Présentation du projet - API Node.js + MySQL

# 1. Introduction

- Présentation rapide du projet (Node.js + Express + MySQL).

- Objectif : gérer utilisateurs (bénévoles & associations) et missions.

# 2. Architecture du code

- routes/ → URLs de l’API (/user, /mission).

- controllers/ → logique métier (CRUD).

- middlewares/ → ex : authMiddleware (JWT).

- db.js → connexion MySQL.

- seed.js / seeduser.js → données factices avec Faker.

# 3. Sécurité

- bcrypt → hachage des mots de passe.

- JWT → génération de tokens après login.

- authMiddleware → bloque l’accès aux routes protégées.

# 4. Choix technologiques

- Pourquoi MySQL et pas MongoDB ?

- → Relationnel, contraintes, intégrité des données.

- → Adapté aux relations missions ↔ utilisateurs.

- Pourquoi Node.js / Express ?

- → Même langage front/back (JavaScript).

- → Léger, flexible, rapide, gros écosystème npm.

# 5. Démo (Thunder Client)

- 1. GET /user/get → liste des utilisateurs (mots de passe hachés).

- 2. POST /user/login → connexion → token JWT.

- 3. POST /mission/post → création d’une mission avec Authorization: Bearer token.

# 6. Conclusion

- API REST sécurisée, claire et fonctionnelle.

- Mots de passe protégés, routes sécurisées par JWT.

- Prête à évoluer avec un futur front-end.