

TP : Book Reading Tracker



Réalisée par :
Laila Marqoum

Encadrée par :
Mme Amal Ourdou



Introduction

Ce projet intitulé **Book Reading Tracker** a pour objectif de créer une application web simple et intuitive permettant de suivre ses lectures, enregistrer ses livres, et visualiser ses statistiques personnelles. Ce travail a été réalisé dans le cadre du module de développement web, en utilisant principalement les technologies suivantes :

- **Node.js** et **Express.js** pour le backend,
- **MongoDB** pour la base de données,
- **HTML, TailwindCSS et JavaScript** pour le frontend.

Objectif principal : permettre à l'utilisateur de gérer facilement sa bibliothèque numérique.

Structure du projet

Le projet est organisé selon une structure claire et modulaire :

- **app.js** : Point d'entrée principal de l'application.
- **routes/books.js** : Définit les routes liées aux livres.
- **public/** : Contient les fichiers statiques (HTML, CSS, JS).
- **views/** : Interface utilisateur.

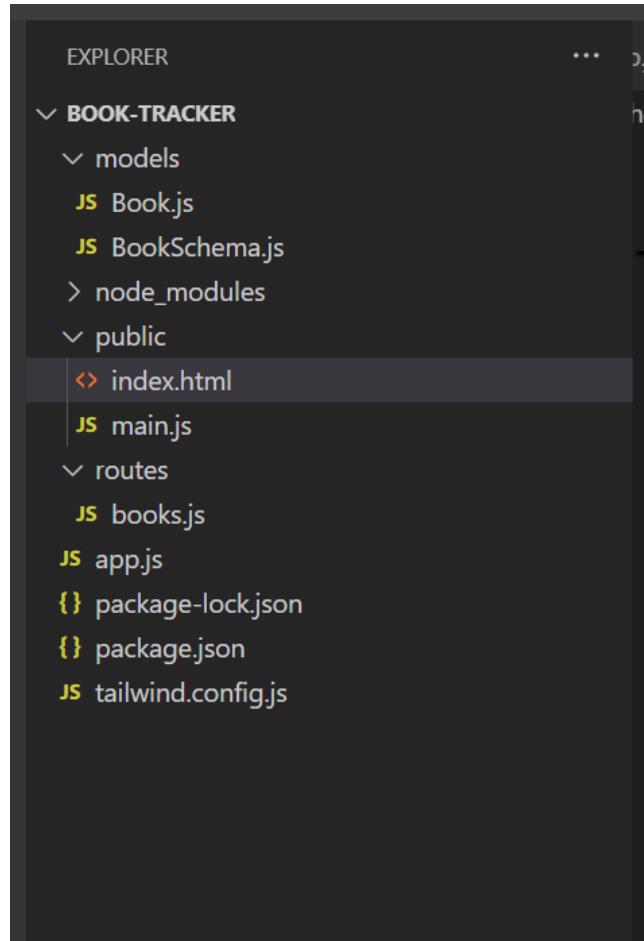


FIGURE 1 – Structure du projet Book Tracker.

Backend avec Node.js et MongoDB

Le fichier `app.js` initialise le serveur Express et se connecte à la base MongoDB. Il définit également les routes principales pour gérer les livres.

Extrait du code `app.js`

```
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const booksRouter = require('./routes/books');

const app = express();

// MongoDB connection
mongoose.connect('mongodb://127.0.0.1:27017/bookTracker', {
  useNewUrlParser: true,
  useUnifiedTopology: true
}).then(() => console.log("MongoDB connected"))
  .catch(err => console.log(err));

app.use(express.static('public'));
app.use(bodyParser.json());
app.use('/books', booksRouter);

const PORT = 3000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```

Interface utilisateur (Frontend)

L'interface a été conçue avec **TailwindCSS** pour offrir un design moderne, minimaliste et responsive.

The screenshot shows a user interface for adding a new book. At the top, there is a header with the text "Suivez vos lectures, votre progression et vos statistiques personnelles." and a footer with "Créé avec ❤️ par Marqoum Laila". Below this, the main form area has a title "Ajouter un nouveau livre". The form consists of several input fields arranged in a grid:

Atomic Habits	James Clear
100	120
Re-read	Print
200	Friend

At the bottom of the form is a blue button labeled "Ajouter le livre" (Add book). To the right of this button is a smaller button labeled "Activer Windows".

FIGURE 2 – Interface du formulaire d'ajout d'un livre.

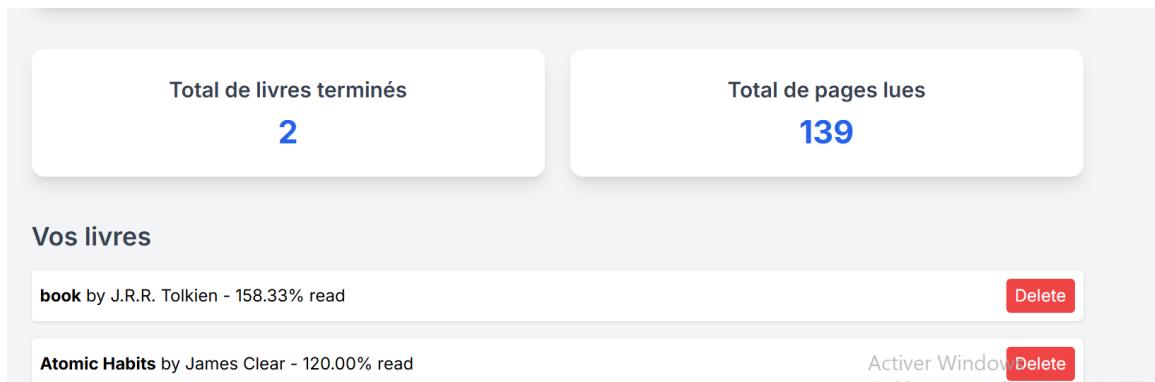


FIGURE 3 – Interface du formulaire d'ajout d'un livre.

Ajout d'un livre dans la base de données MongoDB

Cette partie illustre le fonctionnement de la communication entre le client et le serveur lors de l'ajout d'un nouveau livre. L'utilisateur saisit les informations dans le formulaire, puis ces données sont envoyées au serveur via une requête HTTP POST. Le serveur, à son tour, les enregistre dans la base de données MongoDB.

The figure shows the Compass MongoDB interface. On the left, the connection tree shows "localhost:27017" connected, with "bookTracker" selected. Under "bookTracker", "bookmodels" is selected. The main area shows the "Documents" tab with two results. The first document is for "Atomic Habits" by James Clear, with fields like _id, title, author, pages, status, price, format, and suggestedBy. The second document is for "book by J.R.R. Tolkien", also with similar fields. Both documents have a status of "Read".

FIGURE 4 – Affichage du document inséré dans la base de données MongoDB.

Conclusion

Ce TP m'a permis d'acquérir une expérience pratique dans la conception d'applications web modernes intégrant un backend avec MongoDB et Express, et un frontend élégant avec TailwindCSS. L'application **Book Reading Tracker** constitue une base solide pour un système de gestion de lectures personnelles, extensible vers une version multi-utilisateurs à l'avenir.