



# Rapport de projet :

## Application Web de Réservation des Ressources d'un Data Center

Réalisé par :

Ararou israe

Tigharmte hajar

Marzouq saad

Zarkik mohamed

## I-Introduction générale :

- L'application développée dans ce projet a pour but de fournir une plateforme Web permettant de gérer les ressources informatiques d'un Data Center. Elle doit répondre aux besoins des différents types d'utilisateurs, tout en assurant une gestion efficace et transparente des ressources telles que les serveurs, les machines virtuelles, le stockage et les équipements réseau.
- La plateforme est conçue pour accueillir un grand nombre d'utilisateurs, chacun avec des droits et des fonctionnalités spécifiques, définis selon son rôle. Parmi ces utilisateurs figurent l'invité, l'utilisateur interne, le responsable technique et l'administrateur, chacun pouvant accéder aux fonctionnalités adaptées à ses besoins et responsabilités.
- La plateforme doit permettre aux utilisateurs de consulter les ressources disponibles, de soumettre des demandes de réservation en temps réel et de suivre l'état de leurs demandes (en attente, approuvées, refusées, actives ou clôturées). Elle doit également gérer les périodes de maintenance et permettre la déclaration et le traitement des incidents techniques.
- Des fonctionnalités supplémentaires, telles que la sécurité, la traçabilité, les notifications et les statistiques, sont prévues pour garantir le bon fonctionnement de la plateforme et offrir un meilleur contrôle sur l'infrastructure

## I-Introduction générale :

### a-Objectif :

L'objectif de ce projet est de concevoir et de développer une application Web permettant la gestion des ressources informatiques d'un Data Center. Cette application vise à faciliter la réservation, l'allocation et le suivi des ressources (serveurs, machines virtuelles, stockage, équipements réseau), tout en assurant une organisation claire, efficace et transparente pour les différents utilisateurs.

### b-Language et paradigme utilisé :

Le projet est développé en PHP à l'aide du framework Laravel. Il repose sur le paradigme de la programmation orientée objet afin d'assurer une meilleure organisation du code, une séparation claire des responsabilités et une meilleure maintenabilité de l'application. L'utilisation des models Eloquent permet également une gestion efficace des données à travers des modèles bien structurés.

### c-Visualisation et outille :

Le développement s'appuie sur une base de données MySQL pour le stockage des informations. Des technologies Web telles que HTML, CSS et JavaScript , tandis que Laravel assure la gestion des routes, de la sécurité, de l'authentification et de la logique métier de l'application.

## 2-conception et développement

### I structure de projet :

Le projet suit l'architecture MVC (Modèle-Vue-Contrôleur) imposée par Laravel, ce qui permet une séparation claire entre la logique métier, la base de données et l'interface utilisateur. Pour une meilleure organisation, nous avons structuré l'application autour de plusieurs interfaces distinctes correspondant aux différents profils et fonctionnalités des utilisateurs :

**Page d'accueil** : point d'entrée accessible à tous les utilisateurs et invités.

**Page d'inscription** : pour permettre aux nouveaux utilisateurs de créer un compte.

**Page de login** : pour l'authentification des utilisateurs.

**Interface Utilisateur interne** : espace personnel pour les ingénieurs, enseignants ou doctorants, permettant de visualiser les ressources disponibles, effectuer et suivre les réservations, consulter l'historique et signaler des problèmes.

**Interface Responsable** : pour le responsable technique, afin de gérer les ressources sous sa supervision, approuver ou refuser les demandes, et modérer les incidents.

**Les Interfaces Administrateur** : espace complet pour gérer les utilisateurs, rôles, permissions, ressources, catégories et consulter les statistiques globales du Data Center.

Au niveau technique, le projet suit la structure standard Laravel :

**app/** : contient les modèles Eloquent (Utilisateur, Catégorie, Ressource, Réservation, SigProb) et la logique métier.

**database/migrations/** : contient les migrations pour créer les tables nécessaires à la base de données, par exemple create\_utilisateur\_table.

**database/seeders/** : contient les seeders pour peupler les tables avec des données de test, comme UtilisateurSeeder.

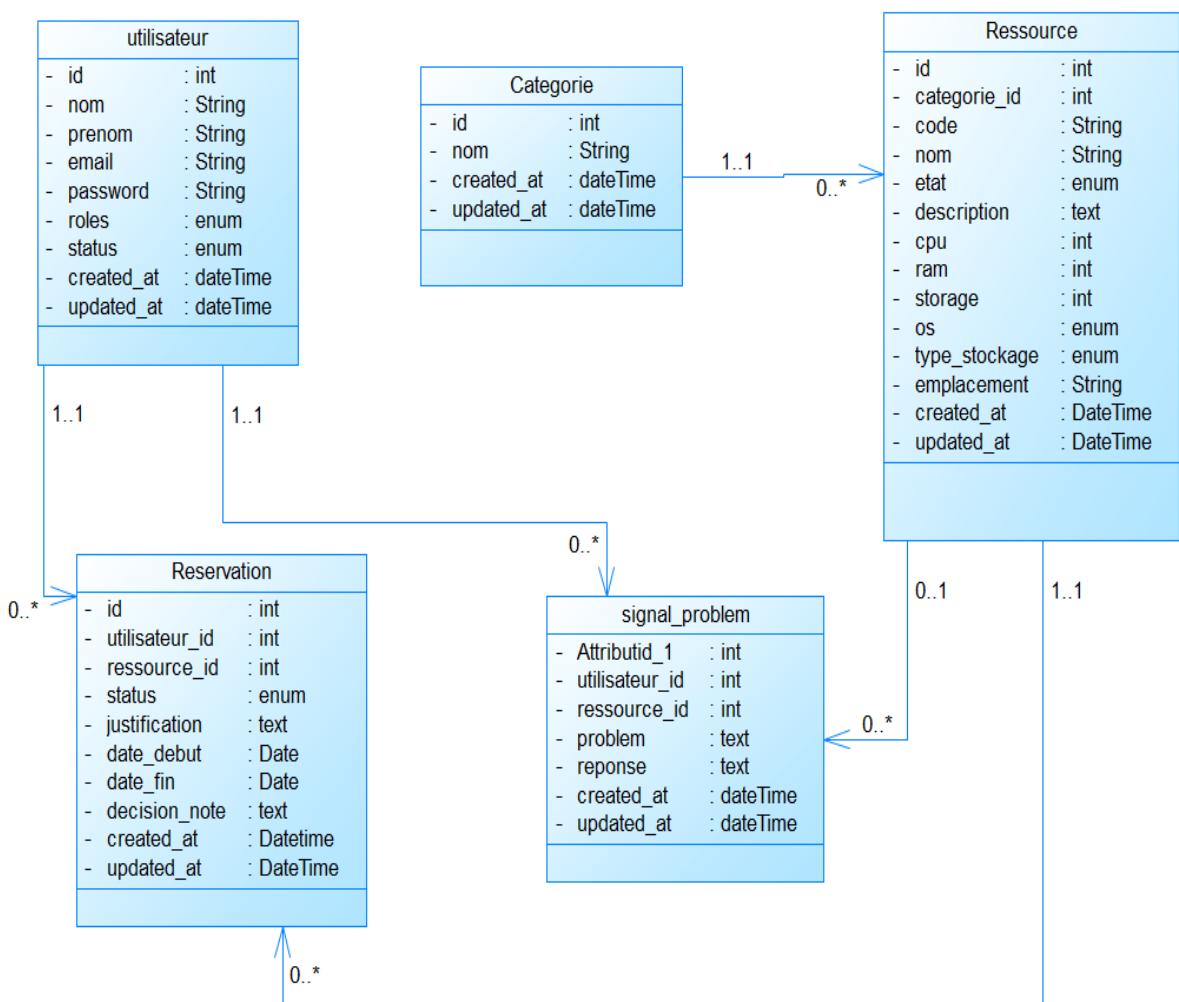
**routes/web.php** : gère toutes les routes accessibles depuis le navigateur.

