



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

Nom de naissance

- Marechal

Nom d'usage

- Marechal

Prénom

- Morgane

Adresse

- °6 rue henri messerer 13001 MARSEILLE

Titre professionnel visé

Concepteur développeur d'applications Niveau 6

MODALITÉ D'ACCÈS :

- Parcours de formation
- Validation des Acquis de l'Expérience (VAE)

DOSSIER PROFESSIONNEL (DP)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.

Ce titre est délivré par le Ministère chargé de l'emploi.

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel (DP)** dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- une déclaration sur l'honneur à compléter et à signer ;
- des documents illustrant la pratique professionnelle du candidat (facultatif)
- des annexes, si nécessaire.

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.

DOSSIER PROFESSIONNEL (DP)

► <http://travail-emploi.gouv.fr/titres-professionnels>

Sommaire

Exemples de pratique professionnelle

Développer une application sécurisée

p. 4

- Safebase
- Pangea

p. 5

p. 10

Concevoir et développer une application sécurisée organisée en couches

p. 16

- Pangea
- Maxaboom
- SIGB

p. 17

p. 22

p. 34

Préparer le déploiement d'une application sécurisée

p. 46

- SIGB

p. 47

Annexes (Si le RC le prévoit)

p. 57

DOSSIER PROFESSIONNEL ^(DP)

EXEMPLES DE PRATIQUE

PROFESSIONNELLE

DOSSIER PROFESSIONNEL (DP)

Développer une application sécurisée

Activité-type 1

Exemple n°1 - Safebase

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Safebase est un projet qui vise à sauvegarder des bases de données sql et postgres à un instant T. Les sauvegardes des différentes bases de données peuvent se faire manuellement mais aussi de manière automatique avec un cron. Il est également possible de restaurer ces dernières via l'interface utilisateurs.

J'ai fait ce projet seule, j'ai donc développé le back-end et le front-end.

Ce projet comporte une partie interface graphique utilisable sur un navigateur web.

J'ai utilisé une image docker pour postgres et adminer pour le développement.

Comme template et library, j'ai utilisé fastify.js pour le back et de react.js pour le front.

J'ai également utilisé les outils de versioning git et github.

Le back avec node et fastify.js

Pour la partie back-end, j'ai utilisé le mini framework fastify.js qui fonctionne avec Node.js.

Node.js est un environnement d'exécution JavaScript côté serveur, basé sur le moteur V8 (Google Chrome). Il permet de développer des applications réseau scalables et performantes en utilisant JavaScript.

Fastify.js un framework optimisé pour la vitesse. On peut utiliser des routes et l'envoie des données sous forme d'API REST, c'est pour cela que je l'ai choisi.

Ce framework dispose également de certains outils pour la gestion des erreurs et la sécurité. Pour la gestion des erreurs, j'ai utilisé les méthodes try/catch pour envoyer le message d'erreur approprié. Et pour la sécurité, les opérations CRUD sont protégées par des verbes HTTP spécifiques (GET, POST, PUT, DELETE). Les champs obligatoires des requêtes POST sont aussi vérifiés. Pour une meilleures vérifications des données entrées par les utilisateurs, on utilise également JSON Schema pour valider les requêtes.

DOSSIER PROFESSIONNEL (DP)

```
fastify.post('/api/database', {
  schema: {
    body: {
      type: 'object',
      required: ['user', 'password', 'host'],
      properties: {
        user: { type: 'string', minLength: 3 },
        password: { type: 'string', minLength: 8 },
        host: { type: 'string', format: 'hostname' }
      }
    }
  },
  (request, reply) => [
    const newDatabase = request.body;
    if (!newDatabase || !newDatabase.user || !newDatabase.password || !newDatabase.host) {
      return reply.status(400).send({ error: "Données de base de données manquantes ou invalides" });
    }
    const databases = new DatabasesManagement();
    databases.insertNewDatabase(newDatabase)
    .then(result => {
      console.log('Insertion réussie:', result);
      reply.status(201).send({ message: 'Base de données ajoutée avec succès', result });
    })
    .catch(error => {
      console.error('Erreur lors de l\'insertion:', error);
      reply.status(500).send({ error: 'Erreur lors de l\'insertion de la base de données' });
    });
  ]
});
```

Le front avec React.js

La partie front est développé en React.js. Elle communique avec le back via une API REST et des requêtes asynchrone fetch.

React est une bibliothèque JavaScript (maintenue par Meta/Facebook) pour construire des interfaces utilisateur (UI) dynamiques et modulaires.

Elle possède plusieurs caractéristiques clés :

- Composants réutilisables : Structure l'UI en briques isolées et combinables.
- Virtual DOM : Optimise les performances en minimisant les mises à jour du DOM réel.
- JSX : Syntaxe proche du HTML pour décrire l'UI directement dans le JavaScript.
- Hooks (comme useState, useEffect) : Gère l'état et les effets secondaires sans classes.
- Unidirectionnel : Flux de données parent → enfant (via les props).

Pour les composants de l'interface utilisateur, j'ai également utilisé la library Material-UI. Material-UI est une bibliothèque de composants React qui implémente le Material Design (système de design de Google). Cela permet d'avoir des composants prêts à l'emploi, responsables et qui respectent des standards d'accessibilité.

Ci dessous, voici quelques visuels de l'application de backups de bases de données :

DOSSIER PROFESSIONNEL (DP)

≡ Menu

Liste des bases de données

AJOUTER UNE BDD

User	Name	Host	Port	Type	Container	Nouvelle sauvegarde	Sauvegarde automatique
Momo	momo_data	localhost	3002	postgres	data_container1	<button>+ Backup</button>	
admin	myproject_data2	localhost	1234	sql	container_2	<button>+ Backup</button>	
test	test	localhost	1233	postgres	container_3	<button>+ Backup</button>	

Nouvelle base de données

Ajoutez les informations de connexion à une base de données existante.

type *

user *

name *

mot de passe *

host *

port *

container *

ANNULER AJOUTER LA CONNEXION

Port Type

3002 postgres

Définir une sauvegarde automatique

Ajoutez les informations pour la sauvegarde automatique.

Choisir une fréquence

Toutes les minutes

Toutes les heures

Tous les jours (à 1:00)

Tous les mois (le premier de chaque mois à 1:00)

ANNULER AJOUTER LA SAUVEGARDE AUTOMATIQUE

DOSSIER PROFESSIONNEL (DP)

Liste des backups

Nom de la BDD	Type	Path	Date de sauvegarde	Restaurer cette version	Supprimer
momo_data	postgres	Sauvegardes/SauvegardesPosteGres/savebase_postgres_25_24_04_14_35.sql	24/04/2025 16:35	<button>Restaurer cette version</button>	<button>Supprimer</button>
test	postgres	Sauvegardes/SauvegardesPosteGres/savebase_postgres_25_24_04_14_47.sql	24/04/2025 16:47	<button>Restaurer cette version</button>	<button>Supprimer</button>
test	postgres	Sauvegardes/SauvegardesPosteGres/savebase_postgres_25_24_04_14_48.sql	24/04/2025 16:48	<button>Restaurer cette version</button>	<button>Supprimer</button>

Page 1 sur 1

Ce projet couvre les compétences :

- Installer et configurer son environnement de travail en fonction du projet
- Développer des interfaces utilisateur
- Développer des composants métiers

2. Précisez les moyens utilisés :

Visual Studio Code : environnement de développement

Git et Github : versionning

Languages : javascript, sql

Framework : pour le back-end utilisation du framework javascript Fastify.js et pour le front-end, utilisation de React.js

Utilisation de docker pour la base de données. Utilisation d'une image d'adminer (application web permettant de gérer des bases de données via une interface graphique.) avec postgres, un Système de Gestion de Bases de Données Relationnelles (SGBDR) .

3. Avec qui avez-vous travaillé ?

J'ai travaillé seule sur ce projet

4. Contexte

DOSSIER PROFESSIONNEL (DP)

Nom de l'entreprise, organisme ou association ➔ *Morgane Maréchal*

Chantier, atelier, service ➔ *Safebase*

Période d'exercice ➔ Du : *22/09/2024* au : *02/12/2024*

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Activité-type 1 Développer une application sécurisée

Exemple n° 2 - Moonshot - Projet de plateforme interculturel

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Le projet est la création d'une plateforme multiculturelle qui propose diverses activités pour que les publics se rencontrent.

Ce projet de groupe prend place au sein d'un cursus de formation à LaPlateforme.

Le back-end est fait avec Strapi. Strapi est un cms headless open source facile, personnalisable et flexible pour les développeurs. headless signifie qu'il ne gère que la partie back-end de l'application.

Dans ce projet j'ai participé au développement de plusieurs fonctionnalités, au design sur figma et à l'élaboration du cahier des charges.

Par exemple: je me suis chargée de la page profil. J'ai également fait la partie graphique avec Tailwind de plusieurs fonctionnalité comme les formulaires d'enregistrement, de connexion et de contact.

Outils de collaboration

Le travail se fait en groupe à l'aide d'outils de versionning : Git et Github



Github est une plateforme de versionning qui permet de centraliser un projet et d'y participer à plusieurs. Le répertoire a été créé sous le nom de Maxaboom (le nom de la boutique en ligne). Qui a ensuite été cloné en local par les membres du groupe.



Git est un logiciel de gestion de version décentralisé. C'est le plus populaire parmi les développeurs. Il permet de communiquer entre la version local de l'application et la version remote qui est centralisée sur github.

Pour chaque feature du projet nous avons créé des branches afin que chacun puisse travailler sur un élément différent sans empiéter sur le travail des autres. Je réalise régulièrement des commits pour ne pas perdre mon travail en cours. Quand une feature est terminée, j'envoie une pull request sur la branche en remote qui est ensuite validée par un membre de l'équipe. Le code est ainsi mutuellement vérifié et cela permet d'éviter les conflits de versioning lors du merge entre les versions.

DOSSIER PROFESSIONNEL (DP)

L'installation du projet est documenté dans le répertoire Github :

The screenshot shows a GitHub README file with the following content:

- Getting Started**
- Requirements**
Ensure you have the following dependencies installed on your system:
- Installation**
 - Clone the moonshot repository:

```
git clone https://github.com/tchessi-pre/moonshot
```
 - Change to the project directory:

```
cd moonshot
```
 - Install the dependencies:

```
cd back && npm install
```



```
cd front && npm install
```
- Running moonshot**
Use the following command to run moonshot:
 - Back:

```
npm run develop
```
 - Front:

```
npm run dev
```

Une fois le projet installé, les collaborateurs travaillent sur des branches locales afin de ne pas empiéter sur le travail des autres.

Les modifications et nouvelles features sont ensuite envoyées (push) sur des branches distantes

avant de créer des demandes pour ajouter ces modifications au projet commun (pull request).

Le chef de projet s'assure ensuite d'incorporer les modifications sans conflits.

DOSSIER PROFESSIONNEL (DP)

Les développeurs s'assure ensuite d'avoir leur master local à jour en faisant des pull réguliers.

