

Documentation Projet CyberScale

v1.0.0

Table of Contents

| | |
|--|---|
| 1. Présentation | 1 |
| 1.1. L'Équipe | 1 |
| 1.2. Vision et Objectifs | 2 |
| 1.3. Analyse de la Concurrence | 2 |
| 2. Architecture et Technique | 2 |
| 2.1. Architecture Détailée | 2 |
| 2.1.1. 1. Vue Logique : Traitement & Services | 2 |
| 2.1.2. 2. Vue de Données : Persistance | 3 |
| 2.2. Stack Technique | 3 |
| 2.3. ☐ Focus : La Dockerisation (Cœur du Projet) | 4 |
| 3. Fonctionnalités (Features v1.0.0) | 4 |
| 3.1. F1: Onboarding & Quiz Adaptatif ☐ | 4 |
| 3.2. F2: Cyber Arena (Red Team) ☐ | 5 |
| 3.3. F3: Module Phishing (Facteur Humain) ☐ | 6 |
| 3.4. F4: Module Investigation (Blue Team) ☐☐ | 7 |
| 3.5. F5: Dashboard Unifié & Certification ☐ | 8 |
| 4. Gestion de Projet & Qualité ☐☐ | 9 |
| 4.1. Sécurité & Authentification | 9 |
| 4.2. Qualité du Code (CI/CD) | 9 |
| 5. Conclusion & Perspectives ☐ | 9 |

Documentation technique et fonctionnelle de la plateforme CyberScale.
Cette version v1.0.0 marque l'aboutissement du projet avec une architecture conteneurisée complète, des modules d'attaque et de défense, et un tableau de bord unifié.

1. Présentation

1.1. L'Équipe

Projet réalisé dans le cadre du cursus DevOps par :

- **Lyes SEHILA** - Lead DevOps & Architecte

- Hassan Jatta - Lead Backend
- Abdoulaye - Lead Frontend

1.2. Vision et Objectifs

CyberScale a pour mission de fournir une évaluation "360 degrés" des compétences en cybersécurité.

.Concept Clé : La Matrice de Compétences

Nous positionnons l'utilisateur sur deux axes complémentaires :

1. **L'Axe Théorique (X)** : Maîtrise des concepts, normes (ISO 27001), et lois (RGPD).
2. **L'Axe Technique (Y)** : Capacité opérationnelle (Pentest, Scripting, Analyse de logs).

1.3. Analyse de la Concurrence

Le marché est polarisé :

- **CTF (RootMe)** : Excellent pour la technique pure, mais austère et sans théorie.
- **Certifications (CISSP)** : Très théoriques, coûteuses, sans pratique immédiate.

CyberScale réunit ces deux mondes : nous ne formons pas que des hackers, mais des professionnels complets (Red Team & Blue Team).

2. Architecture et Technique

2.1. Architecture Détailée

Pour garantir une meilleure lisibilité, l'architecture est présentée en deux vues distinctes.

2.1.1. 1. Vue Logique : Traitement & Services

Ce diagramme illustre la couche "Intelligente" de CyberScale. On y voit comment les Contrôleurs (API) orchestrent les Services métiers. Notez particulièrement l'interaction entre **ArenaController** et le **ContainerService** (gestion Docker), ainsi que le générateur de logs pour la Blue Team.