**resources/views/** : contient les vues correspondant aux pages d'accueil, login, inscription, interface interne, responsable et administrateur.

**public/** : (contient le css et java script des vues. images).

## 2-choix des tables de bases de données

Pour réaliser ce projet, on a d'abord commencé par penser à la structure de la base de données pour bien organiser toutes les informations sur la gestion des ressources du Data Center. On a identifié les entités principales : Utilisateur, Catégorie, Ressource, Réservation et SignalProblem. Pour chacune, on a choisi les champs importants et leurs types, tout en respectant les relations entre les tables (par exemple, chaque ressource appartient à une catégorie et chaque réservation est liée à un utilisateur et une ressource). Dans ce sens on a élaboré ce diagramme de classe .



### Migrations :

Ensuite, on a créé les migrations Laravel correspondantes :

create\_utilisateur\_table,

create\_categorie\_table,

create\_ressource\_table,

create\_reservation\_table et create\_sig\_prob\_table pour générer automatiquement les tables dans la base de données à travers la commande `php artisan make :migration create_utilisateur_table`.

### seeders :

Pour faciliter les tests, on a mis en place des seeders comme UtilisateurSeeder, CategorieSeeder, RessourceSeeder et ReservationSeeder en utilisant la commande `php artisan make :seeder utilisateurSeeder` afin de remplir les tables avec des données d'exemple .

### Models :

Après ça, on a développé les modèles à travers la commande `php artisan make :model` : Utilisateur, Categorie, Ressource, Reservation et SigProb, en définissant les relations entre eux (hasMany, belongsTo) et en configurant les champs fillable et hidden. Cette structure nous permet de gérer facilement toutes les opérations CRUD et de maintenir les données cohérentes.

On a migré les tableaux à la base de données par la commande `php artisan migrate :fresh`

Et on a rempli les tableaux par la commande `php artisan db :seed`

## 3-Page d'accueil

### **a. Présentation générale de l'interface principale**

L'interface principale du projet Data Center Manager constitue le cœur fonctionnel de l'application.

Elle permet à l'utilisateur d'accéder de manière centralisée aux ressources du data center, de

visualiser leur état et d'interagir avec les différents modules du système. Cette page a été pensée

comme un tableau de bord moderne, orienté clarté, efficacité et expérience utilisateur.

### **b. Structure HTML et organisation des vues**

L'interface principale est implémentée dans la vue interface-main.blade.php. La structure HTML est

organisée en plusieurs sections distinctes : une barre de navigation supérieure, une section de

présentation (hero section), un système d'onglets pour le filtrage des ressources, et une zone

d'affichage des cartes. Cette structuration hiérarchique facilite la maintenance du code et son

intégration avec l'architecture MVC de Laravel.

### **c. Style graphique et mise en forme CSS**

La mise en forme repose sur une charte graphique sombre afin de renforcer le contraste et mettre

en valeur les éléments interactifs. Les cartes utilisent des bordures arrondies, des ombres douces

et des couleurs distinctes pour indiquer l'état des ressources (disponible ou en maintenance). Ce

choix de design vise à offrir une lecture rapide des informations tout en conservant un rendu

professionnel.

### **d. Concept des cartes de ressources**

Chaque ressource du data center est représentée sous forme de carte. Ces cartes affichent les

informations essentielles telles que le type de ressource, les caractéristiques techniques (CPU,

RAM, stockage, système d'exploitation) et un bouton permettant d'accéder à plus de détails. Ce

concept visuel permet une comparaison rapide entre les ressources disponibles.

#### **e. Intégration du robot graphique**

Un robot graphique a été intégré dans la section supérieure de l'interface afin de donner une

identité visuelle forte au projet. Il sert d'élément distinctif et renforce l'aspect moderne de

l'application, tout en conservant une cohérence avec le thème technologique du data center.

#### **f. Interactions JavaScript**

Les interactions dynamiques de l'interface sont gérées à l'aide de JavaScript. Celui-ci permet

notamment la navigation entre les catégories de ressources, le défilement horizontal des cartes à

l'aide de flèches, et l'actualisation fluide du contenu sans rechargement complet de la page.

#### **g. Lien avec la base de données, modules et seeders**

Les données affichées dans l'interface proviennent de la base de données via les contrôleurs

Laravel. Chaque carte correspond à une entité stockée (serveur, machine virtuelle, stockage, etc.).

Les seeders ont été utilisés pour initialiser la base de données avec des données de test cohérentes, permettant de valider le bon fonctionnement de l'interface et des modules.

#### **h. Conclusion**

Cette interface principale constitue une page centrale et structurante du projet Data Center

Manager. Elle combine une organisation HTML claire, un design CSS professionnel, des

interactions JavaScript fluides et une connexion efficace avec la base de données. L'ensemble

permet une gestion intuitive et visuelle des ressources du data center.

