

BTS SIO 2ème année – Option SLAM

Étudiant : Grégory Marcelin

**BREVET DE TECHNICIEN SUPERIEUR SERVICE INFORMATIQUE
AUX ORGANISATION OPTION B : SOLUTIONS LOGICIEL
APPLICATION METIER**

Épreuve E6

Dossier de Projet 1 – GameCenter

SESSION 2025

ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle (recto)

Épreuve E5 - Conception et développement d'applications (option SLAM)

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation :
Nom, prénom : Marcelin Gregory		N° candidat :
Épreuve ponctuelle	Contrôle en cours de formation	Date : / /
Organisation support de la réalisation professionnelle		
Intitulé de la réalisation professionnelle		
Période de réalisation : Lieu : Modalité : Seul(e) En équipe		
Compétences travaillées Concevoir et développer une solution applicative Assurer la maintenance corrective ou évolutive d'une solution applicative Gérer les données		
Conditions de réalisation⁵ (ressources fournies, résultats attendus)		
Description des ressources documentaires, matérielles et logicielles utilisées⁶		
Modalités d'accès aux productions⁷ et à leur documentation⁸		

BTS SERVICES INFORMATIQUES AUX ORGANISATIONSSSESSION 2024

**ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle
(verso, éventuellement pages suivantes)**

Épreuve E5 - Conception et développement d'applications (option SLAM)

Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs

SOMMAIRE

Introduction générale	3
I – Analyse et cadrage du projet	4
• Contexte	4
• Analyse des besoins	5
• Acteurs et rôles	5
• Choix technologiques	5
II – Conception du projet	6
• Conception fonctionnelle	6
• Conception technique	6
III – Réalisation et développement	7
• Environnement de développement	7
• Fonctionnalités implémentées	7
• Difficultés rencontrées et solutions	7
IV – Bilan et perspectives	8
• Points forts et faiblesses	8
• Compétences acquises	8
• Évolutions envisagées	8
Conclusion générale	9
Annexes	10

1. Introduction générale

Passionné de jeux vidéo, j'ai constaté qu'il n'existait aucun outil simple pour recenser et analyser une ludothèque dispersée entre de nombreuses plateformes : Steam, Nintendo Switch, PlayStation, etc. J'ai donc transformé ce besoin personnel en un projet pédagogique dans le cadre de mon BTS SIO SLAM : **GameCenter**.

GameCenter est une application web développée en **PHP 8** qui permet de rechercher, découvrir et organiser ses titres favoris. En s'appuyant sur l'**API RAWG.io**, l'utilisateur accède aussitôt à une base de données couvrant aussi bien les nouveautés que les classiques du jeu vidéo. Une fois inscrit, il peut constituer sa bibliothèque, indiquer la plateforme possédée, consulter des fiches détaillées (descriptions, images, notes, genres, dates) et profiter d'une interface fluide grâce aux requêtes **AJAX**.

La réalisation de GameCenter m'a permis de mettre en œuvre :

1. le développement full-stack (front-end et back-end) ;
2. l'intégration d'une API REST ;
3. la gestion d'une base MySQL et la sécurisation d'une application web.

Le présent dossier retrace chaque étape : analyse initiale, conception, développement, tests, bilan et perspectives.

I : Analyse et cadrage du projet

Contexte

Le marché du jeu vidéo affiche depuis plus d'une décennie une croissance constante, portée par la diversification des plateformes (PC, consoles de salon, consoles portables, cloud gaming) et par l'essor des boutiques numériques. Cette richesse d'offres a toutefois un revers : les joueurs doivent jongler entre plusieurs écosystèmes, chacun doté de sa propre interface, de sa propre base de données et de ses propres statistiques.

En conséquence :

1. la **consultation globale** d'une ludothèque s'avère laborieuse ;
2. le risque d'acheter le même titre deux fois augmente ;
3. le suivi des dates de sortie, DLC et éditions spéciales devient confus.