🔥 page Create event	20478cd	🔗	🔗
catherine-tranchand committed on Jul 15, 2024			
design for informations private page	932b387	🔗	🔗
morgane-marechal committed on Jul 15, 2024			
Merge pull request #40 from tchessi-pre/tch-modif-events-pages	02a0f07	🔗	🔗
tchessi-pre authored on Jul 15, 2024			
Added cuisine page merge	fc2bd45	🔗	🔗
Tchessy committed on Jul 15, 2024			
Merge pull request #39 from tchessi-pre/cuisine	c9a7f66	🔗	🔗
tchessi-pre authored on Jul 15, 2024			
ajout de la page Cuisine	915ca0e	🔗	🔗
j-eGallo committed on Jul 15, 2024			
Ajout de la page musique	d704d71	🔗	🔗
j-eGallo committed on Jul 15, 2024			
Merge branch 'main' of https://github.com/tchessi-pre/moonshot into informations_private	e4c43d8	🔗	🔗
morgane-marechal committed on Jul 15, 2024			
last change	f73c8cd	🔗	🔗
morgane-marechal committed on Jul 15, 2024			

Il est aussi possible sur Github d'assurer un suivi de l'avancement du projet et d'assigner des tâches à chaque développeur.

The screenshot shows a GitHub project board with the following details:

- Assignees:**
 - axel-vair (3 tasks)
 - catherine-tranchand (2 tasks)
 - j-eGallo (1 task)
 - jean-emmanuel-gallo (1 task)
 - rabaou-ibrahim (1 task)
- Filter:** Filter by keyword or by field
- Columns:** Title, Status
- Tasks:**
 - P0:** moonshot-front #4 (Done)
 - P0:** moonshot - Création et Gestion de Profils #3 (Done)
 - P1:** moonshot-back - Creation des entités #5 (In Progress)
 - No Priority:** Création de page forgotPassword et resetPassword #6 (In Progress)
 - No Priority:** Création de la page des événements #14 (In Progress)
 - No Priority:** Configuration de la base de donnée #10 (In Progress)
 - No Priority:** Creation de la page de creation d'un événement #31 (Done)

Objectifs globaux de la plateforme web

Cette plateforme a pour but de mettre en relation les utilisateurs autour de projets culturels dans la ville de Marseille. Ces derniers peuvent créer des évènements sur la plateforme et communiquer avec les autres utilisateurs.

Les évènements ont pour but de mettre en avant et de permettre à des communautés culturels de

DOSSIER PROFESSIONNEL (DP)

se rassembler autour de divers intérêts (exemple : café linguistiques, cours de danse, cours de cuisine, ...).

Public Cible

Toutes personnes de plus de 13 ans (âge minimum pour s'inscrire) qui désire rencontrer d'autres personnes autour d'intérêts culturels.

Pour le moment les rendez-vous se situe à Marseille.

Fonctionnalités principales

Création d'un compte

Création d'une page profil

Créer et publier des évènements

S'inscrire à un évènement

Créer et publier des articles

Pouvoir poster des images et des vidéos

Liker, commenter et partager des articles

Appartenir à un ou plusieurs groupes centrés sur des centres d'intérêts

Un tchat pour communiquer directement avec d'autre utilisateurs

Limites du projet (ce qui ne sera pas inclus)

La géolocalisation de l'utilisateur et des évènements n'est pas incluse pour le moment.

La version application mobile ne sera pas développée en raison de contraintes de temps.

L'accessibilité ne pourra pas non plus être développée à son maximum.

Ce projet couvre les compétences :

- **Installer et configurer son environnement de travail en fonction du projet**
- **Développer des interfaces utilisateur**
- **Développer des composants métiers**
- **Contribuer à la gestion d'un projet informatique**

2. Précisez les moyens utilisés :

Visual Studio Code : comment environnement de développement

Git et Github : versionning

Languages : HTML, CSS, javascript

Utilisation du framework [Next.js](#) et de Strapi pour gérer le backend. Utilisation de Tailwind pour le style.

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

J'ai fait ce projet de groupe avec :

Tchessi Pres (lead)
Jean-Emmanuel Gallo
Catherine Tranchand
Axel Vaire
Rabaou Ibrahim

4. Contexte

Nom de l'entreprise, organisme ou association ➔ **La Plateforme**

Chantier, atelier, service ➔ **Moonshot - Pangea**

Période d'exercice ➔ Du : **01/07/2024** au : **01/09/2024**

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Activité-type 2

Concevoir et développer une application sécurisée organisée en couches

Exemple n° 1 - Moonshot - Projet de plateforme interculturel

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Visual Studio Code : environnement de développement

Git et Github : versionning

Languages : javascript, sql

Framework : utilisation de Next.js

Utilisation de docker pour la base de données. Utilisation d'une image d'adminer (application web permettant de gérer des bases de données via une interface graphique.) avec postgres, un Système de Gestion de Bases de Données Relationnelles (SGBDR).

Dans cette partie, on va détailler le maquettage et la charte graphique :

Charte Graphique

Le logo :



Couleurs utilisées :

6B7280	111827	EA580C
374151	111827	3B82F6

Police utilisée :

Inter Inter
Inter Inter

DOSSIER PROFESSIONNEL (DP)

Maquettes basse fidélité :

Inscription

PANGEA

Email
Nom
Prénom
Choisir un mot de passe
Conformer mot de passe

You avez déjà un compte ?

Inscription

Connexion

PANGEA

Email
Choisir un mot de passe

J'ai oublié mon mot de passe

Inscription

Frame 1

PANGEA



Loreum ipsum dolor sit amet, consectetur adipiscing elit. Etiam neque sed metus sed gravida. Nullam est enim, neugit ac mattis nec, dignissim a dolor. Integer condimentum tempor, mauris vel ultrices enim convallis sagittis. Pedentesque ex ut eu ex suspendisse tellus quis fringilla. Duis quis tristique et, euismod vulputate tellus. Nulla ornare metus ac metus sapien, non imperatur arcu vehicula. Donec rhoncus, mauris ut efficit finibus, nulli nunc consequat odio, fringilla gravida tenter diam estatue ipsa. Phasellus posuere, magna sed varius metus, velit etiam natus odio, ut lectura diam ipsum fermentum turpis.

Événements à venir

Meilleurs restaurants

Cours de langue

Mon espace

PANGEA



Mes dernières soirées

Mes restaurants favoris

Les meilleurs restaurants

DOSSIER PROFESSIONNEL (DP)

Description détaillée des fonctionnalités

Inscription et authentification des utilisateurs

Register Page

INSCRIPTION

Nom
Prénom
Email
Mot de passe
Mot de passe
Confirmez votre mot de passe

S'INSCRIRE

Connexion

Connexion :

Login Page

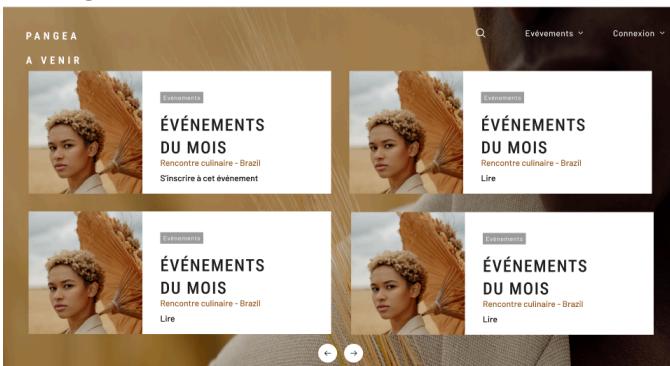
CONNEXION

Email
Mot de passe

SE CONNECTER

Mot de passe oublié?
Pas de compte? S'enregistrer

Affichage des évènements :



DOSSIER PROFESSIONNEL (DP)

Profil utilisateur :

The screenshot shows a user profile for "PANGEA". The profile picture is a circular portrait of a man. Below it, the name "John Doe" is displayed, along with language skills: "Parle : Anglais, Italien, Espagnol" and a location pin indicating "Marseille, France". Under "Mes spécialités", there are four categories: Capoeira, Cuisine Italienne, Calligraphie Arabe, and Médecine Chinoise. In the "Groupes dans lesquels je suis" section, there are four small circular icons representing different groups: Capoeira Marseille, Témoins de Jéhovah, Lycée Fénelon, and Ecole et Université.

Affichage des évènements sur la page principale et possibilité de s'inscrire à ces derniers

Ce projet couvre les compétences :

- Analyser les besoins et maquetter une applications
- Définir l'architecture logicielle d'une applications
- Concevoir et mettre en place une base de données relationnelle
- Développer des composants d'accès aux données SQL (la compétence NoSQL n'est pas présente ici)

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Visual Studio Code : comment environnement de développement

Git et Github : versionning

Languages : HTML, CSS, javascript

Utilisation du framework [Next.js](#) et de Strapi pour gérer le backend. Utilisation de Tailwind pour le style.

3. Avec qui avez-vous travaillé ?

J'ai fait ce projet de groupe avec :

Tchessi Pres (lead)

Jean-Emmanuel Gallo

Catherine Tranchand

Axel Vaire

Rabaou Ibrahim

4. Contexte

Nom de l'entreprise, organisme ou association ▶

La Plateforme

Chantier, atelier, service ▶

Moonshot - Pangea

Période d'exercice

▶ Du : *01/07/2024*

au : *01/09/2024*

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Activité-type 2 Concevoir et développer une application sécurisée organisée en couches

Exemple n° 1 - La boutique en ligne - Maxaboom

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Trello pour l'organisation des tâches

Trello est un outil de gestion de projet en ligne. Il repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignable à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement. Les cartes sont rédigées en anglais.

The screenshot shows a Trello board titled "MaxaBoom Design Concept". The board is divided into three main sections: "To Do", "Doing", and "Done". Each section has a sub-section for "Current Issues" and "Solved Issues". A central smartphone card displays a mobile application interface. The background of the board features a scenic view of green hills and mountains.

Les tâches sont réparties en trois colonnes, celles qui sont à faire, celles qui sont en cours et celles qui sont terminées. Cette méthode permet de suivre plus facilement l'état d'avancement du projet en cours. Chacun peut rédiger de nouvelles étiquettes ou s'en attribuer selon les besoins du projet.

DOSSIER PROFESSIONNEL (DP)

Figma pour le travail commun sur le design



Figma est un éditeur de graphiques vectoriels et un outil de prototypage. C'est un outils collaboratif qui permet à plusieurs personnes de travailler sur un même projet. Chaque utilisateur peut travailler sur une page ou un composant du projet en suivant la charte graphique.

Descriptif fonctionnel

Le projet de boutique en ligne comprend une page d'accueil attractive avec plusieurs sections dont une mise en avant des produits phares sur la page d'accueil.

Le design est contemporain et respecte la charte graphique de l'entreprise. Le site est responsive.

Le site doit comprendre une barre de recherche avec autocomplétion.

La boutique présente tous les produits avec la possibilité de les filtrer par catégorie / sous-catégories sans rechargement de page.

Un clic sur chaque produit renvoie à une page "détail" complète générée dynamiquement avec le nom, l'image, le prix, la description et un bouton ajouter au panier.

L'utilisateur peut se créer un compte avec un module Inscription / Connexion en asynchrone avec javascript.

L'administrateur possède son propre tableau de bord ce qui permet une gestion des produits à l'aide de back office (Ajout / Suppression /Modifications de produits, stocks...), une gestion des catégories et des sous catégories de produits (Ajout / Suppression / Modifications...)

