

Rapport Projet

PLATEFORME DE GESTION DU DATA CENTER

Application Web de Réservation et de Suivi des
Ressources Informatiques



Réalise Par :

AYADI Yassine

CHRIAA Zakariae

JABIR Oussama

ABAKAR MOUSSA Hamit

Encadré par :

- M. M'hamed AIT KBIR
- M. Yasyn EL YUSUFI

1. Introduction générale :

Ce projet consiste à développer une plateforme web dédiée à la gestion, à l'allocation et au suivi des ressources d'un **Data Center**.

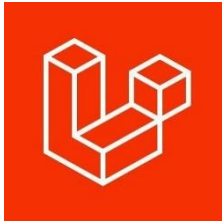
L'application permet de centraliser tout le parc informatique pour offrir une visibilité en temps réel sur la disponibilité du matériel.

Elle gère le cycle de vie complet d'une ressource : depuis sa **consultation** dans le catalogue et sa **réservation** en ligne par les utilisateurs, jusqu'à la **validation** par les responsables et le **suivi technique**

L'objectif est d'assurer une utilisation efficace du matériel tout en évitant les conflits de réservation grâce à un système automatisé et sécurisé



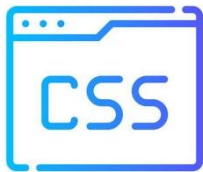
2. Les Outils utilisé :



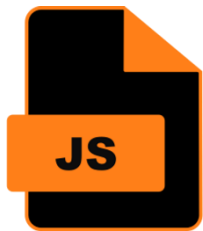
Laravel : Est un Framework web open-source écrit en PHP respectant le principe modèle-vue-contrôleur (MVC). Il est utilisé pour le développement d'applications web robustes et sécurisées.



MySQL : Est un système de gestion de bases de données relationnelles (SGBDR). Il est utilisé pour stocker et structurer les données de l'application.



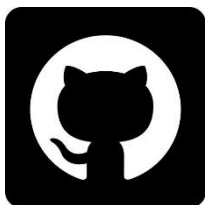
CSS : Langage de feuilles de style utilisé pour la mise en forme et le design de l'interface utilisateur



JavaScript : Langage de programmation de scripts utilisé pour rendre les pages web interactives et dynamiques



VS code : Est un éditeur de code extensible développé par Microsoft. C'est l'environnement de développement principal utilisé pour écrire et déboguer le code.