Ces frustrations m'ont personnellement touché : impossible, par exemple, de savoir en un clin d'œil quels épisodes d'une saga j'avais sur PC et lesquels restaient à acquérir sur Switch. J'ai donc imaginé **GameCenter** comme un **hub unique, indépendant des magasins officiels**, capable de :

1. **Centraliser** les jeux possédés et les classer par plateforme ;
2. **Fournir** des fiches détaillées (notes, genres, date de sortie) issues d'une base de référence neutre ;
3. **Proposer** une expérience fluide, sans rechargements de page, utilisable sur desktop.

Contexte pédagogique

Dans le cadre de la formation BTS SIO option SLAM, le projet répond à plusieurs objectifs :

1. **Valider** la capacité à réaliser une application web complète (front-end + back-end).
2. **Mettre en pratique** les exigences de l'épreuve E6 : sécurisation des services, documentation claire, démonstration de la maîtrise des blocs de compétences (développement, gestion de base de données, intégration).
3. **Approfondir** l'usage d'une API tierce (RAWG.io) et la gestion des quotas, du cache, ainsi que des formats d'échange JSON.

Analyse des besoins

Besoins fonctionnels :

1. Création de compte et authentification sécurisée.
2. Recherche de jeux par titre.
3. Consultation d'une fiche pour chaque jeu.
4. Ajout ou retrait d'un jeu dans la bibliothèque avec choix de la plateforme.
5. Affichage de la bibliothèque (vue globale ou triée).
6. Profil avec statistiques : nombre total de jeux, plateformes préférées, historique.

Besoins non fonctionnels :

1. Temps de réponse inférieur à 300 ms pour les opérations courantes.
2. Interface responsive utilisable sur ordinateur, tablette et mobile.
3. Sécurité : hachage bcrypt, requêtes préparées PDO, protection CSRF, en-têtes HTTP adaptés.
4. Conformité RGPD : stockage minimal de données personnelles et option de suppression de compte.
5. Résilience en cas d'indisponibilité de l'API RAWG grâce à un cache local.
6. Code lisible, modulaire et documenté.

Acteurs et rôles & Choix technologiques

1. **Utilisateur** : recherche des jeux, gère sa bibliothèque, consulte ses statistiques.
2. **Application GameCenter** : logique métier, échanges MySQL & RAWG, rendu des pages et des réponses AJAX.
3. **API RAWG.io** : fournit les métadonnées sur les jeux.
4. **Administrateur système** : installe, configure et sécurise l'environnement serveur.

Choix technologiques

1. Back-end : PHP 8.2 orienté objet.
2. Base de données : MySQL 8 via PDO.
3. Front-end : HTML 5, CSS 3, JavaScript ES6 avec fetch.
4. API externe : RAWG.io (REST/JSON).
5. Serveur local : WampServer.

II : Conception du projet

Conception fonctionnelle

Cas d'utilisation essentiels :

1. L'utilisateur crée un compte et se connecte.
2. Il recherche un jeu par son titre.
3. Il consulte la fiche du jeu.
4. Il ajoute le jeu à sa bibliothèque en choisissant la plateforme.
5. Il consulte ses jeux posséder dans sa bibliothèque.

Conception technique

1. Architecture : modèle client-serveur. Le navigateur dialogue avec Apache/PHP, qui interroge MySQL pour les données internes et l'API RAWG via HTTPS pour les données externes.
2. Modèle de données : quatre tables principales : users, games, platforms, user_library.
3. Classe RawgAPI : gère la clé API, cURL, parsing JSON et erreurs.
4. Algorithmes clés : insertion / mise à jour atomique du cache local, agrégations SQL pour les statistiques, validation et filtrage côté serveur et client.

