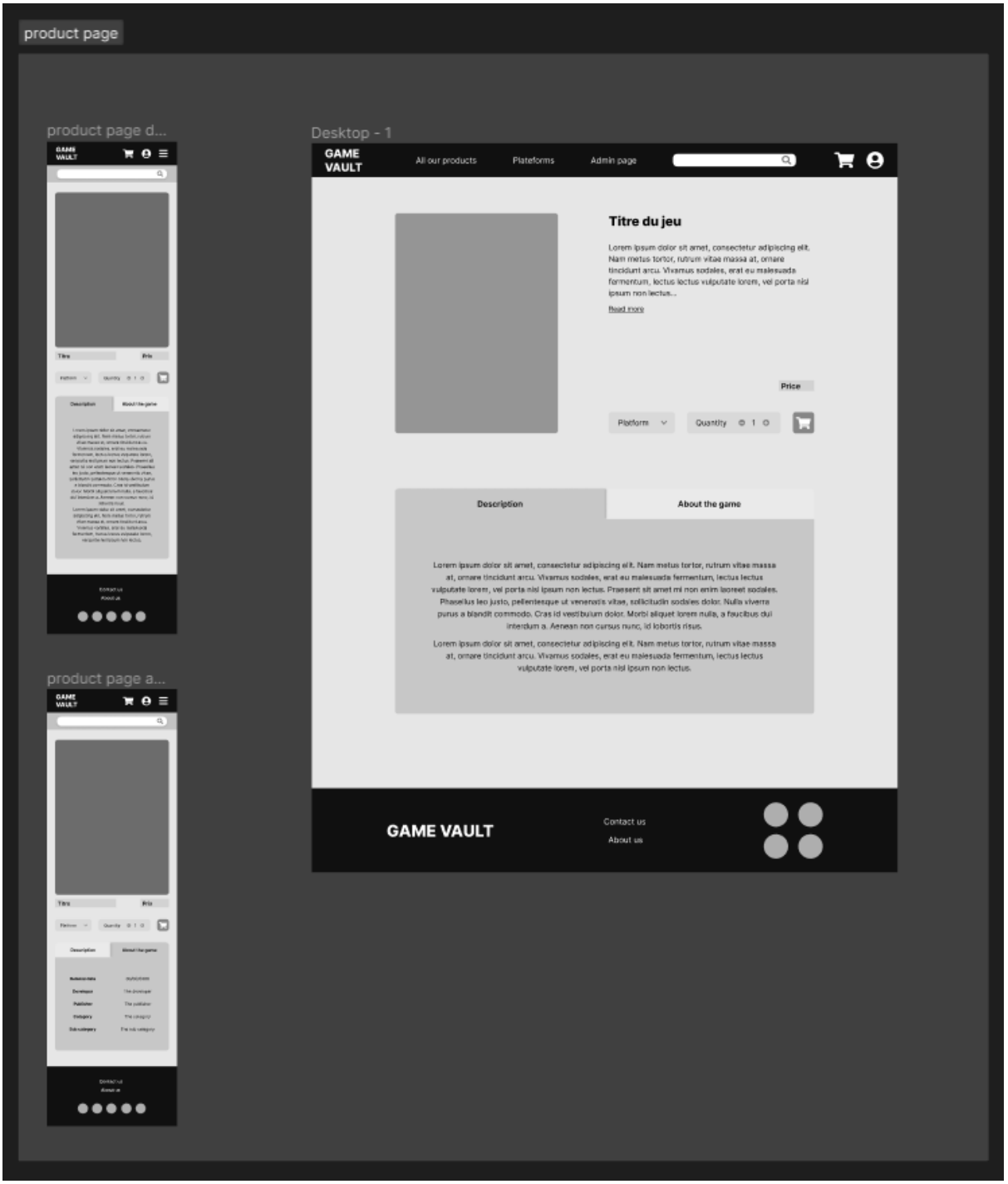
Pour la conception visuelle de notre site web, nous nous sommes inspirés de la charte graphique de Instant Gaming en raison de sa beauté et de son adéquation avec notre vision.

Nous avons trouvé que leur utilisation des couleurs sombres, des polices modernes et des éléments visuels dynamiques correspondait parfaitement à l'atmosphère que nous souhaitions créer pour notre propre site. Cela nous a permis de créer une expérience utilisateur cohérente et attrayante, tout en préservant notre identité unique.