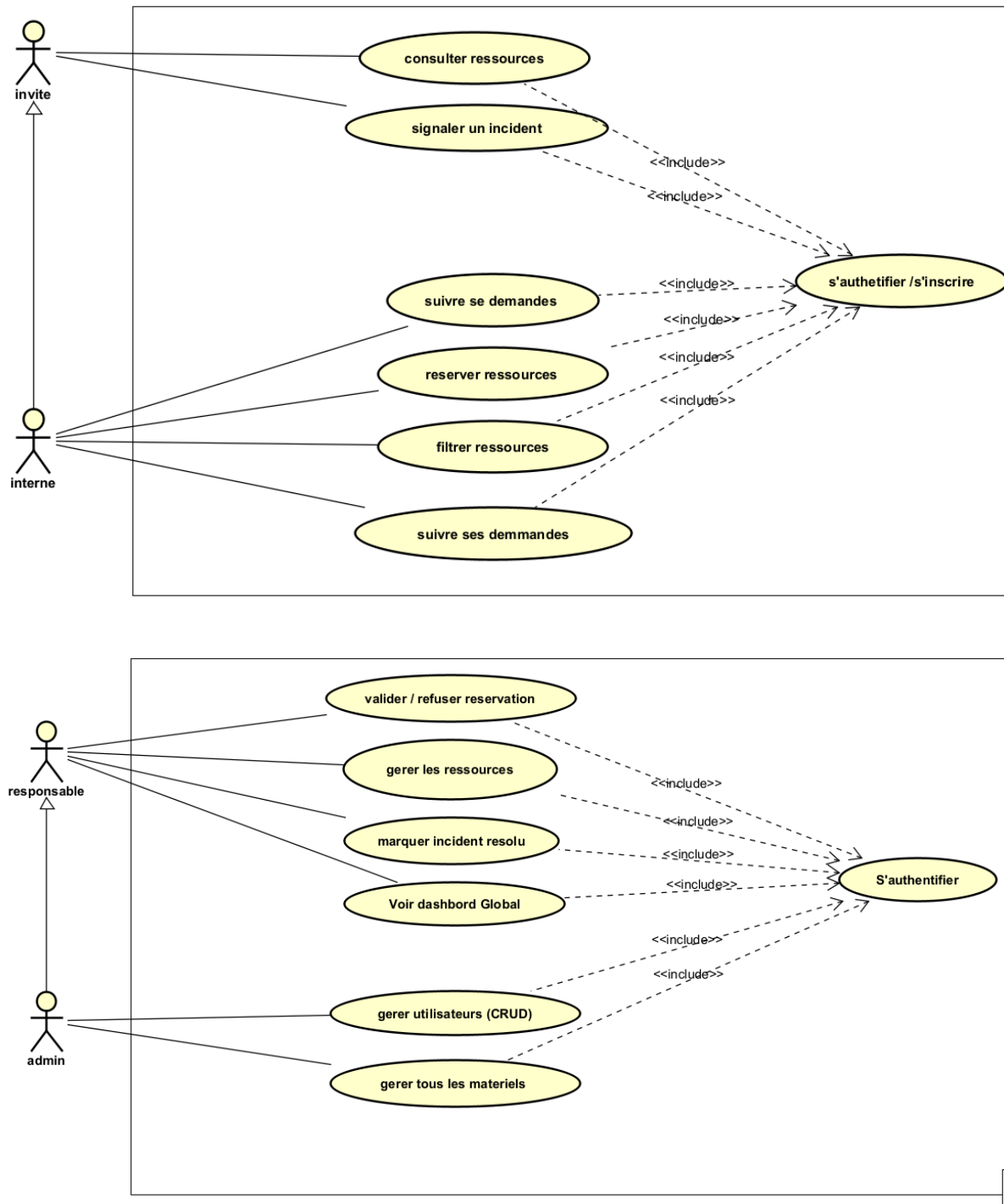


GitHub : Est une plateforme d'hébergement de code pour le contrôle de version avec Git

3. Les acteurs et cas d'utilisations :

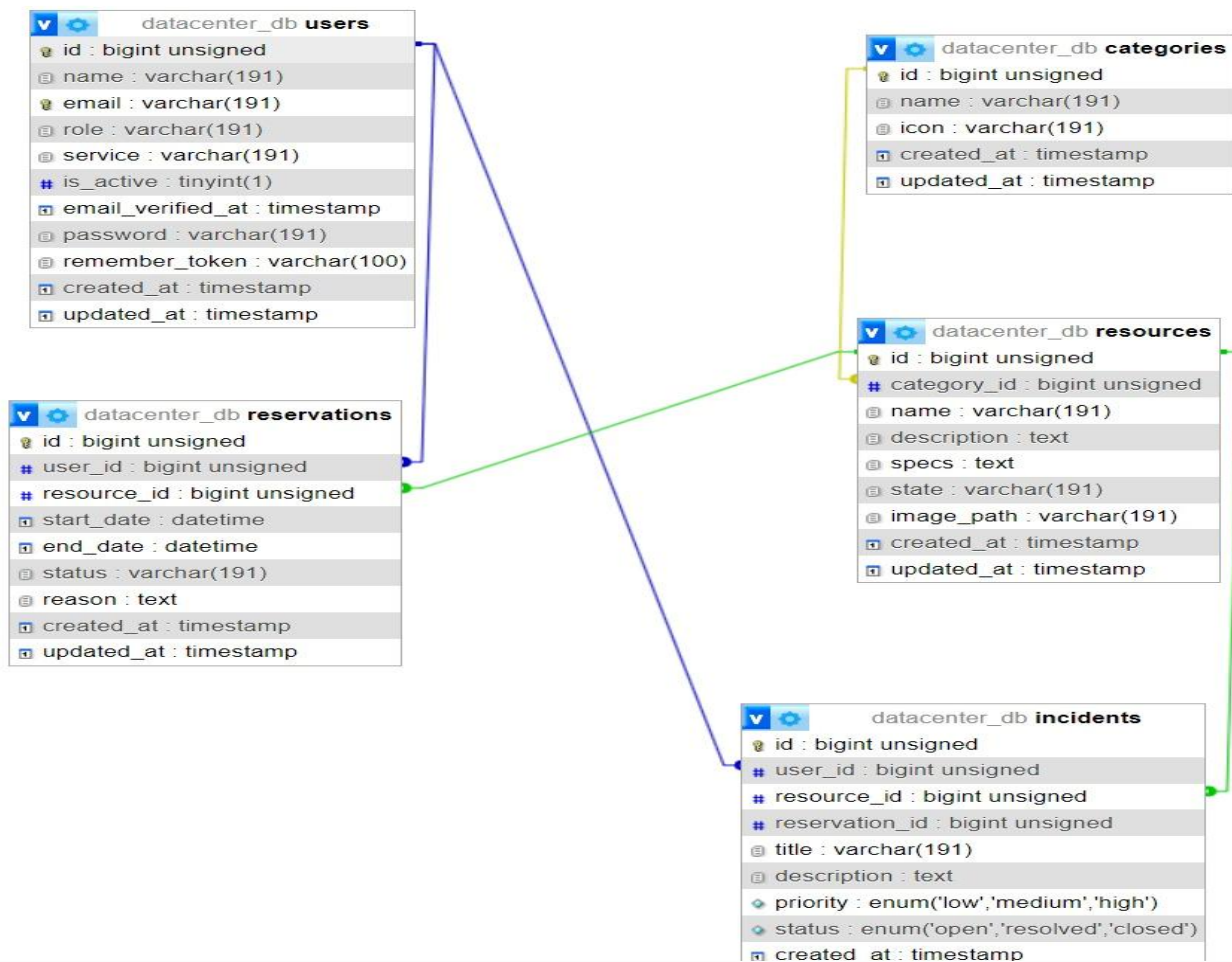
Pour bien comprendre comment les différents profils interagissent avec cette plateforme, **voici le diagramme des cas d'utilisation (Use Case)**.

