



Par Robert DIASSÉ

# Mini Évaluation – Amélioration Sécurisée du Projet Authentification

---

## Objectif

Améliorer la **V1 de votre projet existant** en y intégrant :

- Le **hashage des mots de passe**
- Une **connexion sécurisée**
- L'**affichage structuré des utilisateurs**
- Un **design soigné**
- Un **travail collaboratif efficace avec Git & GitHub**

## Contexte

Chaque groupe a déjà réalisé une première version avec :

- Une inscription fonctionnelle (enregistrement dans un fichier `users.json`)
- Un dépôt GitHub avec le projet

**Vous devez maintenant améliorer ce projet sans le recommencer, mais en suivant une série d'étapes bien structurées.**

## Feuille de route de l'amélioration

### Étape 1 – Mise à jour de la fonctionnalité d'inscription

- Dans le backend, installez `bcrypt` :

```
npm install bcrypt
```

- Modifiez votre route `/register` pour :
  - Générer un identifiant unique (`id`)
  - Hasher le mot de passe avec `bcrypt.hash(...)`
  - Enregistrer les données dans `users.json`

### Étape 2 – Implémenter la connexion sécurisée

- Dans la route `/login` :
  - Recherchez l'utilisateur via son email
  - Vérifiez le mot de passe avec `bcrypt.compare(...)`

- Répondez avec un message clair selon le succès ou l'échec

## Étape 3 – Ajouter une route `/users` pour l'affichage

- Créez une route **GET** `/users` dans votre backend
- Cette route doit :
  - Lire `users.json`
  - Retourner les utilisateurs (sans exposer les mots de passe)

## Étape 4 – Affichage dynamique et design

- Dans une nouvelle page ou section HTML :
  - Utilisez Bootstrap et/ou DataTables
  - Affichez les utilisateurs sous forme de tableau dynamique
  - Colonnes : ID, Nom, Email
  - Le design doit être **professionnel et responsive**
  - Attention aux détails : marges, typographie, contrastes, alignement

## Travail en équipe (obligatoire)

- Chaque membre doit **travailler sur une branche dédiée**

```
git checkout -b feature/secure-login
```

- Commits fréquents et explicites :
  - `git commit -m "Ajout du hash sur mot de passe"`
  - `git commit -m "Connexion avec bcrypt"`
- Utilisation de **Pull Requests** pour les intégrations
- Chaque dépôt final doit refléter un **vrai travail collaboratif**

## Présentation finale

- Chaque groupe présentera son projet.
- Un membre **choisi au hasard** :
  - Expliquera le fonctionnement des routes `/register`, `/login`, `/users`
  - Démontrera le design et l'affichage
  - Montrera le dépôt GitHub (historique des commits, branches, PR)

## Barème d'évaluation

Critère	Points
Ajout du hash dans l'inscription ( <code>bcrypt.hash</code> )	3 pts
Connexion avec vérification hashée ( <code>bcrypt.compare</code> )	3 pts
Route <code>/users</code> fonctionnelle	2 pts
Affichage dynamique avec DataTables/Bootstrap	3 pts

Critère	Points
Design propre, et cohérent	4 pts
Utilisation correcte de Git/GitHub (branches, PR, commits)	3 pts
Présentation orale claire	2 pts

**Total : 20 points**

## Contraintes et rappels

- Aucun mot de passe **en clair** dans le fichier `users.json`
- Aucun membre ne doit "tout faire" à la place des autres
- Le travail **doit évoluer à partir du projet existant**