Nous avons commencé par créer une maquette basse fidélité de notre site :

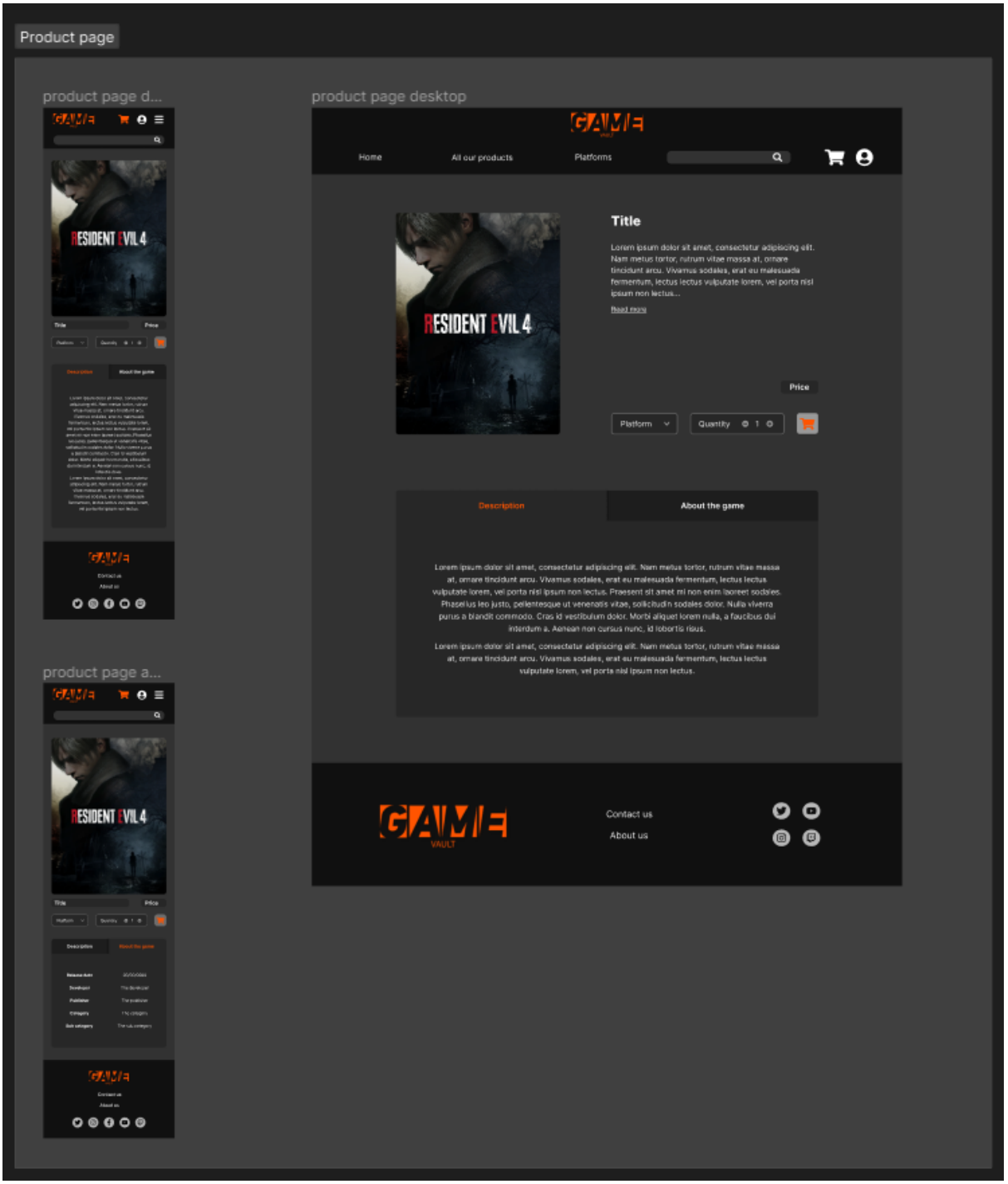


Pour chaque page, nous avons créé une représentation visuelle simplifiée de l'interface utilisateur avec quelques détails. Elle est en nuance de gris pour ne pas se concentrer sur les couleurs et le design, mais plutôt sur comment sont agencés les éléments sur notre page. Nous avons tout de même ajouté quelques détails comme des icônes de base ou des textes.



Puis, quand la maquette basse fidélité a été validée par toute l’équipe, nous sommes passés à la maquette haute fidélité, en ajoutant des couleurs, des images et des logos.