Ce schéma montre les droits spécifiques de chaque utilisateur, de l'invité à l'administrateur



4. Dictionnaire de Données :

- **Users** : Stocke les informations des utilisateurs (invités, personnels, admins). Le champ **role** définit les permissions.
- **Categories** : Permet de classer le matériel (Serveurs, PC, Switchs...).
- **Resources** : Contient l'inventaire matériel. Chaque ressource est liée à une catégorie.
- **Reservations** : Table pivot enregistrant les demandes. Elle relie un utilisateur à une ressource sur une période donnée.
- **Incidents** : Enregistre les pannes signalées. Elle est liée à un utilisateur (signaleur) et potentiellement à une ressource spécifique.



5. Réalisation Technique et Implémentation :

Cette section détaille comment les fonctionnalités principales ont été développées, en mettant l'accent sur la logique métier et l'architecture MVC de Laravel.

1. Architecture et Structure :

L'application respecte l'architecture MVC (Modèle-Vue-Contrôleur) imposée par Laravel:

Modèles (App\Model) : Représentent les tables de la base de données (User, Resource, Réservation, Incident) et leurs relations

Vues (resources/views) : Interfaces utilisateurs développées avec le moteur de template Blade, intégrant un design Responsive CSS personnalisé (**style.css**).

Contrôleurs (App\Http\Controllers) : Contiennent la logique métier.

2. Gestion des Rôles et Sécurité (Auth) :

Nous avons implémenté un système d'authentification avec gestion des rôles pour sécuriser l'accès.

- Chaque utilisateur possède un attribut **role** dans la base de données (admin, responsable, user).
- Un Middleware personnalisé (CheckRole) intercepte les requêtes HTTP pour vérifier si l'utilisateur a le droit d'accéder à une route spécifique par exemple : seul un admin peut supprimer une ressource

3. Gestion des Ressources (Matériel) :

Le module de gestion des ressources permet aux administrateurs de maintenir l'inventaire à jour.

- CRUD: Implémenté dans **ResourceController** (index, create, store, edit, update, destroy).
- Filtrage Avancé : Nous avons ajouté une fonctionnalité de recherche dynamique permettant de filtrer la liste par **Catégorie** (Serveurs, ordinateurs portables, switchs reseau) et par État(active/Hors Service). Cela se fait via des requêtes conditionnelles (query->where(...)).

4. Système de Réservation Intelligent :

Le cœur de l'application est le système de réservation qui empêche physiquement les conflits.

- Gestion des conflits de disponibilité : Avant de valider une réservation, le système vérifie s'il existe déjà une réservation active pour la même ressource sur le créneau choisi.
- Algorithme : Une requête vérifie si $(DateDébut1 < DateFin2)$ ET $(DateFin1 > DateDébut2)$. Si c'est le cas, la réservation est bloquée et un message d'erreur est retourné.

5. Gestion des Incidents :

Pour assurer la maintenance, un module d'incidents a été développé IncidentController.