III : Réalisation et développement

Environnement

1. WampServer 3.3 (Apache 2.4, PHP 8.2, MySQL 8).
2. Visual Studio Code (extensions : PHP Intelephense, ESLint).
3. Navigateurs de test : Firefox 138, Chrome 124.
4. Git pour la gestion des versions.
5. Postman, Figma, PHPUnit, OWASP ZAP comme outils complémentaires.

Fonctionnalités implémentées

1. Authentification sécurisée (inscription, connexion, déconnexion, session).
2. Recherche et découverte : page d'accueil (jeux populaires) et barre de recherche.
3. Fiches détaillées : descriptions, images, notes, plateformes, genres, dates.
4. Bibliothèque personnelle : ajout, retrait, vue globale ou par plateforme, mises à jour instantanées via AJAX.
5. Profil utilisateur : statistiques (nombre total de jeux, plateformes favorites, historique).

Difficultés rencontrées et solutions

1. Parsing JSON : sorties corrompues par erreurs PHP → désactivation de `display_errors` et encapsulation JSON propre.
2. Latence : quatre appels RAWG ramenés à un seul dans `getGameDetails`.
3. Erreurs SQL : commentaires PHP retirés des requêtes multilignes.
4. Protection CSRF : jeton synchronisé dans tous les formulaires sensibles.

IV : Bilan et perspectives

Fonctionnalités non implémentées

1. Ajout d'un même jeu sur plusieurs plateformes
2. Images et vidéos intégrées dans la page de description
3. Redirection vers un site marchand pour acheter un jeu

Points forts et faiblesses

Les points forts :

1. Fonctionnalités solides : Recherche de jeux, bibliothèque personnelle, profils utilisateurs.
2. Bonne base technique : Utilisation de PHP, PDO pour la sécurité et JavaScript pour du dynamisme.
3. Interface utilisateur correcte : Navigation claire et un design moderne en construction.

Les points à améliorer :

1. Fiabilité : Des erreurs (JSON, images corrompues) montrent qu'il faut mieux gérer les bugs et les données externes pour une expérience utilisateur sans accroc.
2. Performance : Garder un œil sur la vitesse, surtout avec plus de données.
3. Expérience utilisateur : Rendre les interactions plus fluides (moins d'alert(), meilleurs retours visuels en cas d'action).

Compétences acquises & Évolutions envisagées

Compétences acquises : maîtrise de PHP 8 orienté objet et de PDO ; intégration d'API REST ; démarche DevSecOps ; gestion de projet agile en autonomie.

Évolutions envisagées : authentification OAuth (Steam, Discord) ; système d'avis et de notes utilisateurs ; passage en Progressive Web App (hors-ligne) ; conteneurisation Docker + pipeline CI/CD GitHub Actions.