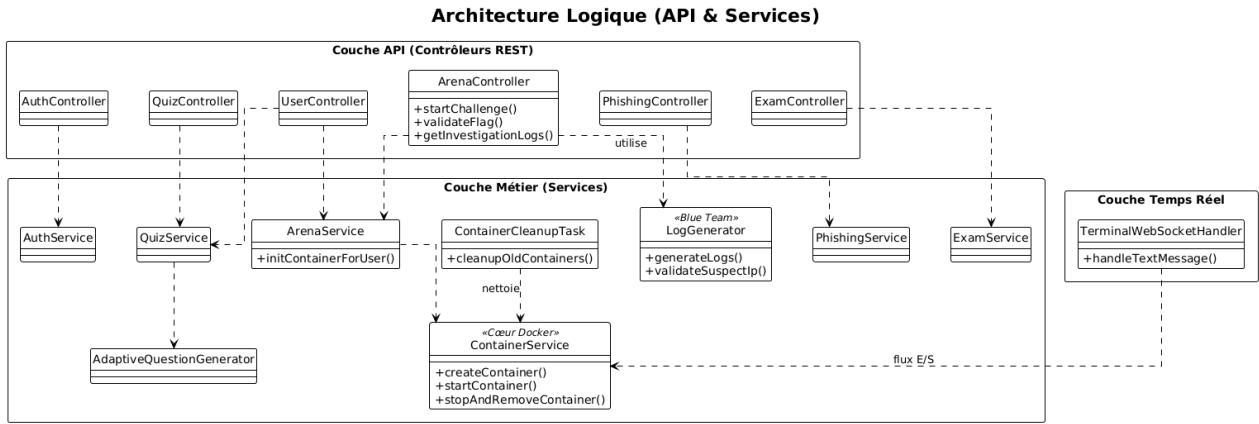


Figure 1. Architecture Logique (Contrôleurs & Services)

2.1.2. 2. Vue de Données : Persistance

Ce diagramme présente la structure de la base de données PostgreSQL (Entités JPA). Il met en évidence la centralisation des données autour de l'entité **User**, qui est liée à ses résultats de quiz, ses challenges réussis et ses examens.

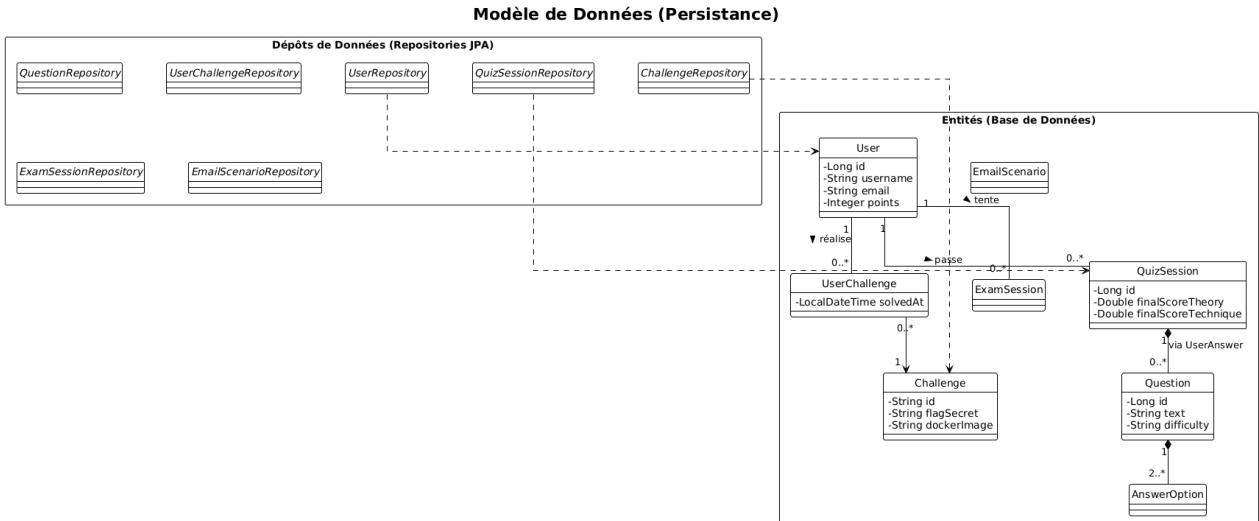


Figure 2. Modèle de Données (Entités JPA)

2.2. Stack Technique

| Domaine | Technologies |
|-----------------|--|
| Backend | Java 21, Spring Boot 3 (Security, JPA, WebSocket) |
| Données | PostgreSQL 16 (Production) |
| Frontend | HTML5, CSS3, JavaScript (Vanilla), Xterm.js (Terminal Web) |
| DevOps | Docker, Docker Compose, GitHub Actions (CI/CD), SonarCloud |
| Tests | JUnit 5, Selenium (E2E), Cucumber (BDD) |

2.3. □ Focus : La Dockerisation (Cœur du Projet)

C'est l'innovation technique majeure de cette release. Pour permettre aux utilisateurs de s'entraîner sans risque, nous avons développé un orchestrateur de conteneurs interne.