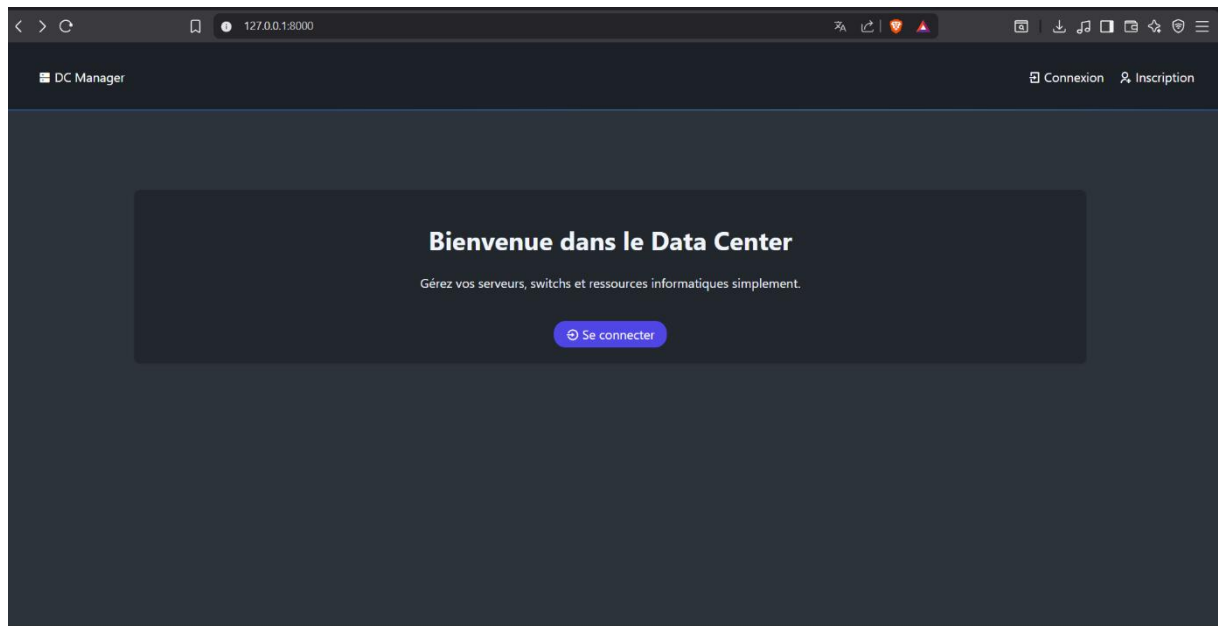
- Les utilisateurs peuvent signaler une panne via un formulaire dédié
- Les administrateurs reçoivent une alerte et voient l'incident dans leur tableau de bord.
- Une fois réparé, l'admin change le statut de 'Open' à 'Resolved'.

Présentation de l'Application :

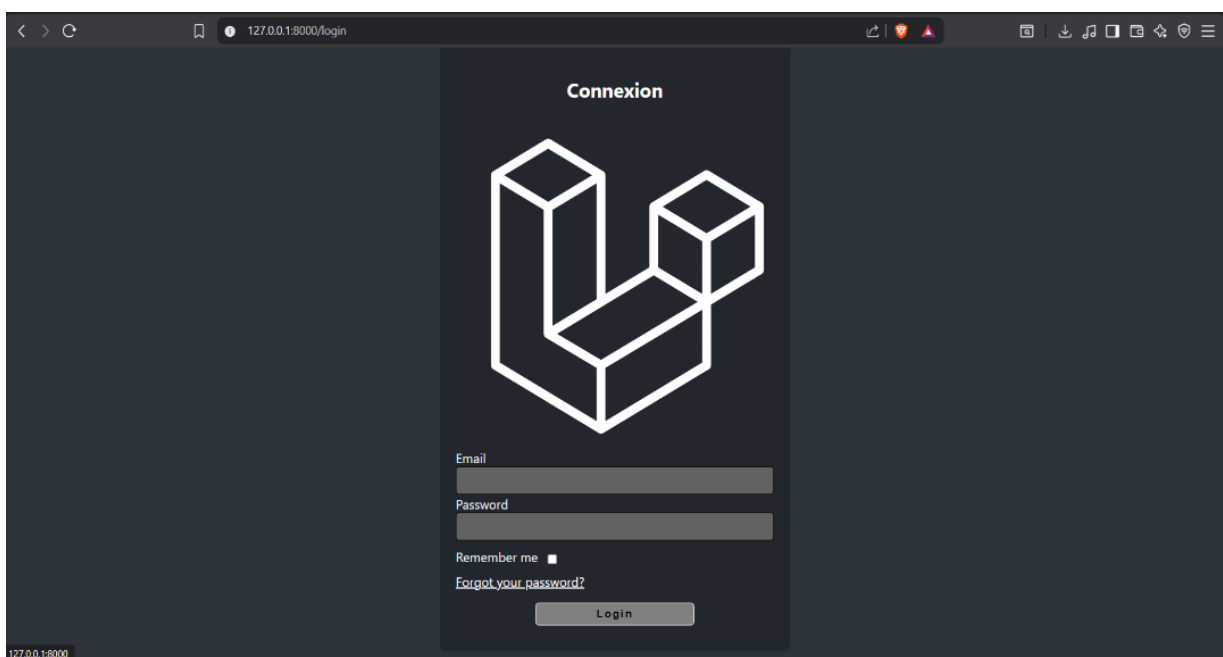


Authentification :