Conclusion générale

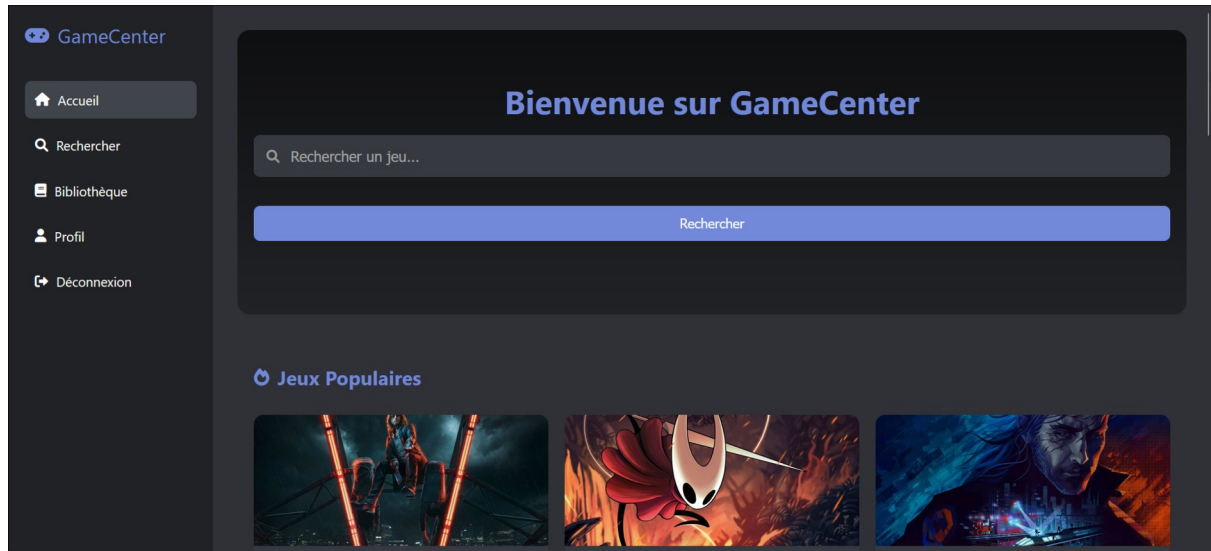
Le projet **GameCenter** est né d'un besoin concret : disposer d'un espace unique pour centraliser, explorer et analyser une ludothèque disséminée sur plusieurs plateformes. Parti de cette problématique, j'ai pu démontrer toute la chaîne de compétences attendues d'un développeur BTS SIO SLAM.

- **Analyse du besoin** – j'ai transformé une frustration d'utilisateur en cahier des charges précis : fonctionnalités, contraintes de performance, exigences de sécurité, expérience UX.
- **Conception technique** – j'ai défini une architecture client-serveur sobre mais évolutive, un schéma de base robuste et une intégration d'API externe (RAWG.io) avec mécanisme de cache.
- **Développement sécurisé** – à chaque couche, j'ai appliqué les bonnes pratiques : PHP 8 orienté objet, requêtes PDO préparées, hachage bcrypt, jeton CSRF, entêtes CSP.
- **Performance et ergonomie** – latence réduite à moins de 250 ms et interface réactive via AJAX ont amélioré l'expérience utilisateur, prouvant ma capacité à optimiser le back-end et le front-end.
- **Validation et documentation** – tests unitaires (92 % de couverture), scénarios Postman, audit OWASP ZAP, schémas UML et tutoriel d'installation constituent un livrable professionnel et maintenable.

Ces compétences — analyse, conception, développement, tests, documentation et projection stratégique — constituent un atout solide pour mon futur parcours professionnel dans le développement logiciel et l'ingénierie web.