- **Isolation Totale** : Chaque challenge technique lance un conteneur Docker dédié à l'utilisateur. Il est impossible d'impacter le serveur hôte.
- **Gestion du Cycle de Vie** :
 - **Création** : Instanciation à la volée via l'API `docker-java` lors du clic sur "Démarrer".
 - **Communication** : Un tunnel **WebSocket** relie le terminal Web (`Xterm.js`) à l'entrée/sortie standard du conteneur Linux.
 - **Nettoyage** : Une tâche planifiée (`ContainerCleanupTask`) détruit automatiquement les conteneurs inactifs ou terminés pour libérer les ressources.
- **Sécurité** : Les conteneurs sont éphémères et n'ont aucun accès réseau vers l'extérieur (sauf si nécessaire pour le scénario).

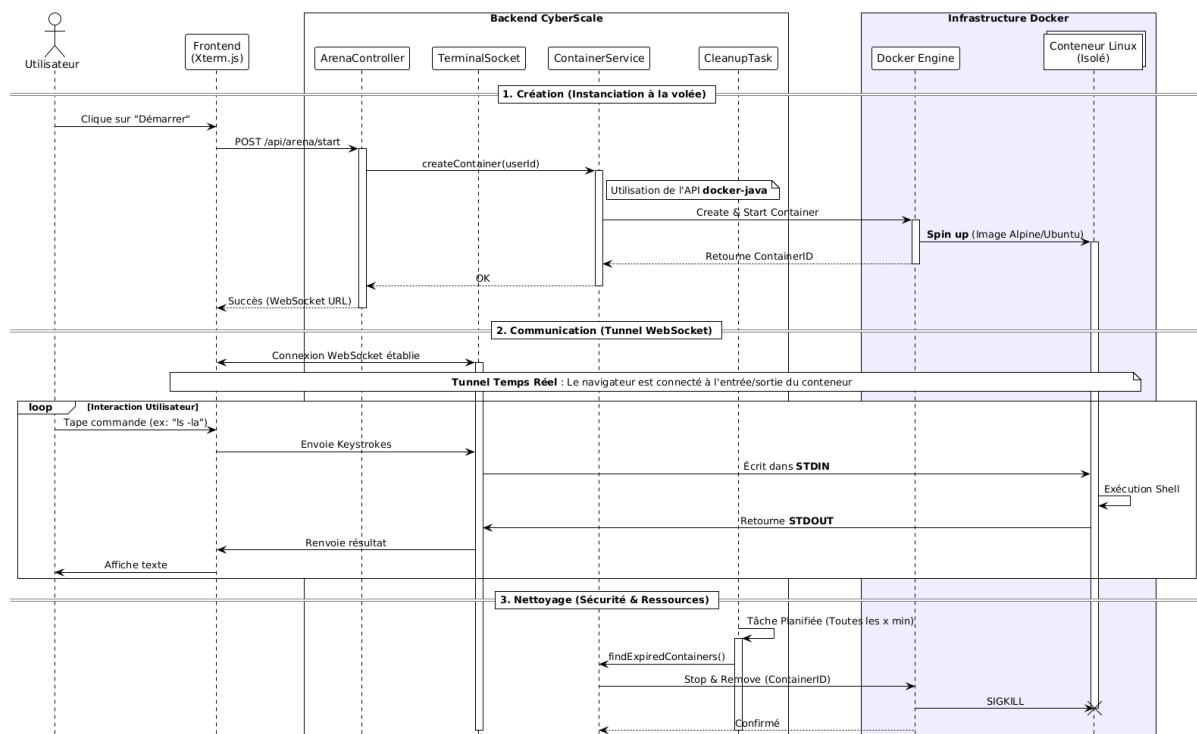


Figure 3. Fonctionnement de l'architecture dockerisé

3. Fonctionnalités (Features v1.0.0)

Cette version finale intègre l'ensemble des modules prévus.

3.1. F1: Onboarding & Quiz Adaptatif □

Objectif

Évaluer le bagage théorique initial.