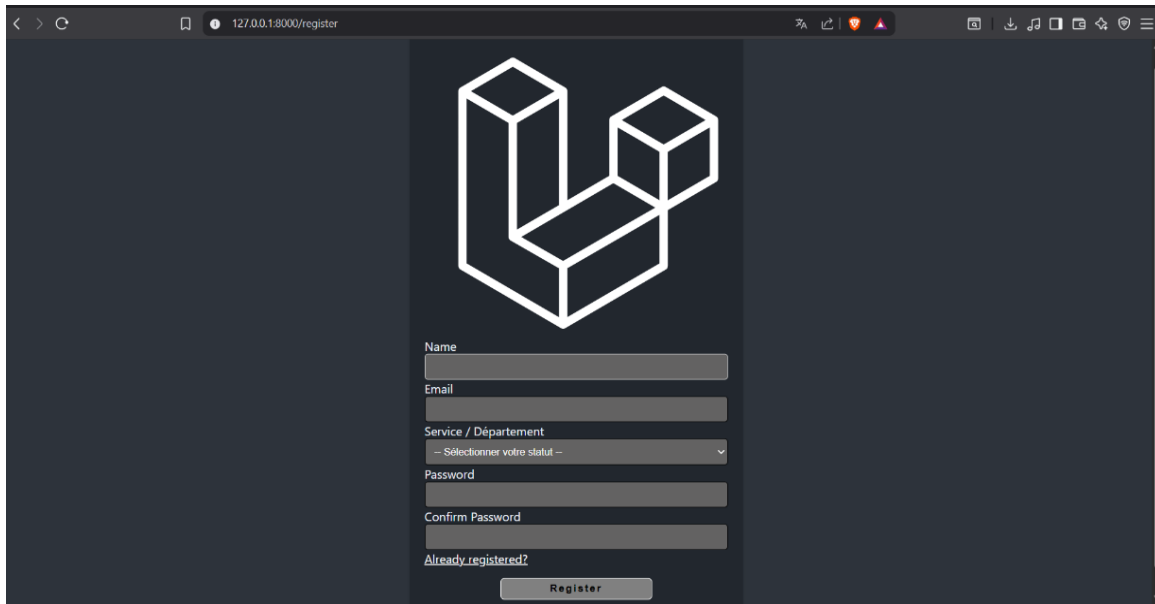
- **Welcome** : Page d'accueil publique qui présente l'application Data Center Manager avec des liens de connexion et d'inscription.



- **Login** : Formulaire de connexion pour authentifier les utilisateurs du Data Center.



- **Register** : Création de compte utilisateur.



127.0.0.1:8000/register

Name

Email

Service / Département
– Sélectionner votre statut –

Password

Confirm Password

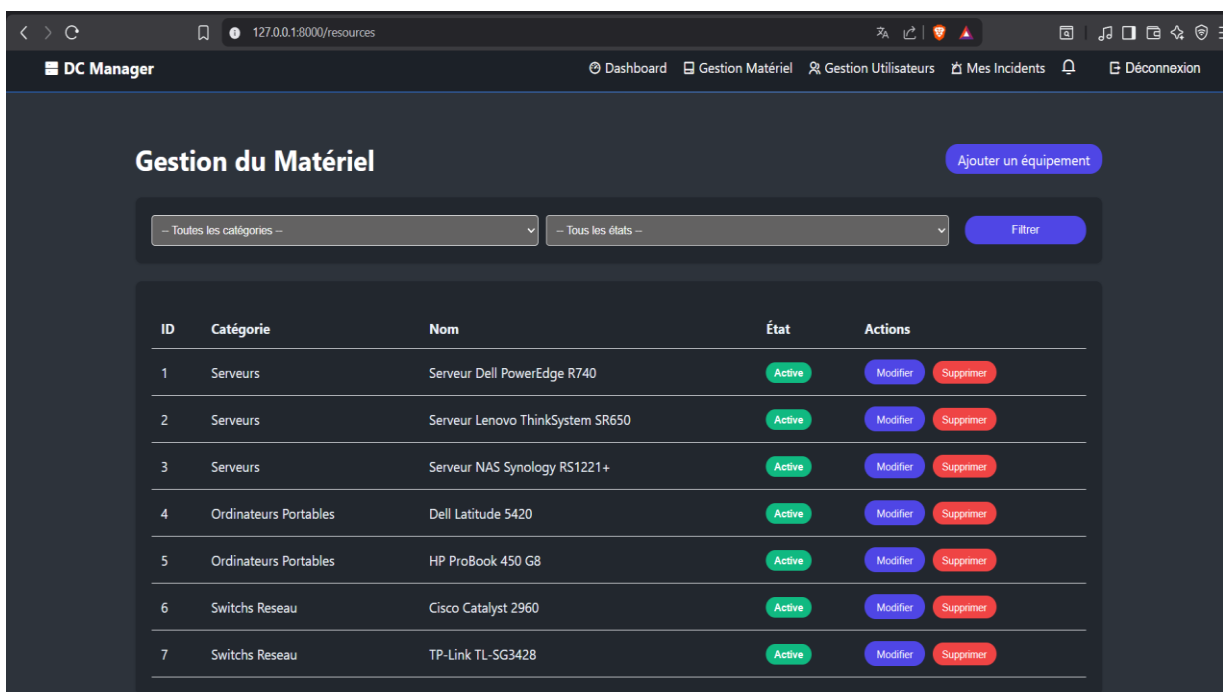
[Already registered?](#)