Figure 1 : Vue générale de l'interface principale

The screenshot shows a dark-themed web application for managing a data center. At the top left is the university logo. To its right are navigation links: Accueil, Ressources, Règles, and Support. On the far right are buttons for Se connecter (Log in) and S'inscrire (Sign up). Below the navigation is a large, stylized graphic of a server rack with a monitor on top. A purple bar at the top contains the text "INSTRUCTING & WEBBAR". Below this is a prominent title "EMAIL FOR DEVELOPPER". A subtext below the title reads: "Consultez les ressources du Data Center, vérifiez la disponibilité et lancez vos demandes de réservation." A "GET started >" button is located in the middle-left area. At the bottom, there are four tabs: Serveurs, Machines virtuelles, Stockage, and Équipements réseau. Below these tabs are three cards representing server resources:

SRV-001 — Serveur principal	SRV-002 — Serveur backup	VM-001 — VM Ubuntu
Available	Available	Available
CPU : 8 cores	CPU : 12 cores	CPU : 4 cores
RAM : 32 GB	RAM : 64 GB	RAM : 8 GB

## DATA CENTER RESOURCES, BUILT FOR RELIABLE OPERATIONS

Designed for environments where performance, availability, and control matter, this platform centralizes the management of data center resources across servers, virtual machines, storage, and network equipment. Every component is tracked, structured, and accessible through a single, consistent interface.

From academic infrastructures to professional IT environments, the system supports controlled reservation and allocation of resources based on clearly defined roles. Users request what they need. Administrators validate, supervise, and maintain full visibility over usage, capacity, and availability—without manual workarounds.

Built on structured data management, the platform enforces traceability at every step. Each reservation, modification, and assignment is logged, making resource usage transparent and auditable over time.

The platform is designed to scale with real infrastructure. Whether managing a limited training lab or a multi-resource data center, it supports efficient utilization, reduces administrative overhead, and ensures resources are used responsibly and securely.

RESOURCES	RESERVATIONS	SUPPORT	PLATFORM
Servers	New Reservation	Help Center	About the Platform
Virtual Machines	My Reservations	Usage Rules	User Roles & Permissions
Storage	Pending Requests	Maintenance Calendar	Statistics & Reports
Network Equipment		Incident Reporting	Security & Access Control

Figure 2 : Présentation des cartes de ressources

Serveurs	Machines virtuelles	Stockage	Équipements réseau
<b>SRV-001 — Serveur principal</b> CPU : 8 cores   RAM : 32 GB Stock : 1000 GB   OS : Linux  <b>Serveur principal</b> Serveur physique pour le Data Center. <a href="#" style="color: blue; text-decoration: none;">View More</a>	<b>SRV-002 — Serveur backup</b> CPU : 12 cores   RAM : 64 GB Stock : 4000 GB   OS : Linux  <b>Serveur backup</b> Serveur secondaire pour le Data Center. <a href="#" style="color: blue; text-decoration: none;">View More</a>	<b>VM-001 — VM Ubuntu</b> CPU : 4 cores   RAM : 8 GB Stock : 200 GB   OS : Linux  <b>VM Ubuntu</b> Machine virtuelle pour le développement et les tests. <a href="#" style="color: blue; text-decoration: none;">View More</a>	

### 4-page de login

Introduction : La porte d'entrée du système "DataCenter Manager" repose sur un module d'authentification sécurisé. Ce module ne se contente pas de vérifier les

identifiants ; il gère également les autorisations d'accès basées sur l'état du compte et le rôle de l'utilisateur.

**b- Analyse du Back-end (AuthController.php) :**

-L'authentification utilise les fonctionnalités natives de Laravel avec une couche de sécurité personnalisée :

-Validation des données : Avant toute tentative, le système vérifie que l'email est au bon format et que le mot de passe est présent.

-Condition de compte actif : La tentative de connexion (Auth::attempt) inclut une clause de sécurité cruciale : 'status' => 'active'. Cela permet à l'administrateur de suspendre un compte sans supprimer ses données.

-Gestion des erreurs spécifique : Si la connexion échoue, le contrôleur vérifie si la cause est un compte inactif ou simplement de mauvais identifiants, afin d'informer précisément l'utilisateur.

-Redirection dynamique par Rôles : Une fois connecté, l'utilisateur est redirigé vers son interface spécifique (admin, responsable, ou utilisateur\_interne) grâce à une vérification du champ roles en base de données.

**c- Interface Utilisateur et Design (Front-end) :**

-L'interface a été conçue pour être à la fois moderne et ergonomique :

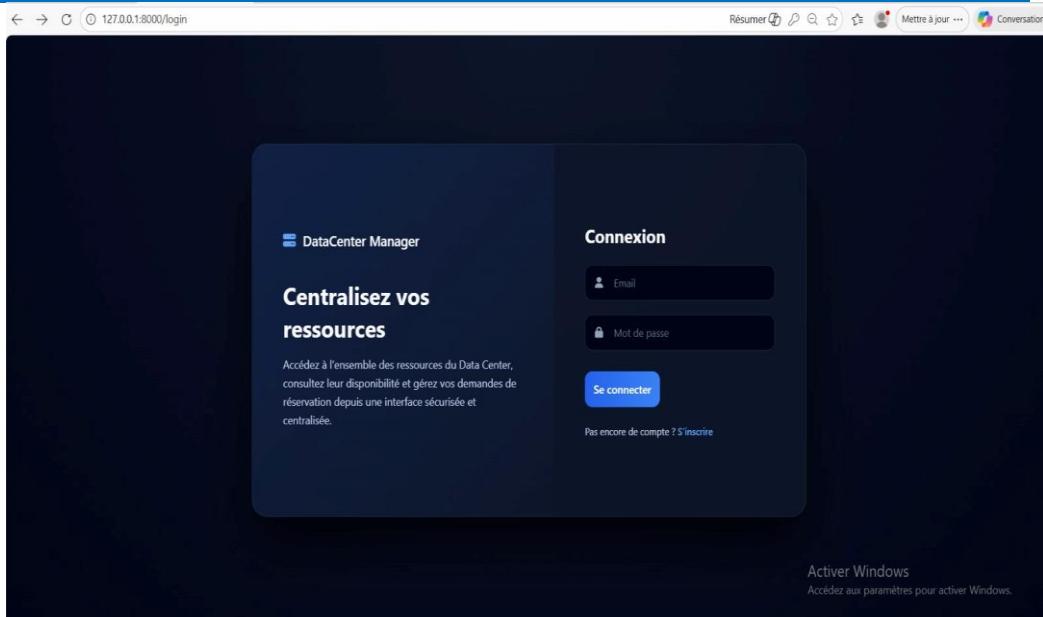
-Design Glassmorphism : La vue utilise des effets de flou (backdrop-filter: blur(16px)) et des transparences pour une esthétique technologique.

-Architecture en deux colonnes : Une colonne visuelle rappelle les objectifs du projet ("Centralisez vos ressources"), tandis que la colonne de droite contient le formulaire de connexion.