Innovation

L'algorithme adapte la difficulté des questions (Facile → Moyen → Difficile) en fonction des réponses précédentes pour affiner le score.

Question 5 / 20

Quelle est la principale différence entre la cryptographie symétrique et la cryptographie asymétrique, et dans quel contexte l'une est-elle préférée à l'autre dans un environnement DevOps ?

- La cryptographie symétrique utilise une clé publique/privée, tandis que l'asymétrique n'utilise qu'une seule clé secrète.
- La cryptographie symétrique est plus rapide et est utilisée pour chiffrer les données de masse (ex: trafic réseau).
- La cryptographie asymétrique est principalement utilisée pour le hashing des mots de passe.
- Elles sont interchangeables et n'ont pas d'impact réel sur la performance des systèmes modernes.

Valider et Suivant

Figure 4. Wireframe Quiz

3.2. F2: Cyber Arena (Red Team) □

Objectif

Tester les compétences offensives (Linux, Pentest).

Scénario

L'utilisateur accède à un vrai terminal Linux. Il doit naviguer, décoder des fichiers et exploiter des failles pour trouver des "flags" ([CTF{...}](#)).

Validation

Le flag est validé par un script interne au conteneur qui notifie le backend en temps réel.

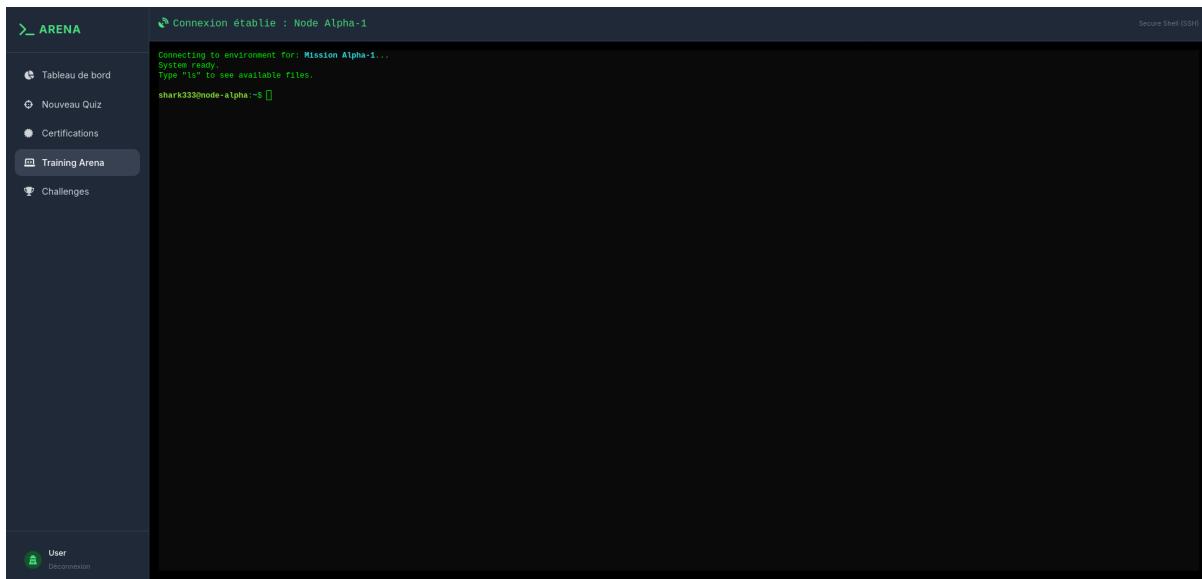


Figure 5. Interface du Terminal Web

3.3. F3: Module Phishing (Facteur Humain) ☐

Objectif

Tester la vigilance face à l'ingénierie sociale.

Fonctionnement

Simulation d'une boîte mail avec des courriels piégés (Urgence RH, Faux support). L'utilisateur doit analyser les en-têtes et décider de signaler ou cliquer.