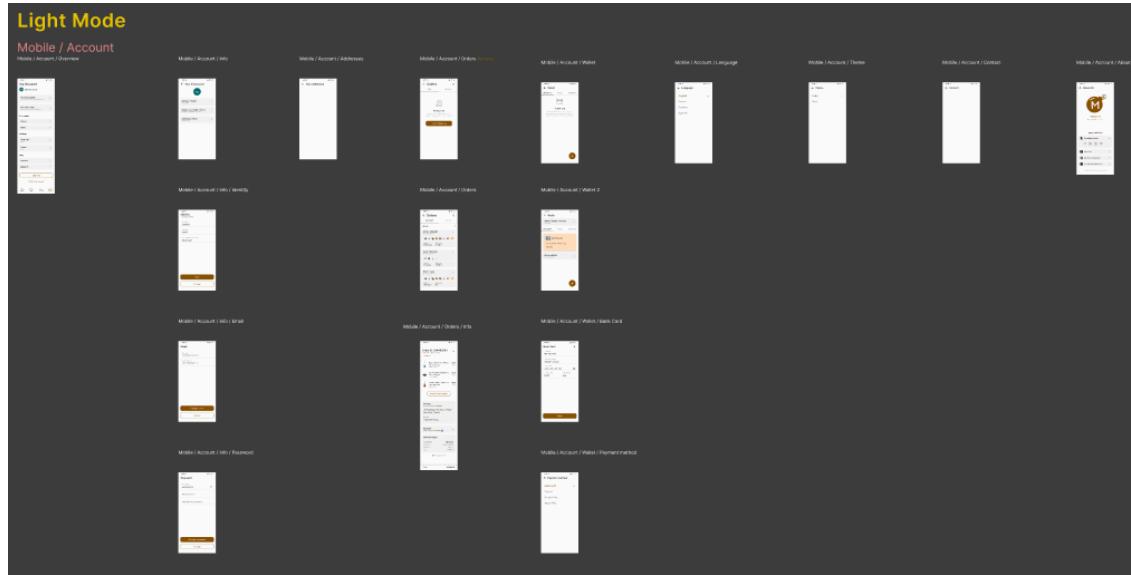
La boutique comprend aussi un système de validation du panier avec ou non une vraie solution de paiement.

Arborescence du projet

- Page d'accueil (home)
- Page de connexion
- Page d'inscription
- Page recherche pour trouver un produit spécifique
- Page pour chaque produit
- Page mon compte (account)
- Page addresses
- Page panier (cart)
- Page paiement

DOSSIER PROFESSIONNEL (DP)

L'ensemble de l'arborescence du projet a été maquetté sur Figma avec la version mobile et la version desktop.



Spécification technique et conception :

Charte graphique

Une charte graphique web est un ensemble de règles et de normes graphiques qui constituent l'identité visuelle d'un site web. La charte graphique web contient des règles d'utilisation de tous les éléments graphiques, et représente donc un élément essentiel du site.

La charte graphique a été explicitée sur le Figma afin que chaque membre de l'équipe puisse s'y référer pour créer des éléments graphiques de la maquette.

Elle comprend plusieurs éléments :

DOSSIER PROFESSIONNEL (DP)



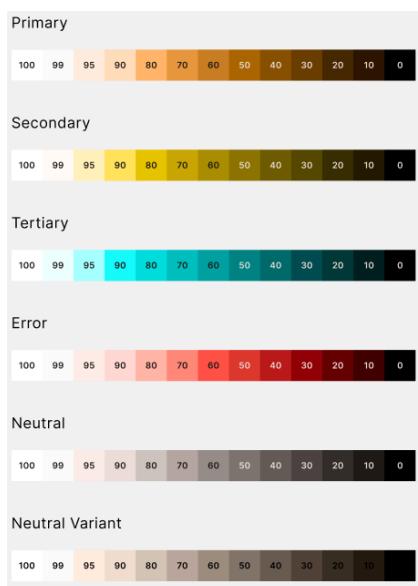
Le logo : la charte doit contenir le logo de l'entreprise ainsi que ses variantes, ses références couleurs et ses différentes tailles.

La typographie : la charte doit contenir la liste de toutes les polices utilisées, et ce pour chaque niveau de titre.

Les polices utilisées pour le site sont Mulish et Roboto.

Typography / Text 1	
Extra Small Font size: 12px Line height: 16px	Large Font size: 18px Line height: 28px
The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.	The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.
Small Font size: 14px Line height: 20px	Extra Large 2 Font size: 24px Line height: 34px
The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.	The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.
Base Font size: 16px Line height: 24px	Extra Large 3 Font size: 28px Line height: 38px
The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.	The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

Les couleurs : la charte doit contenir les références des couleurs utilisées ;



DOSSIER PROFESSIONNEL (DP)

Colors / Main

Light Scheme

Primary	On Primary	Primary Container	On Primary Container
Secondary	On Secondary	Secondary Container	On Secondary Container
Tertiary	On Tertiary	Tertiary Container	On Tertiary Container
Error	On Error	Error Container	On Error Container
Background	On Background	Surface	On Surface
Outline		Surface-Variant	On Surface-Variant

Primary Inverse Inverse Surface Inverse On Surface

Dark Scheme

Primary	On Primary	Primary Container	On Primary Container
Secondary	On Secondary	Secondary Container	On Secondary Container
Tertiary	On Tertiary	Tertiary Container	On Tertiary Container
Error	On Error	Error Container	On Error Container
Background	On Background	Surface	On Surface
Outline		Surface-Variant	On Surface-Variant

Primary Inverse Inverse Surface Inverse On Surface

Les boutons et les icônes :

Les icônes et les boutons ont au préalable été créés sur Figma afin d'avoir une idée précise du rendu final.

Ci-dessous voici une partie des boutons utilisés.

DOSSIER PROFESSIONNEL (DP)



Typography / Overview

Heading

Leading text

Paragraph

Call to action

Learn more

Inter

Aa abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ 123456790 ?!()[]*&%\$#@~

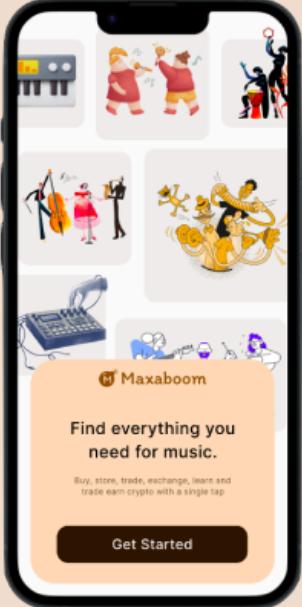
Thin
Extra Light
Light
Normal
Medium
Semi Bold
Bold
Extra Bold
Black

github.com/abraham-ukachi/boutique-en-ligne



The User interface (UI), in the industrial design field of human-computer interaction, is the space where interactions between humans and machines occur.

The international Typographic Style also known as the Sales Styles is a graphic design style that emerged in Russia, the Netherlands, and Germany in 1920s and was developed by designers in Switzerland during the 1960s. It emphasizes cleanliness, readability and objectivity.



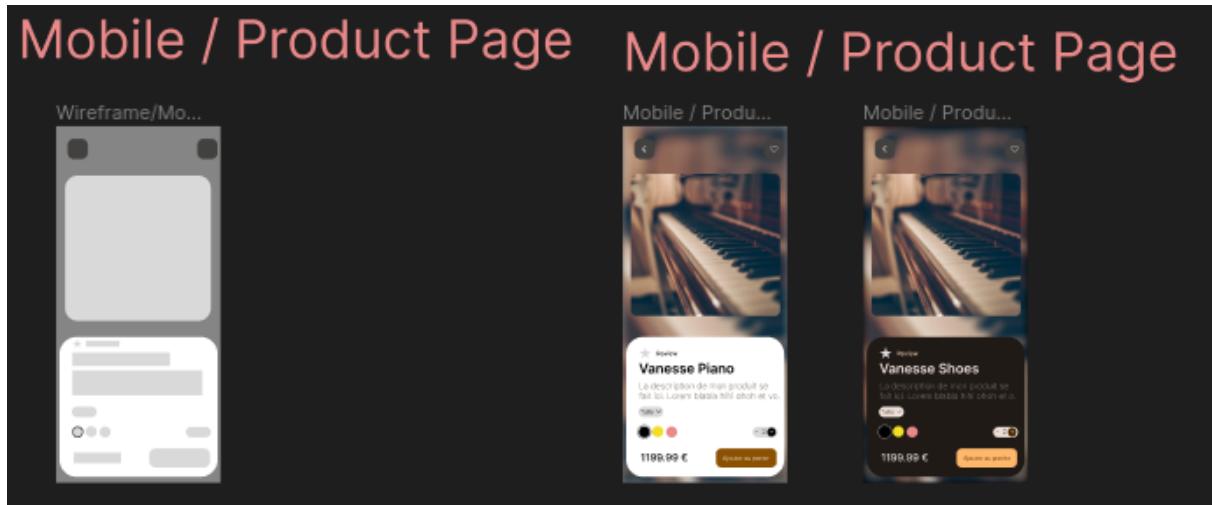
Find everything you need for music.
Buy, store, trade, exchange, learn and trade earn crypto with a single tap

Get Started

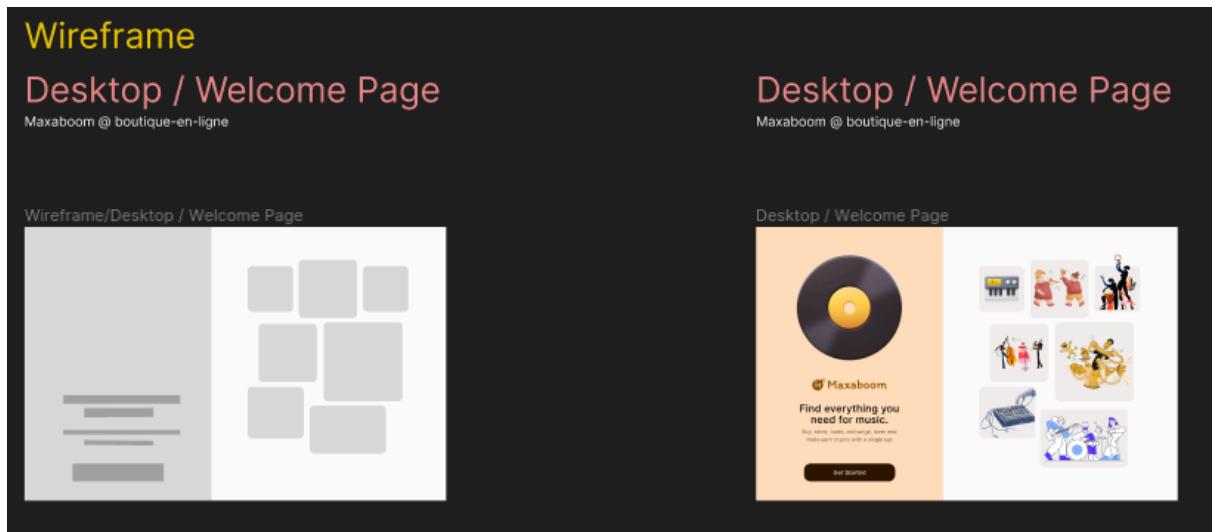
Maquette

DOSSIER PROFESSIONNEL (DP)

Pour chaque page, nous avons fait sur le figma une maquette basse fidélité et une maquette haute fidélité.



