



# Développement d'applications web M1-2025 - Projet (FR)

## Introduction

Ce projet vise à confirmer les différents acquis du cours de développement d'applications web du M1-S1-2025.

La date limite du rendu est le **14 Novembre, 23h59**.

Le projet est à rendre, avec les noms des membres du groupe, à l'adresse **gerald.gallet@ext.junia.com**.

---

Vous travaillerez par **groupe de 4 personnes**, et la note sera commune.

Vous êtes libres de vous séparer le travail comme vous le souhaitez : moitié sur le front, moitié sur le back, ou chacun sa fonctionnalité, tout ce qui fonctionne pour vous.

---

Le rendu se fera via un repository Github.

La propreté de l'historique des commits entrera en compte dans la note.

---

Vous partez d'une codebase commune - qui correspond aux fonctionnalités développées en classe - et vous devrez ajouter le maximum de fonctionnalités.

**La qualité du code produit sera plus importante que le nombre de fonctionnalités livrées.**

Vous implémentez les fonctionnalités dans l'ordre qui vous va. Réfléchissez en amont à l'ordre pour pouvoir apporter les changements petit à petit de manière propre.

La codebase peut être trouvée ici : <https://github.com/GeraldGallet/m1-s1-2025-project>

---

## **Stack technique**

La stack technique du projet est la suivante

- Back-end
  - Typescript
  - NestJS
  - TypeORM
  - SQLite
  - API REST (pas de GraphQL)
- Front-end
  - Typescript
  - React
  - Vite
  - Ant Design
  - `@tanstack/react-router`

**Aucune déviation de la stack vue en cours ne sera tolérée.**

---

Vous suivrez l'architecture de code vue en cours : séparation des différentes couches, organisation par domaine.

---

Les règles ESLint devront être respectées. Si un fichier s'ouvre en rouge sur VSCode, cela fera perdre des points.

---

Chaque `any` (même ignoré), fait perdre des points.

---

Les types suivants seront explicités :

- paramètres de fonctions
  - retour de fonctions
  - types génériques (ex. `useState<string>()`)
- 

Aucun design n'a été préparé : faites-vous plaisir.

La seule règle est d'utiliser le composant demandé lorsque c'est spécifié : si on demande une modale, une modale doit être utilisée.

## Fonctionnalités

Voici la liste des fonctionnalités à implémenter :

### Modèle de données

Vous devez ajouter une gestion de clients.

Un client a

- Un nom
- Un prénom
- Une adresse mail facultative

- Une photo facultative → vous pouvez stocker le lien vers une photo, plutôt que la photo elle-même.

Sur les livres et les auteurs, vous devez ajouter une photo.

Vous devez implémenter un système de ventes : on pourra simplement enregistrer qu'un client X a acheté un livre Y à une date Z.

## Liste des clients

- Une page `/clients` me permet de voir la liste des clients.
  - Une liste s'affiche
    - Je vois le nom / prénom du client
  - Depuis cette liste, je peux supprimer un client
    - Une modal me demandant de confirmer la suppression s'affiche
  - Quand je clique sur un client, je vais sur la page de détails de ce client.
  - Je vois, pour chaque client, le nombre de livres qu'il a acheté
- Un bouton me permet d'ouvrir une modal, me permettant de créer un client.

## Détails d'un client

Une page `/clients/$clientId` me permet de voir les détails d'un client.

- Les informations du client s'affichent
- Je peux modifier les informations du client directement dans la page
- Je vois la liste des livres que le client a acheté
  - Je vois le nom du livre, l'auteur, et la date de l'achat
  - Quand je clique sur un livre, je navigue vers la page de détails de ce livre

## Liste des livres

Une page `/books` me permet de voir la liste des livres.

- Une liste s'affiche
- Je peux supprimer un livre depuis cette liste : j'ai une modale de confirmation avant de supprimer.
- Quand je clique sur un livre, j'arrive sur la page de détails de ce livre
- Je vois le nombre de clients ayant acheté ce livre

## Détails d'un livre

Une page `/books/$bookId` me permet de voir les détails d'un livre.

- Je vois toutes les informations du livre.
- Je peux éditer les informations du livre.
- Je vois la liste des clients qui l'ont acheté.
- J'ai un bouton me permettant d'ouvrir une modal d'achat
  - Cette modale me permet de choisir un client, une date, et d'enregistrer l'achat

## Liste des auteurs

Une page `/authors` me permet de voir la liste des auteurs.

- Je vois la liste des auteurs
- Je peux supprimer un auteur depuis cette liste : j'ai une modale de confirmation avant de supprimer.
- Pour chaque auteur, je vois le nombre de livres écrits.
- J'ai un bouton me permettant d'ouvrir une modale de création d'auteur.
- Quand je clique sur un auteur, j'arrive sur la page de détails de cet auteur.

## Détails d'un auteur

Une page `/authors/$authorId` me permet de voir les détails d'un auteur.

- Je vois toutes les informations d'un auteur.
- Je peux éditer les informations de l'auteur.
- Je vois la liste des livres écrits par l'auteur.
  - Quand je clique sur un des livres, j'arrive sur la page de détails de ce livre.
- Je vois le nombre moyen de ventes des livres de cet auteur.

## Navigation & Layout

Chaque domaine dispose d'un breadcrumb :

<https://ant.design/components/breadcrumb>

Quand je suis dans la page des livres, ce breadcrumb affiche seulement "Books".

Quand je suis dans les détails d'un livre, "Books" → "<name of the book>".

Quand je clique sur "Books", je reviens sur la page des livres.

Et ce pour toutes les ressources.

Le menu me permet de naviguer

- vers la page d'accueil
- vers la liste de chaque ressource