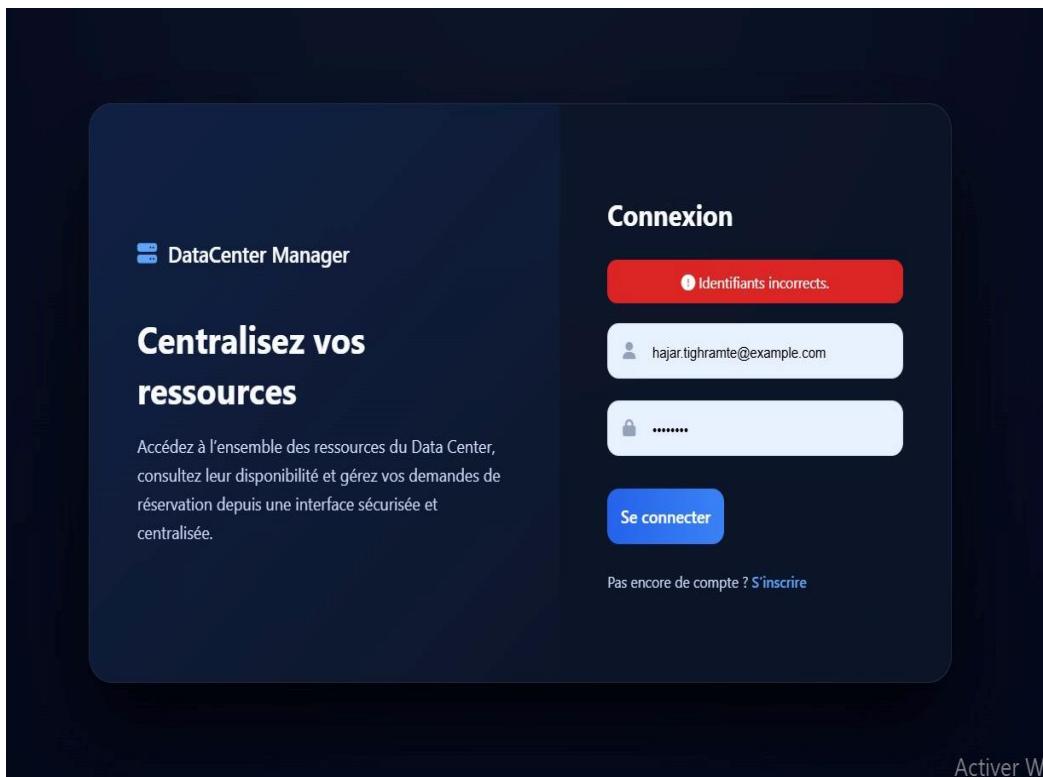
-Feedback en temps réel : Le système affiche des boîtes d'erreur rouges (error-box) si les identifiants sont erronés, permettant une correction immédiate par l'utilisateur.

**d- Analyse visuelle :**

1. Interface de Connexion



## 2. Gestion des erreurs d'accès



### 5-page inscription

#### a-Le code source :

Le code de cette page est générée dans :  
 app/http/Controllers/inscriptionController.php pour back-end et  
 Resources/views/inscription.blade.php et Public/css/inscription.css pour front-end.

### b-Introduction de la page :

La page d’inscription permet aux nouveaux utilisateurs de créer un compte afin d’accéder aux services de la plateforme du Data Center. Elle s’adresse principalement aux utilisateurs internes et aux responsables techniques, et a été conçue de manière simple et sécurisée afin de faciliter la saisie des informations.

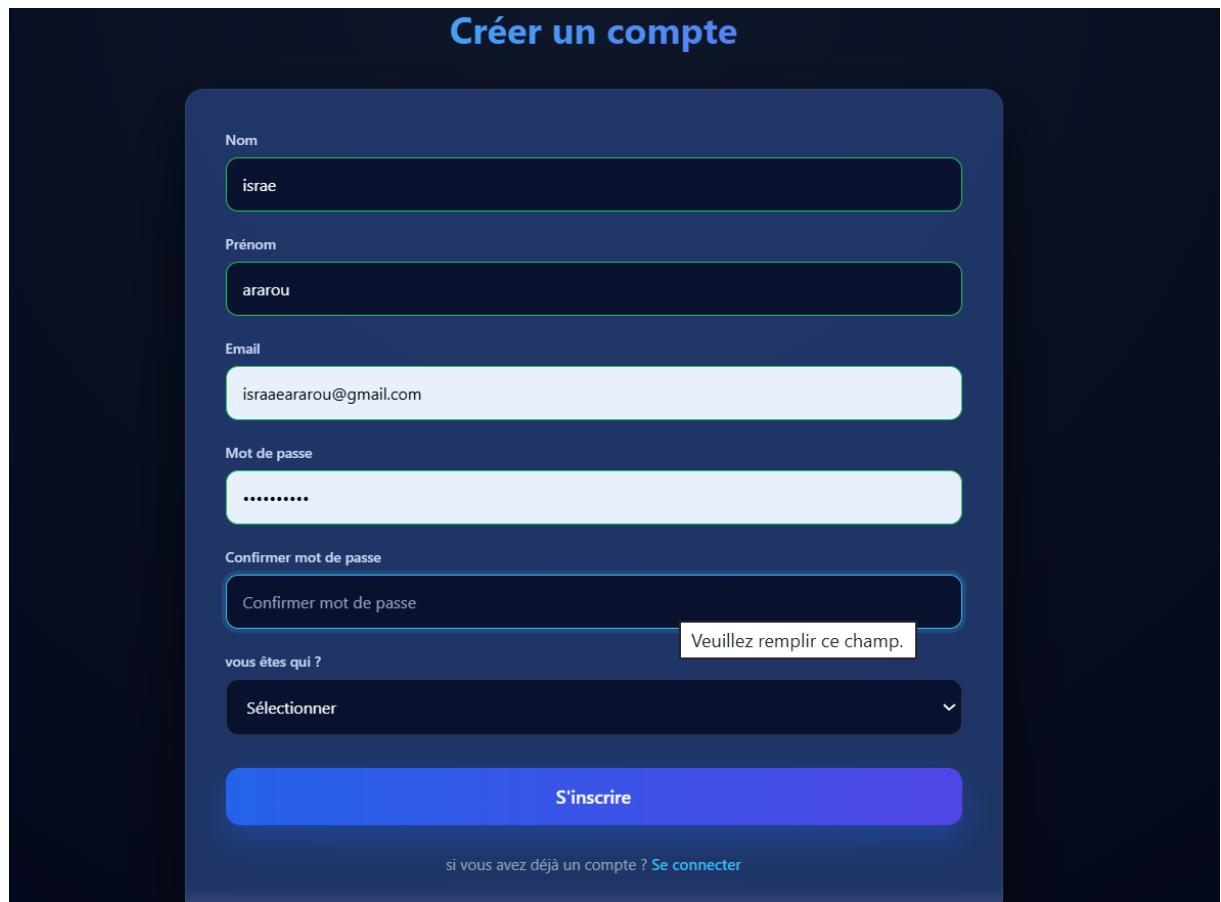
### c-Front-end :Resources/views/inscription.blade.php et Public/css/inscription.css :

L’interface d’inscription est développée en HTML et mise en forme avec du CSS personnalisé, sans recours à des frameworks externes.