Voici la même page que précédemment, en haute fidélité



Extraits de code significatifs

Comme dans le PP mais avec plus d’explications écrites

DISPLAY DES PRODUITS ET PAGINATION

Model :

```php
public function GetProductByFilter($platform, $category, $subcategory) {

        $sqlParam = [];

        $platform === 'all' ? ($sqlPlat1 = "") . ($sqlPlat2 = "") : ($sqlPlat1 = " INNER JOIN compatibility ON product.id =
compatibility.id_game INNER JOIN platform ON compatibility.id_platform = platform.id") . ($sqlPlat2 = " AND id_platform = :platform") .
($sqlParam[':platform'] = $platform);

        $category === 'all' ? $sqlCat = "" : ($sqlCat = " WHERE id_category = :category") . ($sqlParam[':category'] = $category);

        $subcategory === 'all' ? $sqlSubcat = "" : ($sqlSubcat = " AND id_subcategory = :subcategory") . ($sqlParam[':subcategory'] =
$subcategory);

        $sql = "SELECT *,product.id FROM product" . $sqlPlat1 . $sqlCat . $sqlSubcat . $sqlPlat2;
        $req = $this->conn->prepare($sql);
        $req->execute($sqlParam);

        $tab = $req->fetchAll(PDO::FETCH_ASSOC);

        return $tab;

}
```

- requête traitée comme une string
- si les filtres sont à zéro on récupère tout
- conditions pour vérifier chaque filtre et créer le bout de requête qui correspond
- tableau pour les param

Controller :

```php
$productsTable = $product->GetProductByFilter($_GET['platform'], $_GET['category'], $_GET['subcategory']);

$pages = [];

$numPage = 1;

for ($i=0; $i < sizeof($productsTable); $i+=18) {

    for ($j=$i; $j < $i+18; $j++) {

        if(isset($productsTable[$j])) {
            $productsTable[$j]['price'] = substr_replace($productsTable[$j]['price'], ".", -2, 0) . "€";

            strlen($productsTable[$j]['title']) > 10 ? $productsTable[$j]['titleMobile'] = substr($productsTable[$j]['title'], 0, 10) . "..." :
$productsTable[$j]['titleMobile'] = $productsTable[$j]['title'] ;

            strlen($productsTable[$j]['title']) > 20 ? $productsTable[$j]['titleDesktop'] = substr($productsTable[$j]['title'], 0, 20) . "..." :
$productsTable[$j]['titleDesktop'] = $productsTable[$j]['title'] ;

            $pages[$numPage][$j] = <!-- html pout un jeu -->;
        }
    }

    $pages['numPage'][$numPage] = '<p class="changePage" id="page' . $numPage . '">' . $numPage . '</p>' ;

    $numPage++;
}

$json = json_encode($pages, JSON_PRETTY_PRINT);
echo $json;
```

- création du HTML pour chaque jeu
- premier for : représente une page, i = le num du jeu ou on en est à la fin de la page
- deuxième for : les jeux pour la page en cours. On formate le prix et le titre comme on veux et on crée le HTML du jeu avec ça
- on a aussi la pagination qui se crée à chaque tour de boucle du premier for
- puis on encode en json et on le echo pour pouvoir l'utiliser en js
- on stock les prix en int et pas en float (pourquoi?)(ici ou au niveau de la BDD?)

View :

```
const displayGames = async(selectPlatform, selectCategory, selectSubcategory, pageNum) => {

    const divDisplayArticles = document.getElementById('displayArticles');

    const response = await fetch('all_products.php?games=1&platform=' + selectPlatform + '&category=' + selectCategory + '&subcategory='
+ selectSubcategory);
    let schemaPages = await response.json();

    divDisplayArticles.innerHTML = "";

    if(schemaPages[pageNum] === undefined) {

        divDisplayArticles.innerHTML = "There is no products that fits you choices";

    }else{

        schemaPages[pageNum] = Object.values(schemaPages[pageNum])

        schemaPages[pageNum].forEach(gameHTML => {

            divDisplayArticles.innerHTML = divDisplayArticles.innerHTML + gameHTML;

        });
    }
}
```

- on cible la div ou on veux mettre nos jeux, on fetch le HTML des jeux qu'on acréé juste avant et on précise que on recoit du json
- on vide la div pour etre sur et si on a pas de jeux on affiche un message, et si il y a des jeux, on les ajoute un par un dans la div

Veille sur les vulnérabilités de sécurité

présenter les différentes failles
comment s’en prémunir

Il existe de nombreuses failles de sécurité possibles dans les sites web et les applications, voici quelques-unes des plus courantes :

Injection de code : L'injection de code se produit lorsque des données non validées sont insérées dans une requête ou une commande qui est ensuite exécutée par le serveur. Cela peut permettre aux attaquants d'exécuter du code malveillant, tels que des injections SQL, des injections de commandes système ou des scripts intersites.

Cross-Site Scripting (XSS) : L'attaque XSS se produit lorsque des données non fiables sont affichées dans une page web sans être correctement échappées ou filtrées. Cela permet aux attaquants d'injecter du code JavaScript malveillant dans les pages visitées par les utilisateurs, ce qui peut entraîner le vol de données sensibles ou la manipulation du contenu affiché.