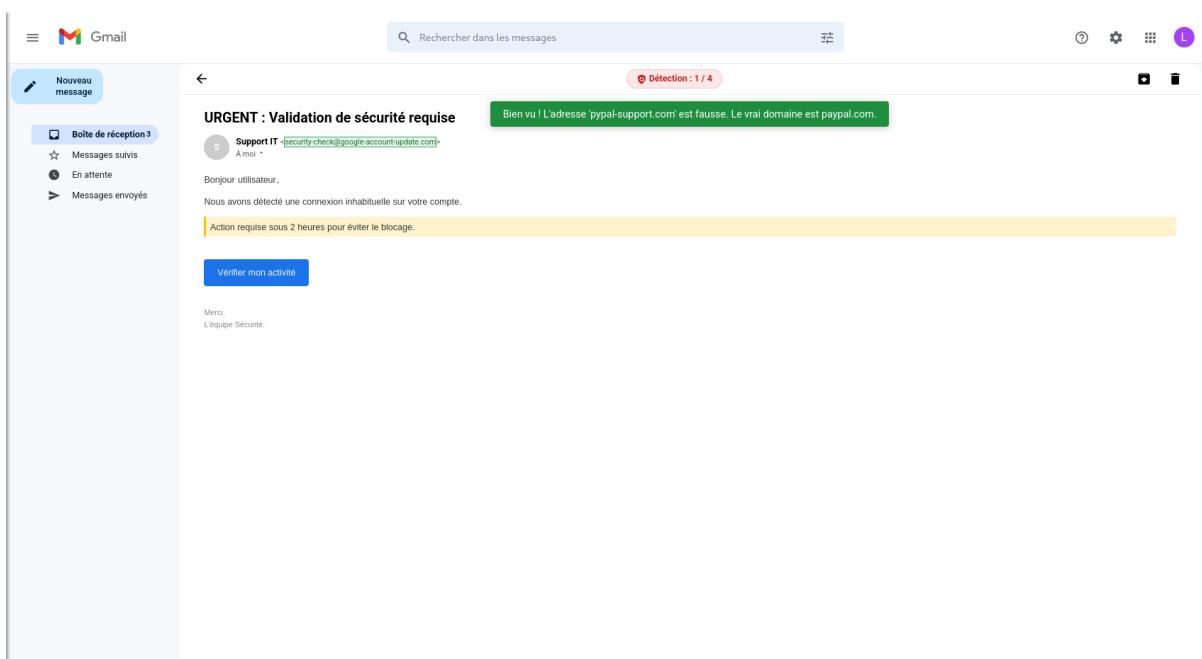


Figure 6. Interface de Phishing

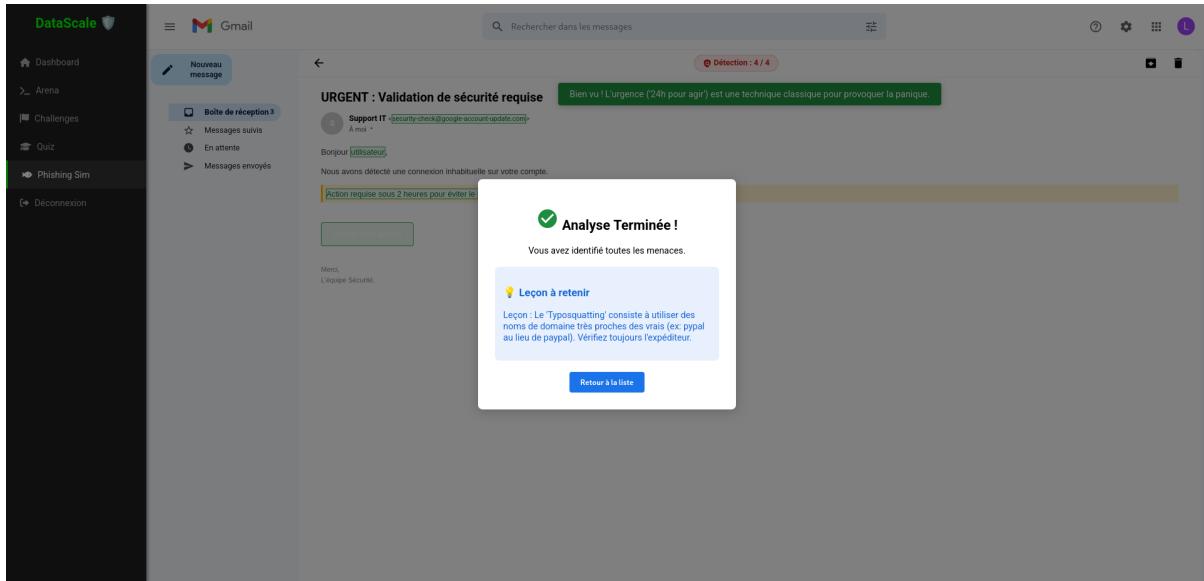


Figure 7. Phishing Réussi

3.4. F4: Module Investigation (Blue Team) ☠

Objectif

Former à la défense et à l'analyse de logs (Forensics).

Technique

Un générateur de logs ([LogGenerator](#)) crée dynamiquement des milliers de lignes de trafic web réaliste, en y cachant une attaque unique (ex: Injection SQL). L'utilisateur doit filtrer et identifier l'IP malveillante.

| TIMESTAMP | IP | STATUS | SIZE |
|----------------------------|----------------|--------|-------------------|