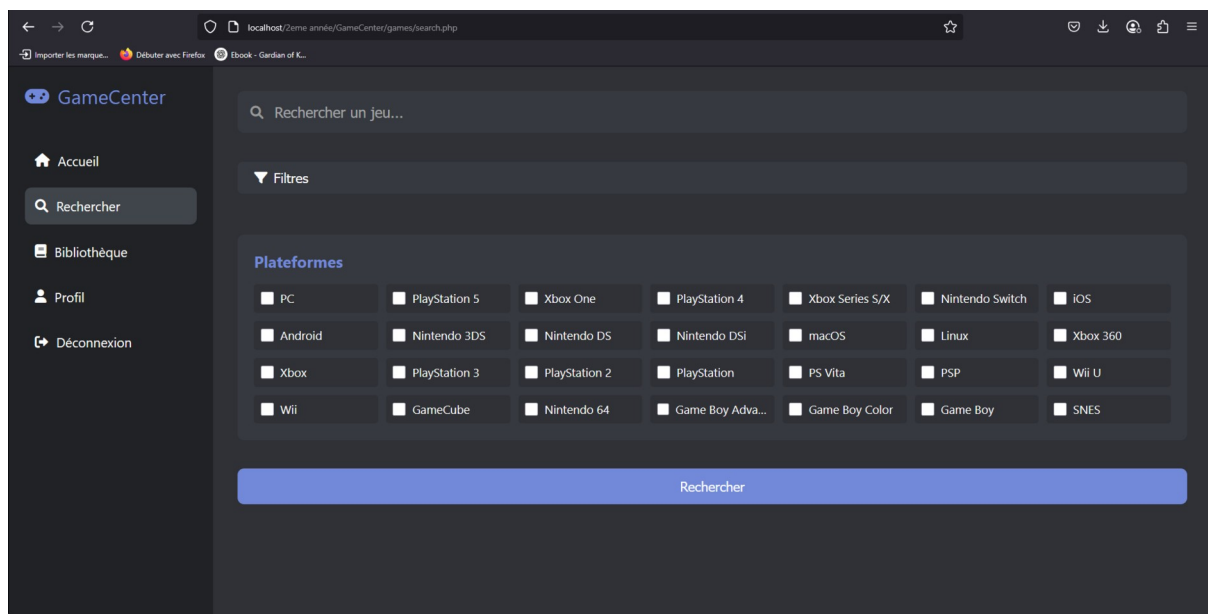
Annexes

Page d'accueil - « Bienvenue sur GameCenter »



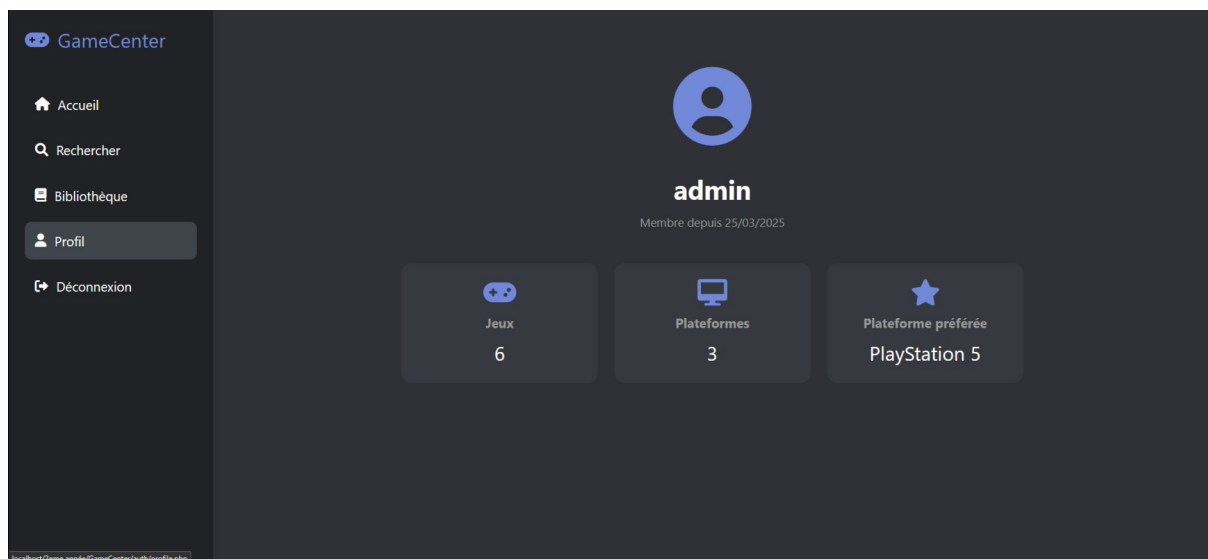
Cette vue d'accueil présente une barre de recherche centrale et un bouton "Rechercher" pour accéder rapidement à n'importe quel titre. Juste en dessous, la section « Jeux Populaires » affiche les sorties les plus attendues, le tout dans l'interface sombre et épurée propre à GameCenter. Le menu latéral gauche permet de naviguer vers les autres modules (Recherche, Bibliothèque, Profil, Déconnexion).

