

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE MINISTÈRE  
DE

L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ MOULOUD MAMMERI TIZI-OUZOU

FACULTÉ DE GÉNIE ÉLECTRIQUE ET DE L'INFORMATIQUE

DÉPARTEMENT D'INFORMATIQUE



**Niveau :** Master 2 en Systèmes Informatiques

**Module :** Projet technique de développement web



**Réalisé par :**

- Mebrek Amar
- Helali Slimane
- Aouam Madjid

# Plateforme de réservation de salles

## Rôles

- Visiteur : consulte les salles sans compte
- Client : réserve des salles, consulte son historique, laisse des avis après réservation
- Propriétaire : ajoute/modifie/supprime ses salles, consulte les avis reçus
- Administrateur : active/désactive des comptes, supprime des annonces et modère les avis

## Fonctionnalités principales

- Inscription / connexion (JWT stocké en cookie httpOnly)
- Gestion des rôles (client, propriétaire, admin)
- CRUD sur les salles (propriétaires + admin)
- Réservation de salles avec contrôle de conflits
- Historique des réservations (client)
- Les clients ne peuvent laisser un avis que si leur réservation est terminée
- Les nouveaux comptes sont créés inactifs et doivent être activés par l'administrateur
- Interface d'administration pour gérer utilisateurs, salles et avis

## Technologies utilisées

- Node.js + Express
- Vue côté serveur : EJS
- Base de données : MongoDB via Mongoose
- Authentification : JWT
- Hashing des mots de passe : bcryptjs
- Autres : body-parser, cookie-parser

## Structure importante

- serveur.js : point d'entrée
- modeles/ : schémas Mongoose (Utilisateur, Salle, Reservation, Avis)
- routes/ : routes Express (auth, salles, reservations, avis, admin...)
- vues/ : fichiers EJS (frontend minimal)
- public/style.css : styles

## Pré-requis

- Node.js (v16+ recommandé)
- MongoDB en local (url attendue : mongodb://127.0.0.1:27017/reservation)

## Installation

1. Ouvrir un terminal dans le dossier du projet
2. Installer les dépendances :

npm install

3. S'assurer que MongoDB tourne (ex. service local ou mongod)

## Lancer l'application

node serveur.js

*# serveur écoute par défaut sur le port 3000*

Puis ouvrir <http://localhost:3000>.

## Comptes et flux importants

- Admin prédefini : admin@admin.com / admin (le compte est créé automatiquement au premier login admin)
- Nouveaux utilisateurs : actif: false par défaut — l'administrateur doit activer le compte via /admin/utilisateurs avant que l'utilisateur puisse se connecter.
- Un client ne peut poster un avis que si une réservation existe pour la même salle et que sa dateFin est passée.
- La validation de réservation vérifie que dateFin >= dateDebut et empêche les conflits existants.

## Notes pédagogiques

- Le code est volontairement simple et lisible pour un rendu étudiant.
- Pas d'interface riche ni d'API de cartographie intégrée (optionnel).

## Prochaines améliorations possibles

- Ajout de notifications / messages d'alerte côté UI
- Intégration d'une carte (Mapbox/Google) pour la localisation
- Pagination / recherche avancée des salles
- Tests automatisés