Le formulaire contient des inputs obligatoires tels que le nom, le prénom, l’adresse email, le mot de passe, la confirmation du mot de passe et le rôle de l’utilisateur.

Les données sont envoyées via la méthode POST, et un système d’affichage des messages d’erreur permet d’informer l’utilisateur en cas de saisie invalide après l’envoi du formulaire.

Les champs remplis sont en vert



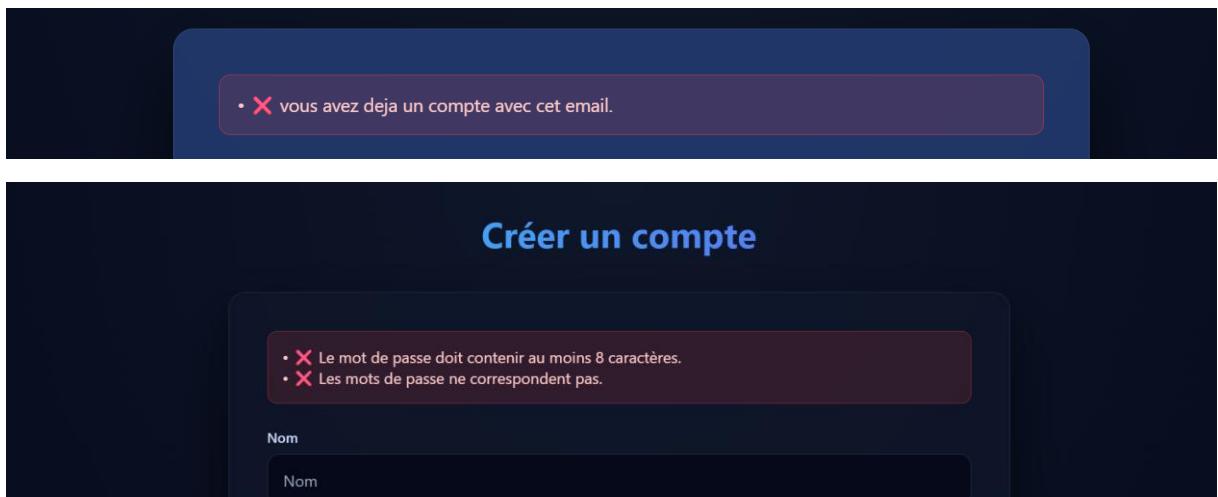
The screenshot shows a dark-themed web form titled "Créer un compte". The form fields are as follows:

- Nom:** israe (green placeholder)
- Prénom:** ararou (green placeholder)
- Email:** israeararou@gmail.com (green placeholder)
- Mot de passe:** ..... (green placeholder)
- Confirmer mot de passe:** (empty field)
- vous êtes qui ?** (dropdown menu showing "Sélectionner")
- S'inscrire** (blue button)

A validation message "Veuillez remplir ce champ." is displayed next to the password confirmation field. At the bottom, a link "si vous avez déjà un compte ? Se connecter" is visible.

## d-Gestion des conflits

Le système d'inscription vérifie l'unicité de l'adresse email ainsi que la validité des mots de passe saisis. En cas d'erreur (email déjà existant ou mot de passe non conforme), des messages explicites sont affichés afin de guider l'utilisateur. Cette gestion des conflits garantit la cohérence et la sécurité des données.



## E-Back-end : inscriptionController

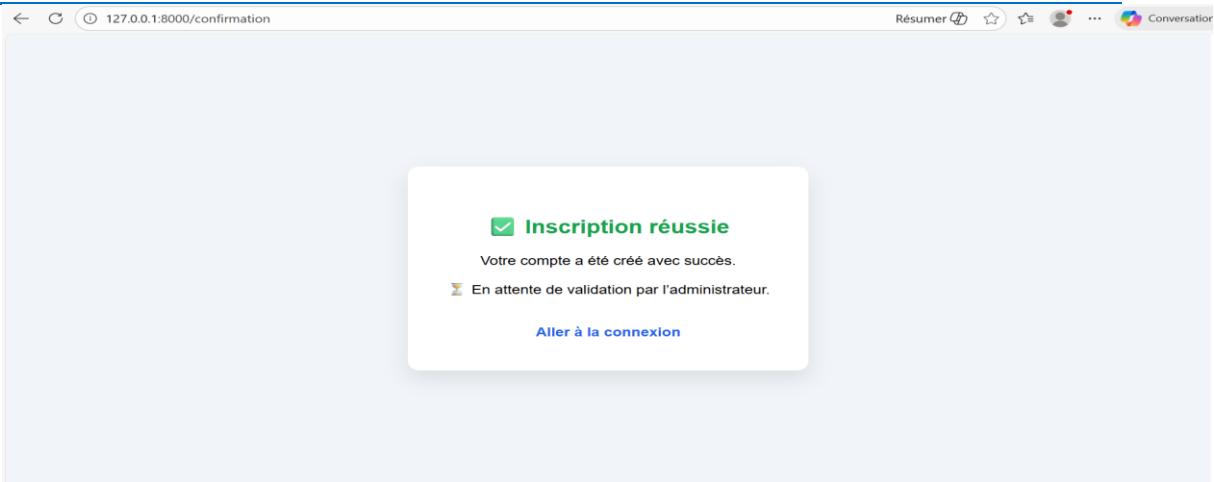
Le contrôleur `InscriptionController` assure le traitement du formulaire d'inscription.

La méthode `create()` permet d'afficher le formulaire, tandis que la méthode `store()` valide les données saisies, vérifie l'unicité de l'email, la conformité du mot de passe et le rôle choisi, puis crée un nouvel utilisateur dans la base de données.

Le mot de passe est chiffré avant l'enregistrement et le compte est créé avec un statut « en attente ».

## Page de confirmation :

Une page de confirmation est ensuite affichée pour informer l'utilisateur que sa demande a bien été enregistrée. (`resources/views/confirmation.blade.php`)



## 6- Interface Utilisateur interne

### a-code source

La page Utilisateur Interne est développée dans :

**Back-end :** app/Http/Controllers/reservController.php

**Front-end :** resources/views/UtilisateurInterne.blade.php et  
public/css/utilisateurinterne.css

L'accès à cette page est réservé aux utilisateurs authentifiés ayant le rôle Utilisateur interne, via le système d'authentification et de middleware de Laravel.