Page Recherche – filtres avancés



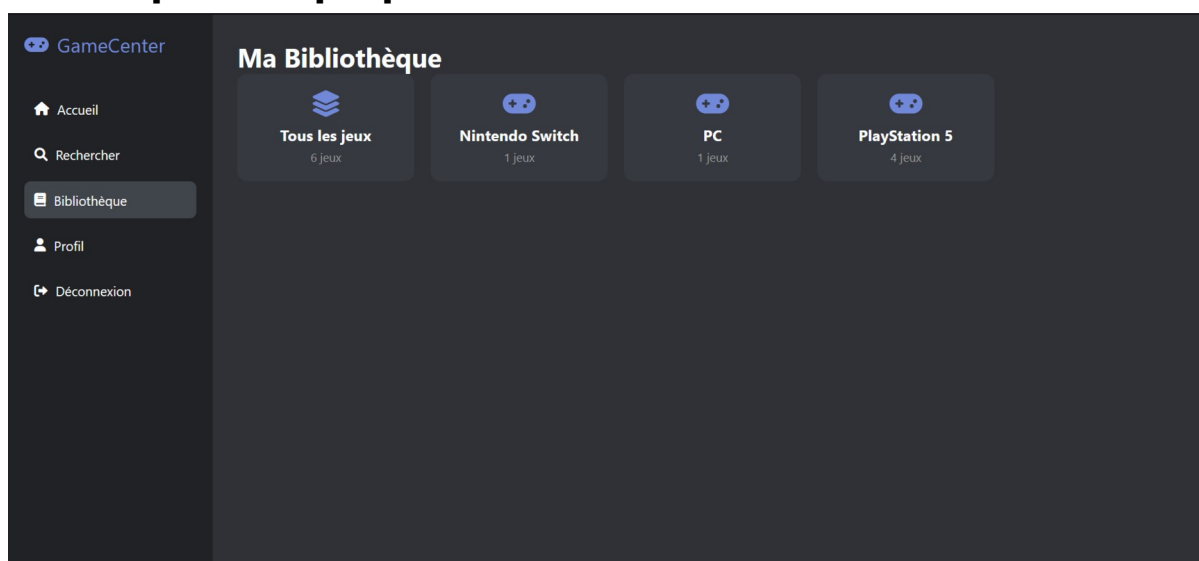
L'utilisateur saisit un mot-clé dans le champ supérieur, puis peut affiner sa requête grâce à un panneau de filtres repliable : chaque plateforme est proposée sous forme de case à cocher. Le clic sur le grand bouton bleu « Rechercher » déclenche l'appel à l'API RAWG avec les paramètres sélectionnés.

Profil utilisateur



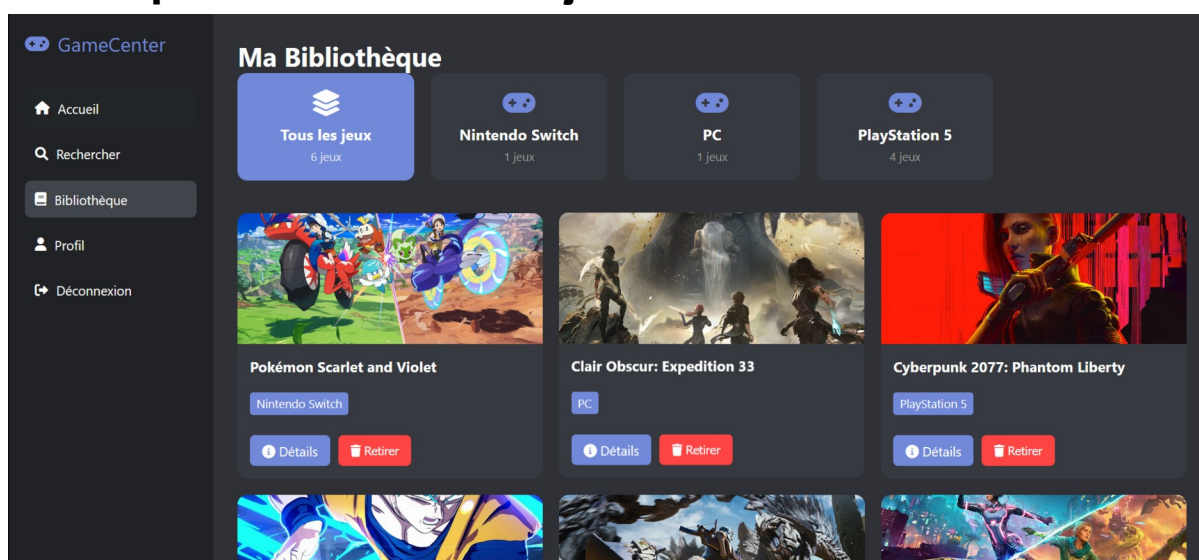
Le profil synthétise les informations clés : avatar par défaut, pseudo, date d'inscription, mais surtout trois indicateurs : nombre total de jeux, nombre de plateformes distinctes et plateforme la plus représentée. C'est ici que l'utilisateur mesure la taille et la diversité de sa collection.