Nous avons fait cela aussi bien pour la version mobile (ci-dessus) que desktop (ci-dessous).



Nous avons également créé des composants réutilisables et détaillés à assembler sur les différentes maquettes (voir ci-dessous).

DOSSIER PROFESSIONNEL (DP)

Card/Product



Card/Product/Desktop



Dans un premier temps, toute l'équipe à travaillé sur le figma. Nous avons créé des wireframes (maquettes) basse fidélité et haute fidélité.

Exemple :

The image shows two side-by-side comparisons of wireframes and final designs for a user interface.

Top Left: A wireframe labeled "Wireframe" for a "Desktop / Account Page". It features a sidebar with a user profile icon and several menu items, followed by a main content area with a grid of horizontal bars representing content blocks.

Top Right: The corresponding high-fidelity design labeled "Desktop / Account Page". It includes a header with the brand name "Maxaboom", a sidebar with a user profile picture and sections for "Votre profil", "Informations de paiement", and "Historique d'achat", and a main content area with sections for "Mes préférences", "Mes méthodes de paiement", and "Mes adresses".

Bottom Left: A wireframe labeled "Wireframe" for a "Desktop / Order Success" page. It shows a sidebar with a user profile icon and a main content area with a large green checkmark icon and the message "La commande a été enregistrée !".

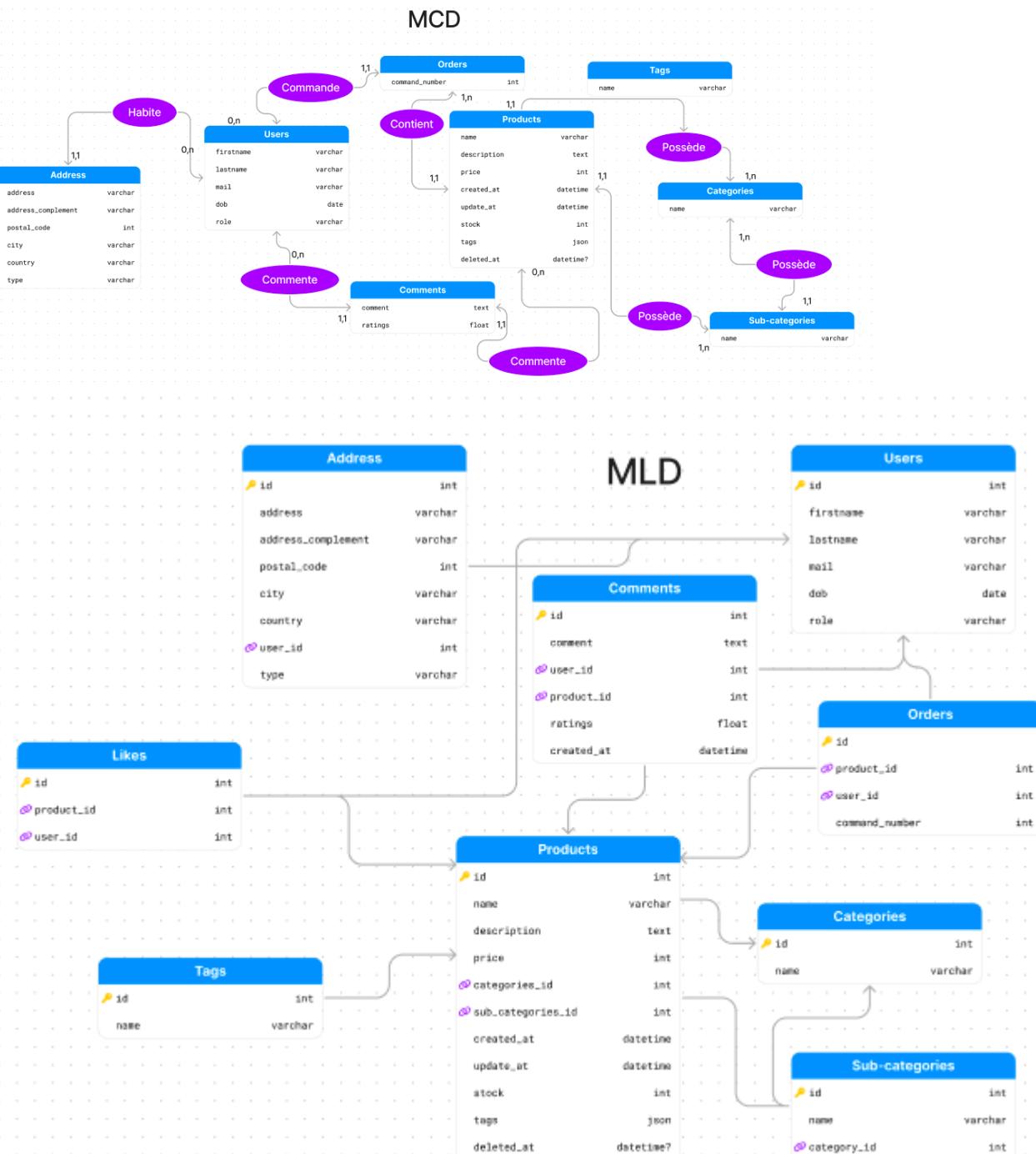
Bottom Right: The corresponding high-fidelity design labeled "Desktop / Order Success". It features a header with the brand name "Maxaboom", a sidebar with sections for "Résumé de la commande", "Informations du destinataire", and "Informations de sécurité", and a main content area with a large green checkmark icon and the message "La commande a été enregistrée !".

DOSSIER PROFESSIONNEL (DP)

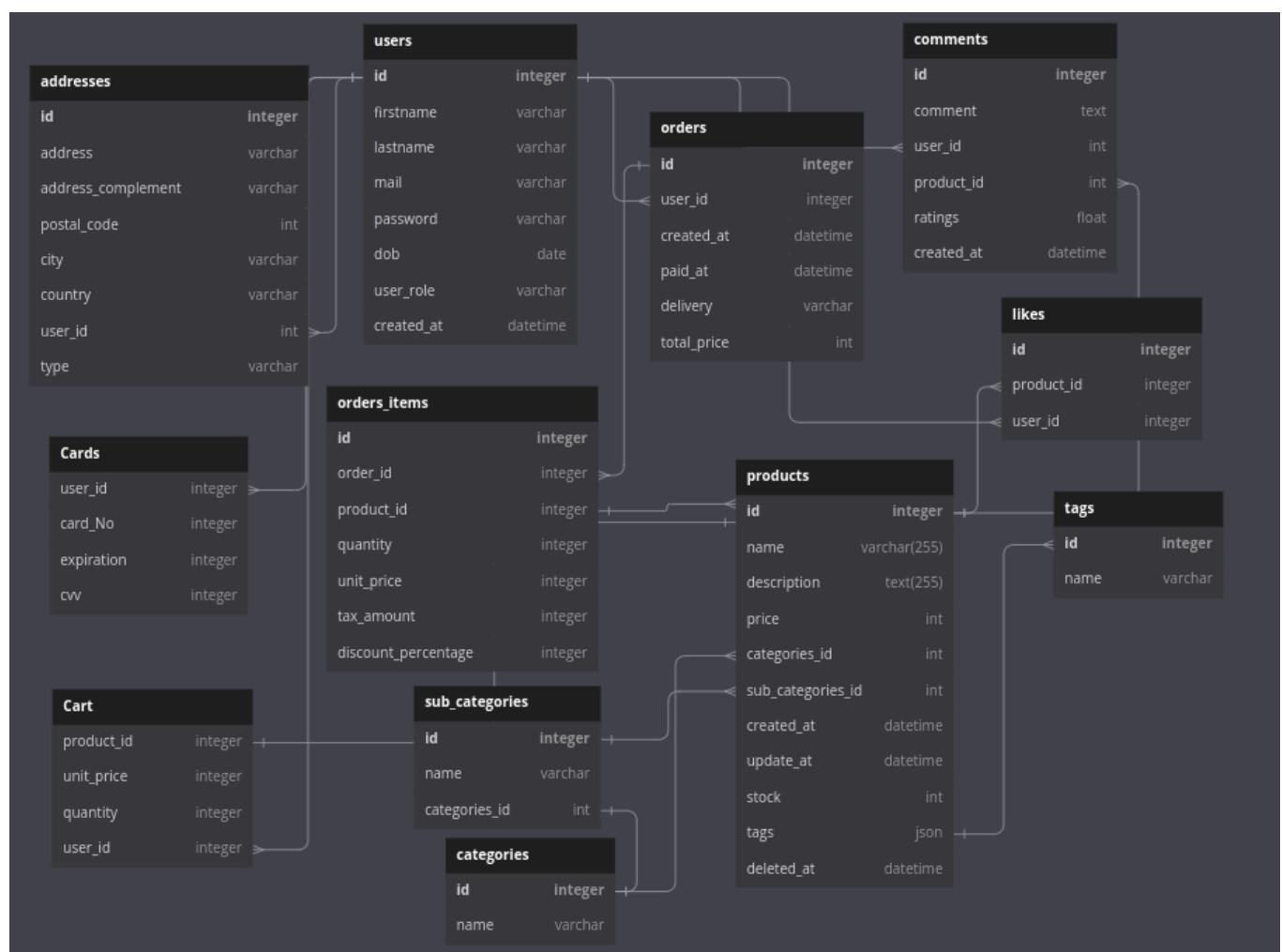
Boutique en ligne - Back end

Très vite, je me suis orientée sur la partie back-end.

J'ai créé la première version du MCD, MLD, MPD. Et j'ai commencé à représenter le MCD et le MLD sur figma et le MPD sur dbDiagram.io.



DOSSIER PROFESSIONNEL (DP)



Par la suite, je me suis surtout occupée de la programmation de la partie Admin en PHP avec la POO et en javascript avec des requêtes asynchrones.

En utilisant l'architecture MVC, j'ai créé les classes pour traiter les données nécessaires à la gestion de la partie admin, ainsi que le controller et les views afin d'afficher les informations aux utilisateurs.. Pour cela, j'ai utilisé les quatre opérations de base CRUD (create, read, update, delete).

J'utilise javascript et l'api fetch afin d'exécuter les requêtes de manière asynchrone sans recharge de page.

DOSSIER PROFESSIONNEL (DP)

The screenshot displays a user interface for managing a music store's inventory and categories. On the left, a sidebar titled 'Admin' includes a 'Panel d'administration' section with links for 'Gestion des utilisateurs', 'Gestion des produits', and 'Gestion des catégories et des sous-catégories'. The main content area is titled 'Les catégories' and lists various musical instruments with edit and delete options. A prominent orange '+' button is located in the center of the category list. The right sidebar contains a form for creating new sub-categories, including a text input field and an 'Ajouter Une Nouvelle Catégorie' button.

Dans la partie admin, il est possible de gérer les utilisateurs, les produits, les catégories et les sous-catégories.

Pour la partie utilisateurs :

On peut modifier ou supprimer des utilisateurs, leur changer de rôle en leur donnant les privilèges d'un administrateur.

Pour la partie produits :

On peut ajouter des produits avec obligatoirement une catégorie, une sous-catégorie, et une image pour l'illustrer. On peut également supprimer et modifier un produit déjà existant. Les produits supprimés restent en base de données avec la colonne deleted_at remplie.

