



# Rapport TP2

# Application

# “BookTracker”

---

ZINEB BENNISS

# Objectif du TP

L'objectif de ce TP est de développer une application web complète en Node.js avec Express.js et MongoDB, permettant de :

- Gérer des utilisateurs (inscription et connexion)
- Ajouter, consulter et lister des livres dans une base de données MongoDB
- Manipuler un moteur de templates (Pug) pour générer dynamiquement les pages HTML
- Gérer les routes backend selon le modèle MVC (Modèle - Vue - Contrôleur)

# Architecture du Projet

The screenshot shows a file explorer window with the following structure:

- TP2-BOOKTRACKER
  - config
  - models
    - Book.js
    - User.js
  - node\_modules
  - public
    - styles.css
  - routes
    - authRoutes.js
    - bookRoutes.js
  - views
    - addBook.pug
    - books.pug
    - layout.pug
    - login.pug
    - register.pug
- app.js
- package-lock.json
- package.json
- tailwind.config.js

Le projet est structuré autour de trois parties principales :

- Les modèles (Models) : définissent la structure des données, notamment les utilisateurs et les livres.
- Les routes (Routes) : gèrent la logique de navigation et les interactions entre le client et le serveur.
- Les vues (Views) : utilisent le moteur de templates Pug pour afficher les pages de connexion, d'inscription et la liste des livres.

The screenshot shows the MongoDB Compass interface. On the left, the 'CONNECTIONS' sidebar lists databases: tp2, tp2.auth, admin, booktracker, config, local, and tp2\_auth. Under tp2\_auth, there are 'books' and 'users' collections. The 'users' collection is selected. On the right, the main pane displays two documents from the 'users' collection:

```
_id: ObjectId('6910eaf35bedf227d3af2fb8')
username : "zineb"
password : "$2b$10$MhWDx9jg72gx6oqPoURHPuTeEB4xU0l570/i4aEw.ST9H8e0IP1oy"
__v : 0
```

```
_id: ObjectId('6910f34515607815ed6c7216')
username : "zineb1"
password : "$2b$10$.eHx0NhM2/kL3W6chWI9d.OAIgvpvfU38.f4J0.k/nu23YW5VD4Qy"
__v : 0
```

L'application se connecte à une base de données MongoDB locale pour stocker les informations, et utilise Mongoose pour faciliter la manipulation des données.



**Connexion**

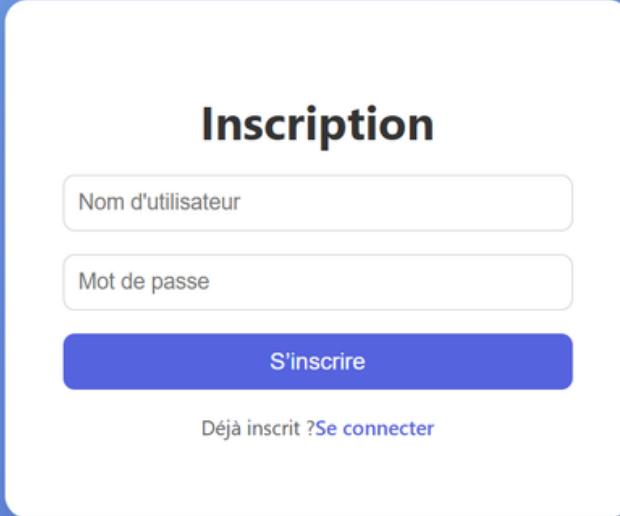
zineb1

•••

Se connecter

Pas encore de compte ?[S'inscrire](#)

This image shows a login form titled "Connexion". It contains two input fields: one for the username "zineb1" and one for the password "•••". Below the password field is a blue "Se connecter" button. At the bottom of the form, there is a link for users who have not yet registered: "Pas encore de compte ?[S'inscrire](#)".



**Inscription**

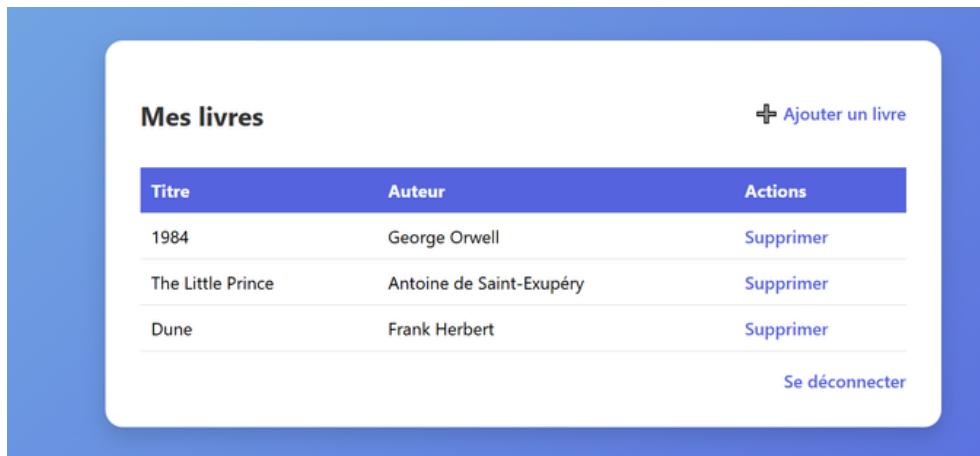
Nom d'utilisateur

Mot de passe

S'inscrire

Déjà inscrit ?[Se connecter](#)

This image shows a registration form titled "Inscription". It contains two input fields: one for the "Nom d'utilisateur" and one for the "Mot de passe". Below these fields is a blue "S'inscrire" button. At the bottom of the form, there is a link for users who have already registered: "Déjà inscrit ?[Se connecter](#)".



Ce TP a permis de mettre en pratique les concepts fondamentaux du développement web côté serveur, notamment la gestion des routes, des modèles de données et la communication entre le front-end et le back-end.

Le résultat final est une application fonctionnelle, claire et extensible, constituant une base solide pour des fonctionnalités futures comme la mise à jour ou la suppression des livres.