### b-fonctionnalités principales

Cette page représente l'espace personnel des ingénieurs, enseignants et doctorants du Data Center.

**Ses fonctionnalités principales :**

Consultation des ressources disponibles avec champ de recherche.

Soumission de demandes de réservation (période, ressource, justification).

Gestion automatique des conflits de réservation (chevauchement de dates).

Suivi et historique des réservations avec filtres (date, ressource, statut).

Signalement des incidents techniques liés aux ressources.

### c- Front-end

L'interface est développée en HTML avec du CSS personnalisé, sans frameworks externes.

Elle comprend :

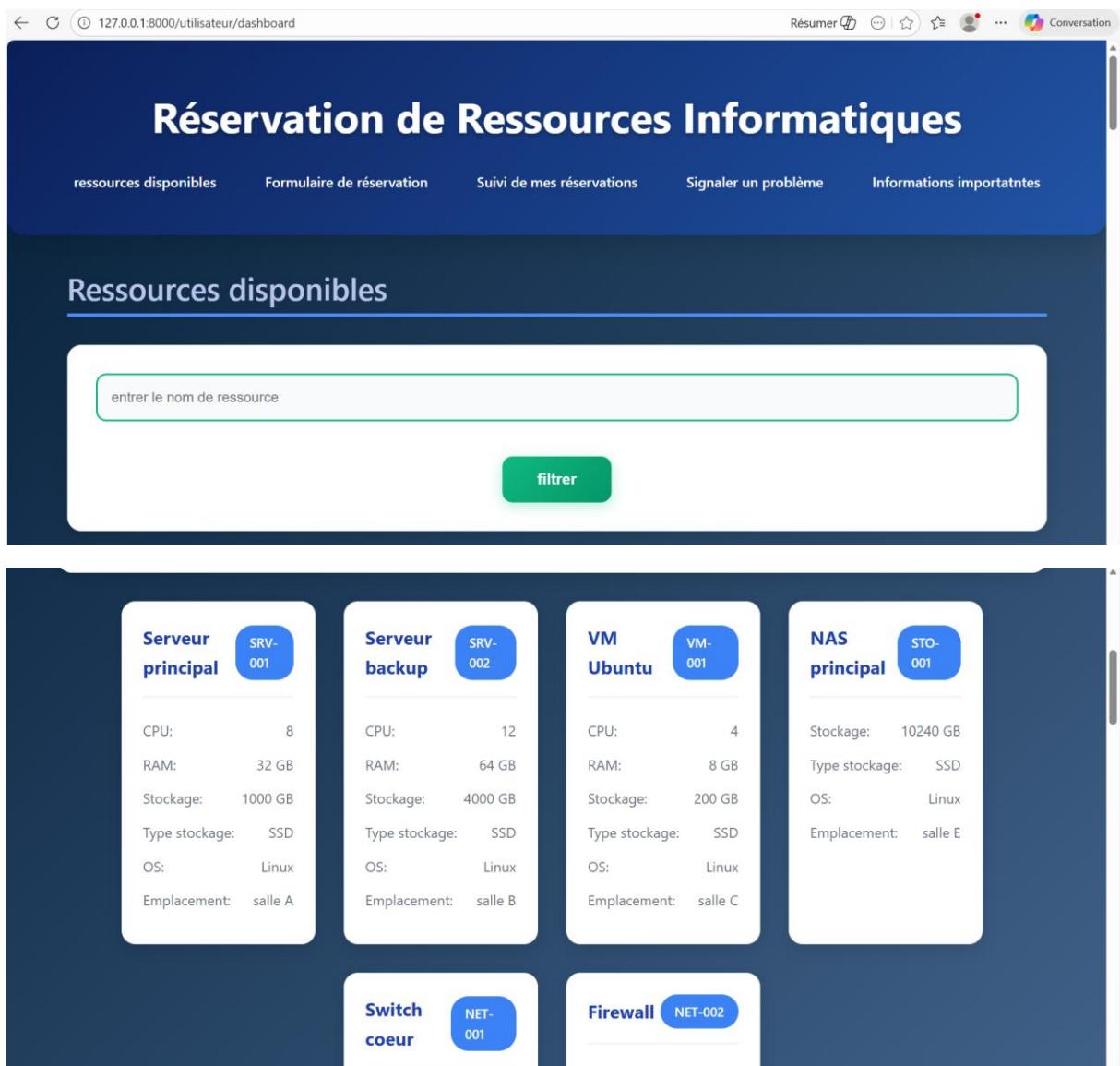
L'affichage des ressources disponibles avec un champ de recherche.

Un formulaire de réservation (ressource, dates, justification).

L'historique des réservations avec filtres (ressource, statut, dates).

Un espace de signalement et de suivi des incidents.

L'affichage dynamique des ressources, réservations et incidents est réalisé à l'aide de boucles @foreach dans le vue Blade.



The screenshot displays a web application interface for managing computer resources. At the top, there's a header bar with links for 'ressources disponibles', 'Formulaire de réservation', 'Suivi de mes réservations', 'Signaler un problème', and 'Informations importantes'. Below this, a search section titled 'Ressources disponibles' contains a search input field ('entrer le nom de ressource') and a green 'filtrer' button. The main content area shows five resource cards:

- Serveur principal (SRV-001)**: CPU: 8, RAM: 32 GB, Stockage: 1000 GB, Type stockage: SSD, OS: Linux, Emplacement: salle A.
- Serveur backup (SRV-002)**: CPU: 12, RAM: 64 GB, Stockage: 4000 GB, Type stockage: SSD, OS: Linux, Emplacement: salle B.
- VM Ubuntu (VM-001)**: CPU: 4, RAM: 8 GB, Stockage: 200 GB, Type stockage: SSD, OS: Linux, Emplacement: salle C.
- NAS principal (STO-001)**: Stockage: 10240 GB, Type stockage: SSD, OS: Linux, Emplacement: salle E.
- Switch cœur (NET-001)**
- Firewall (NET-002)**



### Formulaire de réservation

Choisir une ressource

Serveur backup - SRV-002

Date début

01/01/2026

Date fin

02/01/2026

Projet/usage

Décrivez brièvement le projet ou l'usage de la ressource

Envoyer la demande

Votre demande a été envoyée

### Suivi de mes réservations

RESSOURCE	PÉRIODE	STATUS	DÉCISION
Serveur principal	Du 01/01/2026 Au 03/01/2026	En attente	---

### Signaler un problème

Sélectionnez la ressource (optionnel)

-- Aucune --

Votre Problème

Entrez votre problème...

