



# **Code Challenge – Ma Mini Bibliothèque**

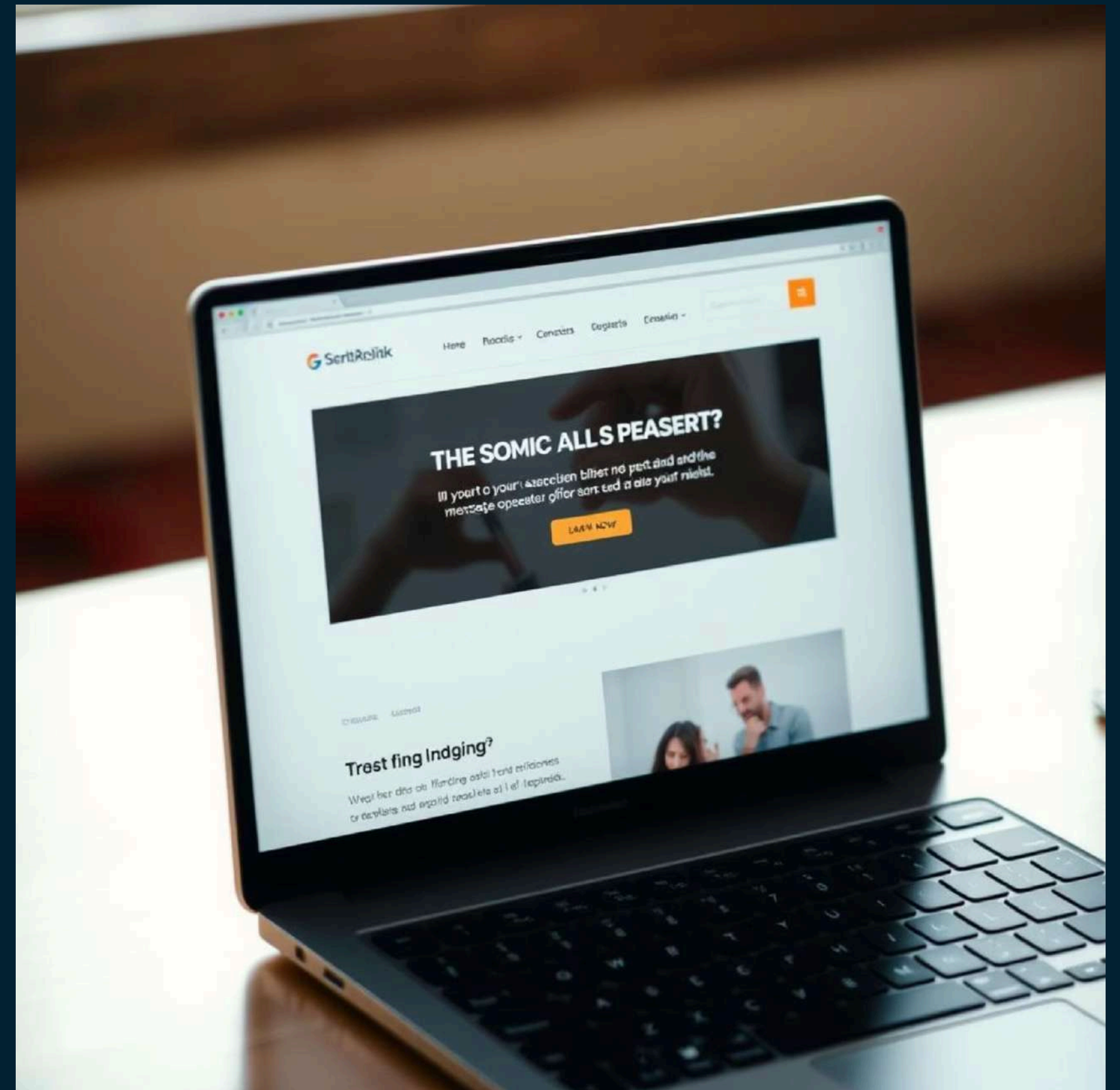


# Objectif général

Développer une API REST complète pour gérer un système de bibliothèques locales. Chaque utilisateur possède une bibliothèque. Chaque bibliothèque contient des livres. Un utilisateur peut emprunter des livres d'autres bibliothèques.

## Compétences visées :

- Créer des entités TypeORM
- Établir des relations MySQL (1-n, n-1, n-n)
- Implémenter des contrôleurs RESTful
- Réaliser des requêtes avancées (jointures, filtres)
- Organiser un projet NestJS en modules



## ✓ Structure attendue du projet

```
src/  
├── users/  
│   ├── user.module.ts  
│   ├── user.controller.ts  
│   └── user.service.ts  
├── libraries/  
│   ├── library.module.ts  
│   ├── library.controller.ts  
│   └── library.service.ts  
├── books/  
│   ├── book.module.ts  
│   ├── book.controller.ts  
│   └── book.service.ts  
├── loans/  
│   ├── loan.module.ts  
│   ├── loan.controller.ts  
│   └── loan.service.ts  
├── app.module.ts  
├── app.service.ts  
├── app.controller.ts  
└── main.ts
```





# ✍️ Spécifications fonctionnelles

Fonction	Description
Lister les utilisateurs	GET /users
Créer un utilisateur	POST /users
Créer une bibliothèque	POST /libraries (avec user_id)
Ajouter un livre à une bibliothèque	POST /books
Lister tous les livres disponibles	GET /books?available=true
Lister tous les livres d'un genre	GET /books?genre=roman
Emprunter un livre	POST /loans
Terminer un prêt	PATCH /loans/:id/return
Lister tous les prêts d'un utilisateur	GET /users/:id/loans



# SCHÉMA BASE DE DONNÉES

## 🧱 Tables :

User (id, name, email, password)

Library (id, name, location, user\_id)

Book (id, title, author, genre, available, library\_id)

Loan (id, book\_id, borrower\_id, start\_date, end\_date, returned)

## 🔄 Relations :

- 1 User → 1 Library
- 1 Library → \* Books
- 1 Book → 0 ou 1 Loan actif
- 1 User → \* Loan (en tant qu'emprunteur)



# LIVRABLES ATTENDUS

Éléments	Détails
 Repo GitHub	Organisation par module
 README.md	Description + instructions d'installation
 Données de test (Optionnel)	Import via script SQL
 Swagger	Documenter l'API
 Export BDD (Optionnel)	Fichier .sql
 Présentation (Obligatoire)	Pitch + démo en 5 min (Reviews)

 **Bonne chance !**

Montre ton professionnalisme :

- écris un code lisible, bien structuré,
- gère proprement les erreurs,