Cross-Site Request Forgery (CSRF) : Le CSRF est une attaque où un site web malveillant force un utilisateur à effectuer une action non intentionnelle sur un autre site web sur lequel il est authentifié. Cela peut conduire à des actions non autorisées, telles que la modification des paramètres de compte, la soumission de formulaires ou le vol de données.

Vulnérabilités de gestion des sessions : Les vulnérabilités de gestion des sessions peuvent permettre aux attaquants d'usurper l'identité d'un utilisateur authentifié ou de voler des informations de session. Cela peut se produire en raison d'une mauvaise génération ou d'une mauvaise gestion des identifiants de session, de l'utilisation de cookies non sécurisés, ou de la réutilisation de sessions après la déconnexion.

Vol de données sensibles : Le vol de données sensibles peut se produire lorsque des informations confidentielles, telles que des identifiants de connexion, des informations financières ou des informations personnelles, sont stockées ou transmises de manière non sécurisée. Cela peut résulter d'une mauvaise configuration de la sécurité, de l'absence de chiffrement des données ou de la vulnérabilité des bases de données.

Vulnérabilités du système de gestion de contenu (CMS) : Les CMS populaires tels que WordPress, Drupal ou Joomla peuvent présenter des failles de sécurité si les versions utilisées ne sont pas mises à jour régulièrement. Des vulnérabilités dans les extensions, les thèmes ou les paramètres de configuration peuvent également être exploitées par les attaquants.

Déni de service (DoS) : Les attaques par déni de service visent à rendre un site web ou une application indisponible en surchargeant les ressources du serveur. Cela peut être réalisé en

inondant le serveur avec un trafic excessif, en exploitant des vulnérabilités logicielles ou en épuisant les ressources système.

Il est important de noter que cette liste n'est pas exhaustive et que de nouvelles failles de sécurité peuvent apparaître à mesure que de nouvelles technologies et méthodes d'attaque émergent. Il est essentiel de mettre en place des mesures de sécurité appropriées et de suivre les meilleures pratiques de développement pour atténuer ces risques.

Pour se prémunir des différentes failles de sécurité mentionnées précédemment, voici quelques mesures de prévention recommandées :

Injection de code :

Utilisez des requêtes préparées ou des mécanismes d'ORM pour éviter les injections SQL.
Échappez ou filtrez correctement toutes les données utilisateur avant de les utiliser dans des requêtes ou des commandes.
Cross-Site Scripting (XSS) :

Validez et filtrez les entrées utilisateur pour empêcher l'exécution de code non autorisé.
Échappez correctement les données avant de les afficher sur les pages web.
Utilisez des en-têtes de sécurité HTTP, tels que Content Security Policy (CSP), pour limiter l'exécution de scripts externes.
Cross-Site Request Forgery (CSRF) :

Utilisez des jetons anti-CSRF (CSRF tokens) pour vérifier l'origine des requêtes.
Exigez une double authentification pour les actions sensibles.
Vérifiez les en-têtes HTTP Referer ou Origin pour valider l'origine des requêtes.
Vulnérabilités de gestion des sessions :

Générez des identifiants de session aléatoires et uniques.
Stockez les identifiants de session de manière sécurisée, en utilisant des cookies sécurisés et en les configurant correctement.
Renouvelez les identifiants de session après la connexion ou la déconnexion.
Vol de données sensibles :

Utilisez des protocoles de chiffrement tels que HTTPS pour sécuriser les communications entre le client et le serveur.
Stockez les données sensibles de manière sécurisée, en utilisant des techniques de chiffrement appropriées.

Suivez les meilleures pratiques de sécurité pour la protection des bases de données, telles que la limitation des accès et la sécurisation des connexions.
Vulnérabilités du système de gestion de contenu (CMS) :

Mettez à jour régulièrement votre CMS et ses extensions, en installant les correctifs de sécurité disponibles.
Utilisez des thèmes et des extensions provenant de sources fiables.
Restreignez les permissions d'accès aux fichiers et aux répertoires du CMS pour limiter les risques de compromission.
Déni de service (DoS) :

Utilisez des pare-feux et des systèmes de détection d'intrusion (IDS) pour surveiller et limiter le trafic suspect.
Optimisez les performances du serveur et mettez en place des mécanismes de limitation de la bande passante.
Planifiez des tests de charge et des simulations de déni de service pour évaluer et renforcer la résistance du système.
Il est également recommandé de suivre les meilleures pratiques de sécurité tout au long du cycle de développement, d'implémenter des mécanismes de surveillance et de journalisation des activités suspectes, et de sensibiliser les utilisateurs aux bonnes pratiques de sécurité, telles que la création de mots de passe forts et la protection de leurs informations personnelles.

La sécurité étant un domaine en constante évolution, il est important de rester informé des dernières vulnérabilités et des meilleures pratiques en matière de sécurité