Signaler

### Suivi de mes signalements

RESSOURCE	PROBLÈME	RÉPONSE	DATE
Serveur backup	cette machine ne fonctionne pas	En attente	28/01/2026



## Informations importantes

### ▼ Politique de réservation

- Les réservations doivent être faites au moins 24h à l'avance
- Durée maximale : 14 jours
- Annulation possible jusqu'à 6h avant le début

### Contact

Département Data

Téléphone: +212(0)666123456

administrateur : saad.med@example.com

© 2025 Data Center. Tous droits réservés.

## d- Gestion des conflits

si la ressource entrée dans le filtre ne se trouve pas

Aucune ressource disponible trouvée.

Avant toute réservation, le système vérifie automatiquement la disponibilité de la ressource.

En cas de chevauchement de dates avec une réservation existante, la demande est refusée et un message d'erreur explicite est affiché à l'utilisateur.(disponibilité de ressource dans cette période)

demande pas enregistrée. Veuillez choisir une autre date. La ressource est déjà réservée sur ces dates.

## e- Back-end : reservController

Le contrôleur reservController assure la logique métier de l'utilisateur interne :

La méthode **index()** récupère les ressources disponibles, l'historique des réservations et les incidents, avec application de filtres, puis retourne la vue UtilisateurInterne.

La méthode **store()** valide la demande, gère les conflits et crée une réservation avec le statut « En attente ».

La méthode **storeSigProb()** permet d'enregistrer les signalements d'incidents et d'en assurer le suivi

## 7- Interface Responsable

### Espace Responsable : Gestion et Supervision du Parc

Le Dashboard Responsable est l'outil de pilotage central pour la gestion des ressources matérielles et le suivi des demandes.

The screenshot shows a dark-themed web interface titled "Supervision du Parc". At the top, it says "Gérez vos ressources et validez les demandes en temps réel." Below this is a form for adding a new resource:

- Fields for "Nom de la ressource" (Resource name), "Code unique" (Unique code), "Catégorie" (Category: Sérvier), and "Système" (System: Linux).
- Fields for "CPU (Cores)", "RAM (Go)", "Stockage (Go)", and "Type" (Type: SSD).
- Fields for "Bande passante" (Bandwidth) and "Emplacement" (Location: salle A, Rack 4).
- A large blue button at the bottom labeled "ENREGISTRER LA RESSOURCE DANS LE PARC" (Register resource in the park).

The screenshot shows a web interface titled "Parc des ressources supervisées". It displays a table of monitored resources:

CODE	NOM / OS	SPECS	ÉTAT	ACTIONS DE MAINTENANCE
SRV-001	Sérvier principal Linux	8 Cores / 32 Go	indisponible	<button>Maintenance</button> <button>Activer</button>
SRV-002	Sérvier backup Linux	12 Cores / 64 Go	disponible	<button>Maintenance</button> <button>Désactiver</button>
VM-001	VM Ubuntu Linux	4 Cores / 8 Go	maintenance	<button>Sortir</button> <button>Désactiver</button>
VM-002	VM Windows Windows	4 Cores / 16 Go	maintenance	<button>Sortir</button> <button>Désactiver</button>
STO-001	NAS principal Linux	Cores / Go	disponible	<button>Maintenance</button> <button>Désactiver</button>
STO-002	SAN	Cores / Go	disponible	<button>Maintenance</button> <button>Désactiver</button>
NET-001	Switch cœur	Cores / Go	disponible	<button>Maintenance</button> <button>Désactiver</button>
NET-002	Firewall	Cores / Go	disponible	<button>Maintenance</button> <button>Désactiver</button>

To the right of the table, there is a note: "Activer Windows" and "Accédez aux paramètres pour activer Windows." (Activate Windows) with a link.

The screenshot shows a dark-themed web application interface. At the top, there's a header bar with a back arrow, a search icon, a user profile 'hajar', and a 'Mettre à jour ...' button. Below the header, the title 'DataCenter Manager | Dashboard Responsable' is displayed. The main content area is divided into two main sections:

- Section 1: Consultation des demandes entrantes**
  - Sub-section: Firewall request from 'ararou israe'.
  - Action buttons: 'Approuver' (Approve), 'Planifier' (Schedule), and 'Refuser' (Reject).
- Section 2: Modération & Alertes Problèmes Réservations**
  - Sub-section: Problem report from 'israe' regarding a 'Serveur backup' stating 'La machine ne fonctionne pas correctement.'
  - Action buttons: 'Répondu : Le problème a été résolu.' (Answered: The problem has been resolved) and 'Supprimer' (Delete).

At the bottom right of the interface, there's a note: 'Activer Windows' and 'Accédez aux paramètres pour activer Windows.'

### a- Ajout et Inventaire des Ressources

Le cycle de gestion commence par l'intégration de nouveaux équipements via un formulaire dédié.

- Saisie technique : Le responsable enregistre le nom, le code unique, la catégorie et les specs (CPU, RAM, OS).
- Méthode store() : Le contrôleur valide l'unicité du code et utilise Auth::id() pour lier la ressource au responsable actuel.
- Disponibilité : À la création, la ressource est automatiquement marquée comme disponible pour les utilisateurs internes

### b- Pilotage du Parc et Maintenance (Back-end)

Le responsable gère le cycle de vie des machines via la méthode updateStatus(). Cette fonction permet de basculer instantanément l'état d'un serveur en base de données :

- Mode Maintenance : Verrouille la ressource pour des interventions (badge orange).
- Activation/Désactivation : Permet de rendre une machine disponible ou de la retirer du parc (badges vert/rouge). Cette logique garantit qu'aucune ressource défaillante ne peut être réservée.



+ Ajouter une nouvelle ressource

Routeur	NET-015	Catégorie : Réseau (Switch/Router)	Système : Windows
5	20	8	Type : HDD
10Gbps	salleA	Matériel Dell PowerEdge	

**ENREGISTRER LA RESSOURCE DANS LE PARC**

### c- Arbitrage des Demandes de Réservation

Lorsqu'une demande est soumise, le responsable utilise la méthode `decider()`. Le back-end filtre les demandes pour ne montrer que celles concernant ses propres ressources. Le responsable analyse le motif et doit obligatoirement saisir une note de décision avant d'approuver ou de refuser, mettant ainsi à jour la table reservation.

### d- Résolution des Incidents et Modération

Le système intègre une gestion des conflits techniques. Grâce à la méthode `repondreSignal()`, le responsable répond aux alertes envoyées par les utilisateurs. Une fois le problème traité, un badge " Répondu" apparaît. Le responsable a également le pouvoir de supprimer (modérer) les signalements via la méthode `detruireSignalement()`

CODE	NOM / OS	SPECS	ÉTAT	ACTIONS DE MAINTENANCE
SRV-001	Serveur principal Linux	8 Cores / 32 Go	disponible	<span>Maintenance</span> <span>Désactiver</span>

État de la ressource mis à jour.