Register



Gestion des Ressources :

- **Ressources** : Gestion du matériel avec les nouveaux filtres (Catégorie & État) activés.



DC Manager

Dashboard | Gestion Matériel | Gestion Utilisateurs | Mes Incidents | Déconnexion

Gestion du Matériel

Ajouter un équipement

– Toutes les catégories – | – Tous les états – | Filtrer

ID	Catégorie	Nom	État	Actions
1	Serveurs	Serveur Dell PowerEdge R740	Active	Modifier Supprimer
2	Serveurs	Serveur Lenovo ThinkSystem SR650	Active	Modifier Supprimer
3	Serveurs	Serveur NAS Synology RS1221+	Active	Modifier Supprimer
4	Ordinateurs Portables	Dell Latitude 5420	Active	Modifier Supprimer
5	Ordinateurs Portables	HP ProBook 450 G8	Active	Modifier Supprimer
6	Switchs Réseau	Cisco Catalyst 2960	Active	Modifier Supprimer
7	Switchs Réseau	TP-Link TL-SG3428	Active	Modifier Supprimer

- **Ressources/create** : Formulaire d'ajout d'un nouvel équipement (nom, catégorie, description, spécifications)

Ajouter un équipement

The screenshot shows the 'Ajouter un nouvel équipement' (Add new equipment) form in the DC Manager application. The form is located at the URL 127.0.0.1:8000/resources/create. It contains the following fields:

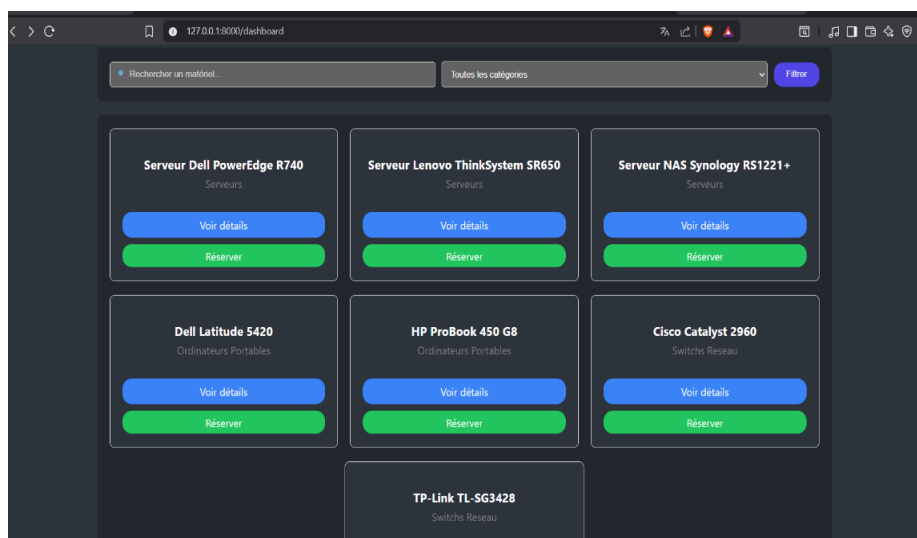
- Nom de l'équipement :** A text input field with the example 'Ex: Serveur Dell PowerEdge'.
- Catégorie :** A dropdown menu with 'Serveurs' selected.
- Description :** A text area with the placeholder 'Description générale...'.
- Caractéristiques Techniques (Specs) :** A text area with the example 'Ex: CPU i7, 32GB RAM...'.

At the bottom of the form, there are two buttons: 'Enregistrer le matériel' (Save equipment) and 'Annuler' (Cancel).