Pour la partie catégories et sous-catégories, on peut ajouter des catégories, des sous-catégories ou les supprimer.

Ce projet couvre les compétences :

- Analyser les besoins et maquetter une applications
- Définir l'architecture logicielle d'une applications
- Concevoir et mettre en place une base de données relationnelle
- Développer des composants d'accès aux données SQL (la compétence NoSQL n'est pas présente ici)

2. Précisez les moyens utilisés :

Versioning avec Git et Github

HTML, CSS, SQL, PHP, Javascript

Utilisation de la POO

Architecture MVC

Dépendances PHP avec composer : altorouter, autoloader

DOSSIER PROFESSIONNEL (DP)

Responsiv design
Maquettage avec Figma
Visualisation de la base de données avec dbDiagram.io
Création d'une base de donnée commune avec un fichier SQL
Trello pour l'organisation d'équipe
Méthode agile : Kanban

3. Avec qui avez-vous travaillé ?

Abraham Ukachi
Axel Vair
Catherine Tranchand

4. Contexte

Nom de l'entreprise, organisme ou association ➔ **La Plateforme**
Chantier, atelier, service ➔ **Boutique en ligne**
Période d'exercice ➔ Du : **05/04/2023** au : **05/05/2023**

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Activité-type 2 Concevoir et développer une application sécurisée organisée en couches

Exemple n° 3 - SIGB (Système informatique de gestion des bibliothèques)

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Ce projet a pour objectif de créer un site web pour la gestion d'une bibliothèque.

Il est composé d'une partie SIGB (système informatique de gestion des bibliothèques) et une partie portail pour consulter les documents qui sont disponibles en bibliothèque.

La partie gestion de bibliothèque se fait uniquement via l'interface web. Avec un compte administrateur. Cette partie permet de :

- Enregistrer des notices
- Enregistrer des livres associés aux notices
- Enregistrer des auteurs
- Enregistrer des catégories de livres.
- Enregistrer des utilisateurs avec différents niveaux de droits
- Gérer et suivre les emprunts des utilisateurs.

Ces différentes fonctions nécessitent l'utilisation des fonctionnalités CRUD (create, read, update, delete) des données.

La partie portail se concentre sur la consultation et les emprunts des usagers de la bibliothèque. Cette partie est accessible par un navigateur et par une application mobile.

La partie navigateur :

- Permet de consulter les livres de la bibliothèque
- Permet à l'utilisateur de voir son compte (nombre d'emprunts, livre à rendre, réservation,...)
- Il permet de réserver un livre
- De faire des recherches selon l'auteur ou la catégorie de livres.

La partie mobile :

- En plus des fonctionnalités précédentes, permet de scanner le code ISBN d'un livre pour voir s'il est enregistré en bibliothèque et disponible.

Pour la charte graphique :

Utilisation des composants de Material UI en respectant la charte graphique suivante.

La charte graphique est commune à la partie réservée aux administrateurs et à la partie réservée aux usagers de la bibliothèque.

DOSSIER PROFESSIONNEL (DP)

Charte Graphique

1. Logo



2. Palette de Couleurs

Rôle	Couleur	Code
Primaire	Bleu profond	#2A5C8D
Secondaire	Vert sage	#4B8F8C
Accent Or doux		#A98E36
Arrière-plan	Blanc cassé	#F5F5F0
Texte	Noir doux	#333333

Pourquoi ces couleurs ?

Bleu : Évoque la confiance, le savoir (parfait pour une bibliothèque).

Vert : Apporte de la sérénité, lien avec la nature (livres anciens/bois).

Or : Rappelle les tranches de livres dorées, ajoute du luxe.



une mauvaise perception des couleurs (daltonisme, ...):

Site ressource pour vérifier l'accessibilité des couleurs par rapport au contraste :

<https://www.skynettetechnologies.com/color-contrast-checker>

On peut ainsi tester les couleurs définies plus haut et voir

A screenshot of a web-based tool for WCAG compliance. It shows two color swatches: a white circle labeled "Foreground Color" with hex code #ffffff and a dark blue circle labeled "Background Color" with hex code #2A5C8D. Below these are sections for "WCAG Compliance Results" and "Color Contrast". The "Color Contrast" section displays a score of 6.97 and a "Pass" status. Under "Color Contrast", there are three green boxes with checkmarks: "Small Text", "Large Text", and "UI Components". At the top right, there are buttons for "AA" and "AAA" accessibility levels. On the left, there's a "Swap Colors" button and some small text about sample text sizes.

3. Typographie

Utilisation de la Google font Marriweather

Titres : Merriweather (serif) → Élégant et lisible.

DOSSIER PROFESSIONNEL (DP)

Bold 700

Whereas recognition of the inherent dignity

Bold 700 Italic

Whereas recognition of the inherent dignity

font-family: 'Merriweather', serif;

font-weight: 700;

Corps de texte : Open Sans (sans-serif) → Moderne et accessible.

Éléments UI (composants de MaterialUI)

Boutons :

Bordure arrondie (border-radius: 8px).

Ombre portée légère (box-shadow: 0 2px 4px rgba(0,0,0,0.1)).

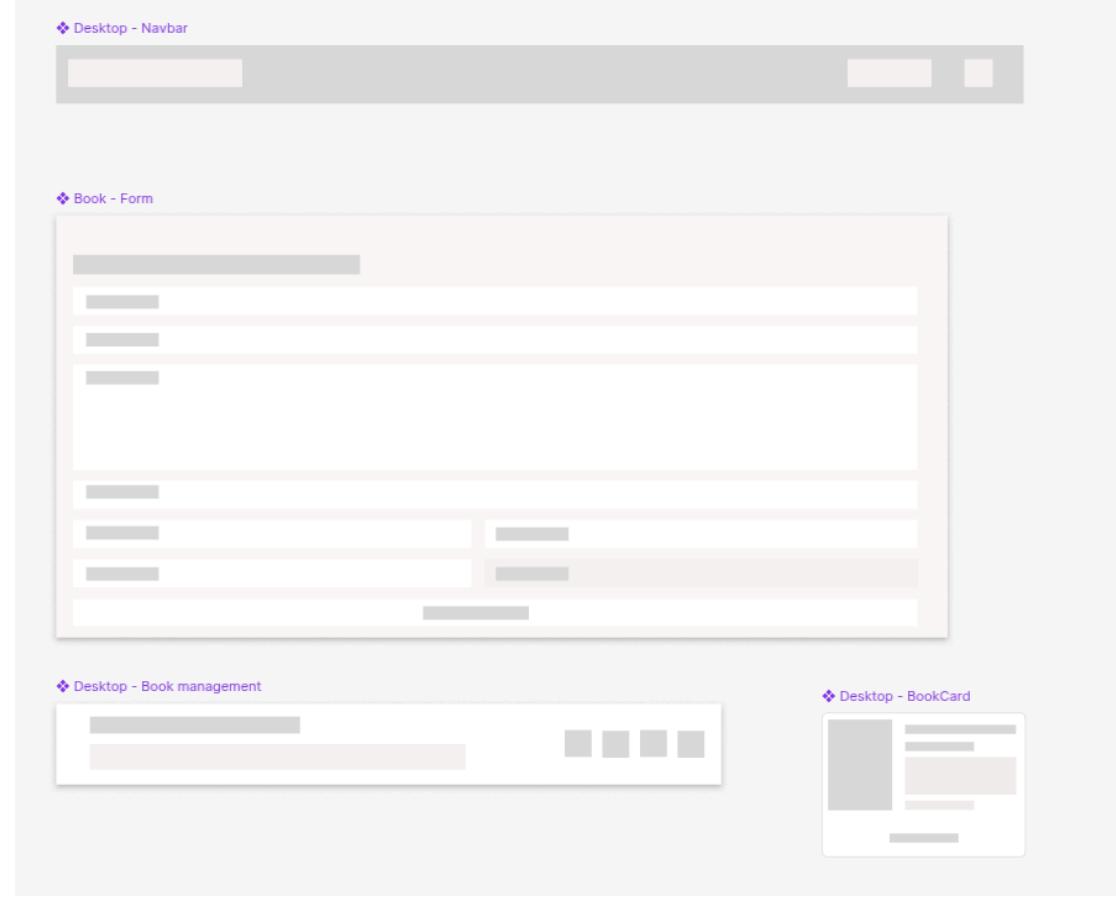
Cartes de livres :

Fond blanc, ombre subtile, bordure 1px solid #E0E0E0.

Titre et informations importantes en bleu.

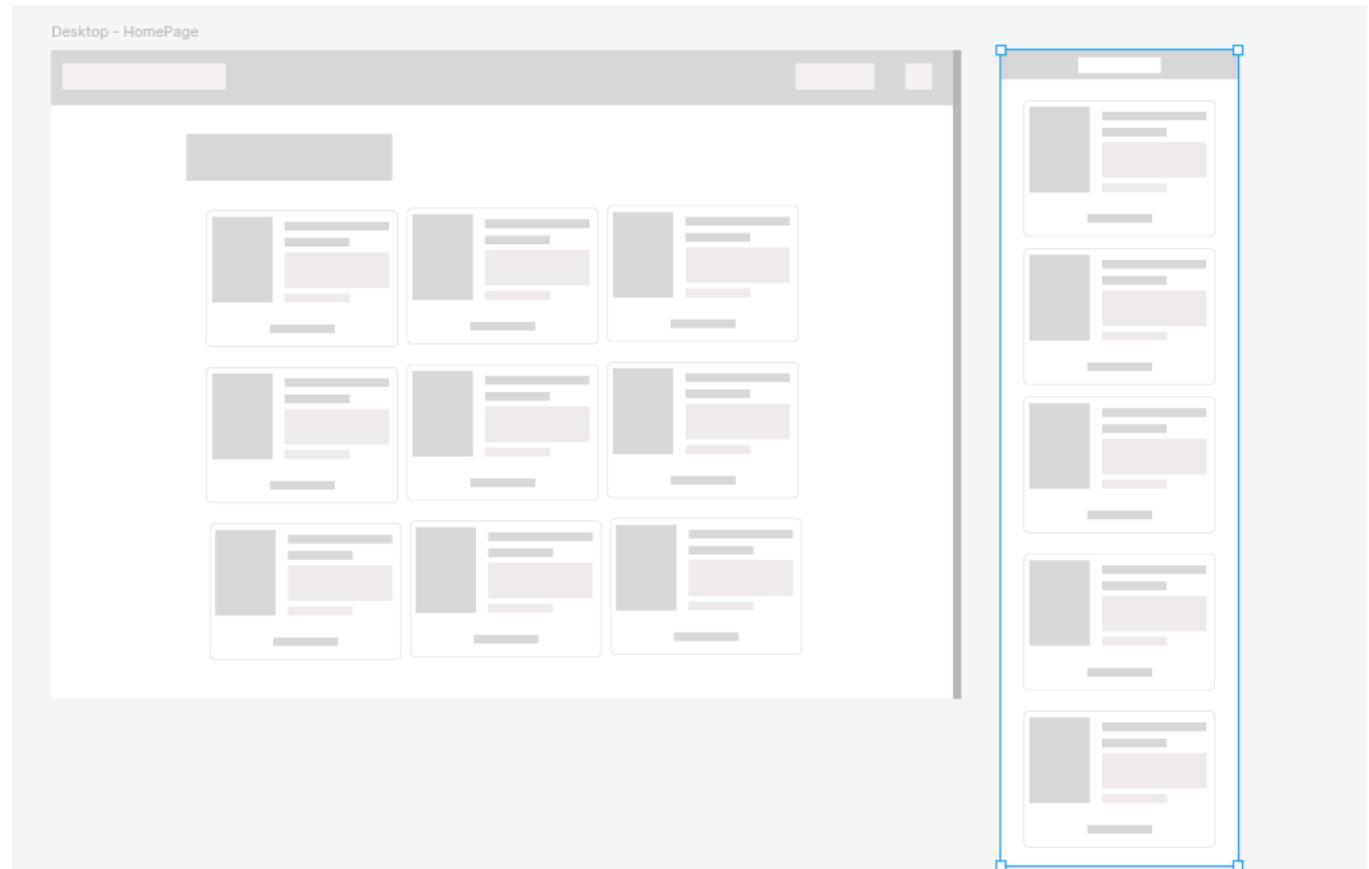
L'application a été maquettée sur figma :