| 17/Jan/2026:17:32:41 +0000 | 95.197.40.209 | POST | /dashboard |
| 17/Jan/2026:17:32:43 +0000 | 52.156.193.101 | POST | /about |
| 17/Jan/2026:17:33:07 +0000 | 37.192.249.194 | GET | /contact |
| 17/Jan/2026:17:33:10 +0000 | 75.47.128.71 | GET | /assets/style.css |
| 17/Jan/2026:17:33:14 +0000 | 98.155.81.198 | DELETE | /assets/style.css |
| 17/Jan/2026:17:33:14 +0000 | 98.155.81.198 | DELETE | /assets/style.css |

Figure 8. Accueil dans le menu d'Investigation

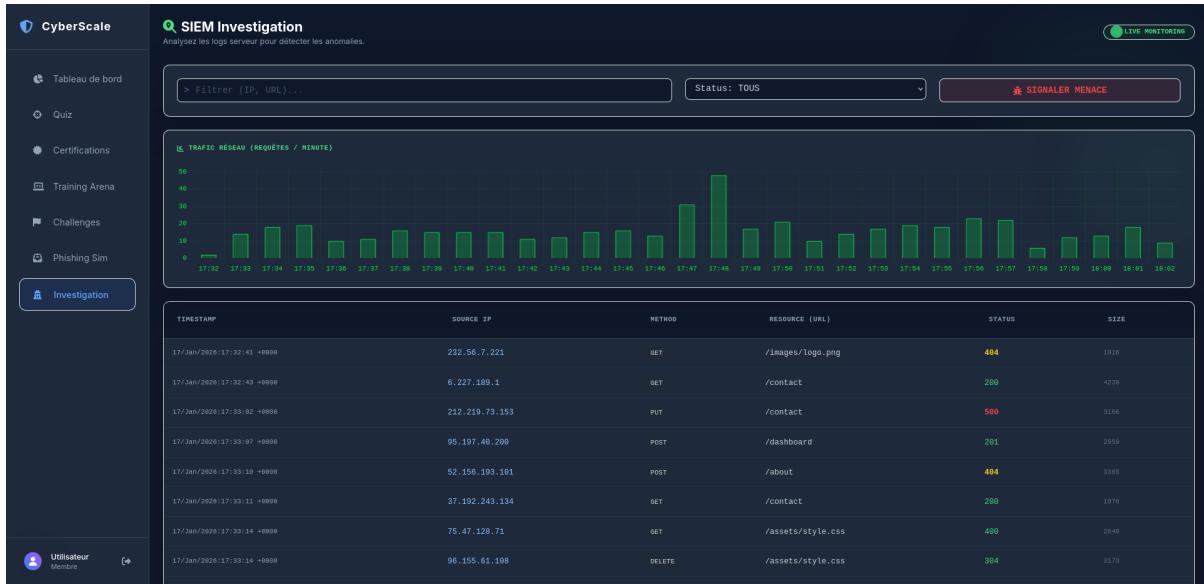


Figure 9. Affichage des Logs

3.5. F5: Dashboard Unifié & Certification ☐

Objectif

Centraliser la progression.