Reservation :

- **Dashboard (utilisateur) :** Interface du tableau de bord affichant le catalogue des équipements informatiques disponibles (serveurs, ordinateurs, switch) avec des fonctionnalités de recherche et de réservation



- **/reserve** : Formulaire de demande de réservation permettant à l'utilisateur de définir la période d'emprunt et de justifier sa demande pour le matériel sélectionné.

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/reserve/1'. The page title is 'DC Manager'. The main heading is 'Nouvelle Réservation'. Below this, there is a form titled 'Réserver : Serveur Dell PowerEdge R740'. The form includes a 'Catégorie : Serveurs' label, a 'Date de début :' field with a date picker showing 'mm/dd/yyyy --:-- --', a 'Date de fin :' field with a date picker showing 'mm/dd/yyyy --:-- --', and a 'Motif de la demande :' text area with the placeholder text 'Pour quel projet ? Pourquoi ce matériel ?'. At the bottom of the form are two buttons: 'Envoyer la demande' (blue) and 'Annuler' (red).

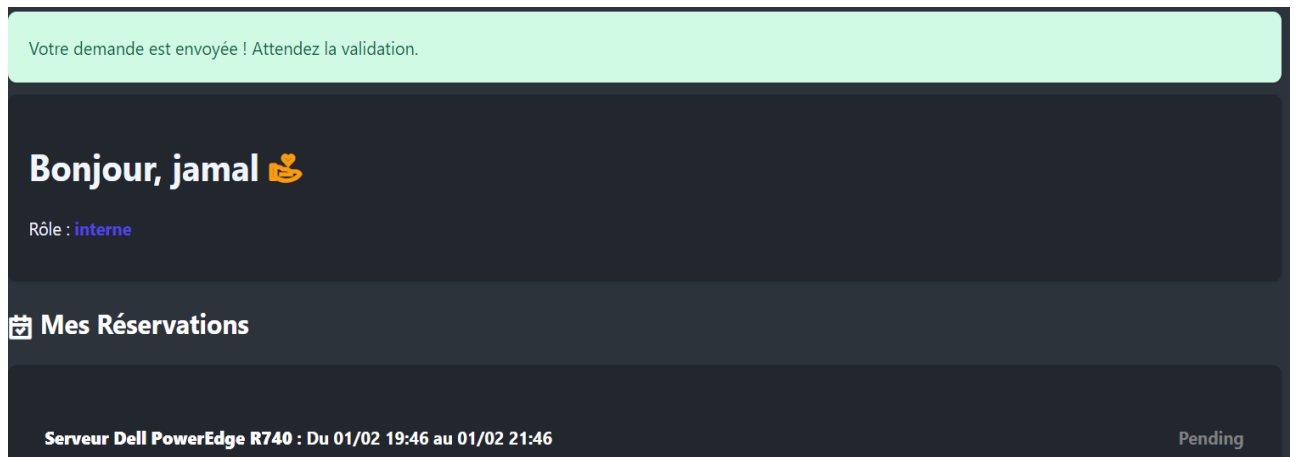
a) Gestion des conflits de disponibilité :

Le système détecte automatiquement un chevauchement de dates et affiche une alerte rouge bloquante, empêchant la validation si la ressource est déjà réservée sur la période sélectionnée.

This screenshot shows the same reservation form as above, but with a red alert message displayed at the top: 'Impossible ! Cette ressource est déjà réservée du 29/01 08:43 au 29/01 10:48.' The form fields and buttons remain the same.

b) Confirmation et suivi de la demande :

Après soumission valide, une notification verte confirme l'envoi de la requête et la réservation apparaît dans le tableau de bord de l'utilisateur avec le statut "Pending" (En attente de validation)