Voici une maquette base fidélité de quelques composants:



DOSSIER PROFESSIONNEL (DP)

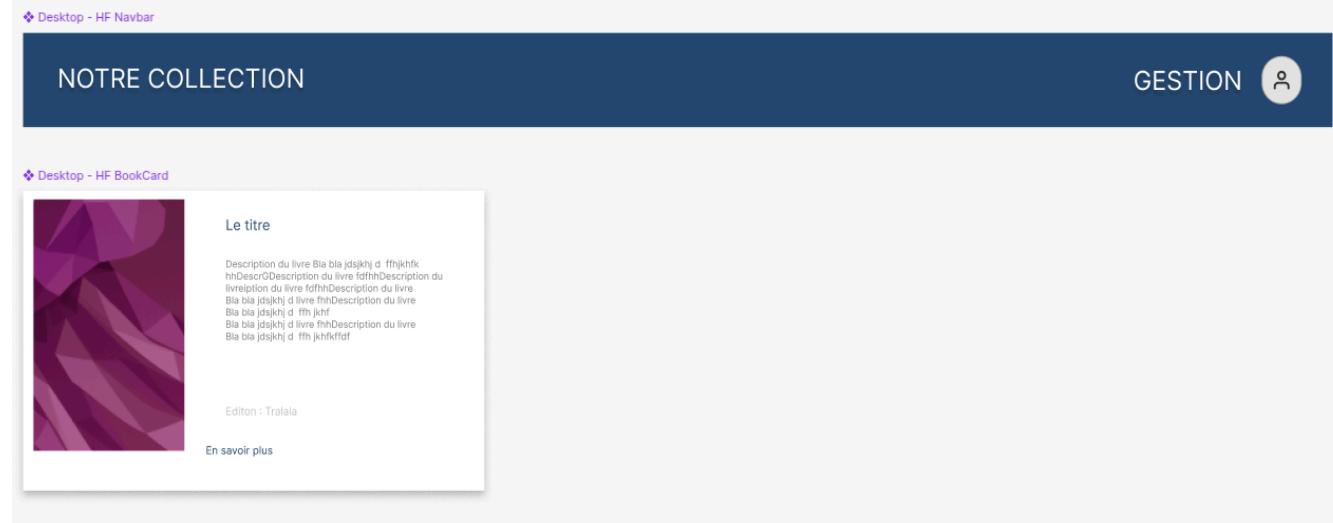
Maquettage basse fidélité de l'interface :



DOSSIER PROFESSIONNEL (DP)

OBJ

Maquettage haute fidélité de composants :



Architecture logicielle :

C'est un logiciel 3 tiers, ou 3 couches décomposé comme suit :

- couche de présentation avec la View en React;
 - couche de traitement ; Logique métier avec le Model, Controleur dans Adonis.js
 - couche d'accès aux données ; avec la base de données Postgres consultable sur Adminer et l'utilisation de Redis, un SGBD noSql, pour envoyer des notifications.

La couche de présentation est séparée des autres couches et communique avec la logique métier via une API REST.

API REST : L'API (Application programming Interface) est, une interface de programmation. Les APIs sont un ensemble de classes, méthodes, fonctions et constantes qui sert d'interface par laquelle un logiciel peut offrir ses services à d'autres logiciels. Elles servent concrètement à accéder aux données d'une application et à utiliser ses fonctionnalités.

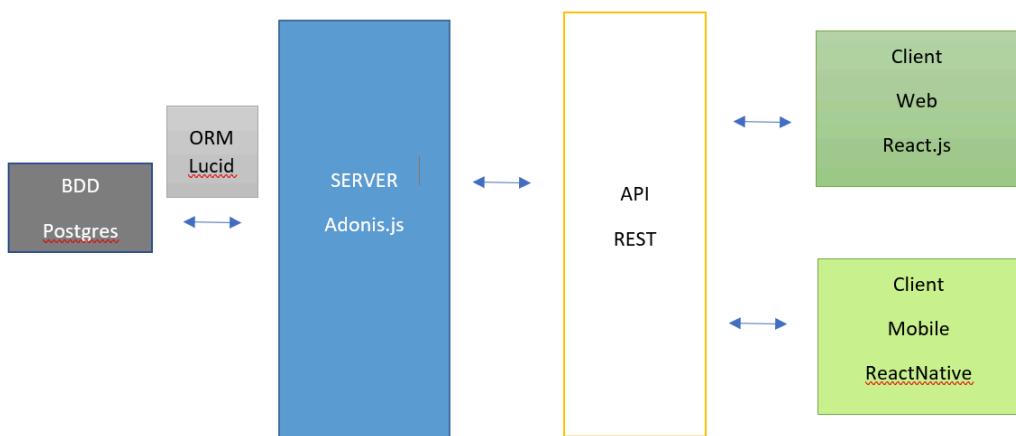
REST API (Representational State Transfer Application Program Interface) est un style architectural qui permet aux logiciels de communiquer entre eux sur un réseau ou sur un même appareil. Le plus souvent les développeurs utilisent des API REST pour créer des services web. Souvent appelés services web RESTful, REST utilise des méthodes HTTP pour récupérer et publier des données entre un périphérique client et un serveur.

DOSSIER PROFESSIONNEL (DP)

En utilisant le protocole HTTP, les REST API permettent aux logiciels d'un appareil de communiquer avec les logiciels d'un autre appareil (ou du même appareil) même s'ils utilisent des systèmes d'exploitation et des architectures différents. Le serveur renvoie la ressource au format JSON (JavaScript Object Notation), XML (Extensible Markup Language) ou texte par exemple.

L'API REST obéit à certains principes comme l'Architecture client-serveur: Les clients de l'API utilisent des appels HTTP pour demander une ressource (une méthode GET) ou envoyer des données au serveur (

une méthode POST), ou l'une des autres méthodes HTTP prises en charge par l'API. GET et POST sont les méthodes les plus fréquemment utilisées, mais d'autres méthodes comme HEAD, PUT, PATCH, DELETE, CONNECT, OPTIONS et TRACE peuvent également être prises en charge. La documentation de l'API montre les méthodes disponibles prises en charge



Création du back-end:

Adonis.js est un framework pour le back-end high-scope. Ils offrent de nombreuses fonctionnalités intégrées avec des structures établies.

Modules intégrés dans le projet :

- lucid : ORM pour les interactions avec les bases de données ;
- auth : Gestion de l'authentification des utilisateurs ;

DOSSIER PROFESSIONNEL (DP)

- bouncer : Système de gestion des permissions et des rôles ;
- drive : Gestion du stockage de fichiers, avec support de divers services (S3, GCS) ;
- japa : Framework de tests ;
- ace : Création d'interfaces de ligne de commande
- Vine.js : validation des données

Lucid : Lucid est un ORM (Object-Relational Mapping). C'est une technique de programmation informatique qui crée un pont entre les bases de données relationnelles et le paradigme de la programmation orientée objet. L'ORM permet aux développeurs d'interagir avec une base de données à l'aide d'objets, sans avoir à écrire directement des instructions SQL ou à gérer les échanges de données de manière manuelle. Les ORM permettent d'effectuer en général toutes les opérations du CRUD et d'intégrer certaines requêtes spécialisées.

Auth avec Access tokens guard : Les tokens authentifient les requêtes HTTP dans les contextes API où le serveur ne peut pas conserver les cookies sur l'appareil de l'utilisateur final, par exemple pour l'accès tiers à une API ou l'authentification pour une application mobile. Comme à terme, l'application sera aussi disponible sur mobile avec une possibilité de s'authentifier, l'utilisation de token semblait adéquate pour ce projet. Les jetons d'accès peuvent être générés dans n'importe quel format, ici les tokens sont conformes à la norme JWT. AdonisJS utilise des jetons d'accès opaques structurés et stockés comme suit :

Un jeton est représenté par une valeur aléatoire cryptographiquement sécurisée, suivie d'une somme de contrôle CRC32. Un hachage de la valeur du jeton est conservé dans la base de données. Ce hachage permet de vérifier le jeton au moment de l'authentification. La valeur finale du jeton est codée en base64 et préfixée par oat_. Le préfixe est personnalisable. Le préfixe et le suffixe de la somme de contrôle CRC32 aident les outils d'analyse secrète à identifier un jeton et à empêcher sa fuite dans une base de code. (voir ce que ça veut dire) Le token est ensuite envoyé côté client pour indiquer que l'utilisateur est connecté. Il est possible de décider le temps de validité d'un token.

Japa : framework de test pour Node.js

Ace : framework de ligne de commande pour simplifier l'usage du framework.

Exemples :

- node ace list (liste des commandes)
- node ace serve -hmr (lancer le server)
- node ace make:migration users (créer une migration pour la table users)

Vine.js : VineJS est l'une des bibliothèques de validation de données les plus rapides dans l'écosystème Node.js. Exemple d'utilisation de Vine.js :