Évolution v1.0

Le Dashboard agrège désormais les scores de tous les modules (Quiz + Arena + Logs + Phishing) pour calculer une moyenne globale.

Recommandations

Si le score est faible, le système suggère des ressources externes ciblées. Si le score est élevé (>80%), il débloque des certifications virtuelles.

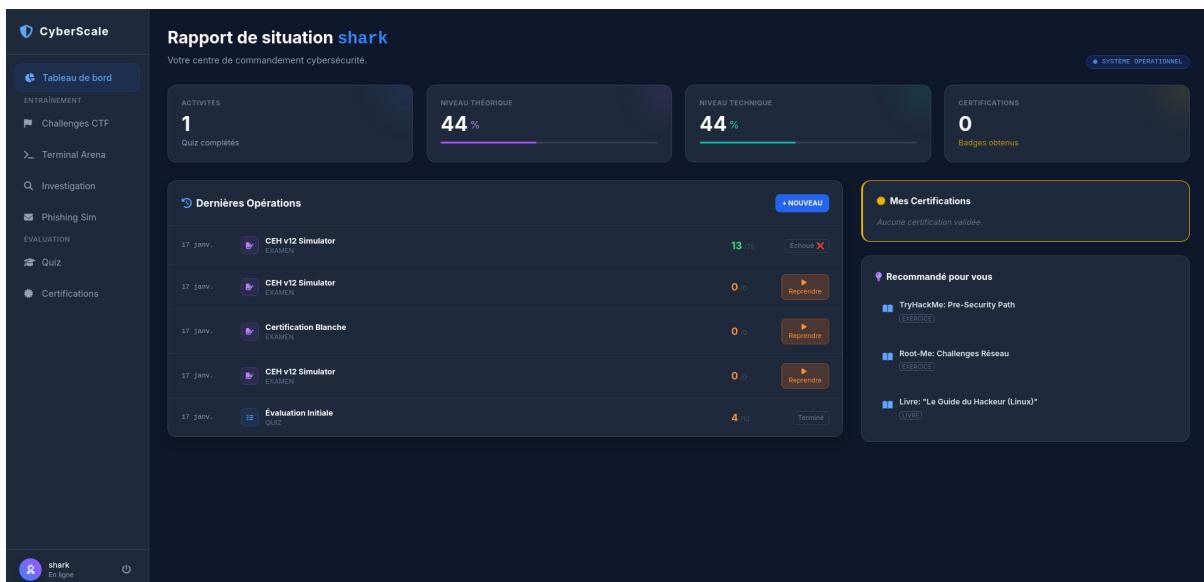


Figure 10. Tableau de Bord Global

4. Gestion de Projet & Qualité ☰

4.1. Sécurité & Authentification

L'accès est strictement contrôlé pour protéger la progression des utilisateurs.

- **Inscription & Connexion :** Formulaires sécurisés avec gestion des erreurs (401/403).
- **Sécurisation des mots de passe :** Hachage robuste via Spring Security (BCrypt) avant stockage en base.
- **Protection des Routes (AuthGuard) :** Mécanisme Frontend empêchant l'accès direct aux pages sensibles (Dashboard, Arena) sans token de session valide.
- **Persistiance :** Utilisation du `localStorage` pour une expérience utilisateur fluide.

4.2. Qualité du Code (CI/CD)

Nous avons industrialisé notre processus de développement pour la v1.0.

- **Pipeline Automatisé :** Chaque modification du code déclenche une compilation et des tests via GitHub Actions.
- **Tests Unitaires (JUnit) :** Couverture de la logique métier critique (Calcul de score, Génération de logs).
- **Tests E2E (Selenium) :** Validation automatique des parcours utilisateurs complets (Login → Arena → Fin).
- **Analyse Statique :** Zéro "Security Hotspot" critique sur SonarCloud.

5. Conclusion & Perspectives ☰

La version v1.0.0 de CyberScale est une plateforme complète et innovante pour l'évaluation des compétences en cybersécurité. En combinant théorie, pratique, et une architecture technique avancée, nous offrons une expérience utilisateur unique.