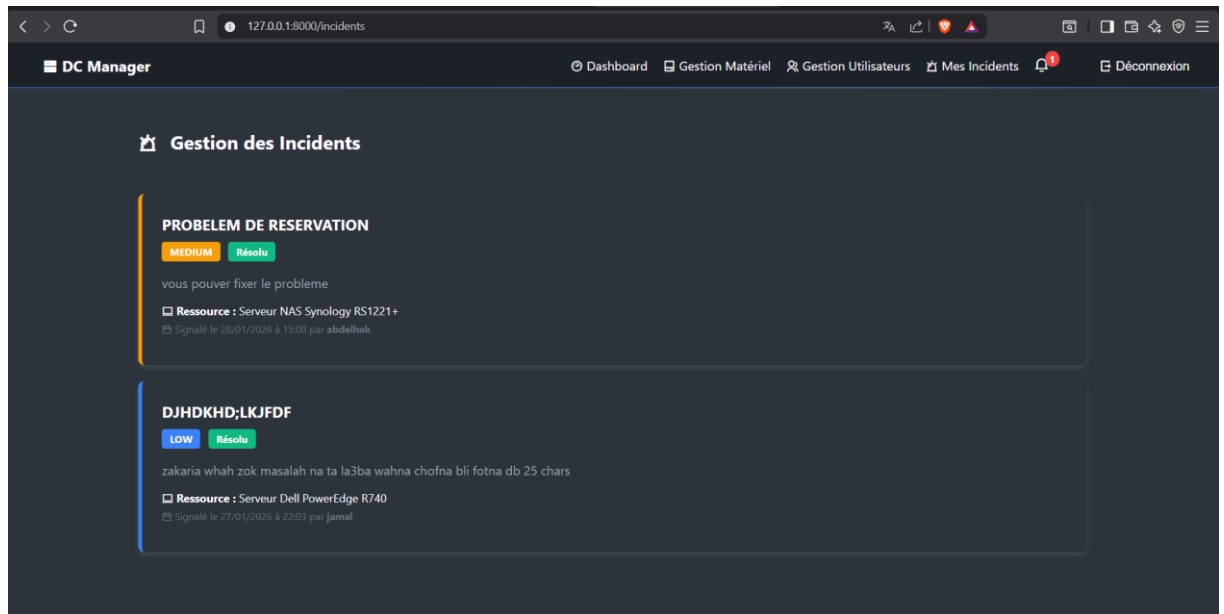
Incidents :

- **/incidents/create (utilisateur)** : Interface de création de ticket d'incident où l'utilisateur doit renseigner le titre et la priorité du dysfonctionnement, avec la possibilité de lier l'alerte à une réservation ou un équipement matériel précis avant de soumettre une description détaillée.

A screenshot of a web browser showing the "Signaler un Incident" form. The form includes fields for "Titre du problème" (with an example "Ex: Serveur inaccessible, Panne réseau..."), "Priorité" (a dropdown menu), "Ressource concernée (optionnel)" (a dropdown menu), and "Réservation concernée (optionnel)" (a dropdown menu). There is a large text area for "Description détaillée" with a note to provide as much detail as possible. At the bottom, there are two buttons: "Signaler l'incident" (blue) and "Annuler" (red).

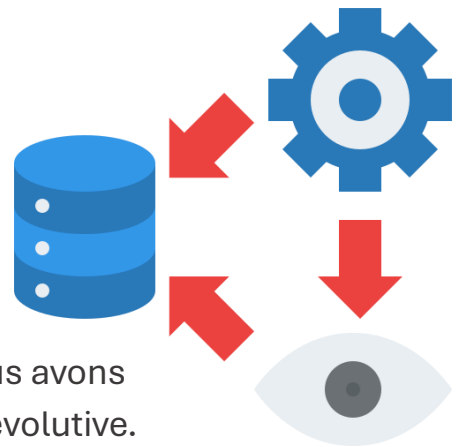
- **/incidents (administrateur)** : Cette vue permet à l'administrateur de surveiller les problèmes signalés en temps réel. Les nouvelles alertes sont signalées par une notification visuelle (badge rouge sur la

cloche), invitant l'administrateur à consulter la liste, à prendre connaissance des détails (ressource impactée, priorité) et à traiter le ticket.



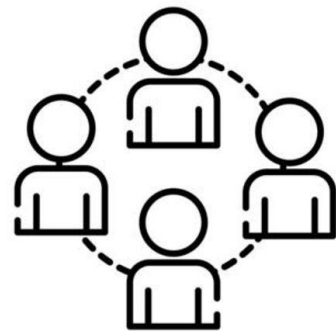
Conclusion :

Ce projet de réalisation d'une **Plateforme de Gestion du Data Center** nous a permis de répondre efficacement aux problématiques de centralisation et de suivi des ressources informatiques. Grâce à l'utilisation du framework **Laravel** et d'une architecture **MVC**, nous avons pu développer une solution robuste, sécurisée et évolutive.



L'application garantit désormais une gestion fluide des réservations, éliminant les conflits de disponibilité grâce à notre algorithme de vérification, tout en assurant un suivi réactif des incidents techniques.

D'un point de vue pédagogique, ce projet a été l'occasion de renforcer nos compétences en développement **Full Stack**, en gestion de base de données relationnelles et en travail collaboratif via **GitHub**.



Pour l'avenir, plusieurs améliorations sont envisageables, notamment l'intégration de notifications par email en temps réel, un tableau de bord statistique pour analyser l'usure du matériel, ou encore le développement d'une application mobile pour faciliter les signalements d'incidents sur le terrain.