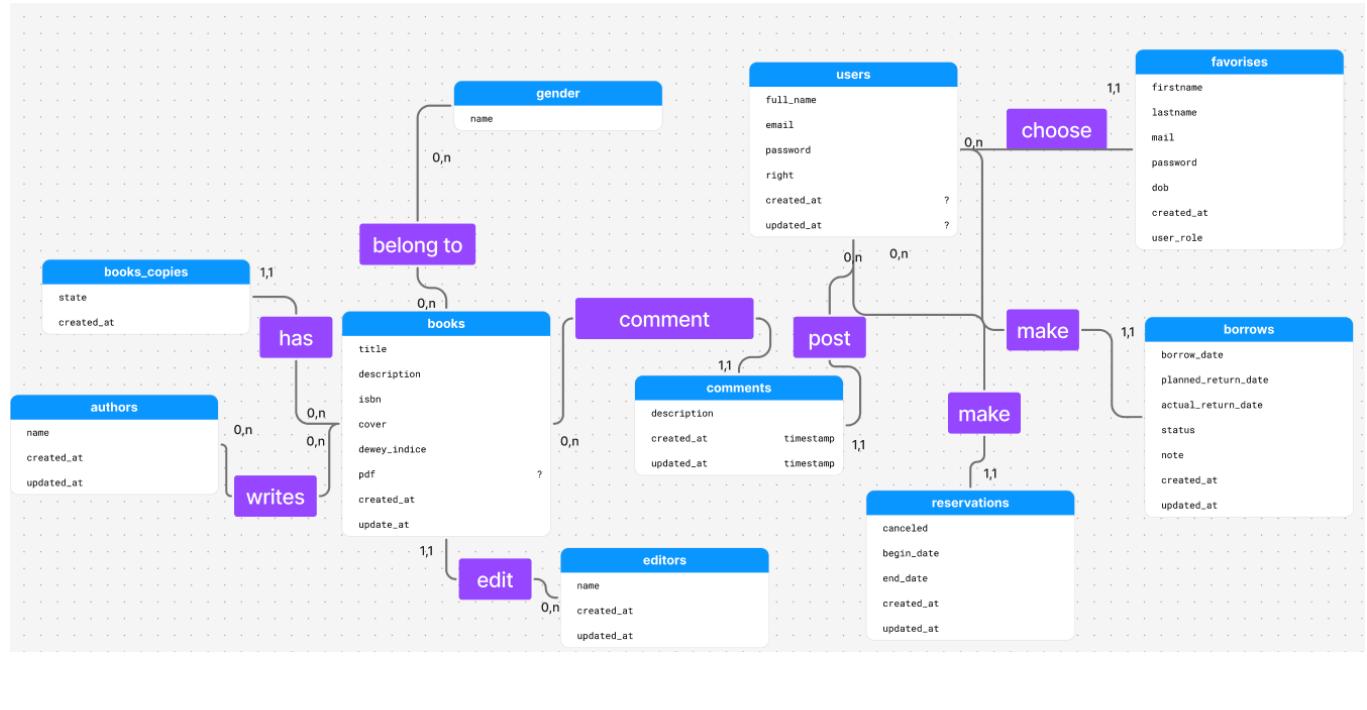
Création du front-end:

Création d'un front-end desktop avec React et (si temps) d'une partie mobile) avec react Nativ.

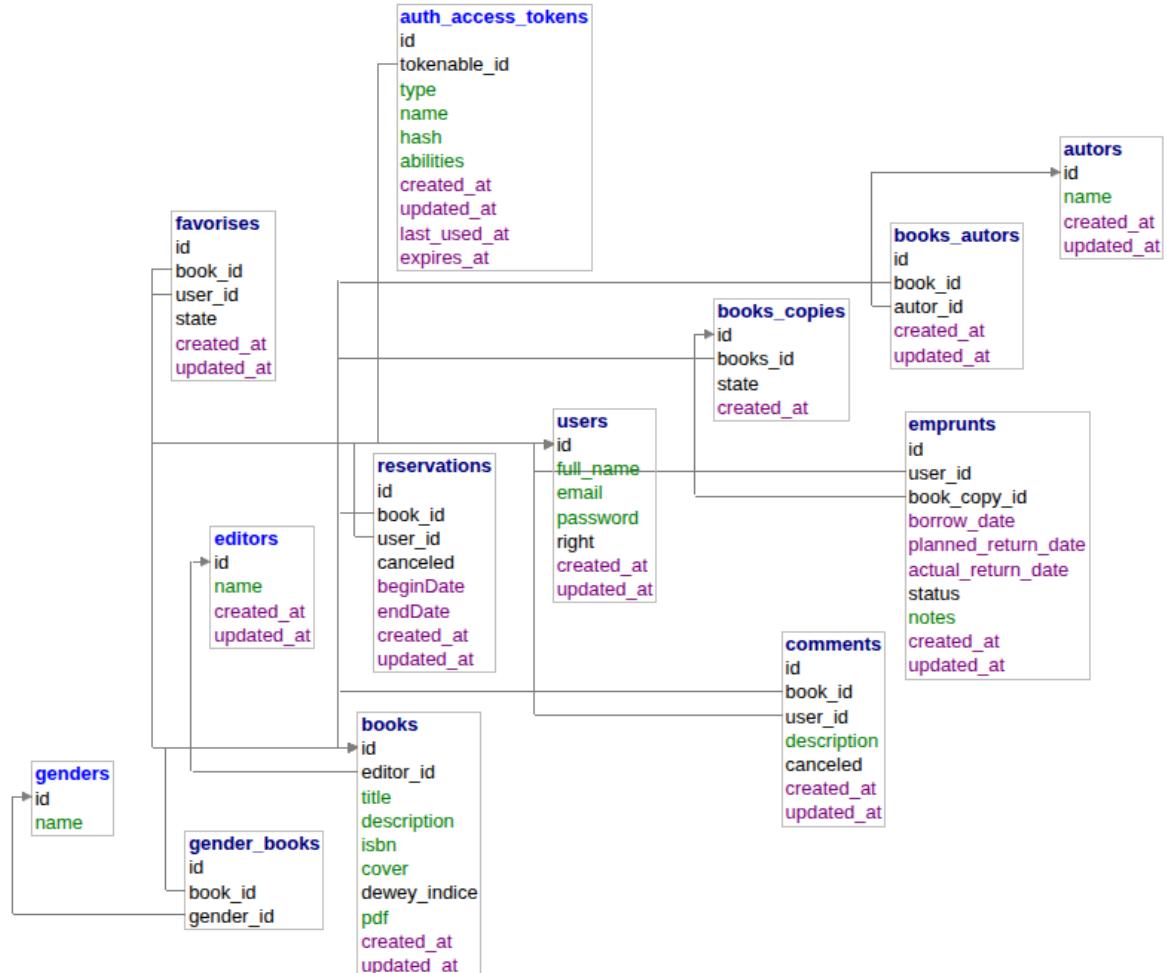
DOSSIER PROFESSIONNEL (DP)

Les bases de données :

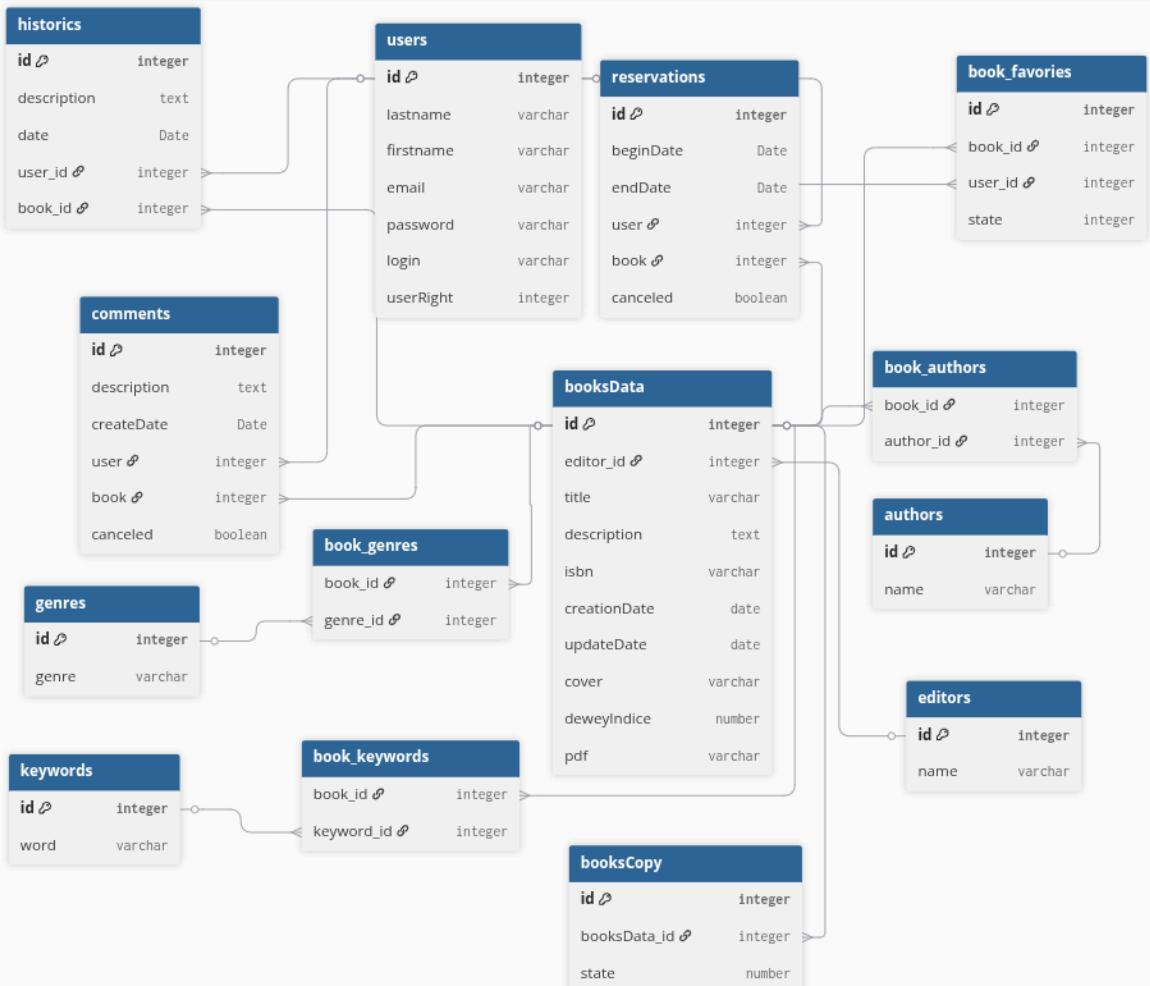
Model conceptuel de données



DOSSIER PROFESSIONNEL (DP)



DOSSIER PROFESSIONNEL (DP)



Utilisation de Redis avec Transmit dans Adonis.js pour envoyer des données en temps réel comme les notifications. Utilisation du Server-Sent-Event (SSE).

 **redis** Redis est système de gestion de base de données clé-valeur extensible, NoSQL, vise à fournir les performances les plus élevées possible. Pour visualiser mes données Redis, j'utilise Redis Commander, outil web pour gérer mes bases de données.

Ce projet couvre les compétences :

Analyser les besoins et maquetter une applications

Définir l'architecture logicielle d'une applications

Concevoir et mettre en place une base de données relationnelle

Développer des composants d'accès aux données SQL

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Visual Studio Code : environnement de développement

Git et Github : versionning

Languages : javascript, sql

Framework : pour le back-end utilisation du framework javascript Adonis.js et pour le front-end, utilisation de React.js

Utilisation de docker pour la base de données. Utilisation d'une image d'adminer (application web permettant de gérer des bases de données via une interface graphique.) avec postgres, un Système de Gestion de Bases de Données Relationnelles (SGBDR), ainsi que d'une image de Redis et de Redis Commander .

3. Avec qui avez-vous travaillé ?

Travail seule

4. Contexte

Nom de l'entreprise, organisme ou association ➤ *La Plateforme*

Chantier, atelier, service ➤ *SIGB*

Période d'exercice ➤ Du : *01/02/2025* au : *01/06/2025*

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Activité-type 3

Préparer le déploiement d'une application sécurisée

Exemple n° 1 - SIGB

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Dans mon projet de système de gestion de bibliothèque, j'ai utilisé cypress pour faire des test sur la partie en react. Cela me permet de tester différentes situations pour mes formulaires d'inscription et de connexion.

Le projet SIGB couvre également les compétences de l'activité 3 mentionnées ci-dessus.

Pour les tests, j'ai utilisé deux outils de test, Cypress et Japa.

Cypress est un outil open-source permettant d'écrire les tests, il fonctionne avec très bien avec React.js et propose une interface graphique.