CODE	NOM / OS	SPECS	ÉTAT	ACTIONS DE MAINTENANCE
SRV-001	Serveur principal Linux	8 Cores / 32 Go	indisponible	<span>Maintenance</span> <span>Activer</span>

Consultation des demandes entrantes			1 demande(s) à examiner
DEMANDEUR	RESSOURCE DEMANDÉE	PÉRIODE / MOTIF	ACTION RESPONSABLE
ararou israe	Routeur	Cette ressource est nécessaire pour effectuer un traitement intensif de données (Big Data). Les spécifications de ce serveur (8 Cores / 32 Go RAM) sont indispensables pour faire tourner nos scripts d'analyse sans saturer les machines locales du département	<input type="text" value="la demande est acceptée"/> <span>Approver</span> <span>Planifier</span> <span>Refuser</span>

## Supervision du Parc

Gérez vos ressources et validez les demandes en temps réel.

Action réussie : Statut mis à jour en 'Approuvée'.

### Consultation des demandes entrantes

1 demande(s) à examiner

DEMANDEUR	RESSOURCE DEMANDÉE	PÉRIODE / MOTIF	ACTION RESPONSABLE
ararou israe	SAN	klmsklxmklsnksax pas dispo	<input checked="" type="checkbox"/> Approuver <input type="checkbox"/> Planifier <input type="checkbox"/> Refuser

### Modération & Alertes Problèmes Réservations

Contrôle des problèmes signalés par les utilisateurs

RESSOURCE	UTILISATEUR	PROBLÈME SIGNALÉ	ACTION / RÉPONSE
Serveur backup	israe	La machine ne fonctionne pas correctement.	<input checked="" type="checkbox"/> Répondu : Le problème a été résolu. <input type="button" value="Supprimer"/>

Activer Windows  
Accédez aux paramètres pour

## 8- Interface Administrateur

### a-code source

Les pages administrateur est développée dans :

**Back-end :** app/Http/Controllers/AdminController.php

**Front-end :**

- resources/views/ADministrateur.blade.php
- resources/views/admin\_ressource.blade.php
- resources/views/admin\_user.blade.php
- resources/views/admin\_edit.blade.php

L'accès à cette page est réservé aux utilisateurs authentifiés ayant le rôle admin, via le système d'authentification et de middleware de Laravel.

### **b-fonctionnalités principales**

Cette page représente l'espace personnel des administrateurs pour surveiller le Data Center.

#### **Ses fonctionnalités principales :**

- Gestion complète des **utilisateurs**, rôles et permissions.
- Gestion du **catalogue des ressources**, catégories et caractéristiques.
- Consultation des **statistiques globales** et du taux d'occupation global du Data Center.
- Gestion des périodes de maintenance planifiée.
- Activation/désactivation d'une ressource ou d'un utilisateur.

### **c-les views**

Chaque vue est responsable d'une partie de la gestion, et vous pouvez naviguer entre elles depuis la barre latérale.

#### - ADministrateur

Il est chargé de présenter les statistiques globales.

### Consultation des statistiques globales

28/01/2026

Aperçu rapide

RESSOURCES SUPERVISÉES <b>9</b>	RESSOURCES EN MAINTENANCE <b>3</b>	RESSOURCES indisponibles <b>1</b>	INCIDENTS_TOTAL <b>1</b>
ACTIVE USERS <b>3</b>	DEMANDES D'INSCRIPTION <b>1</b>	RÉSERVATIONS ACTIVES <b>2</b>	RÉSERVATIONS EN ATTENTE <b>1</b>

**Réservations créées (l'année dernière)**

Suivi quotidien basé sur les réservations créées sur les 12 derniers mois.

#### - Admin\_ressource admin\_ressource

Il est responsable de la gestion du catalogue des ressources. Activer,Modifier,Desactiver

### Gestion des ressource

28/01/2026

**les ressources disponibile**

ACTION	CATEGORIE	CODE	NOM	ETAT	DESCRIPTION	CPU	RAME	STOREG	OS	TYPE_ST
<b>mes en maintenance</b>	serveur		Serveur principal	disponible		8			Linux	
<b>désactiver</b>										
<b>mes en maintenance</b>	machine virtuelle		VM Ubuntu	disponible		4			Linux	
<b>désactiver</b>										
<b>mes en maintenance</b>	machine virtuelle		VM Windows	indisponible		4			Windows	
<b>désactiver</b>										
<b>mes en</b>	Baies de		NAS	disponible					Linux	
<b>mes en</b>										

et vous pouvez faire défiler vers la droite pour voir le reste du tableau

ACTION	CATEGORIE	CODE	NOM	ETAT	DISCRITION	CPU	RAME	STOREG	OS	TYPE_STOCKAGE
<button>modifier</button>	serveur		Serveur backup	maintenance		12				Linux
<button>active</button>										
<button>modifier</button>	Baies de stockage		SAN	maintenance						
<button>active</button>										
<button>modifier</button>	equipements réseau		Routeur	maintenance						
<button>active</button>										

## -Admin\_user

Il est responsable de la gestion des utilisateurs . Activer,Modifier,Desactiver

NOM	EMAIL	ROLE	STATUS	CREATED_AT	UPDATED_AT	ACTION
tighramte hajar	hajar.tighramte@example.com	responsable	active	2026-01-27 08:07:25	2026-01-27 08:07:25	<button>modifier</button>
med saad	saad.med@example.com	admin	active	2026-01-27 08:07:25	2026-01-27 08:07:25	<button>désactiver</button>
Bennani Yasmine	yasmine.bennani@example.com	responsable	active	2026-01-28 15:57:54	2026-01-28 15:57:54	<button>modifier</button>
						<button>désactiver</button>

NOM	EMAIL	ROLE	STATUS	CREATED_AT	UPDATED_AT	ACTION
ararou israe	israe.ararou@example.com	utilisateur_interne	en attente	2026-01-27 08:07:24	2026-01-27 08:07:24	<button>Approuver</button>

## -Admin\_edit

Elle n'est visible que lorsque vous essayez de modifier un utilisateur ou une ressource.



كلية العلوم والتكنولوجيات، طنجة

## Développement Web

2025/2026

**MGMT DataCenter**

Responsable : saad med

Statistiques

Gère les utilisateurs

Gère les ressources

Demandes (1)

Modification

Déconnexion

To exit full screen, press and hold Esc 28/01/2026

### Edit un utilisateur

Table des utilisateurs

NOM	EMAIL	ROLE		
actuel tighramte hajar	hajar.tighramte@example.com	responsable		
new tighramte	hajar	hajar.tighramte@example.com	responsable	Enregistrer