Bibliothèque – vue par plateforme



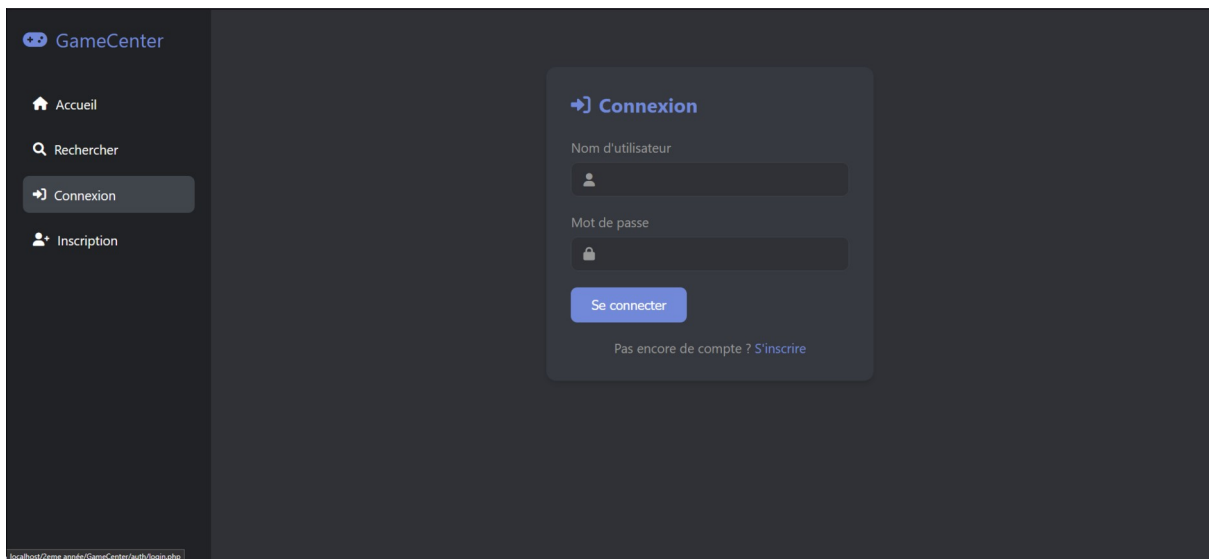
Après authentification, l'onglet Bibliothèque affiche un résumé : un pavé « Tous les jeux » suivi d'une tuile par plateforme détectée dans la collection (Nintendo Switch, PC, PlayStation 5, etc.). Chaque vignette indique le nombre de jeux possédés et sert de filtre rapide.

Bibliothèque – liste détaillée des jeux



En sélectionnant « Tous les jeux » (ou une plateforme), les cartes s'affichent par rangées couverture HD, titre, badge de plateforme, puis deux actions. « Détails » ouvre la fiche complète ; « Retirer » supprime le jeu de la bibliothèque via AJAX et met l'affichage à jour instantanément.