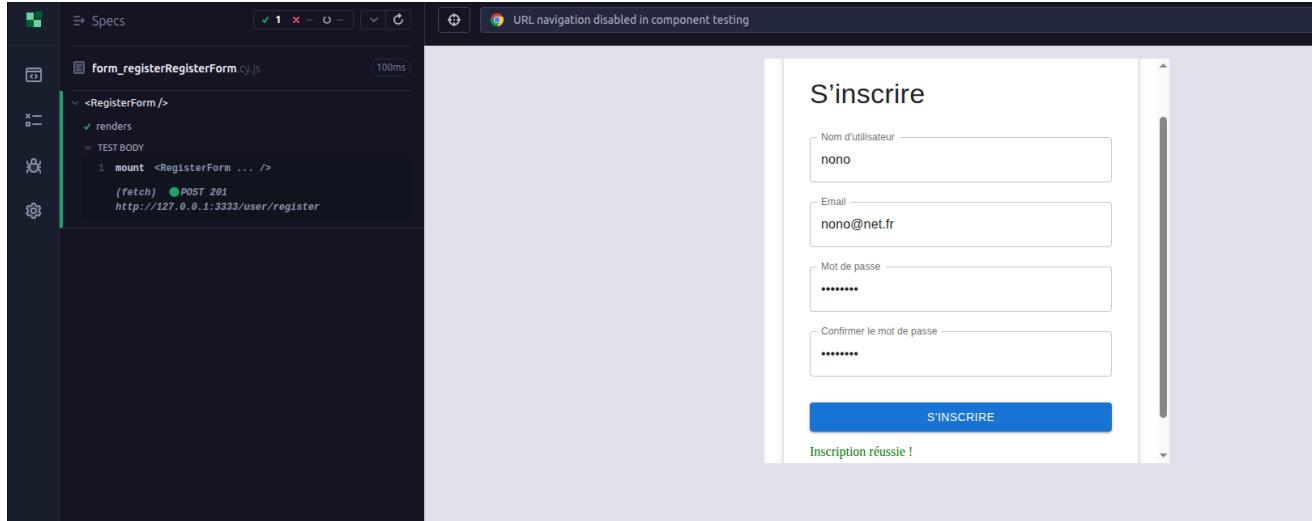
Qu'est-il possible de tester avec Cypress ?

Cypress permet d'effectuer une grande variété de tests, mais la librairie a été développée pour permettre d'effectuer des tests de composants et des tests end-to-end.

- Les tests de composant vont permettre d'isoler un composant du projet web et de le tester en modifiant les données d'entrées. Le but va être de s'assurer du bon fonctionnement du composant, quel que soient les données reçues.
- Les tests End to End (de bout en bout) sont une technique utilisée pour vérifier si une application (web, mobile...) se comporte comme prévu, du début à la fin. Ils consistent à vérifier que l'utilisateur final puisse achever les principaux scénarios d'utilisation de l'application. Les tests end-to-end vont permettre de tester des scénarios de test, c'est-à-dire des chemins prédéfinis que pourrait suivre un utilisateur sur l'application. Par exemple, pour une création de compte, l'utilisateur pourrait se déplacer sur plusieurs pages et interagir avec plusieurs composants, ce qu'il nous serait impossible de tester avec les seuls tests de composants.

Les tests de composants seront généralement développés en parallèle du composant lui-même. Il est d'ailleurs parfaitement possible d'écrire les tests d'abord dans une logique [TDD](#). A l'inverse, il vaudra mieux attendre une certaine stabilité de l'application pour commencer les tests end-to-end pour ne pas modifier les tests à chaque fois qu'un bouton est déplacé ou qu'un composant est modifié.

DOSSIER PROFESSIONNEL (DP)



E2E specs Component specs

src

- NavBarNavbar.cy.js
- form_connectionLoginForm.cy.js
- form_registerRegisterForm.cy.js

Avant de créer un test fonctionnel, il faut rédiger un scénario de test. C'est une séquence d'actions comprenant des données de test et des résultats attendus pour chaque action, qui peuvent être effectuées pour vérifier la fonctionnalité du logiciel . Voici un exemple de scénario de test possible.

DOSSIER PROFESSIONNEL (DP)

Composant testé	Description	Etapes du test	Données testée	Résultat reçu	Succès ou échec
Formulaire de Connexion	Connexion avec de bons identifiants	Cliquer sur le bouton se connecter. Entrer les données et Cliquer sur Valider		Redirection vers Dashboard	succès
Formulaire de Connexion	Connexion avec un mauvais mot de passe	Cliquer sur le bouton se connecter. Entrer les données et Cliquer sur Valider		Inscription en rouge mauvais identifiants	échec
Formulaire de Connexion	Connexion avec un mauvais email	Cliquer sur le bouton se connecter. Entrer les données et Cliquer sur Valider		Inscription en rouge Mauvais identifiants	échec
Formulaire d'enregistrement	Enregistrement avec des données valides	Cliquer sur le bouton s'enregistrer. Entrer les données et Cliquer sur Valider		Inscription réussie!	succès
Formulaire d'enregistrement	Enregistrement avec un email qui existe déjà	Cliquer sur le bouton s'enregistrer. Entrer les données et Cliquer sur Valider			échec
Formulaire d'enregistrement	Enregistrement avec un mot de passe nom valide.	Cliquer sur le bouton s'enregistrer. Entrer les données et Cliquer sur Valider		Afficher en rouge conditions du MP	succès
Formulaire d'enregistrement	Format email non valide	Cliquer sur le bouton s'enregistrer. Entrer les données et Cliquer sur Valider			

Avec Github Actions, il est également possible d'automatiser les tests lors d'une action définie (par exemple un push).

GitHub Actions est une plateforme d'intégration continue et livraison continue (CI/CD) qui permet d'automatiser le pipeline de génération, de test et de déploiement. Les actions sont définis dans un fichier .yml.

Voici un exemple pour lancer les test avec cypress :

DOSSIER PROFESSIONNEL (DP)

```
<describe('Inscription et connexion', () => {
  <const user = {
    full_name: 'Jean Dupont',
    email: 'lebossdu02@example.com',
    password: 'MotDePasse123',
  };

  it('permet de s'inscrire puis de se connecter', () => [
    // Incription
    cy.visit('/register');

    cy.get('input[name="full_name"]').type(user.full_name);
    cy.get('input[name="email"]').type(user.email);
    cy.get('input[name="password"]').type(user.password);
    cy.get('input[name="confirmPassword"]').type(user.password);

    cy.get('button[type="submit"]').click();

    cy.wait(2000);
    cy.contains('Inscription réussie !', { timeout: 10000 }).should('be.visible');

    cy.visit('/login');

    cy.get('input[name="email"]').type(user.email);
    cy.get('input[name="password"]').type(user.password);

    cy.get('button[type="submit"]').click();

    cy.url().should('include', '/dashboard');
    cy.contains('Bienvenue').should('be.visible');
  ]);
});
```

The screenshot displays two side-by-side windows. On the left is a terminal window showing a Cypress test file named 'register.cy.js'. The file contains JavaScript code for performing a registration and login process. On the right is a browser window titled 'Mon App' showing the application's user interface. The UI includes a welcome message, user information, and several buttons for managing loans and reservations.

DOSSIER PROFESSIONNEL (DP)

Utilisation de github actions pour automatiser certaines tâches

Dans un dossier .github/workflows, on peut automatiser des tâches

Dans le cas ci-dessous par exemple, dès qu'on push sur la branche distante, on fait un backup des fichiers de dump de base de données :

```
name: Upload Existing Backup

on:
  push:
    branches: [updateDatabase]

jobs:
  upload:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout repo
        uses: actions/checkout@v4

      - name: Upload latest backup
        uses: actions/upload-artifact@v4
        with:
          name: postgres-backup
          path: backups/backup_*.sql
```

All workflows

Showing runs from all workflows

6 workflow runs

✓ Merge branch 'updateDatabase' of https://github.com/morgane-marechal/...

Docker PostgreSQL Backup (SQL + Artefact) #19: Commit [d27797b](#) pushed by [morgane-marechal](#)

updateDatabase

✓ add artefact

Docker PostgreSQL Backup (SQL + Artefact) #18: Commit [daccd4b](#) pushed by [morgane-marechal](#)

updateDatabase

✓ Merge branch 'updateDatabase' of https://github.com/morgane-marechal/...

Docker PostgreSQL Backup (SQL + Artefact) #17: Commit [cded0f7](#) pushed by [morgane-marechal](#)

updateDatabase

✓ add auto backups1

Docker PostgreSQL Backup (SQL + Artefact) #16: Commit [8a48ae8](#) pushed by [morgane-marechal](#)

updateDatabase

DOSSIER PROFESSIONNEL (DP)

On peut également lancer des tests automatiques avec Githubs actions, par exemple quand on veut push le code sur Github.

```
name: Cypress Tests

on: push

jobs:
  cypress-run:
    runs-on: ubuntu-24.04
    steps:
      - name: Checkout
        uses: actions/checkout@v4
      # Install npm dependencies, cache them correctly
      # and run all Cypress tests
      - name: Cypress run
        uses: cypress-io/github-action@v6
        with:
          build: npm run build
          start: npm start
```

Ce projet recouvre les compétences :

- Préparer et exécuter les plans de test d'une application
- Préparer et documenter le déploiement d'une application
- Contribuer à la mise en production dans une démarche devops

2. Précisez les moyens utilisés :

DOSSIER PROFESSIONNEL (DP)

Visual Studio Code : environnement de développement

Git et Github : versionning

Languages : javascript, sql

Framework : pour le back-end utilisation du framework javascript Adonis.js et pour le front-end, utilisation de React.js

Utilisation de docker pour la base de données. Utilisation d'une image d'adminer (application web permettant de gérer des bases de données via une interface graphique.) avec postgres, un Système de Gestion de Bases de Données Relationnelles (SGBDR).

Utilisation de Cypress pour les tests de la partie front-end avec React et de Japa pour les tests back-end.

Utilisation de github Actions pour le déploiement et les tests automatiques.

3. Avec qui avez-vous travaillé ?

Travail seule

4. Contexte

Nom de l'entreprise, organisme ou association ▶

Chantier, atelier, service ▶

SIGB

Période d'exercice ▶ Du : 01/02/2025 au : 01/06/2025

DOSSIER PROFESSIONNEL^(DP)

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
TOEIC : score 835	ETS Global	01/05/2025
Développeur web et web mobile Titre rncp niveau 5	Ministère du Travail	31/10/2023
Master Sciences humaines et sociales Mention Information et communication	Université de Toulon	28/08/2014
Master Sciences humaines et sociales Etudes européennes et internationales spé Géopolitique	Université de Reims	04/06/2012
Licence Sciences humaines et sociales Mention histoire	Université de Reims	2010

DOSSIER PROFESSIONNEL^(DP)

Déclaration sur l'honneur

Je soussigné(e) Morgane Maréchal ,

déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis l'auteur(e) des réalisations jointes.

Fait à Marseille

le 30 juillet 2025

pour faire valoir ce que de droit.

Signature : Morgane Maréchal

DOSSIER PROFESSIONNEL (DP)

Documents illustrant la pratique professionnelle

(*facultatif*)

Intitulé

Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL ^(DP)

ANNEXES

(Si le RC le prévoit)