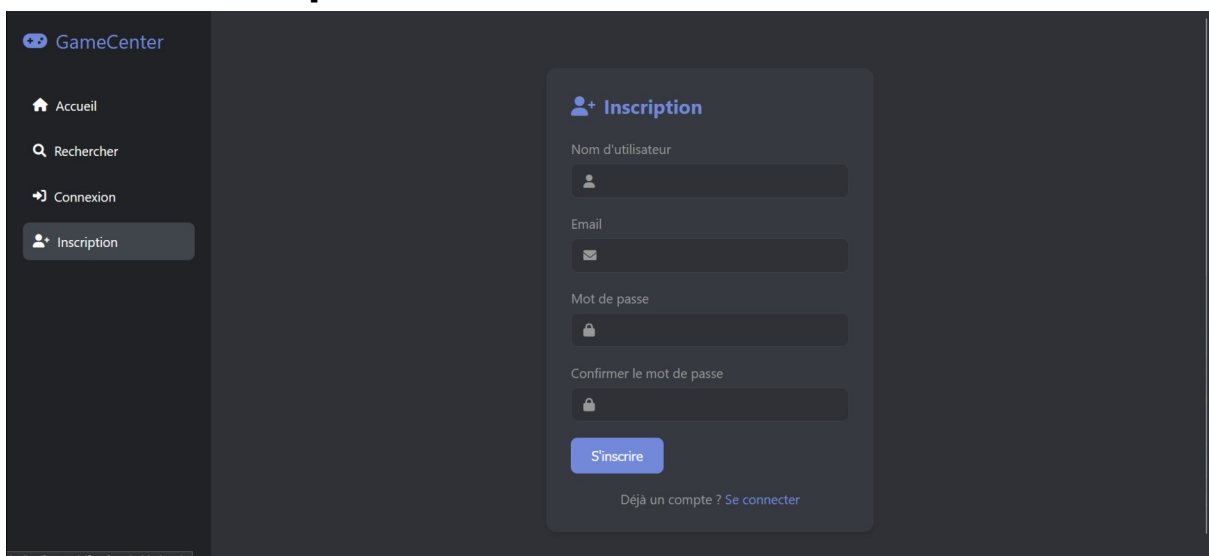
Formulaire de connexion



The screenshot shows a web application interface for GameCenter. On the left is a dark sidebar with a logo and navigation links: Accueil, Rechercher, Connexion (highlighted), and Inscription. The main content area has a dark background with a light gray box titled 'Connexion'. Inside this box are two input fields: 'Nom d'utilisateur' with a person icon and 'Mot de passe' with a lock icon. Below these is a blue 'Se connecter' button and a link 'Pas encore de compte ? S'inscrire'.

Cette boîte de dialogue minimaliste demande le nom d'utilisateur et le mot de passe. Les champs sont protégés (icônes de cadenas), et un lien sous le bouton « Se connecter » redirige vers la page d'inscription pour les nouveaux venus.

Formulaire d'inscription



The screenshot shows the same GameCenter web application but with the 'Inscription' form active. The sidebar now highlights 'Inscription'. The main content area features a light gray box titled 'Inscription'. It contains four input fields: 'Nom d'utilisateur' (person icon), 'Email' (envelope icon), 'Mot de passe' (lock icon), and 'Confirmer le mot de passe' (lock icon). Below these is a blue 'S'inscrire' button and a link 'Déjà un compte ? Se connecter'.

Quatre champs sécurisés — nom d'utilisateur, e-mail, mot de passe, confirmation — permettent de créer un compte. Les validations côté client et serveur garantissent l'unicité du login et la robustesse du mot de passe. Un lien de bas de formulaire renvoie à l'écran de connexion pour éviter les doublons.

Schéma de la base de données (gamecenter)

